

**SOME JACK-KNIFING RESULTS FOR  
REGRESSION WITH NON-HOMOGENEOUS  
LOSS FUNCTIONS**

**Peter Verboon  
Willem J. Heiser**

**Department of Data Theory  
University of Leiden**

# Some Jack-knifing results for regression with non-homogeneous loss functions

Peter Verboon

Willem J. Heiser

## Abstract

*To study the properties of some robust loss functions a jack-knife experiment with generated data has been performed including four different functions. Special attention has been paid to the tuning constants. These constants were found to be of major importance for the performance of the applied loss functions.*

*Key words: robustness, Jack-knife, regression*

## 1. Introduction

In robust statistics one studies the behavior of a method under a broad range of circumstances. The technical term "robust" is usually accompanied by the term "resistant". From Andrews (1974) we quote: "Techniques of fitting are said to be resistant when the result is not greatly altered in the case a small fraction of the data is altered; techniques of fitting are said to be robust of efficiency when their statistical efficiency remains high for conditions more realistic than the utopian cases of Gaussian distributions with errors of equal variance. These properties are particularly important in the formative stages of model building when the form of the response is not known exactly".

Loss functions that are designed to make the associated data analysis technique more robust and resistant frequently involve one or more so-called tuning constants (Hoaglin et. al., 1983, p.346). These constants determine the shape of the loss function in such a way that the solution may critically depend on them. Therefore it is of interest to study systematic ways for choosing these constants (cf. Holland & Welsch, 1977).

Most loss functions involving such tuning constants are *non-homogeneous*. A homogeneous function  $S$  of the residuals  $r_i, i = 1, \dots, n$  satisfies

$$S(\alpha r_i) = \alpha S(r_i) \quad \text{with } \alpha \geq 0 \quad (1)$$

for the entire range of its argument  $r$ . This characteristic naturally leads to the property

$$\alpha \operatorname{argmin} S(r_i) = \operatorname{argmin} S(\alpha r_i), \quad (2)$$

which implies that it doesn't matter how the residuals are scaled, since it will always be possible to rescale the solution afterwards. This convenient property is no longer valid for non-homogeneous loss functions.

In this paper three non-homogeneous loss functions are discussed: the Huber function and two so-called redescending functions. The term redescender points to the first derivative of such functions. We distinguish a hard redescender for which the derivative of the function becomes exactly zero beyond a finite bound and a soft redescender for which the derivative merely becomes asymptotically zero. These functions will be compared to the least squares and the least absolute residuals loss functions, which are the building blocks of the former ones, and which are homogeneous themselves. Our approach will be to determine the stability of parameter estimates under Jack-knifing. In a Monte Carlo experiment it turns out that this procedure yields reasonable choices.

## 2. Description of the loss functions

In order to study the qualities of these loss functions, consider the well-known linear regression problem with the following model

$$\mathbf{y} = \mathbf{X}\mathbf{b}, \quad (3)$$

with  $\mathbf{y}$  as the dependent variable and  $\mathbf{X}$  the matrix with independent variables, which are assumed to be fixed. The objective is to find the parameter vector  $\mathbf{b}$  that minimizes some function of the residual vector  $\mathbf{r}$ , which is defined as

$$\mathbf{r} = \mathbf{y} - \mathbf{X}\mathbf{b}. \quad (4)$$

The most common way to tackle this problem is based on Least Squares (LS), which minimizes the following loss function:

$$S_{LS}(r_i) = \sum_i r_i^2. \quad (5)$$

The minimum of this function can directly be found by computing  $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ . In the following sections we will describe four alternative loss functions. These functions supposedly are more robust and resistant, which means that they are suited for data with long tailed distributions or when outliers are involved.

*Least Absolute Residuals loss function*

The first alternative loss function is based on the absolute values of  $r$ . This Least Absolute Residuals loss function (LAR) is written as

$$S_{\text{LAR}}(r_i) = \sum_i |r_i| \quad (6)$$

Now suppose we have any known set of residuals  $r_i$ , associated with some previous solution for  $\mathbf{b}$ . As shown in Heiser(1986) the function  $S_{\text{LAR}}$  is majorized by a family of quadratic functions in the sense that we always have

$$S_{\text{LAR}}(r_i) \leq 1/2 S_{\text{LAR}}(r_i) + 1/2 \sum_i \underline{w}_i r_i^2. \quad (7)$$

The  $\underline{w}_i$  are variable weights defined as

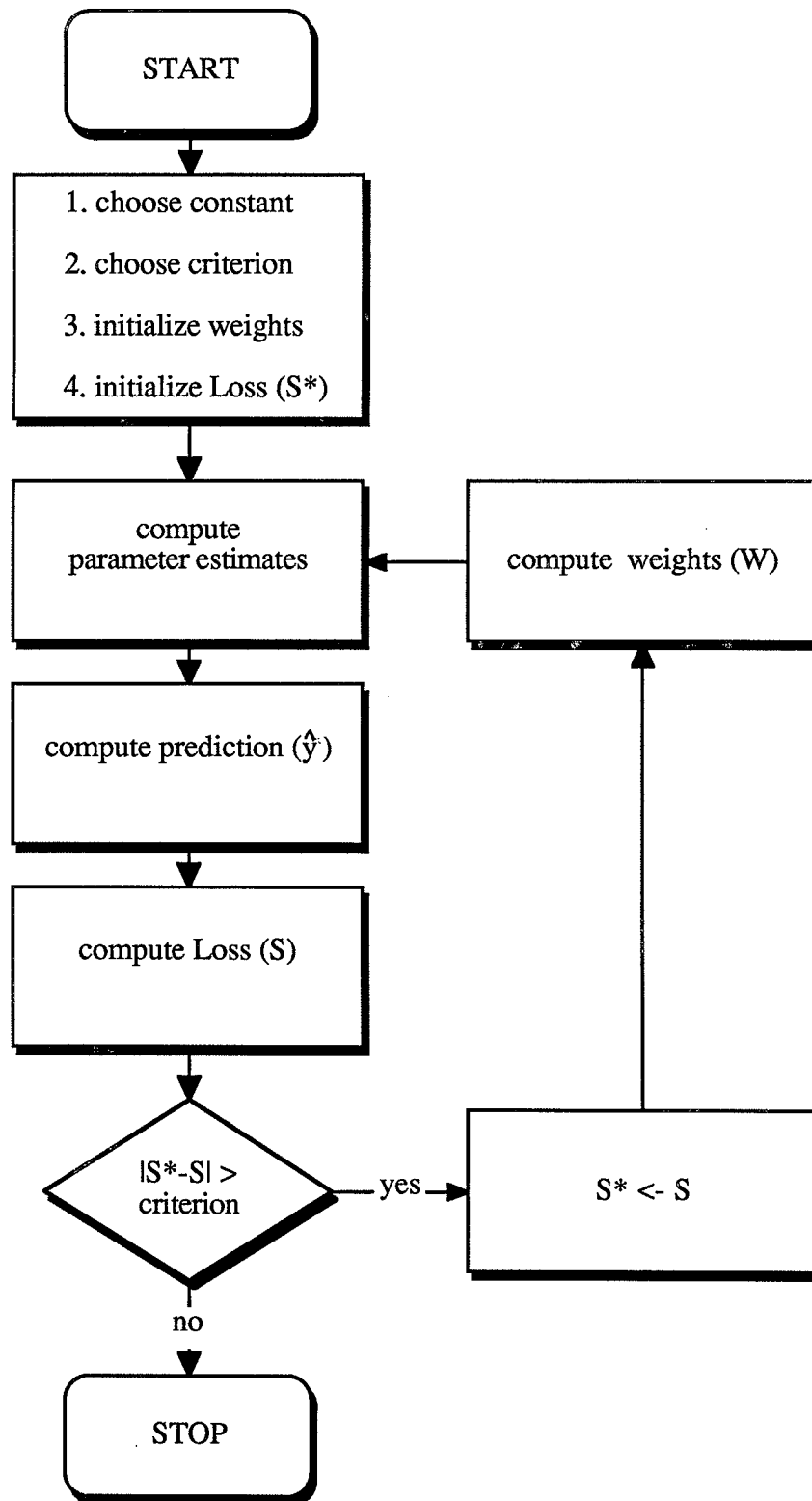
$$\underline{w}_i = 1 / |r_i| \quad \text{if } |r_i| > \epsilon, \quad (8a)$$

$$\underline{w}_i = 1 / \epsilon \quad \text{if } |r_i| \leq \epsilon, \quad (8b)$$

where  $\epsilon$  is some very small constant that is needed to make the algorithm converge smoothly (see Heiser 1987a).

The majorization algorithm to minimize this function can actually be looked upon as a reweighted least squares algorithm. The scheme for such an algorithm is displayed in Figure 1, with the parameter vector  $\mathbf{b}$  computed as  $\mathbf{b} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}$  and  $\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}$ . The matrix  $\mathbf{W}$  is a diagonal matrix with weights  $\underline{w}_i$ , for the  $S_{\text{LAR}}$  defined in (8a,b).

The LAR loss function has a long history, but has not been applied too often. Heiser (1987b) formulated a LAR version for Correspondence Analysis. More applications and theoretical considerations about LAR can be found in Dodge (1987).



**Figure 1.** Flow chart of reweighted least squares algorithm for non-homogeneous loss functions.

### *Huber's Loss Function*

From 8a it is seen that the variable weight for point  $i$  becomes very large when the residual  $r_i$  is very small. To cope with this problem Huber(1973) introduced another loss function, which is a mixture of least squares and least absolute residuals. The function is actually a family of loss functions depending on the tuning constant  $c$ . For some constant  $c$  the Huber loss function is defined as:

$$S_{HL}(r_i) = \sum_i r_i^2 \quad \text{for } |r_i| < c \quad (9a)$$

$$S_{HL}(r_i) = \sum_i [2c|r_i| - c^2] \quad \text{for } |r_i| \geq c \quad (9b)$$

Note that for  $c \rightarrow 0$  we obtain LAR and for  $c \rightarrow \infty$  we have LS (cf. Ekblom, 1987). The idea behind Huber's loss function (HL) is that small residuals (smaller than  $c$ ) will relatively contribute more to the loss than large residuals. This will diminish the disturbing effects of outliers to some extent. As shown in Heiser(1987a), we can still minimize this function by using a majorization function defined as

$$\kappa(r_i) = \sum_i r_i^2 \quad \text{for } |r_i| < c \quad (10a)$$

$$\kappa(r_i) = \sum_i [(c/|r_i|) r_i^2 + c|r_i| - c^2] \quad \text{for } |r_i| \geq c \quad (10b)$$

Again we will consider this as a reweighted least squares problem with variable weights computed as

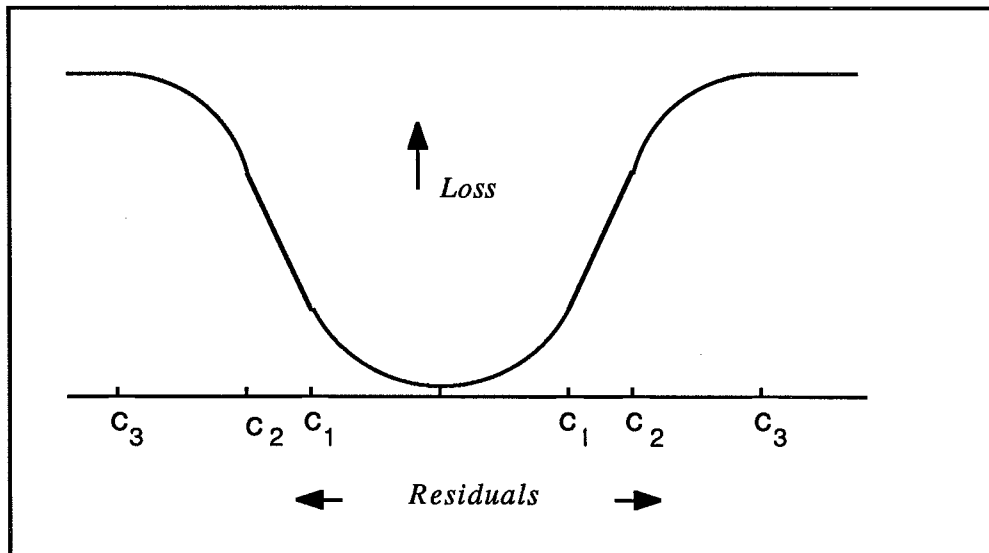
$$\underline{w}_i = 1 \quad \text{for } |r_i| < c. \quad (11a)$$

$$\underline{w}_i = c / |r_i| \quad \text{for } |r_i| \geq c \quad (11b)$$

Note that  $0 < \underline{w}_i \leq 1$  and since  $c/|r_i|$  is always smaller than 1, the procedure amounts to degrading the weights when the residuals are large. Contrary to the straightforward reweighted approach to LAR fitting the weights now have an upper bound of 1, so they cannot become unreasonably large.

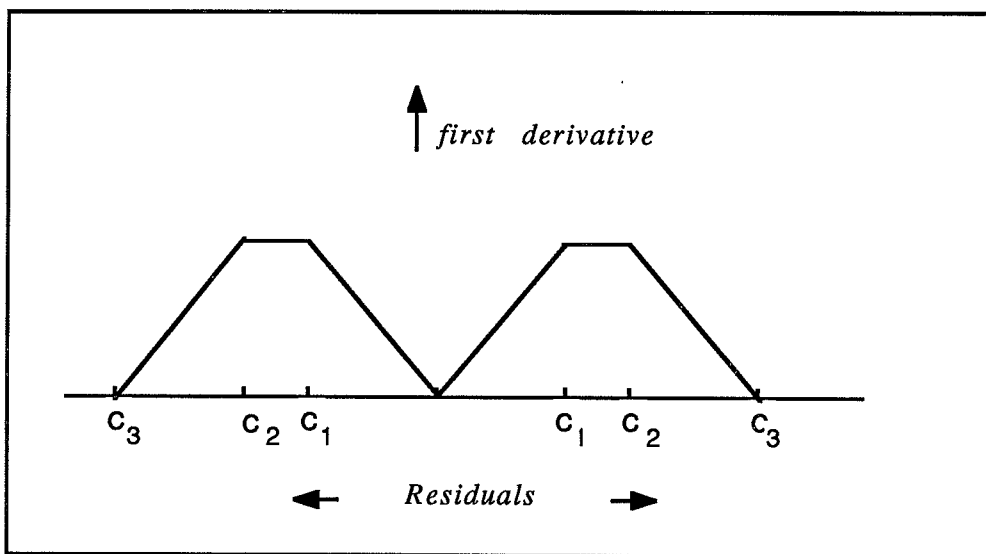
### *A Hard Redescending Loss Function*

The effect of using a hard redescending loss function (HR) is shown in Figure 2a. For small residuals the function shown is quadratic. When the residuals exceed some constant  $c_1$  the function becomes linear. So far we are dealing with Huber's function. When the residuals also exceed a constant  $c_2$  the function obtains, besides a linear part, a quadratic part with a negative second derivative. Finally for residuals exceeding a constant  $c_3$  the function becomes a constant function.



**Figure 2a.** Example of hard redescending loss function.

The Figure clearly shows that for  $r_i > c_3$  there is no further increase in loss. So large residuals will be treated like moderately large ones. This makes the HR more radical than HL. In Figure 2b the first derivative of the function is displayed. It clearly shows the four parts of the function and that the derivative becomes zero for large residuals.



**Figure 2b.** First derivative of hard redescender.

In the first column of Table 1 a redescending function is described with equally spaced intervals for the different function parts. The second column gives the majorization function. A similar function has been described by Hampel (1980).

Table 1. Class of hard redescenders.

$S_{HR}(r_i)$	$\kappa(r_i)$	interval
$\sum_i r_i^2$	$\sum_i r_i^2$	$ r_i  < c$
$\sum_i [2c  r_i  - c^2]$	$\sum_i [(c/ r_i ) r_i^2 + c  r_i  - c^2]$	$c \leq  r_i  < 2c$
$\sum_i [6c r_i  - r_i^2 - 5c^2]$	$\sum_i [(3c/ r_i ) r_i^2 - r_i^2 + 3c  r_i  - 5c^2]$	$2c \leq  r_i  < 3c$
$4\sum_i c^2$	$4\sum_i c^2$	$3c \leq  r_i $

The variable weights for  $|r_i| < 2c$  are similar to HL. For the other parts of the function the weights are defined as

$$\underline{w}_i = (3c - |r_i|) / |r_i| \quad \text{for } 2c \leq |r_i| < 3c, \quad (12a)$$

$$\underline{w}_i = 0 \quad \text{for } |r_i| \geq 3c. \quad (12b)$$

Now we have variable weights which are 1 for the positive quadratic part, between 1 and .5 in the linear part, between .5 and 0 in the quadratic part with negative second derivative and finally which are 0 in the constant part of the function.

In estimating the location of a distribution a well-known resistant estimator is the trimmed mean. The trimmed mean can also be seen as a special kind of a hard redescender. The large values of a distribution obtain weights 0 and the other values weights 1.

#### *A Soft Redescending Loss Function*

As an example of a soft redescender (SR) we will take Eilers' adaptive weights approach (Eilers, 1987). Here the loss function is a function of both the residuals and the weights:

$$S_{SR}(w_i, r_i) = \sum_i w_i^2 r_i^2 + k^2 \sum_i (1 - w_i)^2. \quad (13)$$

In this function two parts can be distinguished. The second part of the function prevents a trivial solution in which all weights are zero. The function yields small weights for large residuals and weights near 1 for small residuals. Again we use reweighted least squares to minimize this function. The variable weights are

$$\underline{w}_i = (1 + r_i^2 / k^2)^{-1}. \quad (14)$$

The  $k$  is a prefixed constant. Notice that for large values of  $k$  the function will make all weights to be approximately equal to 1, which reduces the function to LS. For very small  $k$



values the weights will approximately become 0. It's evident that this is not what we want, therefore we introduce the restriction  $k \geq 1$ .

### 3. Description of the data

To study the robustness of the different loss functions, a dataset was generated by taking some vector of x-coordinates on the interval [1,40]. The dependent variable y was computed according to  $y = x + e$ , with e uncorrelated with x and  $\sim U(0, .05\sigma_y^2)$ . Some y-coordinates (10%) were computed according to  $y = -x + 30 + e$  and another 10% according to  $y = .5x + e$ . A scatterplot of the data is displayed in Figure 3.

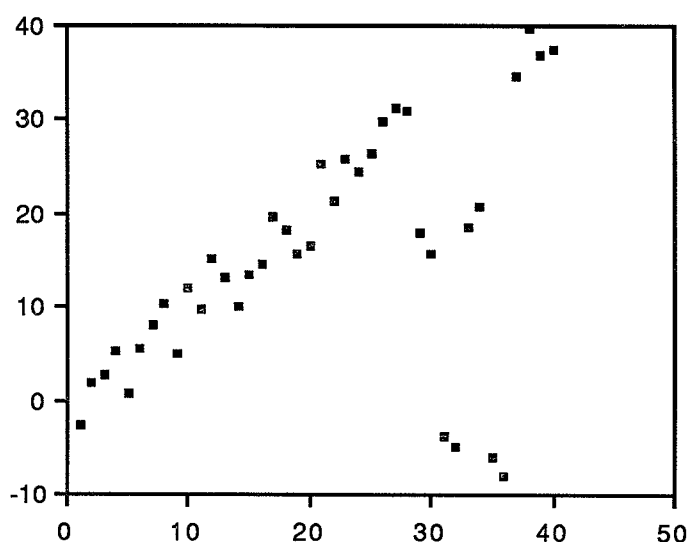


Figure 3. Scatterplot of simulated data.

### 4. Parameter estimates

The data have been analyzed by each loss function. In the first place we are interested in the ability of each function to find the slope parameter of the largest part of the data, which has value 1. Besides the slope the intercept has been simultaneously computed, but this is of no further interest. The slope estimates of LS and LAR are .526 and .935 respectively. The results for the other functions are shown in Table 2.

Table 2. Effects of the constant c and k on slope estimates

c/k	.25	1	5	10	20	40
HR	.466	.349	1.005	.892	.629	.526
HL	.934	.915	.858	.756	.629	.526
SR		.837	.960	.892	.763	.625

If there were no effect of outliers then the parameter estimate should be around 1.0. From Table 2 it can be seen that the HR closely approximates this value for values of c that are around c = 5. With c = 5 the diagonal weights matrix of HR has values 0 for the extreme and less extreme outliers. For large c values all weights are equal to 1 and the results become like LS, as it should. With small c values many weights become 0 and the slope depends on just a few points.

The second function, HL, runs from the LAR results for small c values to the LS results for large ones.

The SR has an optimal result for k = 5. For larger k the results will also go to the LS ones but this process will be slower than for the other functions.

Table 2 has given us an indication about the range of the tuning constant within which the optimal slope values are estimated. The next step is to study the stability of the estimates for some values of the tuning constant.

## 5. Determination of stability by Jack-knifing

In the present example the optimal solution was known and therefore we could easily decide upon which technique and constant showed the best result, but of course usually there is no such external validation criterion present.

For this reason a jack-knife study has been done to test the stability of the loss function for different values of c and k respectively. The idea is that for "too large" values of the tuning constant the functions behave like least squares, which is known to be non-robust in the presence of outliers. The spread of parameter estimates under Jack-knifing will then become large. On the other hand, for "too small" values of the tuning constant the functions essentially assign uniform loss to the majority of the residuals, so that their behavior may become erratic under dropping parts of data. This erratic behavior will also be reflected in a large spread of the parameter estimates. The data were analyzed forty times, each time leaving out another point.

One way to define spread is as the standard deviation of the estimated parameter,

$$\sigma_b = (\sum_i (b_{(i)} - \bar{b})^2 / n)^{1/2}, \quad (15)$$

where the summation is over the  $n$  points. The notation  $b_{(i)}$  indicates the estimate of the slope parameter when point  $i$  has been left out. A technique will be stable if its slope parameter is relatively little influenced by individual points. The standard deviations of the redescending and Huber's estimates for different values of  $k$  and  $c$  are shown in Table 3. Notice that the constants for both techniques have different meanings. The standard deviations for the Least Squares approach is .029 and for the Least Absolute Residuals it is .002.

Table 3. Standard deviations of HR, HL and SR for different constants

	1	5	10	20	40
HR	.206	.010	.013	.027	.029
HL	.005	.017	.017	.027	.029
SR	.116	.008	.013	.020	.027

Table 3 shows the instability of the redescenders for small constants. These functions have the least spread when they compute the optimal solution. Consistent with former results HL tends towards LAR for small  $c$  values.

Another statistic to measure spread is the Median Absolute Deviation (MAD), defined as

$$\text{MAD} = \text{med}\{|b_{(i)} - \text{med}\{b_{(i)}\}|\}. \quad (16)$$

The MAD first requires the computation of the median of the estimates  $b_{(i)}$ . Next the  $b_{(i)}$  are taken in deviance of this median and finally the median of these values is called the MAD. Contrary to the spread the MAD is highly resistant to outlying observations, i.e. in this case the outlying parameter estimates. The MAD for LS and LAR are respectively .00612 and .00088. Table 4 shows the other results. For esthetic reasons all values in the Table are multiplied by 100.

Table 4. MAD of HR, HL and SR for different constants

	1	5	10	20	40
HR	.001	.441	.436	.835	.612
HL	.349	.805	.663	.835	.612
SR	2.553	.491	.730	.775	.871

Comparing the MAD and the  $\sigma_b$  we notice the similar patterns of both measures for the SR. Both measures show a clear minimum dispersion for  $k=5$ . For the other techniques the patterns are not the same. The apparent conflicting results for the HR with  $c=1$  is probably due to the

fact that the estimates are rather stable except when one of the outliers is left out. Leaving out an outlier causes the slope estimate to become highly deviant from the others. The MAD however is resistant to these "outliers".

Because of its property of resistance the MAD isn't completely suited for a measure of dispersion of parameter estimates, since the interest is actually in outlying parameter estimates. In addition to the  $\sigma_b$  however the MAD gives insight into the distribution of the slope estimates.

For instance if a particular technique with tuning constant  $c$  comes up with both a large MAD and a large  $\sigma_b$ , the technique could be resistant after all. In this case the technique is not able to provide a proper fit to the data, which is obviously something quite different than being non-resistant.

In general when both measures yield low values, it is still not possible to draw inferences about the resistance of the technique. Such a result could only tell us that there are no outliers in the data. In this study however, since we know that there are outliers, low values for both of the spread measures indeed indicate resistance of a technique for a given constant.

## 6. Economical aspects

Another interesting aspect of robust algorithms is the cpu-time needed to reach the solution. For HR, HL and SR the values in Table 5 are the average over 30 analyses with different tuning constants. Besides the cpu-time the mean number of iterations to reach convergence has been given.

Table 5. Mean cpu-time and number of iterations

	cpu	iterations
LS	.009	1
LAR	.236	42
HR	.098	10.0
HL	.065	9.1
SR	.396	17.9

The low value of LS is obvious, because LS is computed in one step only. Furthermore we see that LAR and SR need considerably more time than HR and HL. The long cpu time for LAR is caused by the large number of iterations it takes to reach convergence. The SR on the contrary needs more computing time within each separate step.

## 7. Discussion

First of all it should be noticed that the robust functions from this study indeed appear to be resistant to outliers if a proper choice can be made for the tuning constant. It has been shown that a Jack-knife experiment can be performed for different tuning constants and that a minimal spread of the estimated parameter yields a good choice of the tuning constant.

Furthermore it was seen that HR and HL need relatively few cpu-time, so these are suited for routine applications.

Further research will be necessary to study whether a stability experiment to determine the tuning constant can be applied to other techniques as well. In particular we are planning to look at Procrustes rotation, i.e. rotation of a configuration of points towards a fixed target.

## References

- Andrews, D.F. (1974). A robust method for multiple linear regression. *Technometrics*, 16, 523-531.
- Dodge, Y (ed.). (1987). *Statistical Data Analysis*. New York: North-Holland.
- Eilers, P.H.C. (1987). Adaptieve gewichten, een exploratieve techniek voor uitbijters en mengsels van regressie modellen. *Kwantitatieve Methoden*, 23, 63-83.
- Ekblom, H. (1987). The  $L_1$ -estimate as limiting case of an  $L_p$ - or Huber-estimate. In: Y. Dodge(ed.). *Statistical Data Analysis*. New York: Wiley.
- Hampel, F.R. (1980). Optimally bounding the gross-error sensitivity and the influence of position in factor space. *Proc. Statist. Comput. Sect.*, Amer. Statist. Assoc., 59-64.
- Heiser, W.J. (1986). *A majorization algorithm for the reciprocal location problem*. Internal Report RR-86-12, Department of Data Theory, University of Leiden
- Heiser, W.J. (1987a). *Notes on the LARAMP Algorithm*. Internal Report RR-87-04, Department of Data Theory, University of Leiden.
- Heiser, W.J. (1987b). Correspondence analysis with least absolute residuals. *Comp. Statist. and Data Analysis*, 5(4), 337-356.
- Hoaglin, C.H., Mosteller, F., Tukey, J.W. (1983). *Understanding robust and exploratory data analysis*. New York, Wiley.
- Holland, P.W. and Welsch, R.E. (1977). Robust regression using iteratively reweighted least-squares. *Commun. statist.-theor. meth.*, A6(9), 813-827.
- Huber, P.J. (1973). Robust regression: Asymptotics, conjectures and Monte Carlo. *Ann. Statist.*, 1, 799-821.