Metrika (2002) 55: 37–51

Metrika

Algorithms to compute CM- and S-estimates for regression

O. Arslan[1], **O. Edlund**[2], **H. Ekblom**[2]

[1] University of Cukurova, Department of Mathematics, 01330 Balcali, Adana, Turkey
[2] Luleå University of Technology, Department of Mathematics, S-97187 Luleå, Sweden

Abstract. Constrained M-estimators for regression were introduced by Mendes and Tyler in 1995 as an alternative class of robust regression estimators with high breakdown point and high asymptotic efficiency. To compute the CM-estimate, the global minimum of an objective function with an inequality constraint has to be localized. To find the S-estimate for the same problem, we instead restrict ourselves to the boundary of the feasible region. The algorithm presented for computing CM-estimates can easily be modified to compute S-estimates as well. Testing is carried out with a comparison to the algorithm SURREAL by Ruppert.

Key words: CM-estimators, S-estimators, High breakdown point estimators for regression, Robust regression, Robustness, Algorithms

1 Statistical background

Consider the usual linear regression model

$$y_i = x_i^T \beta + e_i, \tag{1}$$

where x_i and β are p dimensional vectors of covariates and regression coefficients, and e_i are the errors with $E(e_i) = 0$ and $\mathrm{Var}(e_i) = \sigma^2 < \infty$, for $i = 1, 2, \ldots, n$. Among the several estimation techniques for the unknown regression parameter β, the most popular one is the classical least squares method, which can be defined as the minimum point of the following objective function of residuals $r_i = y_i - x_i^T \beta$,

$$\sum_{i=1}^{n} r_i^2. \tag{2}$$

However, least squares estimates are not robust against outlying observations in the data set. A single outlier (point not obeying the trend of the rest of the data) can spoil the estimates very badly. The breakdown point of the least squares estimator is $1/n$ which approaches zero when n tends to infinity (Dohono and Huber, 1983).

Huber (1973, 1981) proposed the regression M-estimators as an alternative robust regression estimator to the least squares. An M-estimate minimizes the following function

$$\sum_{i=1}^{n} \rho(r_i), \tag{3}$$

where $\rho(t)$ is symmetric, having a unique minimum at zero. If $\rho(t) = |t|$, the corresponding M-estimators are the L_1 estimators for regression. If $\rho(t) = t^2$ we just get the least squares estimators. A widely used $\rho(t)$ function is the Huber function which is unbounded but has bounded derivatives and is less rapidly increasing than the square function. M-estimates obtained from the Huber function are sometimes called Huber type M-estimates or monotone M-estimates. Huber type M-estimates are robust against outliers in y-direction, but they are not robust against leverage points (outliers in x-direction).

Because of this vulnerability to the outliers in x-direction of the regression M-estimators, generalized M-estimators (GM-estimators, for short) were introduced (Maronna et al., 1979). The basic idea behind the GM-estimators is to bound the influence of leverage points by making use of some weight function which only depends on x. The most widely used example of GM-estimators of regression are the Mallows and Schweppe types (see Hampel et al., 1986). Although GM-estimators can be tuned to have good local robustness properties measured by the influence function (they can have bounded influence function), it has been shown that GM-estimators for regression do not have good global robustness properties measured by the breakdown point (Maronna et al., 1979, Donoho and Huber, 1983). The upper bound for the breakdown point of a GM-estimator of regression can not be greater than $1/(p+1)$, which makes the situation worse in higher dimensional problems. The problem of the low breakdown point of the GM estimators has been solved by the one-step GM-estimators which have bounded influence and high breakdown point (see Simpson et al., 1992).

The low breakdown point of the regression M-estimators rises the question whether it is possible to get regression estimators with good global robustness properties, that is estimators with high breakdown point. The first high breakdown regression estimator was the repeated median (RM) proposed by Siegel (1982). However RM is not affine equivariant, it depends on the choice of coordinate axes for x_i. Next high breakdown point estimator was the least median of squares (LMS), proposed by Rousseeuw (1984). LMS possesses 50% breakdown point, is affine equivariant but does not have good asymptotic properties.

Rousseeuw and Yohai (1984) proposed S-estimators for regression which is another high breakdown point estimator having the same asymptotic properties as the M-estimators of regression. To obtain a high breakdown point estimator of regression which is $\sqrt{n}$ consistent and asymptotically normal was the motivation for the S-estimators for regression. Let ρ be a symmetric and

continuously differentiable function, which is bounded and nondecreasing on $[0, \infty)$. Let $k = E_\Phi[\rho]$, where Φ is the standard normal distribution. Formally, an S-estimate of regression can be defined as follows. For any given sample $\{r_1, r_2, \ldots, r_n\}$ of residuals, an M-estimate of scale $\sigma(r_1, r_2, \ldots, r_n)$ is the solution to

$$ave\{\rho(r_i/\sigma)\} = k, \tag{4}$$

where "*ave*" stands for the arithmetic mean over $i = 1, 2, \ldots, n$. For each value of β, the dispersion of the residuals $r_i = y_i - x_i^T\beta$ can be calculated using equation (4). Then, the S-estimator $\hat{\beta}$ of β will be defined as

$$\arg\min_{\beta} \sigma(r_1(\beta), r_2(\beta), \ldots, r_n(\beta)), \tag{5}$$

and the final scale estimate is $\hat{\sigma} = \sigma(r_1(\hat{\beta}), r_2(\hat{\beta}), \ldots, r_n(\hat{\beta}))$. In (4), setting $k = E_\Phi[\rho]$ ensures that the S-estimator of the residual scale $\hat{\sigma}$ is consistent for σ_0 whenever it is assumed that the error distribution is normal with zero mean and σ_0^2 variance.

The breakdown point of an S-estimator is determined by the choice of the ρ function. For the large sample case the breakdown point is $k/\rho(\infty)$, where $\rho(\infty) = \lim_{t\to\infty} \rho(t)$. To have approximately 1/2 breakdown point the ρ function should be bounded and properly tuned.

The ρ function in the constraint (4) will depend on a positive tuning parameter c as $\rho_c(t) = c^2\rho(t/c)$. The tuning parameter plays a very important role for the asymptotic and breakdown properties of S-estimators for regression. For all values of c, $\rho(\infty)$ will be the same so that to obtain 1/2 breakdown point of the S-estimator, c must be chosen as the solution to the equation $E_\Phi[\rho_c(t/c)] = \rho_c(\infty)/2$.

A class of widely used ρ functions is the biweight or Tukey ρ given by

$$\rho(t) = \begin{cases} \dfrac{t^2}{2} - \dfrac{t^4}{2} + \dfrac{t^6}{6}, & \text{for } |t| \le 1 \\ \dfrac{1}{6}, & \text{for } |t| \ge 1 \end{cases}. \tag{6}$$

where $\rho(\infty) = 1/6$. Suppose that the error distribution is the standard normal distribution. The solution of the equation $E_\Phi[\rho_c(t/c)] = \rho_c(\infty)/2$ yields $c = 1.5476$ (see Rousseeuw and Yohai, 1984). That is, to get an S-estimator with the asymptotic breakdown point 1/2, one has to choose $c = 1.5476$. To obtain .25, .20 and .15 breakdown points one can choose $c = 2.937, 3.42$ and 4.00, respectively. Some other values of c can be found in Rousseeuw and Yohai (1984).

Rousseeuw and Yohai (1984) show that if ρ is differentiable, the S-estimators for β and σ satisfy the following redescending M-estimating equations

$$ave\{\psi(r_i/\sigma)x_i\} = 0 \tag{7}$$

$$ave\{\rho(r_i/\sigma)\} = k \tag{8}$$

where $\psi(t) = \rho'(t)$. The above estimating equations may have many solutions, but it has to be noticed that not all the solutions correspond to S-estimates. To find the S-estimates one has to seek for the global minimum under the constraint given in (4). The M-estimating equations are useful to obtain the asymptotic normality and the influence function of the S-estimators. The S-estimator of the regression parameter β has an unbounded influence function.

Setting $c = 1.5476$ gives 1/2 breakdown point but the asymptotic relative efficiency (ARE) of the S-estimator for β is 28.7%, which is very low. On the other hand, for $c = 4.096$ the ARE is 91.7% but the corresponding breakdown point is 15%. This shows that one has a trade-off between high breakdown point and high asymptotic relative efficiency.

Constrained M-estimators (CM-estimators) for the regression parameters β and the scale parameter σ were introduced by Mendes and Tyler (1995). The aim is to have robust regression estimators with high breakdown point and high asymptotic relative efficiency. The CM-estimates for β and σ are defined as the global minimum of the objective function

$$L(\beta, \sigma) = ave\{\rho(r_i/\sigma)\} + \log\sigma \tag{9}$$

over all $\beta \in R^p$ and $\sigma > 0$ subject to

$$ave\{\rho(r_i/\sigma)\} \leq \varepsilon\rho(\infty), \tag{10}$$

where ε is a fixed number between 0 and 1, and $\varepsilon\rho(\infty) = k$ in the constraint (4) for the S-estimators.

The CM-estimators are regression and affine equivariant, and possess, at the same time, the good local properties of the M-estimators for regression and good global robustness properties of the regression S-estimators. The breakdown point of the CM-estimates is approximately $\min(\varepsilon, 1-\varepsilon)$ or approximately 0.5 when $\varepsilon = 0.5$. Also, when ρ is properly tuned the CM-estimates can have good local robustness properties. They are consistent, asymptotically normal and very efficient estimators. One can see Mendes and Tyler (1995) and Kent and Tyler (1996) for the robustness and the asymptotic properties of the CM-estimators for regression and the CM-estimators for multivariate location and scatter, respectively.

When ρ is differentiable, the CM-estimates for the regression parameters β and the scale parameter σ satisfy the following estimating equations:

$$ave\{\psi(r_i/\sigma)x_i\} = 0, \tag{11}$$

and either

$$ave\{\chi(r_i/\sigma)\} = 0, \tag{12}$$

or

$$ave\{\rho(r_i/\sigma)\} - \varepsilon\rho(\infty) = 0, \tag{13}$$

where $\chi(t) = t\psi(t)$. If strict inequality holds in the constraint (10) we get the estimating equations (11) and (12) which are the redescending M-estimating

equations for β and σ. If equality holds in the constraint (10) we get the equations (11) and (13) which are the S-estimating equations for β and σ.

The breakdown point of the CM-estimators is independent of the tuning parameter c. It only depends on ε. On the other hand, the asymptotic efficiency and the gross error sensitivity of the CM-estimators for β depend on the tuning parameter c. If the ρ function is properly tuned the CM-estimator for β can have high asymptotic efficiency and high breakdown point. For example, if we set $c = 1.5476$ in the $\rho_c(t)$ function given above, the corresponding CM-estimator for β will be the same as the S-estimator. It is shown in the Appendix, that for the Tukey biweight ρ function, the CM- and S-estimates will always coincide for $0 < c < 2.598$. Corresponding intervals for other ρ functions are given in Arslan et al. (2001). For $c = 1.547$ the breakdown point will be 50%, but ARE is 28.7%; that is, we get high breakdown point but very low ARE which are not the desired properties. On the other hand, if we set $c = 3.42$, the corresponding CM-estimator for β has 50% breakdown point, 85% ARE, but the corresponding S-estimator has 20% breakdown point and 85% ARE (e.g. see Mendes and Tyler, 1995, and Rousseeuw and Yohai, 1984). That is, we can have the same ARE for both CM- and S-estimators but the breakdown point may be very different. Similarly setting $c = 4.00$, we get 50% breakdown point for the CM-estimator. The CM- and S-estimators have the same 91% ARE but the S-estimator has a breakdown point of only 15%.

To illustrate the performance of CM-estimates we will give a simple example.

Example 1. We will consider the data set of the Hertzsprung-Russell diagram of the star cluster, which contains 47 stars in the direction of Cygnus and is a perfect example of a bad cluster of outliers. This data set can be found in Rousseeuw and Leroy (1987). Here x is the logarithm of effective temperature at the surface of the star, and y is the logarithm of its light intensity. Examining the scatter plot of the data (Figure 1) we can identify two clusters.

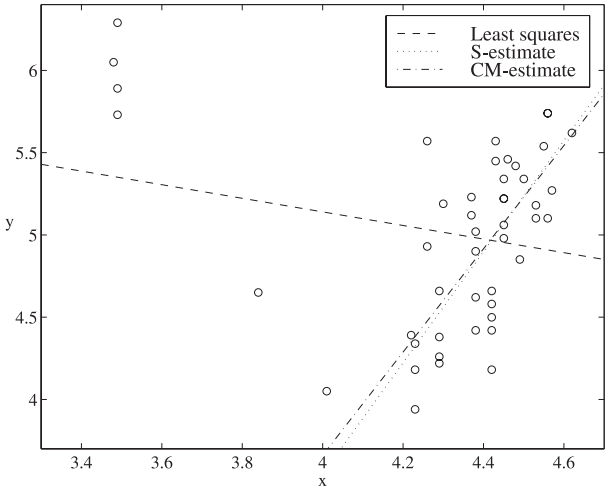

Fig. 1.

Clearly the majority of the stars follow a steep direction on the right of the figure, forming the so-called main sequence. The second cluster is formed by four stars, called giants.

Figure 1 also shows the plot of the data set with LS, S and CM fits. The LS line fits very poorly to the data. The S-estimate from Rousseeuw and Yohai (1984) fits to the main sequence ignoring the second cluster. The CM-estimate from Mendes (1995), gives another good fit to the data set. For the S-estimate, $c = 1.547$ was used, and for the CM-estimate we had $c = 4.00$. The difference between the S- and CM-estimates should be noticed.

2 Algorithms

2.1 Introduction

As described above, a CM-estimate for regression minimizes the function

$$L(\beta, \sigma) = ave\{\rho(r_i/\sigma)\} + \log(\sigma) \tag{14}$$

over $\beta \in R^p$ and $\sigma > 0$ subject to the constraint

$$ave\{\rho(r_i/\sigma)\} \leq \varepsilon\rho(\infty).$$

To find the S-estimate, we solve a similar problem but with equality in the constraint instead, i.e. we are restricted to the boundary of the feasible region. In either case, we seek the solution to a nonlinear minimization problem with a constraint, and this solution cannot be expressed explicitly. Computing S- and CM-estimates numerically is a challenging problem, since we like to minimize an objective function where many local minima may exist. We can divide the computational problem into two parts:

– a strategy for finding good start points
– a reliable and efficient local minimizer

To find the global minimum of an objective function is in general an unsolvable problem. Many strategies, most of them based on heuristics, have been proposed to catch the minimum with high probability (see e.g. Törn and Zilinskas, 1989). If possible, the special character of the problem should be taken into account to locate regions where the global minimum most probably is to be found. For our problem reasonable β-values can be generated by simplified fitting using a small subset of the equations given.

In the algorithm PROGRESS (Rousseeuw and Leroy, 1987), which can be used to compute S-estimates, p of the n equations are picked at random each time, generating a square system to be solved for β. Ruppert (1992) presented a more refined S-algorithm SURREAL. In SURREAL, when a new start point is generated, other points are picked along the line between the generated point and the best point so far. The IRLS-algorithm is used for the local optimization. A simple line search is applied on the IRLS-steps to ensure convergence.

2.2 An algorithm to compute CM-estimates

2.2.1 Finding a local minimum

Obviously a general purpose computer program for minimizing a non-linear objective function with an inequality constraint can be tried, but we can certainly do better utilizing the special character of the problem. First we note that if σ is held fixed in the objective function $L(\beta, \sigma)$ defined in (14), we have a linear model M-estimation problem. Edlund (1997) presents a fast and robust algorithm for this task. It is based on Newton's method with line search, which has a favourable convergence rate compared to the commonly used IRLS method.

We first give a rough sketch of an algorithm where β and σ are updated separately:

given start values β_0 and σ_0;
$k := 0$;
if (β_0, σ_0) is infeasible **then**
    find a point σ_1 such that (β_0, σ_1) is on the border of the feasible region
else
    $\sigma_1 = \sigma_0$;
find $\beta_1 := \text{argmin}\, L(\beta, \sigma_1)$, i.e. go to "the bottom of a valley";
while $(\|\beta_{k+1} - \beta_k\| > \varepsilon_1)$ or $(|\sigma_{k+1} - \sigma_k| > \varepsilon_2)$ **do begin**
    $k := k + 1$;
    find a point σ_{k+1} approximately minimizing $L(\beta_k, \sigma)$ in the feasible region;
    find $\beta_{k+1} := \text{argmin}\, L(\beta, \sigma_{k+1})$, i.e. go to "the bottom of a valley";
end

Here ε_1 and ε_2 are suitably chosen tolerance parameters.

When $L(\beta, \sigma_{k+1})$ is minimized with fixed σ_{k+1} above, one would imagine that there is a risk of slipping outside the feasible region, but it is easy to see that this can never happen. The minimization process will make $ave\{\rho(r_i/\sigma_{k+1})\}$ smaller, and since the feasible region is given by the constraint $ave\{\rho(r_i/\sigma)\} \leq \varepsilon\rho(\infty)$, we will move away from the border, into the interior of the feasible region.

It is well known that separating variables totally may slow down convergence. For this reason an alternative algorithm is designed. Instead of minimizing with respect to σ, another direction is chosen for the one-dimensional line search for a minimum. It is possible to let this direction be along the "the bottom of the valleys" of the objective function. An algorithm along theses lines is the following:

given start values β_0 and σ_0;
$k := 0$;
if (β_0, σ_0) is infeasible **then**
    find a point σ_1 such that (β_0, σ_1) is on the border of the feasible region
else
    $\sigma_1 = \sigma_0$;
find $\beta_1 := \text{argmin}\, L(\beta, \sigma_1)$, i.e. go to "the bottom of a valley";
while $(\|\beta_{k+1} - \beta_k\| > \varepsilon_1)$ or $(|\sigma_{k+1} - \sigma_k| > \varepsilon_2)$ **do begin**

$k := k + 1$;
do a linear approximation of the "valley", with direction given by $\Delta\beta_k$ and $\Delta\sigma_k$;
find a point $(\beta^*_{k+1}, \sigma_{k+1}) = (\beta_k + \alpha_k\Delta\beta_k, \sigma_k + \alpha_k\Delta\sigma_k)$ approximately minimizing $L(\beta_k + \alpha\Delta\beta_k, \sigma_k + \alpha\Delta\sigma_k)$in the feasible region;
find $\beta_{k+1} := \text{argmin}\, L(\beta, \sigma_{k+1})$, i.e. go to "the bottom of a valley";
end

The direction $(\Delta\beta_k, \Delta\sigma_k)$ is found by applying the implicit function theorem. Let

$$F(\sigma, \beta) = ave\{\psi(r_i/\sigma)x_i\}.$$

By applying the implicit function theorem on (11), i.e. $F(\sigma, \beta(\sigma)) = 0$, we find that

$$\frac{d\beta}{d\sigma} = -(X^T WX)^{-1} X^T W(r/\sigma),$$

where W is a diagonal matrix with diagonal entries $w_{ii} = \rho''(r_i/\sigma)$, and X is a $n \times p$-matrix with row i equal to x_i. So the direction of the valley is given by

$$\begin{cases} \Delta\beta_k = -(X^T WX)^{-1} X^T W(r/\sigma), \\ \Delta\sigma_k = 1. \end{cases}$$

The calculation of $\Delta\beta_k$ is carried out by solving the system of linear equations $X^T WX\Delta\beta_k = -X^T W(r/\sigma)$. The matrix of this system is identical to the one used in the minimization of β, thus the last factorization from that calculation can be used here, to save computation time.

The step to find $(\beta^*_{k+1}, \sigma_{k+1})$ is carried out by first calculating a Newton step for minimizing $L(\beta_k + \alpha\Delta\beta_k, \sigma_k + \alpha\Delta\sigma_k)$, and then applying steplength control to ensure convergence. If this procedure happens to slip out of the feasible region, a coordinate $(\beta^*_{k+1}, \sigma_{k+1})$ on the border is found instead.

2.2.2 Global minimization

As mentioned above, reasonable β-values can be found by simplified fitting using a small subset of the n equations given. We can pick p equations at random each time, generating a square system to be solved for β. The hope is to get at least some samples without outliers, giving good start points from where the global minimum can be identified. However, it is possible to find examples where, although the subsample consists of "good data", the resulting estimate of the β-values is totally wrong. Referring to Example 1, we may think of choosing two of the points corresponding to non-giant stars but being close together in the diagram. The line through these two points may have almost any direction, i.e. we have an ill-conditioned problem. This motivates the decision to use $s = p + q$ points in the least squares fit, where q is a small number (typically 1 or 2). Since we are just generating start values for the local minimizer, there is no need to solve the overdetermined equation very accurately. Thus we may use the normal equations, which are solved by Cholesky factorization. The computational cost is not much higher than for solving the

square system and also small compared to what is needed for the iterations which follow.

Now let us consider the following question. Assume that we have n equations, out of which $b \cdot n$ include outliers. If we take d samples, each consisting of s equations, to compute start values for the iteration, what chance do we have to get at least one sample free of outliers? If b is the fraction of outliers in data, the probability for a sample to consist of only "good points" is

$$\Pr(\text{"good sample"}) = \frac{(1-b)n}{n} \frac{(1-b)n-1}{n-1} \frac{(1-b)n-2}{n-2} \cdots \frac{(1-b)n-(s-1)}{n-(s-1)}. \quad (15)$$

From this follows that the probability that at least one of the d samples is outlier-free is

$$P = 1 - (1 - \Pr(\text{"good sample"}))^d$$

Thus given values of s, b and P, the required number of start points is limited by

$$d > \ln(1-P)/\ln(1 - \Pr(\text{"good sample"})) \quad (16)$$

Example 2. Consider again Example 1, where $b = \frac{4}{47}$ and $p = 2$. Taking $q = 2$ extra points for smoothing gives $s = 4$. Then $d = 8$ gives the probability 99.99% to have at least one outlier-free sample out of the 8 samples generated. To achieve $P = 99\%$ we only need 4 samples.

In Table 1 we give the number of samples needed to achieve $P = 50\%$, 90% and 99% when $n = 100$, for some values of b and s. If n is very large compared to s (as is often the case), we can make the following simplification of (15)

$$\Pr(\text{"good sample"}) \approx (1-b)^s$$

This leads to some reduction in the number of samples needed, as seen from comparing Table 1 and Table 2.

Table 1. Number of samples needed to get at least one outlier-free sample with probabilities 50%, 90% and 99%, resp., in case $n = 100$

	$b = 10\%$	$b = 25\%$	$b = 40\%$
$s = 5$	1 3 6	3 9 18	10 31 62
$s = 10$	2 6 12	15 47 94	159 528 1055

Table 2. Limits (for n approaching ∞) of the number of samples needed to get at least one outlier-free sample with probabilities 50%, 90% and 99%, resp

	$b = 10\%$	$b = 25\%$	$b = 40\%$
$s = 5$	1 3 6	3 9 17	9 29 57
$s = 10$	2 6 11	12 40 80	115 380 760

Another aspect to take into account is whether the s equations should be chosen totally at random each time. As an alternative, the total amount of samples may be divided into blocks. Let us consider the case when the smallest possible block size is used.

Example 3. In Ruppert's paper on algorithms for S-estimates (Ruppert, 1992), a simulation was carried out with 9 out of 50 equations including outliers and $p = 5$. If 10 samples of size 5 are generated at random, the probability to get outliers in every sample is $\left(1 - \frac{41}{50}\frac{40}{49}\frac{39}{48}\frac{38}{47}\frac{37}{46}\right)^{10} \approx 1.3\%$. In contrast, generating 10 non-overlapping samples will of course give at least one sample without outliers.

In general, if we have $b < \frac{1}{s+1}$ then subdividing the equations into non-overlapping samples is sufficient to produce one "good sample". If b is larger, the non-overlapping strategy has to be repeated.

In Example 3, using non-overlapping samples made a difference, although not very large. In other cases the effect may be more pronounced, as the following example shows.

Example 4. Let $n = 6$, $s = p = 2$, $b = 50\%$ and $d = 3$. The probabilities that all three samples include outliers is $\frac{3}{5}\frac{2}{3} = 40\%$ if the non-overlapping strategy is used but is $0.8^3 = 51.2\%$ otherwise.

In our code CMREGR, non-overlapping is standard, but can be relaxed as an option.

2.3 A new way to compute S-estimates

As stated above, for suitably chosen small values of the tuning parameter c, the CM- and S-estimates coincide, so in those cases the algorithm above can also be used for S-estimates. For greater values of c there is a difference though. If S-estimates are sought rather than CM-estimates, we can still get the desired solution from the above algorithm by setting a special parameter. Following Mendes and Tyler (1995) we can introduce a parameter γ to scale the first term of the objective function (14)

$$L(\beta, \sigma) = \gamma\, ave\{\rho(r_i/\sigma)\} + \log\sigma.$$

This parameter is set to $\gamma = 1$ in all examples where CM-estimates are sought, but by letting $\gamma = 0$ we will always get S-estimates, regardless of the choice of c, because the objective function then reduces to $\log\sigma$. Mendes and Tyler have the parameter γ as a possible alternative to using c. (Note that in Mendes and Tyler (1995) our c is denoted k, and γ denoted c.)

3 Testing

3.1 Experimental design

To study the performance of the algorithms presented above, a Monte Carlo simulation was carried out similar to the one Ruppert used for testing his S-

algorithm SURREAL (Ruppert, 1992). In his design, 200 samples were generated with $n = 50$ and $p = 5$. All samples used the same X matrix (the matrix with ith row equal to x_i). The first column was identically 1 and the remaining columns were filled with independent standard normal variates, except that the last nine entries of the fifth column were $N(10, 1)$. The y_i's were $N(0, (0.25)^2)$ except for the last nine, which were $N(10, (0.25)^2)$. Thus the first 41 observations were thought of as "good" data, corresponding to $\beta = 0$, while the remaining $b = 18\%$ of the data were outliers intended to pull the estimate of β_5 towards 1. The algorithms in the test were applied twice to see whether the same solution was generated or not.

In our simulations, we built X and y along the same lines (i.e. trying to pull β_p away from zero) as above, but the fraction of outliers was also doubled (to $b = 36\%$) in some of the tests. We used the following combinations of p and n: (5, 50), (5, 200), (10, 50) and (10, 200). The number of samples was set to 10 in each case. For each new sample we generated a new matrix X. The outliers were generated by taking the last $b \cdot n$ entries in y from $N(10, (0.25)^2)$ and the last $b \cdot n$ entries in the last column of X from $N(10, 1)$. We also used different c-values, as given below.

3.2 The code CMREGR

The algorithm presented in section 2.2 has been implemented in Matlab as the code CMREGR, and is used in the tests below. The code is downloadable from the internet at web-adress 'http://www.sm.luth.se/~jove/research'.

Preliminary testing showed that the use of least-squares smoothing (with $q = 1$ and 2) instead of interpolation ($q = 0$) does not usually compensate for the decreased probability to find outlier-free samples. For that reason smoothing is an option in the code but not used in the testing below. As to the local minimizer, updating β and σ separately may give a slow "zig-zaging" behaviour in the iterations (like the steepest descent algorithm), the other way to find directions was taken as standard and used below.

In summary, the "standard form" of CMREGR used in the test has the following characteristics:

a) the more advanced search for good directions (not separating β and σ) was used
b) for finding the start points, square systems were solved (i.e. $q = 0$)
c) in sampling equations for startpoints, a non-overlapping strategy was used.

3.3 Computing CM-estimates

The testing was done on a PC, with a 550 MHz Pentium III processor, running the FreeBSD operating system. Matlab version 5.3 was used.

For every set of p, n, b and c, 10 test problems were generated and the algorithm was applied 10 times on each of these test problems. It was recorded in how many of these attempts a good result (i.e. $\beta \approx 0$) was reached. These cases are called "hits" below. Furthermore, average numbers of Newton steps used and changes in the σ-value were computed. Also the average time needed to find the solution was recorded as well as the fraction of time spent in the preparation phase of the algorithm, i.e. where start points are generated.

Table 3. The number of samples used in the preparation phase

p, n	5, 50	5, 200	10, 50	10, 200
nr. of samples	47	42	731	454

Table 4. Result from computing CM-estimates with $c = 4$. In all cases when the solution was found it was at the boundary of the feasible region

p, n, b	5, 50, 18	5, 50, 36	5, 200, 18	5, 200, 36	10, 50, 18	10, 50, 36	10, 200, 18	10, 200, 36
Average hitting %	100%	99%	100%	98%	100%	79%	100%	85%
Average # Newton steps	11.2	8.9	8.7	6.9	20.0	14.8	11.4	8.8
Average # σ changes	3.8	3.3	3.1	3.0	5.0	4.2	4.0	3.3
Average time consumed (s)	0.16	0.14	0.26	0.24	0.89	0.82	0.85	0.77
% time spent on preparation	38%	43%	31%	33%	79%	83%	65%	69%

Table 5. Result from computing CM-estimates with $c = 6$. In all cases when the solution was found it was in the interior of the feasible region

p, n, b	5, 50, 18	5, 50, 36	5, 200, 18	5, 200, 36	10, 50, 18	10, 50, 36	10, 200, 18	10, 200, 36
Average hitting %	100%	98%	100%	99%	100%	79%	100%	89%
Average # Newton steps	8.4	8.9	7.0	7.3	9.4	13.5	8.0	8.4
Average # σ changes	4.1	4.2	3.8	4.2	4.2	5.3	4.0	4.5
Average time consumed (s)	0.14	0.14	0.23	0.24	0.79	0.80	0.78	0.77
% time spent on preparation	50%	43%	35%	33%	89%	85%	72%	69%

The number of samples used in the preparation phase was determined by formula (16) with $P = 99\%$ and $b = 36\%$. So in cases with 18% outliers we are doing some extra work in the preparation phase. This makes sense if one assumes no apriori knowledge of b. The number of samples used are displayed in Table 3.

We used two different c-values, namely $c = 4$ for which the solution was at the boundary of the feasible region with high probability, and $c = 6$, for which the solution most probably was to be found in the interior. The result of the simulation is given in Table 4 and 5.

We can observe in Table 4 and 5 that there is no striking difference in the result for $c = 4$ and $c = 6$. We also note that the hit-rate is generally quite high, something which can be taken as an indication that we have found the global minimum. We can also see in the tables, that in some cases (especially for $p = 10$ and $n = 50$) the hit-rate is much smaller than the expected fraction of "good samples". The way the problems are generated probably explains

Table 6. Result from computing S-estimates with $c = 1.547$

p, n, b	$5, 50, 18$	$5, 200, 18$	$10, 50, 18$	$10, 200, 18$
CMREGR giving best σ-value	99%	100%	97%	100%
SURREAL giving best σ-value	1%	0%	3%	0%
CMREGR/SURREAL σ-values	0.994	0.999	0.985	0.997

this behaviour. If we let p equal two, start solutions are generated in a way which is equivalent to finding the interpolating line $y = \beta_1 + \beta_2 t$ through two points (t_1, y_1) and (t_2, y_2), where t_1 and t_2 are normal deviates. If t_1 and t_2 happen to be close together, the slope β_2 and interception β_1 can certainly be far away from the global solution, also if (y_1, y_2) constitutes a "good sample".

3.4 Computing S-estimates

Problems were generated in the same manner as for computing CM-estimates, but here the tuning parameter c was given the value 1.547, also used in Ruppert (1992), and only $b = 18\%$ was considered. Comparison was made with a Matlab program (modification of a code written by Stefan Van Aelst) based on the algorithm SURREAL (Ruppert, 1992).

For each size, 100 problems were generated. Each problem was solved once by the two programs. The code giving the lowest σ was considered "winner". The result from all the "matches" was compiled and the average of percentage difference between computed best σ-values was also computed. The results are given in Table 6. The aim of the test is to investigate how well CMREGR and SURREAL manage to find the S-estimate in a set of regression problems. What can be seen from Table 6 is that both algorithms behave well, with CMREGR slightly more successful.

4 Conclusion

The algorithm presented works well on the test problems, for finding regression CM-estimates, in spite of the fact that a global optimization problem has to be solved. There are guidelines on how to choose the number of generated points in the preparation phase, but the reliability of the output from the program also depends on properties of the design matrix X. In should be noticed that the user also has the possibility to run the problem at hand many times to assess the accuracy of the estimate produced.

The algorithm can also be modified to compute S-estimates. Although there is still room for improvements in the algorithmic design, the present code works satisfactory on the test problems generated.

Acknowledgement. We are grateful to Stefan Van Aelst for providing his Matlab code of the algorithm SURREAL.

Appendix

The CM-estimation problem is to find the global minimum of

$$L(\beta, \sigma) = ave\{\rho_c(r_i/\sigma)\} + \log(\sigma)$$

over $\beta \in R^p$ and $\sigma > 0$ subject to the constraint

$$ave\{\rho_c(r_i/\sigma)\} \leq \varepsilon\rho_c(\infty). \tag{17}$$

Here $\rho_c(t)$ is a bounded, nondecreasing function of $t \geq 0$ with tuning parameter $c > 0$. To find the S-estimate, we minimize L with respect to σ, implying equality in the constraint (17).

From the definition of the S- and CM-estimates, it is apparent that they will in many cases coincide. This will depend on the positive tuning parameter c of the ρ function used. We derive an interval for c when this happens, for the case that ρ is the Tukey biweight function. We have

$$\rho_c(t) = \begin{cases} \dfrac{t^2}{2} - \dfrac{t^4}{2c^2} + \dfrac{t^6}{6c^4} & \text{if } |t| \leq c \\ \dfrac{c^2}{6} & \text{if } |t| > c \end{cases}$$

and

$$\rho_c(t) = c^2\rho\left(\frac{t}{c}\right)$$

with

$$\rho(t) = \begin{cases} \dfrac{t^2}{2} - \dfrac{t^4}{2} + \dfrac{t^6}{6} & \text{if } |t| \leq 1 \\ \dfrac{1}{6} & \text{if } |t| > 1 \end{cases}$$

Using this notation, the CM-estimate is the minimum of

$$L(\beta, \sigma) = c^2\, ave\left\{\rho\left(\frac{r_i}{c\sigma}\right)\right\} + \log(\sigma)$$

We have

$$\frac{\partial L(\beta, \sigma)}{\partial \sigma} = c^2\, ave\left\{\rho'\left(\frac{r_i}{c\sigma}\right)\left(-\frac{r_i}{c\sigma^2}\right)\right\} + \frac{1}{\sigma} = \frac{c^2}{\sigma}\left[\frac{1}{c^2} - ave\left\{\rho'\left(\frac{r_i}{c\sigma}\right)\frac{r_i}{c\sigma}\right\}\right]$$

Now consider the function $g(t) = \rho'(t)t$. For the Tukey ρ function we get

$$g(t) = \begin{cases} t^2(1 - t^2)^2 & \text{if } |t| \leq 1 \\ 0 & \text{if } |t| > 1 \end{cases}$$

and

$$g'(t) = \begin{cases} t(1-t^2)(1-3t^2) & \text{if } |t| \le 1 \\ 0 & \text{if } |t| > 1 \end{cases}$$

Thus $g(t)$ has its maximum $\frac{4}{27}$ for $t = \pm\frac{1}{\sqrt{3}}$. This implies that $\frac{\partial L(\beta,\sigma)}{\partial \sigma} > 0$ if $\frac{1}{c^2} > \frac{4}{27}$, i.e. if $c < \frac{3\sqrt{3}}{2} \approx 2.598$. For these c-values, σ should be chosen as small as possible. The left hand side of (17) increases when σ gets smaller. This means that σ is limited by the equality case in the constraint, i.e. we have the S-estimate. Thus, for $c < 2.598$ the S- and CM-estimates will always coincide.

References

Arslan O, Edlund O, Ekblom H (2001) When do S- and CM-estimates for regression coincide? Proceedings of 53rd session of the International Statistical Institute

Donoho DL, Huber PJ (1983) The notion of breakdown point. In: Bickel PJ, Doksum KA, Hodges Jr JL (eds.), Festschrift for Lehmann EL. Wadsworth, Belmont, California, pp. 157–184

Edlund O (1997) Linear M-estimation with bounded variables. BIT 37:13–23

Hampel FR, Ronchetti EM, Rousseeuw PJ, Stahel WA (1986) Robust statistics: The approach based on influence function. Wiley, New York

Huber PJ (1973) Robust regression: Asymptotics, conjectures and Monte Carlo. The Annals of Statistics 1:799–821

Huber PJ (1981) Robust statistics. Wiley, New York

Kent JT, Tyler DE (1996) Constrained M-estimation for multivariate location and scatter. The Annals of Statistics 24:1346–1370

Louhaä HP (1989) On the relationship between S-estimators and M-estimators of multivariate location and covariance. The Annals of Statistics 17:1662–1684

Maronna RA, Bustos OH, Yohai VJ (1979) Bias and efficiency robustness of generalized M estimators for regression with random carriers. In: Gasser T, Rosenblatt M (eds.) Smoothing techniques for curve estimation. Springer, New York

Mendes B (1995) Constrained M estimation for linear regression models. Unpublished Ph.D. Thesis. Rutgers University, Statistics Department, NJ, USA

Mendes B, Tyler DE (1995) Constrained M estimates for regression. In: Robust Statistics; Data Analysis and Computer Intensive Methods, Lecture Notes in Statistics 109. Springer, New York, pp. 299–320

Rousseeuw PJ (1984) Least median of squares regression. J. Amer. Statist. Assoc. 79:871–880

Rousseeuw PJ, Yohai VJ (1984) Robust regression by means of S-estimators. In: Frank J, Härdle W, Martin RD (eds.) Robust and Nonlinear Time Series Analysis, (Lecture Notes in Statistics). Springer-Verlag, New York, pp. 256–272

Rousseeuw PJ, Leroy AM (1987) Robust regression and outlier detection. Wiley, New York

Ruppert D (1992) Computing S estimators for regression and multivariate location/dispersion. Journal of Computational and Graphical Statistics 1:253–270

Siegel AF (1982) Robust regression using repeated median. Biometrika 69:242–244

Simpson DG, Ruppert D, Carroll RJ (1992) On one-step GM-estimates and stability of inferences in linear regression. J. Amer. Statist. Assoc. 87:439–450

Törn AA, Zilinskas A (1989) Global optimization, Lecture Notes in Computer Science 350. Springer-Verlag, Berlin, 255 pp