KM 33(1990)
pag 161-173

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,

*Department of Data Theory
University of Leiden
Wassenaarseweg 52
2333 AK Leiden
tel. 273827/ 273828

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

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

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

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

where "argmin" denotes the argument for which the minimum of the function is attained. This 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 This convenient property is no longer valid for non-homogeneous loss functions.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.

The estimators based on the Huber function and on a hard redescender are often called M-estimators in the robust literature. (Hampel et. al., 1986). We have chosen for these functions because they are among the best known in the robust tradition. The soft redescender is included in our study as an example of a rather new approach.

These three 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 and to examine whether an optimal value for the tuning constant can be found on the basis of this stability. In a Monte Carlo experiment it turns out that this procedure yields reasonable choices. Of course, these results can only be tentative due to the modest size of our experiment.

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

$$\mathbf{y} = \mathbf{X}\mathbf{b}, \tag{3}$$

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

$$\mathbf{r} = \mathbf{y} - \mathbf{Xb}. \qquad (4)$$

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

$$S_{LS}(r_i) = \Sigma_i \, r_i^2 \, . \qquad (5)$$

The minimum of this function can directly be found by computing $\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'y}$. 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. Loss functions based on this so called L_1-norm already have a long tradition in robust estimation. The Least Absolute Residuals loss function (LAR) is written as

$$S_{LAR}(r_i) = \Sigma_i \, |r_i| \qquad (6)$$

To minimize this function we use the majorization method. In short this means that we are going to approach (6) iteratively from above by means of a simple weighted quadratic function. Now suppose we have any known set of residuals $\underline{r}_i$, associated with some previous solution for $\mathbf{b}$. As shown in Heiser(1986) the function S_{LAR} is majorized by a family of quadratic functions in the sense that we always have

$$S_{LAR}(r_i) \leq 1/2 \; S_{LAR}(\underline{r}_i) + 1/2 \; \Sigma_i \, \underline{w}_i \, r_i,^2. \qquad (7)$$

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

$$\underline{w}_i = 1 \,/\, |\underline{r}_i| \qquad \text{if } |\underline{r}_i| > \varepsilon, \qquad (8a)$$

$$\underline{w}_i = 1 \,/\, \varepsilon \qquad \text{if } |\underline{r}_i| \leq \varepsilon, \qquad (8b)$$

where ε 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 parameter vector $\mathbf{b}$ is computed as :

$\mathbf{b} = (\mathbf{X'WX})^{-1}\mathbf{X'Wy}$ and $\hat{\mathbf{y}} = \mathbf{Xb}$. The matrix $\mathbf{W}$ is a diagonal matrix with weights $\underline{w}_i$, for the S_{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).

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:

$$S_{HL}(r_i) = \Sigma_i \, r_i^{\,2} \qquad \text{for } |r_i| < c \qquad (9a)$$

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

Note that for *c* -> 0 we obtain LAR and for *c* -> ∞ we have LS (cf. Ekblom, 1987). The idea behind the Huber function 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. In Figure 1 we show the first derivative of the Huber function, which is sometimes called the influence curve (Hampel et. al., 1986). From this Figure we learn that for the Huber function the influence of outliers is bounded

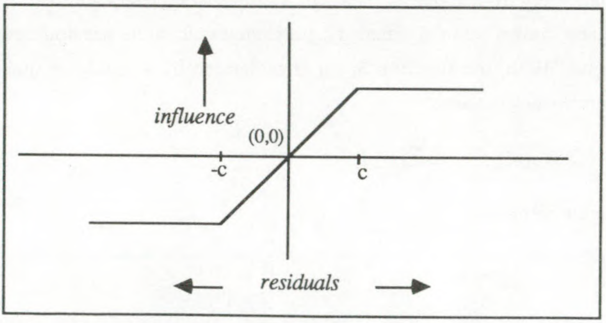

Figure 1. First derivative of the Huber function.

As shown in Heiser(1987a), we can still minimize this function by using a majorization function defined as

$$\kappa(r_i) = \Sigma_i \, r_i^{\,2} \qquad \text{for } |\underline{r}_i| < c \qquad (10a)$$

$$\kappa(r_i) = \Sigma_i \, [(c/|\underline{r}_i|) \, r_i^{\,2} + c|\underline{r}_i| - c^2] \qquad \text{for } |\underline{r}_i| \geq c \qquad (10b)$$

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

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

$$\underline{w}_i = c \,/\, |\underline{r}_i| \qquad \text{for } |\underline{r}_i| \geq c \qquad (11b)$$

Note that $0 < \underline{w}_i \leq 1$ and since $c/|\underline{r}_i|$ 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 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 the Huber 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.

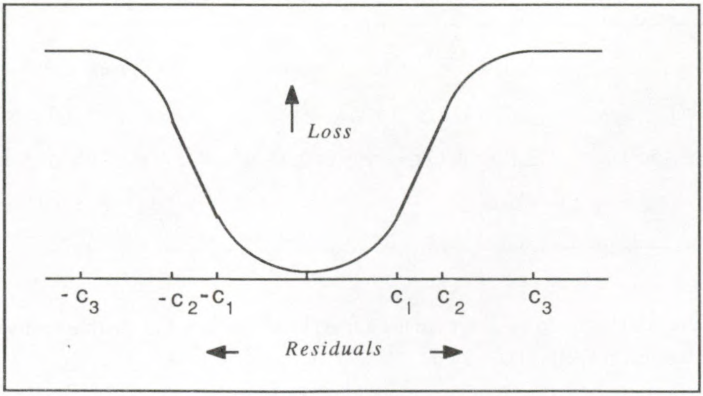

Figure 2a. Example of a hard redescending loss function.

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 hard redescender more radical than the Huber function. 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. So we see that the hard redescender also has a bounded influence curve.

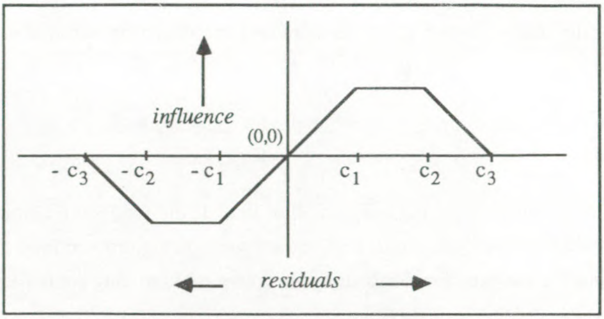

Figure 2b. First derivative of a hard redescender.

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.

loss function	majorization function	interval
$\Sigma_i \, r_i{}^2$	$\Sigma_i \, r_i{}^2$	$\lvert\underline{r}_i\rvert < c$
$\Sigma_i \, [2c \lvert r_i\rvert - c^2]$	$\Sigma_i \, [(c/\lvert\underline{r}_i\rvert) \, r_i{}^2 + c \lvert\underline{r}_i\rvert - c^2]$	$c \leq \lvert\underline{r}_i\rvert < 2c$
$\Sigma_i \, [6c\lvert r_i\rvert - r_i^2 - 5c^2]$	$\Sigma_i[(3c/\lvert\underline{r}_i\rvert) \, r_i^2 - r_i^2 + 3c \lvert\underline{r}_i\rvert - 5c^2]$	$2c \leq \lvert\underline{r}_i\rvert < 3c$
$4\Sigma_i \, c^2$	$4\Sigma_i \, c^2$	$3c \leq \lvert\underline{r}_i\rvert$

The variable weights for $\lvert\underline{r}_i\rvert < 2c$ are similar to the Huber function. For the other parts of the function the weights are defined as

$$\underline{w}_i = (3c - \lvert\underline{r}_i\rvert) / \lvert\underline{r}_i\rvert \qquad \text{for } 2c \leq \lvert\underline{r}_i\rvert < 3c, \qquad (12a)$$

$$\underline{w}_i = 0 \qquad \text{for } \lvert\underline{r}_i\rvert \geq 3c. \qquad (12b)$$

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 robust estimator is the trimmed mean. The trimmed mean can also be seen as a special kind of hard redescender. The large values of a distribution obtain weights 0 and the other values weights 1. The hard redescender, however, rejects more smoothly.

A Soft Redescending Loss Function

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

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

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

$$\underline{w}_i = (1 + \underline{r}_i^2 \, / \, k^2 \,)^{-1}. \tag{14}$$

The k is a prefixed constant, which we will call the penalty constant. Notice that for large values of the penalty constant 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. Especially the first 10% are really heavy outliers. Various studies have shown, however, that the percentage of outliers in routine data can easily be up to 10% and sometimes even more. So the choice of 10% heavy outliers and 10% small outliers may not be too unrealistic.

The outliers were not randomly divided over the x-values, but chosen in such a way as to make their influence large. Furthermore the y is considered a random variable, but the x is considered fixed. This implies that we are interested in outliers in the y-direction and not in so called leverage points (outliers in the x-direction). A scatterplot of the data is displayed in Figure 3.

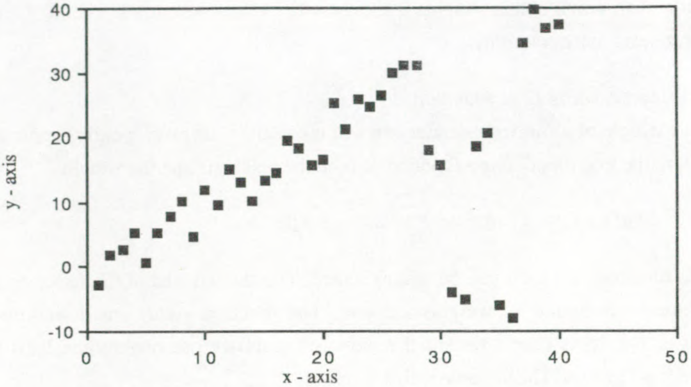

Figure 3. Scatterplot of simulated data.

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 results are shown in Table 2.The slope estimates of LS and LAR don't depend on any constant and are shown below the dotted line in Table 2. The hard and soft redescenders are respectively denoted as HR and SR, the Huber function as HF.

Table 2. Effects of the tuning and penalty constants on the slope estimates

	c or *k*					
function	.25	1	5	10	20	40
HR	.466	.349	1.005	.892	.629	.526
HF	.934	.915	.858	.756	.629	.526
SR		.837	.960	.892	.763	.625
LS	.526					
LAR	.935					

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 the tuning constant 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, HF, runs from the LAR results for small tuning constants to the LS results for large ones.

The SR has an optimal result for $k = 5$. For larger penalty values the results will also go to the LS ones but this proces 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 extern 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 the tuning and penalty constants 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,

$$\sigma_{b} = (\Sigma_{i}\,(b_{(i)} - \bar{b})^{2} / n\,)^{1/2}, \tag{15}$$

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.

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

	c or *k*				
function	1	5	10	20	40
HR	.206	.010	.013	.027	.029
HF	.005	.017	.017	.027	.029
SR	.116	.008	.013	.020	.027
LS	.029				
LAR	.002				

Table 3 shows the instability of the redecenders for small constants. These functions have the least spread when they compute the optimal solution. Consistent with former results HF tends towards LAR for small tuning constants.

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

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

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. Table 4 shows the other results. For esthetic reasons all values in the Table are multiplied by 100.

Table 4. MAD of HR, HF and HR for different constants

	c or *k*				
function	1	5	10	20	40
HR	.001	.441	.436	.835	.612
HF	.349	.805	.663	.835	.612
SR	2.553	.491	.730	.775	.871
LS	.612				
LAR	.088				

Comparing the MAD and the σ_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 σ_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 σ_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, HF 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
HF	.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 HF. 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 robust to outliers if a proper choice can be made for the tuning constant. This finding is consistent with most literature on robust methods. From this example and others that we did not present here, it seemed that after Jack-knifing for a series of different tuning values, the variances of the estimates (in combination with the MAD) can be used to determine an optimal value for the tuning constant. More specific: a minimal spread of the estimated parameter yields a good choice of the tuning constant, in the sense that for this tuning constant the best estimation of the parameter is found.

It would be necessary to study whether these findings are consistent over a wide variety of data and whether they could be applied in more practical situations. For instance if one wants to analyze a data set with the Huber function, a jack-knife could be done for different tuning constants, which would yield something like Figure 4.

Now it can immediately be seen for which tuning constant the results are optimal. Plots like Figure 4 would indeed be a helpful tool in robust estimation.

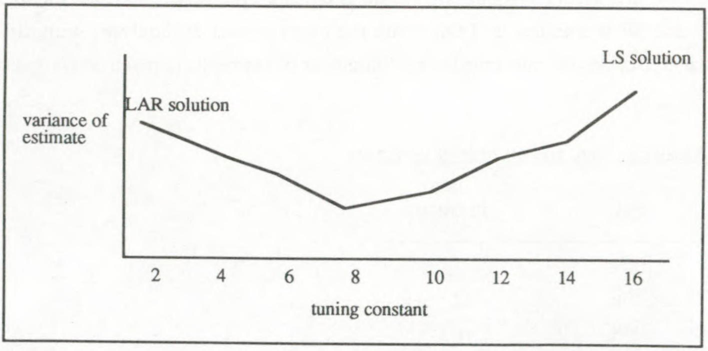

Figure 4. Possible result of jack-knife study for the Huber function.

Further research will be necessary to study whether a stability experiment to determine the optimal tuning constant can be applied to other techniques as well. In particular we are planning to look at some techniques for multivariate analysis. We already found some promising results for the orthogonal Procrustes problem, i.e. the 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. Stastist. Assoc., 59-64.

Hampel, F.R., Ronchetti, G.M., Rousseeuw, P.J., Stahel, W.A. (1986). *Robust Statistics: The approach based on influence functions*. New York: Wiley.

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.

Ontvangen: 08-12-1988
Geaccepteerd: 26-10-1989