

**ROBUST LOSS FUNCTIONS IN THE  
ORTHOGONAL PROCRUSTES PROBLEM**

**Peter Verboon**

**Willem J. Heiser**

**Department of Data Theory**

**University of Leiden**

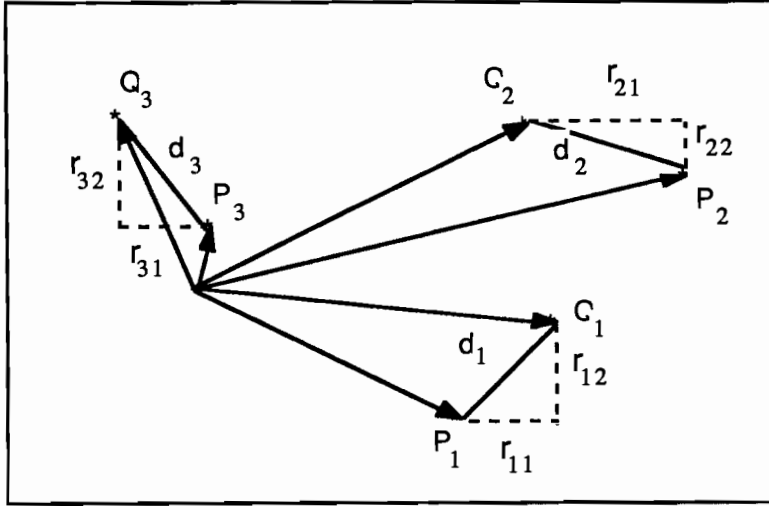


Figure 1. A rotation problem with three points, showing two types of residuals.

The quantities  $r_{ij}$  are defined as

$$r_{ij} = |q_{ij} - \sum_k p_{ik} h_{kj}|. \quad (2)$$

The two equivalent alternatives for the LS loss are :

$$\sigma_1(\mathbf{H}) = \sum_i \sum_j r_{ij}^2 \quad (3a)$$

$$\sigma_2(\mathbf{H}) = \sum_i d_i^2, \quad (3b)$$

where  $d_i$  is defined as

$$d_i = (\sum_j r_{ij}^2)^{1/2} \quad (4)$$

In this paper we will discuss three robust alternatives for the LS approach formulated in (1). Furthermore for each of these robust loss functions both types of residuals which define the loss criterion as shown in Figure 1, will be discussed. We do so because the criteria (3a) and (3b) are not equivalent anymore for robust loss functions and will possibly yield different solutions with respect to the Procrustes problem.

### 3. Alternative loss functions with two distance criteria

The three loss functions which we will examine here are well known in the robust tradition. They are known as functions with bounded influence curves (Hampel et al., 1986), because they bound the influence of the outliers. This means that outliers may yield rather large residuals without letting the loss explode. So the contribution of these large residuals to the loss is limited.

First we will study the function that minimizes the absolute values of the residuals. Functions based on this so called  $L_1$  norm already have a long tradition in robustness. Next we study Huber's loss function, which is a combination of LS and the least absolute values of the residuals and finally a hard-redescender is examined for which the residuals are completely rejected in the loss function after a particular value. These functions have proved to be valuable in the robust estimation of location and in robust regression problems (e.g. Huber, 1981).

Remember the general Procrustes problem, which is the rotation of the points in configuration  $\mathbf{P}$  towards those of  $\mathbf{Q}$  such that "some kind of distance" between the corresponding points is minimized. In analogy to LS the two alternative ways of aggregation for the Least Absolute Residuals (LAR) loss is formulated as:

$$\sigma_1(\mathbf{H}) = \sum_i [ \sum_j |r_{ij}| ] \quad (5a)$$

$$\sigma_2(\mathbf{H}) = \sum_i d_i \quad (5b)$$

The distance that corresponds with the part of (5a) between brackets is called the *city-block distance*. Since the minimization of the sum of city-block distances is not the same as the minimization of the sum of Euclidean distances, we have to study them separately. However, it will be shown that the minimum Euclidean distance criterion is easy to solve, while the minimum city-block criterion yields more problems and different ways to solve them.

The general method that will be used to minimize these robust functions is majorization by quadratics. Practically this means that the algorithms which we propose, are based on iteratively reweighted least squares. However, for each function we have to find another quadratic function and different ways to compute the weights. We will first concentrate on the solutions for the LAR function. Subsequently these results can easily be extended to the Huber function and to the redescending function.

#### 4. Solutions for the least absolute residuals loss function

##### *Minimizing the Euclidean distance*

It has been shown by Heiser (1987a) that the general minimization of a distance function of the form given in (5b) can be minimized by majorization by quadratics. In short this means that the objective function can iteratively be approached from above by a weighted quadratic function. To minimize loss function (5b) we need the following majorization function

$$\kappa(\mathbf{H}) = \sum_i u_i d_i^2, \quad (6)$$

Now let's define  $\underline{d}_i$  as the distance computed after the former step in the iteration process, then the variable weights  $u_i$  can be computed as

$$u_i = 1/\underline{d}_i, \quad \text{for } \underline{d}_i \geq \epsilon \quad (7a)$$

$$u_i = 1/\epsilon, \quad \text{for } \underline{d}_i < \epsilon \quad (7b)$$

where  $\epsilon$  is some very small number that is needed for a proper minimization. The problem in (6) is actually a weighted Procrustes problem which can easily be seen when we write down (6) in matrix notation

$$\kappa(\mathbf{H}) = \text{tr} (\mathbf{Q} - \mathbf{P} \mathbf{H})' \mathbf{U} (\mathbf{Q} - \mathbf{P} \mathbf{H}), \quad (8)$$

where  $\mathbf{U}$  is diagonal matrix containing the weights. Now let  $\mathbf{U} = \mathbf{V}^2$ , then we can define the matrices  $\mathbf{Q}^* = \mathbf{V} \mathbf{Q}$  and  $\mathbf{P}^* = \mathbf{V} \mathbf{P}$ . The loss function now becomes

$$\kappa(\mathbf{H}) = \text{tr} (\mathbf{Q}^* - \mathbf{P}^* \mathbf{H})' (\mathbf{Q}^* - \mathbf{P}^* \mathbf{H}), \quad (9)$$

for which the ordinary LS solution is given above. After having found a solution for  $\mathbf{H}$ , new distances are computed. If the decrease in loss computed by (5b) is larger than a specified criterion new weights are computed according to (7) and a new cycle is started. The iteration process stops when some convergence criterion has been reached.

#### *Minimizing the city-block distance*

The minimization of the LAR loss function with the Euclidean distances criterion didn't cause many problems. However, we will see that for the city-block criterion the problem is more complicated because the weights cannot be put in a diagonal matrix so easily. We now have weights for each element of  $\mathbf{R}$ , which is the matrix containing the residuals. Therefore, we will systematically study different solutions to the problem. The first step is to find a weighted quadratic function that can be used to majorize the loss function in (5a).

Let's start by substituting (2) in (5a), which results in the following loss function

$$\sigma_1(\mathbf{H}) = \sum_i \sum_j |q_{ij} - \sum_k p_{ik} h_{kj}|, \quad (10)$$

where  $i$  runs from 1 to  $n$  and  $j$  and  $k$  from 1 to  $m$ .

If there were no restrictions on the rotation matrix  $\mathbf{H}$  this problem could also be solved quite simply by majorization with a columnwise partitioned quadratic loss function. The majorization function in matrix notation would be of the form:

$$\kappa(\mathbf{H}) = \sum_j (\mathbf{q}_j - \mathbf{P} \mathbf{h}_j)' \mathbf{W}_j (\mathbf{q}_j - \mathbf{P} \mathbf{h}_j), \quad (11)$$

where  $\mathbf{W}_j$  is a diagonal matrix with the  $j^{\text{th}}$  column of  $\mathbf{W}$  as diagonal elements. The elements of  $\mathbf{W}$  are defined as

$$w_{ij} = 1/\underline{\Gamma}_{ij} \quad \text{for } \underline{\Gamma}_{ij} \geq \epsilon \quad (12a)$$

$$w_{ij} = 1/\epsilon \quad \text{for } \underline{\Gamma}_{ij} < \epsilon \quad (12b)$$

The  $\underline{\Gamma}_{ij}$  have been computed in the former step of the algorithm. Since there are no restrictions on  $\mathbf{H}$  we can solve (11) for each column of  $\mathbf{Q}$  separately. In fact, we are repeatedly using a weighted multiple regression solution. However, in the Procrustes problem  $\mathbf{H}$  is restricted to be an orthonormal matrix.

### *Why the columnwise approach fails with an orthogonal $\mathbf{H}$*

The approach of partitioning the columns of  $\mathbf{Q}$  and using (11) for constructing a function that computes optimal updates for  $\mathbf{h}_j$  fails. It appears that an arbitrary solution is found if after each computed update the orthonormality restriction is imposed upon  $\mathbf{h}_j$ . The reason for this is geometrically shown in Figure 2 for three columns of  $\mathbf{H}$ . The vector  $\mathbf{h}$  is shown in the Figure and denotes any column of  $\mathbf{H}$ ; it has length 1 and is updated by  $\mathbf{s}$  to  $\mathbf{h}+\mathbf{s}$ . Next this update is positioned orthogonal to  $\mathbf{G}$ , containing the other columns of  $\mathbf{H}$ . Normalization of this vector will result in the old  $\mathbf{h}$ .

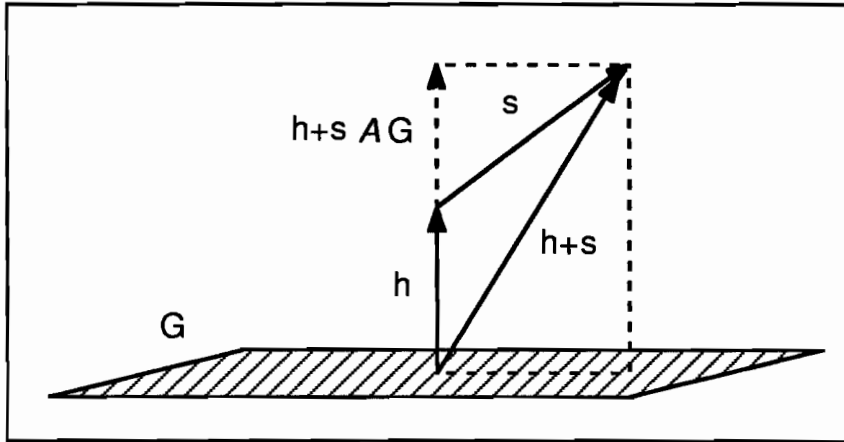


Figure 2. Vector  $\mathbf{h}$  is the  $j^{\text{th}}$  column of  $\mathbf{H}$  and  $\mathbf{G}$  contains all other columns. Vector  $\mathbf{h}+\mathbf{s}$  is the unrestricted update of  $\mathbf{h}$ .

### *The rowwise approach*

Since the columnwise approach does not work we have to find a different majorization function. Another idea<sup>1</sup> is splitting up  $\mathbf{Q}$  into its rows. The problem then becomes

$$\kappa(\mathbf{H}) = \sum_i (\mathbf{q}_i - \mathbf{H}' \mathbf{p}_i)' \mathbf{W}_i (\mathbf{q}_i - \mathbf{H}' \mathbf{p}_i). \quad (13)$$

We want to find a sequence of updates for  $\mathbf{H}$ , by writing  $\mathbf{H} = \underline{\mathbf{H}} + \mathbf{\Delta}$ . This way of attacking the problem is similar to the approach used by De Leeuw and Bijleveld (1988) in a different context. The matrix  $\underline{\mathbf{H}}$  is the best  $\mathbf{H}$  found in the previous step of the algorithm. So now we

<sup>1</sup> We would like to thank Henk Kiers for some helpful comments and suggestions at this point.

also have  $\mathbf{\Delta} = \mathbf{H} - \underline{\mathbf{H}}$ . Substituting these results in (13) yields, after some rewriting the following function,

$$\kappa(\mathbf{H}) = \kappa(\underline{\mathbf{H}}) + \sum_i (\mathbf{p}_i' \mathbf{\Delta} \mathbf{W}_i \mathbf{\Delta}' \mathbf{p}_i) - 2 \text{tr} \mathbf{\Delta}' \mathbf{S}^*, \quad (14)$$

where  $\mathbf{S}^*$  is defined as

$$\mathbf{S}^* = \sum_i (\mathbf{p}_i \mathbf{q}_i' - \mathbf{p}_i \mathbf{p}_i' \underline{\mathbf{H}}) \mathbf{W}_i \quad (15)$$

Next we want to define a majorization function  $\xi(\mathbf{H})$  such that the inequality

$$\kappa(\mathbf{H}) \leq \xi(\mathbf{H}), \quad (16)$$

is always true. For  $\xi(\mathbf{H})$  we can take the following function

$$\xi(\mathbf{H}) = \kappa(\underline{\mathbf{H}}) + \gamma \text{tr} (\mathbf{\Delta} - \mathbf{S})' (\mathbf{\Delta} - \mathbf{S}) - \gamma \text{tr} \mathbf{S}' \mathbf{S}, \quad (17)$$

with  $\mathbf{S} = \gamma^{-1} \mathbf{S}^*$ . Working this result out, yields

$$\xi(\mathbf{H}) = \kappa(\underline{\mathbf{H}}) + \gamma \text{tr} \mathbf{\Delta}' \mathbf{\Delta} + \gamma \text{tr} \mathbf{S}' \mathbf{S} - 2 \gamma \text{tr} \mathbf{\Delta}' \mathbf{S} - \gamma \text{tr} \mathbf{S}' \mathbf{S}, \quad (18)$$

$$\xi(\mathbf{H}) = \kappa(\underline{\mathbf{H}}) + \gamma \text{tr} \mathbf{\Delta}' \mathbf{\Delta} - 2 \text{tr} \mathbf{\Delta}' \mathbf{S}^*. \quad (19)$$

The first and last term of (19) are the same as those in (14), so in order to satisfy (16) we have to choose  $\gamma$  in such a way that the inequality stated in (20) is always true.

$$\sum_i (\mathbf{p}_i' \mathbf{\Delta} \mathbf{W}_i \mathbf{\Delta}' \mathbf{p}_i) \leq \gamma \text{tr} \mathbf{\Delta}' \mathbf{\Delta}. \quad (20)$$

*Lemma.* If we choose  $\gamma$  in (17) as

$$\gamma = \sum_i \mathbf{w}_i^* \mathbf{p}_i' \mathbf{p}_i \quad (21)$$

with  $\mathbf{w}_i^*$  defined as the maximum value of the diagonal matrix  $\mathbf{W}_i$  then the inequality in (16) is satisfied.

*Proof.* We will start with the inequality

$$\mathbf{p}_i' \mathbf{\Delta} \mathbf{W}_i \mathbf{\Delta}' \mathbf{p}_i \leq \mathbf{p}_i' \mathbf{\Delta} (\mathbf{w}_i^* \mathbf{I}) \mathbf{\Delta}' \mathbf{p}_i \quad (22)$$

Since the diagonal elements of the matrix  $\mathbf{w}_i^* \mathbf{I}$  are always larger than or equal to the corresponding elements of  $\mathbf{W}_i$ , the inequality relation from (22) is also true for the complete product. The right-hand term of (22) can be rewritten with a trace operator, which yields, after cyclic permutation of its arguments:

$$\mathbf{p}_i' \mathbf{\Delta} \mathbf{W}_i \mathbf{\Delta}' \mathbf{p}_i \leq \text{tr} \mathbf{\Delta}' (\mathbf{w}_i^* \mathbf{p}_i \mathbf{p}_i') \mathbf{\Delta}. \quad (23)$$

We also know that for any symmetric matrix  $\mathbf{M}_i$  the inequality

$$\text{tr } \mathbf{\Delta}' \mathbf{M}_i \mathbf{\Delta} \leq \gamma_i \text{tr } \mathbf{\Delta}' \mathbf{\Delta}, \quad (24)$$

is always true, if  $\gamma_i$  is chosen as the largest eigenvalue of  $\mathbf{M}_i$ . If  $\mathbf{M}_i$  is actually equal to the matrix  $\mathbf{w}_i * \mathbf{p}_i \mathbf{p}_i'$  then the largest eigenvalue is  $\mathbf{w}_i * \mathbf{p}_i' \mathbf{p}_i$ . If this true for one  $i$ , it is also true for the sum so finally we have to include the summation over  $i$ ; combining (23) and (24) we obtain

$$\sum_i \mathbf{p}_i' \mathbf{\Delta} \mathbf{W}_i \mathbf{\Delta}' \mathbf{p}_i \leq \text{tr } \mathbf{\Delta}' (\sum_i \mathbf{w}_i * \mathbf{p}_i \mathbf{p}_i') \mathbf{\Delta} \leq \gamma \text{tr } \mathbf{\Delta}' \mathbf{\Delta}, \quad (25)$$

where  $\gamma$  is chosen as  $\gamma = \sum_i \gamma_i = \sum_i \mathbf{w}_i * \mathbf{p}_i' \mathbf{p}_i$ , *QED*.

The minimization of (17) in fact means that we have to minimize the following term:

$$\gamma \text{tr } (\mathbf{\Delta} - \mathbf{S})' (\mathbf{\Delta} - \mathbf{S}). \quad (26)$$

Now substituting for  $\mathbf{\Delta}$  leaves us with an orthogonal Procrustes problem of the form

$$\text{tr } (\mathbf{H} - (\mathbf{H} + \mathbf{S}))' (\mathbf{H} - (\mathbf{H} + \mathbf{S})), \quad (27)$$

which is solved by taken the singular value decomposition of the known matrix  $(\mathbf{H} + \mathbf{S})$ , which is written as  $\mathbf{A} \mathbf{\Lambda} \mathbf{B}'$ . The solution for  $\mathbf{H}$  is then given by  $\mathbf{H} = \mathbf{A} \mathbf{B}'$ . Having minimized the majorization function  $\xi(\mathbf{H})$ , new weights can be computed and we continue this process until convergence.

### *The block diagonal matrices approach*

For the third approach let's define the following notation :  $\mathbf{Q}^0$  is a block diagonal matrix of order  $(n \times m) \times m$ , with each column of  $\mathbf{Q}$  used as a block.  $\mathbf{H}^0$  is defined the same way and is of order  $(m \times m) \times m$ .  $\mathbf{P}^0$  is defined differently, the blocks of  $\mathbf{P}^0$  consist of the entire matrix  $\mathbf{P}$ , which makes the order of the matrix  $(n \times m) \times (m \times m)$ . This can also be written in a more formal notation as

$$\mathbf{P}^0 = \mathbf{I}_m \bullet \mathbf{P}, \quad (28)$$

with the symbol  $\bullet$  as the Kronecker product. We also define  $\mathbf{I}^0 = \mathbf{I}_m \bullet \mathbf{e}_n$ , with  $\mathbf{e}_n$  the unit vector of length  $n$  and  $\mathbf{Q}^{+m}$  as the vertical concatenation of  $m$  times  $\mathbf{Q}$ . Now for  $\mathbf{Q}^0$  we have

$$\mathbf{Q}^0 = \mathbf{I}^0 * \mathbf{Q}^{+m}, \quad (29)$$

with the symbol  $*$  as the Hadamard product. See Figure 3.

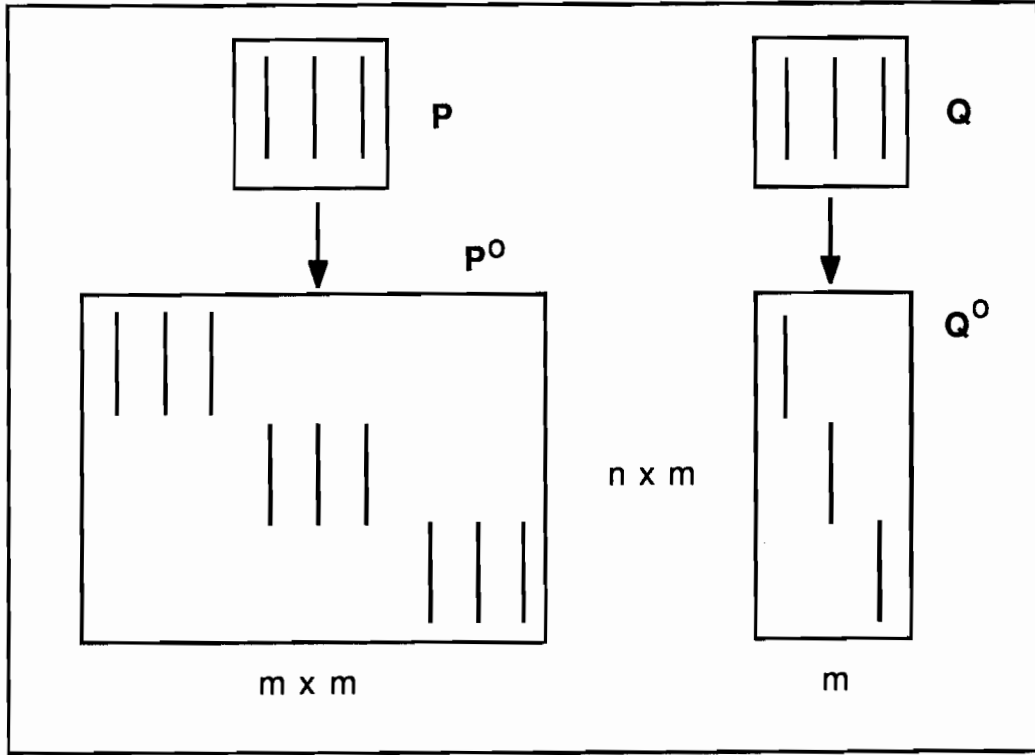


Figure 3. Example of the block diagonal matrices.

The majorization function can now be written as

$$\kappa(\mathbf{H}) = \text{tr} (\mathbf{Q}^0 - \mathbf{P}^0 \mathbf{H}^0)' \mathbf{V} (\mathbf{Q}^0 - \mathbf{P}^0 \mathbf{H}^0), \quad (30)$$

where  $\mathbf{V}$  is the diagonal  $(n \times m) \times (n \times m)$  matrix of weights. So in this way we have introduced a diagonal weights matrix from which each weight corresponds with one element of the residual matrix. To find the minimum of (30) we proceed in the following way. For  $\mathbf{H}^0$  we can always write

$$\mathbf{H}^0 = \mathbf{H}^* + (\mathbf{H}^0 - \mathbf{H}^*). \quad (31)$$

$\mathbf{H}^*$  is an unrestricted matrix. Substitution of (31) into (30) yields a function that consists of two independent parts when  $\mathbf{H}^*$  is chosen as the unrestricted minimum of (30):

$$\begin{aligned} \kappa(\mathbf{H}) = & \text{tr} (\mathbf{Q}^0 - \mathbf{P}^0 \mathbf{H}^*)' \mathbf{V} (\mathbf{Q}^0 - \mathbf{P}^0 \mathbf{H}^*) + \\ & \text{tr} (\mathbf{P}^0 \mathbf{H}^* - \mathbf{P}^0 \mathbf{H}^0)' \mathbf{V} (\mathbf{P}^0 \mathbf{H}^* - \mathbf{P}^0 \mathbf{H}^0). \end{aligned} \quad (32)$$

The first term on the right hand side of (32) is the loss of (30) without restrictions. Because  $\mathbf{H}^*$  is the minimum of this problem the cross product term vanishes. The second term can be written as

$$\xi(\mathbf{H}^0) = \text{tr} (\mathbf{H}^* - \mathbf{H}^0)' \mathbf{P}^0' \mathbf{V} \mathbf{P}^0 (\mathbf{H}^* - \mathbf{H}^0). \quad (33)$$

To minimize (33) we again need a majorization function. We will proceed along the lines of Heiser (1987b). Let  $\mathbf{M}$  be defined as  $\mathbf{M} = \mathbf{P}^0 \mathbf{V} \mathbf{P}^0$  and  $\xi(\mathbf{H}^0)$  denotes the function to be minimized in (33). Furthermore let  $\underline{\mathbf{H}}^0$  be a supporting point. We now want to find a function  $\xi(\mathbf{H}^0, \underline{\mathbf{H}}^0)$  to majorize  $\xi(\mathbf{H}^0)$ .

For  $\mathbf{H}^0$  we can always write  $\mathbf{H}^0 = \underline{\mathbf{H}}^0 + (\mathbf{H}^0 - \underline{\mathbf{H}}^0)$ . Substituting this in  $\xi(\mathbf{H}^0)$  we find

$$\xi(\mathbf{H}^0) = \xi(\underline{\mathbf{H}}^0) + \text{tr}(\mathbf{H}^0 - \underline{\mathbf{H}}^0)' \mathbf{M} (\mathbf{H}^0 - \underline{\mathbf{H}}^0) - 2 \text{tr}(\mathbf{H}^0 - \underline{\mathbf{H}}^0)' \mathbf{M} (\mathbf{H}^* - \underline{\mathbf{H}}^0) \quad (34)$$

The first term on the right-hand side of (34) is constant and the last term is linear in  $\mathbf{H}^0$ . It is therefore the middle term we want to simplify. Let  $\beta$  denote the largest eigenvalue of  $\mathbf{M}$ , i.e.  $\beta$  is the maximum of the Rayleigh quotient:

$$\beta = \max_{\mathbf{b}} \mathbf{b}' \mathbf{M} \mathbf{b} / \mathbf{b}' \mathbf{b}, \quad (35)$$

where the maximum is taken over all  $m$  vectors  $\mathbf{b}$ . It follows that for any  $\mathbf{b}$  we have

$$\beta \mathbf{b}' \mathbf{b} \geq \mathbf{b}' \mathbf{M} \mathbf{b} \quad (36)$$

Substituting the columns of  $\mathbf{H}^0 - \underline{\mathbf{H}}^0$  for  $\mathbf{b}$ , and repeatedly using (36) yields the inequality

$$\beta \text{tr}(\mathbf{H}^0 - \underline{\mathbf{H}}^0)' (\mathbf{H}^0 - \underline{\mathbf{H}}^0) \geq \text{tr}(\mathbf{H}^0 - \underline{\mathbf{H}}^0)' \mathbf{M} (\mathbf{H}^0 - \underline{\mathbf{H}}^0). \quad (37)$$

From this inequality it follows that the function

$$\xi(\mathbf{H}^0, \underline{\mathbf{H}}^0) = \xi(\underline{\mathbf{H}}^0) + \beta \text{tr}(\mathbf{H}^0 - \underline{\mathbf{H}}^0)' (\mathbf{H}^0 - \underline{\mathbf{H}}^0) - 2 \text{tr}(\mathbf{H}^0 - \underline{\mathbf{H}}^0)' \mathbf{M} (\mathbf{H}^* - \underline{\mathbf{H}}^0) \quad (38)$$

has the desired properties for majorizing (34). It can be simplified by collecting all constant terms in the constant  $\alpha$ , and after regrouping all terms involving  $\mathbf{H}^0$ , into

$$\xi(\mathbf{H}^0, \underline{\mathbf{H}}^0) = \alpha + \beta \text{tr}(\mathbf{H}^+ - \mathbf{H}^0)' (\mathbf{H}^+ - \mathbf{H}^0), \quad \text{with} \quad (39a)$$

$$\mathbf{H}^+ = \underline{\mathbf{H}}^0 + (1/\beta) \mathbf{M} (\mathbf{H}^* - \underline{\mathbf{H}}^0). \quad (39b)$$

With this result we are able to formulate an algorithm that minimizes  $\xi(\mathbf{H}^0)$ :

- (1) Find an arbitrary  $\underline{\mathbf{H}}^0$ , satisfying the restrictions.
- (2) Compute  $\beta$  as the largest eigenvalue of  $\mathbf{P}^0 \mathbf{V} \mathbf{P}^0$ .
- (3) Compute an update  $\mathbf{H}^+$  according to  $\mathbf{H}^+ = \underline{\mathbf{H}}^0 + (1/\beta) \mathbf{P}^0 \mathbf{V} \mathbf{P}^0 (\mathbf{H}^* - \underline{\mathbf{H}}^0)$ .
- (4) Change the block diagonal matrix  $\mathbf{H}^+$  into the ordinary matrix  $\mathbf{H}$  and perform a Procrustes orthonormalization, such that  $\mathbf{H}' \mathbf{H} = \mathbf{I}$ .
- (5) If the decrease in loss is more than a particular criterion, then replace  $\underline{\mathbf{H}}^0$  by the "blown up" version of  $\mathbf{H}$  and go to step (3).

If this process has reached convergence, a matrix of absolute values of the residuals is computed according to

$$\mathbf{R} = \text{abs}(\mathbf{Q} - \mathbf{PH}) \quad (40)$$

Next a new matrix  $\mathbf{V}$  is computed as a diagonal matrix with elements computed according to

$$v_s = 1/ r_{ij} \quad \text{for } r_{ij} \geq \epsilon \quad (41a)$$

$$v_s = 1/ \epsilon \quad \text{for } r_{ij} < \epsilon \quad (41b)$$

The index  $s$  runs from 1 to  $(n \times m)$ . Now we are going to minimize (32) again using different weights. If the decrease in loss is less than a preset criterion value we have finished.

The computation of step (3) in the algorithm can be facilitated by substituting in step (3) for  $\mathbf{H}^*$  the expression

$$\mathbf{H}^* = (\mathbf{P}^0 \mathbf{V} \mathbf{P}^0)^{-1} \mathbf{P}^0 \mathbf{V} \mathbf{Q}^0. \quad (42)$$

Step (3) then becomes

$$\mathbf{H}^+ = \mathbf{H}^0 + (1/\beta) (\mathbf{P}^0 \mathbf{V} \mathbf{Q}^0 - \mathbf{P}^0 \mathbf{V} \mathbf{P}^0 \mathbf{H}^0). \quad (43)$$

This result can also be obtained by directly substituting the expression  $\mathbf{H}^0 = \mathbf{H}^0 + (\mathbf{H}^0 - \mathbf{H}^0)$  in the loss function given in (34).

## 5. Huber's loss function

In analogy to LS and LAR it is also possible to define two variants for Huber's loss function (HF). HF consists of two parts: a quadratic part for well fitting points and a linear part for badly fitting points (Huber, 1981).

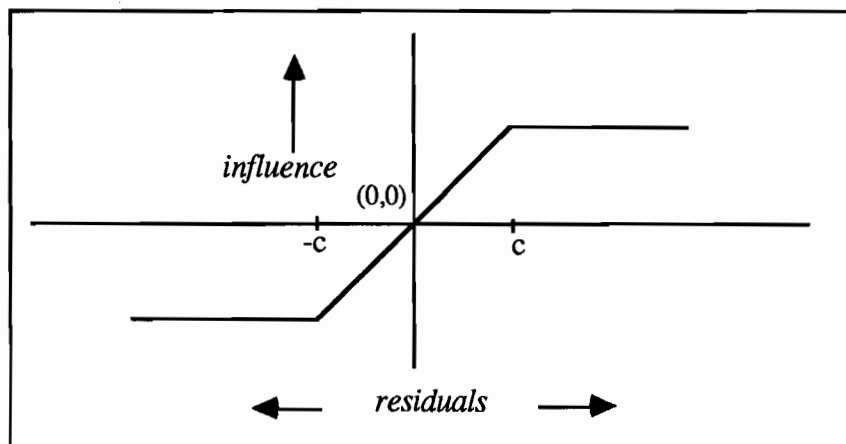


Figure 4. First derivative of Huber's function.

In Figure 4 we have plotted the first derivative of HF. Such plots are made to see the influence of the outliers upon the loss. For HF this influence curve (Hampel et al., 1986) is bounded. So the effect of the outliers cannot become infinitely large.

#### *Minimizing the Euclidean distance*

When we apply HL to minimize the Euclidean distances the loss function is written as

$$\sigma_1(\mathbf{H}) = \sum_i d_i^2 \quad \text{for } d_i < c, \quad (44a)$$

$$\sigma_2(\mathbf{H}) = \sum_i d_i \quad \text{for } d_i \geq c, \quad (44b)$$

where  $c$  is a *tuning constant*, to be chosen by the analyst. For residuals smaller than  $c$  HF fits LS; otherwise, LAR is fitted. Obviously, the LS part of this function can directly be minimized. For the second part we again use majorization by means of iteratively reweighted LS. This is completely analogous to the LAR solution. The weights are computed as

$$u_i = 1 \quad \text{for } \underline{d}_i < c, \quad (45a)$$

$$u_i = c / \underline{d}_i \quad \text{for } \underline{d}_i \geq c. \quad (45b)$$

The  $\underline{d}_i$  are the residuals, computed in the previous step of the iteration process. Positioning these weights in the diagonal matrix  $\mathbf{U}$  brings us back to the weighted quadratic function. So HF basically uses the same algorithm as LAR. The differences are in the computation of the weights and in the computation of the loss.

#### *Minimizing the city-block distance*

To minimize the city-block distances between two configurations HF is written as

$$\sigma_1(\mathbf{H}) = \sum_i \sum_j r_{ij}^2 \quad \text{for } r_{ij} < c, \quad (46a)$$

$$\sigma_2(\mathbf{H}) = \sum_i \sum_j r_{ij} \quad \text{for } r_{ij} \geq c, \quad (46b)$$

Again a reweighted least squares algorithm is used with weights defined as

$$v_{ij} = 1 \quad \text{for } \underline{r}_{ij} < c, \quad (47a)$$

$$v_{ij} = c / \underline{r}_{ij} \quad \text{for } \underline{r}_{ij} \geq c. \quad (47b)$$

With these weights a matrix  $\mathbf{V}$  is defined and we may proceed the same way as we did in one of the variants of the LAR case. So both the Euclidean criterion as the city-block criterion of Huber's loss function are straightforward extensions of the LAR solution.

## 6. A hard redescending loss function

Finally we will consider a so-called hard redescending loss function (HR) which decreases the influence of outliers even more than HF (cf. Hampel, 1980). The qualification "hard redescending" refers to the first derivative of the function, which becomes 0 for large residuals (c.q. distances). In Verboon and Heiser (1988) a hard redescender is discussed with equally spaced intervals and some considerations are studied for the choice of the tuning constant that should be used.

Like for HF we can show (see Figure 5) the first derivative of the HR, from which we learn that influence of the outliers is also bounded.

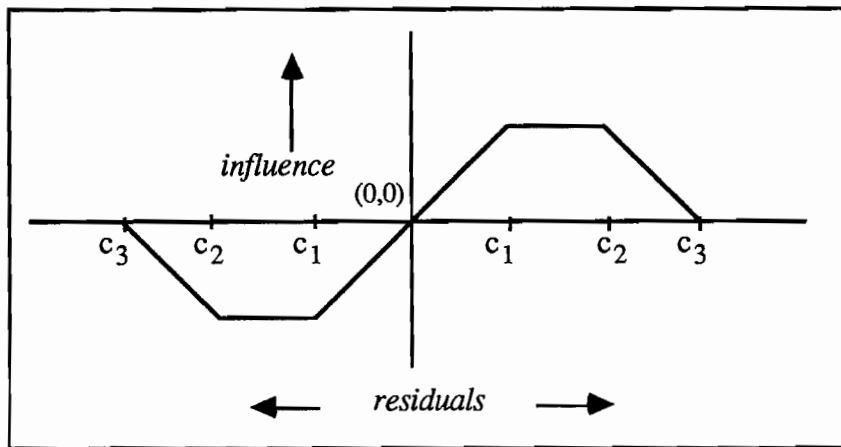


Figure 5. First derivative of the hard-redescending loss functions.

### Minimizing the Euclidean distance

In Table 1 a hard redescender is given with equally spaced intervals. The first column gives the loss function, the second the majorization function that is used for minimization and in the third the interval is given for which the function holds.

Table 1. Class of hard redescenders.

loss function	majorizing function	interval
$\sum_i d_i^2$	$\sum_i d_i^2$	$d_i < c$
$\sum_i [2c d_i - c^2]$	$\sum_i [(c/d_i) d_i^2 + c d_i - c^2]$	$c \leq d_i < 2c$
$\sum_i [6c d_i - d_i^2 - 5c^2]$	$\sum_i [(3c/d_i) d_i^2 - d_i^2 + 3c d_i - 5c^2]$	$2c \leq d_i < 3c$
$4 n m c^2$	$4 n m c^2$	$3c \leq d_i$

In the Table it can be seen that the variable weights for  $|d_i| < 2c$  are similar to HF. For the other parts of the function the weights are defined as

$$w_i = (3c - d_i) / d_i \quad \text{for } 2c \leq d_i < 3c, \quad (43a)$$

$$w_i = 0 \quad \text{for } d_i \geq 3c. \quad (43b)$$

So for large residuals the corresponding points obtain a weight 0, while points that fit well will obtain a weight 1. The other points are somewhere between 0 and 1. Having found the proper way to compute the weights, we proceed along the same lines as we did with LAR and HL. Again the only difference is in the computation of the loss and the weights.

### *Minimizing the city-block distance*

If in Table 1 all  $d_i$  are replaced by  $r_{ij}$  and the summation is over  $i$  and  $j$  than we have a hard redescending loss function to minimize the city-block distances between two configurations. Having found the weights we can apply the same algorithm as with LAR and HL.

So from a computational point of view minimizing a hard redescender, Huber's function or the least absolute residuals loss function are merely simple extensions of each other.

## 7. Three illustrations

### *First illustration: artificial data*

To learn about the merits of robust loss function for rotation problems, we will now present three examples. The first example is with artificial data. See Figure 6.

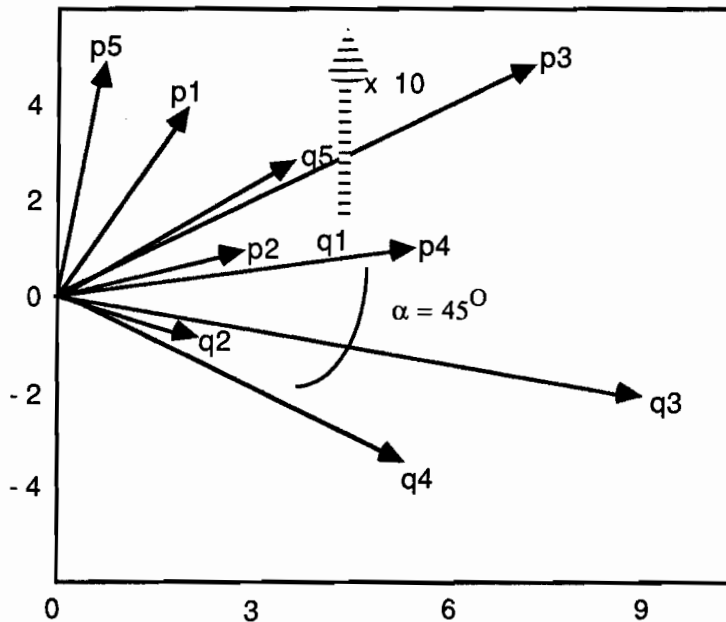


Figure 6. Example of two configurations in two dimensions with one outlier added.

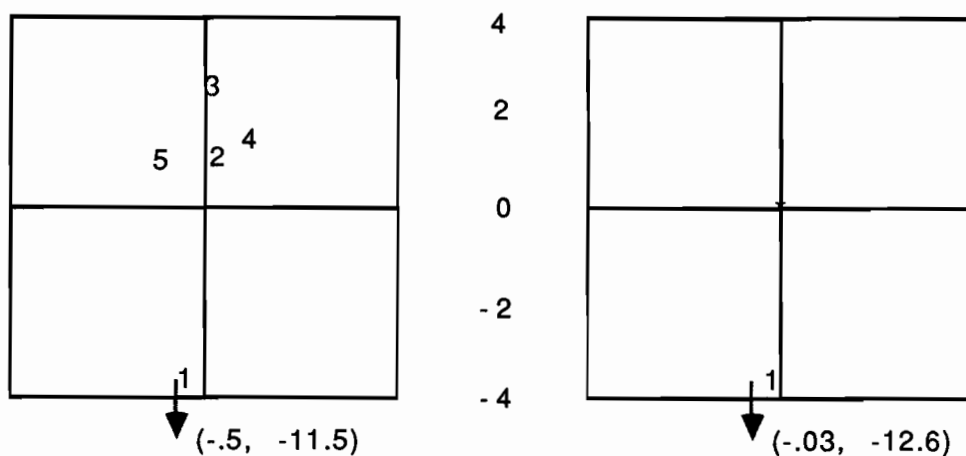
Consider the two configurations from Figure 6 with 5 points each in two dimensions. The rotation that is needed to change configuration **P** into **Q** is approximately  $45^\circ$ . All loss functions will exactly give the proper rotation angle. So if we are not bothered with computer time we can conclude that with "clean" data it doesn't matter which function to use. Next an outlier is created by multiplying the y-coordinate of the first point of **Q** with a factor 10. Again a Procrustes rotation is performed. The results are given in Table 2.

**Table 2.** Results Procrustes analysis with an outlier

	City-block		Euclidean	
	angle	iterations	angle	iterations
LS	29.92	1	29.92	1
LAR	44.79	144	44.86	6
HL	43.42	18	43.43	3
HR	44.90	17	44.92	4

We can see that when the criterion is to minimize the Euclidean distances convergence is much faster than in case of the city-block criterion. Furthermore it is seen that the computed angle is approximately the same for both criteria.

Like in regression analysis the LS solution is severely disturbed by the presence of the outlier. The LAR and HR yield a perfect solution, while the angle computed by HL only slightly differs from the true rotation angle.



**Figure 7.** Plot of residuals LS (left) vs. all other loss functions (right).

In Figure 7 the residuals are plotted. The left plot contains the residuals of the LS analysis. The outlier is immediately spotted. However the other points don't fit very well either. The right plot closely corresponds with the residual plot of all other analyses. In this plot then we

find all the clean points at or very near the origin, while the outlier is very far away. From both plots we not only see which point is the outlier, but also that it is only the y-coordinate that has a deviant value.

From the matrix of weights we can also learn which point is the outlier. For HL and HR all the good points have a weight of 1, the outlier however has a weight that is small for HL and 0 for HR. Finally the weights for the good points with LAR are very large, while the outlier has a small weight.

*Second illustration : empirical data*

For the second illustration we used data collected by Vonk (1989). The data consist of 61 two dimensional configurations each containing 18 points. These points refer to 18 character attributes that one uses to describe the personality of other people. In the original analysis performed by Vonk, the configurations resulted from a multidimensional scaling of the dissimilarity data of 61 subjects. Each of these configurations was then rotated towards a centroid configuration with respect to a least squares criterion. It was found that nearly all subjects fitted very well to the centroid configuration, except for three subjects.

A possible cause for the three subjects to fit rather badly could be the presence of outliers in their data. We therefore rotated the three configurations in a robust way towards the centroid with respect to Huber's loss function. In all three cases the computed rotation angle didn't differ very much from the least squares results.

Examining the weights however clearly showed that the bad fit was mainly due to a small number of points. See Table 3 in which the weights are shown that are unequal to one. We can see that there is no correspondence between the points of different subjects. Furthermore most weights differ merely in one dimension as we can see in the city-block analysis.

**Table 3.** Weights for three subjects.

		Euclidean	City-block	
subject43	timid	.88	1	.89
	persistent	.38	1	.38
	gentle	.81	.87	1
subject47	closed	.61	.80	1
	timid	.86	.83	1
subject54	dependent	.27	.52	.32
	stubborn	.74	1	.77
	independent	.63	.88	.88

*Third illustration: artificial data*

In the third illustration a target configuration was constructed by means of two different rotations. First a random configuration of twenty points was made. Next 75% of the points were rotated  $45^\circ$ , the other 25% were rotated  $-45^\circ$ . The rotated points made up the target configuration. These data were analyzed by a least squares Procrustes and two variants of Huber's loss function, the city-block and the euclidean distance approach.

The rotation angle found by least squares was  $25.5^\circ$ , by the 'city-block Huber'  $44.4^\circ$  and by the 'distance Huber'  $44.1^\circ$ . In table 4 the weights are shown for the 25% points derived from a different rotation. All other points were weighted by 1.

**Table 4.** Weights for the 25% points with a negative rotation.

	Euclidean	City-block	
	.04	.65	.04
	.10	.12	.18
	.03	.05	.04
	.04	.65	.04
	.23	.24	.72

**Table 5.** Means and variances of residuals for each rotation group per dimension.

		Least Squares		Euclidean		City-block	
mean	( $45^\circ$ )	.41	-3.82	-.01	-.17	-.01	-.12
	( $-45^\circ$ )	-7.69	12.04	-6.85	15.95	-6.84	16.00
variance	( $45^\circ$ )	4.72	5.04	.01	.01	.00	.01
	( $-45^\circ$ )	29.04	70.69	39.79	107.57	39.93	108.00

In table 5 the means and variances of the residuals for each technique are shown. It is evident that the points from the second group (negative rotation) are badly fitted. The means and variances of their corresponding residuals are large, while the first group has almost a residual variance of 0.

## 8. Conclusion

Like in regression analysis we have seen that in the orthogonal Procrustes problem outliers can have a very disturbing effect upon the least squares solution. It is therefore no luxury to have some robust tools at one's disposal. We have seen that the loss functions that were used in this study were able to detect the outlier and simultaneously compute the proper solution. Defining two different distance criteria for these functions didn't have any effect upon the solution. Under both conditions the techniques remained robust and yielded approximately the same results. However fitting the city-block distances has the advantage to spot the particular dimension that causes the outlier to be an outlier. While in the Euclidean distance approach the whole point is weighted with 0, in the city-block approach only one dimension might be weighted with 0.

## References

- Cliff, N. (1966). Orthogonal rotation to congruence. *Psychometrika*, 31, 33-42.
- De Leeuw, J. and Bijleveld, C. (1988). *Fitting longitudinal reduced rank regression models by alternating least squares*. RR-88-03, Department of Data Theory, University of Leiden.
- 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.
- Hampel, F.R., Ronchetti, E. M., Rousseeuw, P.J., Stahel, W.A. (1986). *Robust Statistics: the approach based on influence functions*. New York: Wiley.
- 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.
- Heiser, W.J. (1988). Multidimensional scaling with least absolute residuals. In: *Classification and related methods of data analysis*. H. Bock (ed.). Amsterdam: North-Holland.
- Huber, P.J. (1981). *Robust Statistics*. New York: Wiley.
- Rousseeuw, P.J. and Leroy, A. (1987). *Robust regression and outlier detection*. New York: Wiley.
- Verboon, P. and Heiser, W.J. (1990). Some jack-knifing results for regression with non-homogeneous loss functions. *Kwantitatieve Methoden*, 33, 161-173.
- Vonk, R. (1989). personal communication.