Robust Least Squares Multidimensional Scaling

Jan de Leeuw

October 18, 2024

Combining different loss functions with linear models and minimizing loss with iteratively reweighted least squares (IRLS) has a long history in robust statistics. In this paper we use an IRLS version of the smacof algorithm to minimize various robust multidimensional scaling loss functions. Our results use a general theorem on sharp quadratic majorization of De Leeuw and Lange (2009). We relate this theorem to earlier results in robust statistics, location theory, and sparse recovery. Code in R is included.

Table of contents

List of Figures

1 Introduction

The title of this paper is somewhat paradoxical. Least squares estimation is typically not robust, it is sensitive to outliers and pays too much attention to minimizing the largest residuals. What we mean by robust least squares multidimensional scaling (MDS), however, is using the smacof machinery designed to minimize least squares loss functions of the form

$$\sigma_2(X) := \sum \omega_k(\delta_k - d_k(X))^2, \tag{1}$$

to minimize robust loss functions.

The prototypical robust loss function is least absolute value loss

$$\sigma_1(X) := \sum \omega_k|\delta_k - d_k(X)|, \tag{2}$$

which we will call *strife*, because the names *stress*, *sstress*, and *strain* are already taken.

Strife is not differentiable at configurations X for which there is at least one k for which either $d_k(X) = \delta_k$ or $d_k(X) = 0$ (or both). This lack of differentiability complicates the minimization problem. Moreover experience with one-dimensional and city block MDS suggests that having areas where the loss function is not differentiable leads to (many) additional local minima.

In this paper we will discuss (and implement) various variations of σ_1 from (2). They can be interpreted in two different ways. On the one hand we use smoothers of the absolute value function, and consequently of strife. We want to eliminate the problems with differentiability, at least the ones caused by $\delta_k = d_k(X)$. If this is our main goal, then we want to choose the smoother in such a way that it is close to the absolute value function. This is not unlike the distance smoothing used by Pliner (1996) and Groenen, Heiser, and Meulman (1999) in the global minimization of σ_2 from (1).

On the other hand our modified loss function can be interpreted as more robust versions of the least squares loss function, and consequently of stress. Our goal then is to combines the robustness of the absolute value function with the efficiency and computational ease of least squares. If robustness is our goal then there is no reason to stay that close to the absolute value function.

Our robust or smooth loss functions are all of the form

$$\sigma(X) := \sum \omega_k \, f_c(\delta_k - d_k(X)), \tag{3}$$

for a suitable choice of the real valued function f. We will define what we mean by "suitable" later on. The subscript c of f_c is meant to indicate that the loss function may depend on one or more real-valued tuning parameters c that regulate the degree of smoothness and/or robustness. For now, note that loss (1) is the special case with $f_c(x) = x^2$ and loss (2)is the special case with $f_c(x) = |x|$. There is no tuning in both these cases.

2 Majorizing Strife

The idea of minimizing a least absolute value (LAV) to obtain parameter estimates dates back to the work of Boskovitch in the middle of the eighteenth century. Until recently it has been applied mainly to fit linear models, so that we can actually use standard linear programming algorithms to obtain optimal solutions.

The pioneering work in MDS strife minimization using smacof is Heiser (1988), which builds on earlier work of Heiser (1987). It is based on a creative use of the Arithmetic Mean-Geometric Mean (AM/GM) inequality to find a majorizer of the absolute value function. For the general theory of majorization algorithms (now more commonly known as MM algorithms) we refer to their original introduction in De Leeuw (1994) and to the excellent recent book by Lange (2016).

The AM/GM inequality says that for all non-negative x and y we have

$$|x||y| = \sqrt{x^2y^2} \leq \frac{1}{2}(x^2 + y^2), \tag{4}$$

with equality if and only if $x^2 = y^2$. If $y > 0$ we can write (4) as

$$|x| \leq \frac{1}{2}\frac{1}{|y|}(x^2 + y^2), \tag{5}$$

and this provides a quadratic majorization of $|x|$ at y. There is no quadratic majorization of $|x|$ at $y = 0$, which is a problem we will have to deal with.

Using the majorization (5), and assuming $\delta_k \neq d_k(Y)$ for all k, we define

$$\omega_1(X) := \frac{1}{2}\sum \omega_k \frac{1}{|\delta_k - d_k(Y)|}((\delta_k - d_k(Y))^2 + (\delta_k - d_k(X))^2). \tag{6}$$

Now $\sigma_1(X) \leq \omega_1(X)$ for all X and $\sigma_1(Y) = \omega_1(Y)$, and thus ω_1 majorizes σ_1 at Y.

2.1 Algorithm

Define

$$\omega_k(Y) := \omega_k \frac{1}{|\delta_k - d_k(Y)|}. \tag{7}$$

Reweighted smacof to minimize strife computes $X^{(k+1)}$ by decreasing

$$\sum \omega_k(X^{(k)})(\delta_k - d_k(X^{(k)}))^2, \tag{8}$$

using a standard smacof step. It then computes the new weights $\omega_k(X^{(k+1)})$ from (7) and uses them in the next smacof step to update $X^{(k+1)}$. And so on, until convergence.

A straightforward variation of the algorithm does a number of smacof steps before upgrading the weights. This still leads to a monotone, and thus convergent, algorithm. How many smacof steps we have to take in the inner iterations is something that needs further study. It is likely to depend on the fit of the data, on the shape of the function near the local minimum, and on how far the iterations are from the local minimum.

2.2 Zero Residuals

It may happen that for some k we have $d_k(X^{(k)}) = \delta_k$ while iterating. There have been various proposals to deal with such an unfortunate event, and we will discuss some of them further on. Even more importantly we will see that that the minimizer of the absolute value loss usually satisfies $d_k(X) = \delta_k$ for quite a few elements, which means that near convergence the algorithm will become unstable because the weights from (7) become very large.

A large number of somewhat ad-hoc solutions have been proposed to deal with the problem of zero residuals, both in location analysis and in the statistical literature. We tend to agree with the assessment of Aftab and Hartley (2015).

> .. attempts to analyze this difficulty [caused by infinite weights of IRLS for the ℓ_p-loss] have a long history of proofs and counterexamples to incorrect claims.

Schlossmacher (1973) is the first discussion of the majorization method in the statistical literature (for LAV linear regression). His proposal is to simply set a weight equal to zero if the corresponding residual is less than some small positive value ϵ. A similar approach, also used in location analysis, is to cap the weights at some large positive value. In Heiser (1988) all residuals smaller than this epsilon get a weight equal to the weighted average of all these small residuals. Phillips (2002) assumes double-exponential errors in LAV regression and then concludes that the EM algorithm gives the majorization method we have discussed. He uses (7) throughout if all residuals are larger than ϵ. If one of more residuals are smaller than epsilon then the weight for those residuals is set equal to one, while for the remaining residuals the weight is set to epsilon divided by the absolute value of the residual. Often we get the assurance in these papers that the problem is not really important in practice, because it is very rare, and by just wiggling we will get to the unique solution anyway. But both in location analysis and in LAV regression the loss function is convex, however, which guarantees a unique minimum. This is certainly not the case in robust MDS. In this paper we try to follow a more systematic approach that uses smooth parametric approximations to the absolute value function, where the parameter can be used to make the approximation as precise as necessary.

To illustrate the problems with differentiability we compute the directional derivatives of strife.

Let $s_k(X) := \omega_k|d_k(X) - \delta_k|$.

1. If $\delta_k = 0$ and $d_k(X) = 0$ then $ds_k(X;Y) = \omega_k d_k(Y)$.
2. If $\delta_k > 0$ and $d_k(X) = 0$ then $ds(X;Y) = -\omega_k d_k(Y)$.
3. If $d_k(X) > 0$ and $d_k(X) - \delta_k > 0$ then $ds_k(X;Y) = \omega_k \frac{1}{d_k(X)}\text{tr } X'A_kY$.
4. If $d_k(X) > 0$ and $d_k(X) - \delta_k < 0$ then $ds_k(X;Y) = -\omega_k \frac{1}{d_k(X)}\text{tr } X'A_kY$.
5. If $d_k(X) > 0$ and $d_k(X) - \delta_k = 0$ then $ds_k(X;Y) = \omega_k \frac{1}{d_k(X)}\text{tr } |X'A_kY|$.

The directional derivative of σ_1 is consequently the sum of five terms, corresponding with each of these five cases.

In the case of stress the directional derivatives could be used to prove that if $\omega_k\delta_k > 0$ for all k then stress is differentiable at each local minimum (De Leeuw (1984)). For strife to be differentiable we would have to prove that at a local minimum both $d_k(X) > 0$ and $(d_k(X) - \delta_k) \neq 0$ for all k with $\omega_k > 0$. But this is impossible by the following argument. In the one-dimensional case we can partition $\mathbb{R}^n$ into $n!$ polyhedral convex cones corresponding with the permutations of x. Within each cone the distances are a linear function of x. Each cone can be partitioned by intersecting it with the polyhedra defined by the linear inequalities $\delta_k - d_k(x) \geq 0$ or $\delta_k - d_k(x) \leq 0$. Some of these intersections can and will obviously be empty. Within each of these non-empty polyhedral regions strife is a linear function of x. Thus it attains its minimum for the region at a vertex, which is a solution for which some distances are zero and/or some residuals are zero. There can be no
minima, local or global, in the interior of one of these polyhedral regions. We have thus shown that in one dimension strife is not differentiable at a local minimum, and that there is presumably a large number of them. Even for moderate n the number of regions is of course too large to actually compute or draw.

In the multidimensional case linearity goes out the window. The set of configurations $d_k(X) = \delta_k$ is an ellipsoid and $d_k(X) = 0$ defines a hyperplane. Strife is not differentiable at all intersections of these ellipsoids and hyperplanes. The partitioning of $\mathbb{R}^n$ by these ellipsoids and hyperplanes is not simple to describe. It has convex and non-convex cells, and within each cell strife is the difference of two weighted sums of distances. Anything can happen.

2.3 ℓ_0 loss

A somewhat extreme special case of Equation (3) has

$$f(x) = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{otherwise.} \end{cases}$$

This is ℓ_0 loss. Minimizing ℓ_0 loss means maximizing the number of cases with perfect fit, i.e. with $\delta_k = d_k(X)$. The reason we mention it here is that the work of Donoho and Elad (2003) and Candes and Tao (2005) suggests that the minimizer of ℓ_1 loss, i.e. absolute value loss, gives a good approximation to the minimizer of ℓ_0 loss, at least in a number of special cases. In MDS we do not have linearity or convexity, but nevertheless the theoretical results in simpler cases are suggestive. By computing the directional derivatives we have seen that at least in the one-dimensional MDS case a number of residuals will indeed be zero at the optimum LAV solution.

There is an excellent review of the use of ℓ_1 in various sparse recovery fields in Candes, Wakin, and Boyd (2008). In that paper they also propose an iteratively reweighted LAV algorithm, which solves ℓ_1 problems between weight updates. Maybe because of that they go so far as calling ℓ_1 "the modern least squares". But let's not get carried away, in actual ease and frequency of use ℓ_1 still has a long way to go if it wants to replace ℓ_2.

3 Generalizing Strife

We have seen that Heiser (1988) applied majorization to minimize strife, using the AM/GM inequality. We now generalize this approach so that it can easily deal with other robust loss functions. A great number of different loss functions will be discussed. The intention is not to confuse the reader by presenting a large number of alternatives with rather limited information. We show all these loss functions as examples of a general principle of algorithm construction and as examples of loss functions that have been used in statistics, location analysis, image analysis and engineering over the years. They are all implemented in the function smacofRobust(), written in R (R Core Team (2024)).

3.1 Majorization

First some definitions.

Definition 3.1. A function g *majorizes* a function f at y if $g(x) \geq f(x)$ for all x and $g(y) = f(y)$. The point y is the *support point* of the majorization. Majorization of f at y is *strict* if $g(x) > f(x)$ for all $x \neq y$.

Definition 3.2. If $\mathfrak{H}$ is a family of functions that all majorize f at y then $h \in \mathfrak{H}$ is a *sharp majorization* in $\mathfrak{H}$ if $h(x) \leq g(x)$ for all $g \in \mathfrak{H}$. The sharp majorization, if it exists, is by definition unique.

Theorem 3.1. *Suppose f and g are two functions defined on an open interval of the real line and y is a point of the interval where g majorizes f.*

- *If f and g are differentiable at y then $f'(y) = g'(y)$.*
- *If f and g are twice-differentiable at y then $f''(y) \leq g''(y)$.*
- *If $f''(y) < g''(y)$ then g strictly majorizes f in a neighborhood of y.*

Proof. $h = g - f$ is non-negative and has a minimum equal to zero at y. Thus the derivative of h vanishes at y and the second derivative is non-negative at y. If the second derivative is negative then we use the sufficient condition for a local minimum. □

3.1.1 Sharp Quadratic Majorization

The AM/GM inequality was used in Section 2 to construct a quadratic majorization of strife. In this paper we are specifically interested in sharp quadratic majorization, in which $\mathfrak{H}$ is the set of all convex quadratics that majorize f at y. This case has been studied in detail (in the case of real-valued functions on the line) in De Leeuw and Lange (2009). For the loss functions we study there are two problems that have to be solved. First, we need a general procedure to construct quadratic majorizers. Second, we need to show that some of our majorizers are sharp.

If f is differentiable at y, then all quadratics g that majorize f at y have

$$g(x) = f(y) + f'(y)(x-y) + \frac{1}{2}a(x-y)^2 \tag{9}$$

for some a, not necessarily positive. Since $g''(y) \geq f''(y)$ by Theorem 3.1 we must have $a \geq f''(y)$. Note that not all functions have quadratic majorizations. If f is a non-trivial cubic then so is $h = g - f$, and consequently we cannot have $h \geq 0$ on the whole real line.

We now look more closely at a in Equation 9. For $x \neq y$ define

$$\alpha(x) := \frac{f(x) - f(y) - f'(y)(x-y)}{\frac{1}{2}(x-y)^2} \tag{10}$$

If f is two times differentiable at y then by l' Hôpital

$$\lim_{x \to y} \alpha(x) = f''(y), \tag{11}$$

and thus we define $\alpha(y) = f''(y)$ to make α continuous at y. Of course α is a different function of x for each y, but since we are dealing with only one fixed y here we suppress this dependence.

If f is convex, then α is the ratio of two non-negative convex functions of x. If f is two times differentiable then there is a z between x and y such that $\alpha(x) = f''(z)$. If $f''(x) \leq K$ for all x, then $\alpha(x) \leq K$ as well.

Quadratic majorization is equivalent to $a \geq \alpha(x)$ for all x. Sharp quadratic majorization is possible if and only

$$A := \sup_x \alpha(x) < +\infty, \tag{12}$$

in which case we get sharp quadratic majorization by setting $a = A$ in Equation 9. The quadratic g of Equation 9 majorizes f if and only if $a \geq A$.

We illustrate these concepts by using low-degree polynomials. First a cubic. Expand the function around y as $f(x) = f(y) + f'(y)(x-y) + \frac{1}{2}f''(y)(x-y)^2 + \frac{1}{6}f'''(y)(x-y)^3$.

Thus $\alpha(x) = f''(y) + 3f'''(y)(x-y)$ and we have majorization at y if the linear function α is everywhere non-negative. Since this is impossible, no quadratic majorizer exists at any y. Now apply the same reasoning to a quartic. We find $\alpha(x) = f''(y) + 3f'''(y)(x-y) + \frac{1}{12}f^{iv}(y)(x-y)^2$, a quadratic in x. We have quadratic majorization at y if this quadratic is non-negative in $x - y$. It is clearly necessary that $f^{iv}(y)$ is positive. This implies the quadratic is unbounded above, and thus a sharp quadratic majorization does not exist at any y. Non-sharp quadratic majorizations at y exist if the discriminant of the quadratic is non-positive, and this inequality constraint on $(f''(y), f'''(y), f^{iv}(y))$ defines a non-empty cone in $\mathbb{R}^3$. It is sufficient for quadratic majorization at y, for example, that both $f''(y)$ and $f^{iv}(y)$ are positive and that $f'''(y) = 0$. It is necessary that both $f''(y)$ and $f^{iv}(y)$ are positive.

Theorem 3.2. *Suppose α has a local maximum at a point x where α is two times differentiable. Then*

$$\frac{f(x) - f(y)}{x - y} = \frac{1}{2}(f'(x) + f'(y)), \tag{13a}$$

$$\alpha(x) = \frac{f'(x) - f'(y)}{x - y}, \tag{13b}$$

$$f''(x) \leq \frac{f'(x) - f'(y)}{x - y}. \tag{13c}$$

If the inequality in (13c) *is strict then α has a local maximum at x.*

Proof. After some manipulation (13a) and (13c) are the necessary conditions $\alpha'(x) = 0$ and $\alpha''(x) \leq 0$ for a local maximum, and the sufficient condition $\alpha''(x) < 0$. If x satisfies (13a) then substitution in Equation 10 gives (13b). □

Note that we have not shown that α always attains its maximum. De Leeuw and Lange (2009) give the example of the differentiable function

$$f(x) = \begin{cases} x^2 & \text{if } x \leq 1, \\ 2x - 1 & \text{otherwise,} \end{cases}$$

(#eq-dllexam) which has $\alpha(x) = 0$ for $x > 1$ and $\alpha(x) < 2$ for $x \leq 1$, so that $A = \sup_{x \leq 1} \alpha(x) = 2$ and the maximum does not exist.

Also, the conditions of Theorem 3.2 cannot show that α has a *global* maximum at x, and that consequently g of Equation 9 with a given by (13b) is a sharp quadratic majorizer.

3.1.2 Regulated Functions

Theorem 4.5 in De Leeuw and Lange (2009) gives a way to construct quadratic majorizers of a differentiable function on an interval. We generalize it here to regularized functions, which may not be differentiable at all points in the interval. First a short introduction to regularized functions (Dieudonné (1969), sections 7.6 and 8.7).

Definition 3.3. A real-valued function on a closed interval $[a, b]$ is *regulated* if it has a limit from the right for each x in $[a, b)$ and a limit from the left for each x in $(a, b]$.

Note that the end-points of the interval can be $\pm\infty$. Regularized functions can only have discontinuities of the first kind (jump discontinuities). Step functions, monotone functions, functions of bounded variation, and continuous functions are all regulated functions. An alternative definition is that regulated functions are limits of sequences of step functions, where the convergence is uniform on compact sets.

Definition 3.4. A real-valued function f on $[a, b]$ is a *primitive* of a real-valued function g on $[a, b]$ if f is differentiable with $f'(x) = g(x)$, except possibly at a denumerable number of points.

A regulated function has at least one primitive, and primitives are unique up to addition of a constant function. The primitive f of a continuous function g is differentiable everywhere, with $f'(x) = g(x)$.

Definition 3.5. The *integral* of a regulated function f on $[a, b]$, written as $\int_a^b f(x)dx$, is equal to $g(b) - g(a)$, where g is one of the primitives of f.

Theorem 3.3. *Suppose g is a regulated function on $[x, y]$ and f is one of its primitives. Define $h(z) := g(z)/z$ and assume h is non-increasing on $[x, y]$. Then we have the quadratic majorization*

$$f(x) \leq f(y) + \frac{1}{2}h(y)(x^2 - y^2). \tag{14}$$

of f at y.

Proof.

$$f(y) - f(x) = \int_x^y g(z)dz = \int_x^y h(z)zdz \geq h(y)\int_x^y zdz = \frac{1}{2}(y^2 - x^2).$$

{#eq:dllproof} □

Theorem 3.3 implicitly assumes that if $[x, y]$ contains zero then $h(z)/z$ is defined at $z = 0$. Maybe h is of the form $h(z) = zu(z)$ for some function u, or maybe l'Hôpital applies. Since the value of the integral does not change if we change the integrand at a single point this causes no loss of generality.

Suppose g is the sign function, which is obviously regulated, and f is the absolute value function, a primitive of g. Then for $y > 0$

$$|x| \leq |y| + \frac{1}{2}\frac{1}{|y|}(x^2 - y^2) = \frac{1}{2}\frac{1}{|y|}(x^2 + y^2),$$

which is the AM/GM inequality.

3.1.3 Two Support Points

Theorem 3.4. *Suppose the quadratic g majorizes f at y and at $z \neq y$. Then*

$$a = \frac{f'(z) - f'(y)}{z - y} \tag{15}$$

Proof. From Equation 9 we have $g'(x) = f'(y) + a(x - y)$. But because of Theorem 3.1 we must also have $g'(z) = f'(z)$. Thus $g'(z) = f'(y) + a(z - y) = f'(z)$. □

Again, we have not shown that g with a from Equation 15 majorizes f at y and z. Only the reverse implication, which is that if g majorizes f at y and z then g is uniquely determined by Equation 15. In practice, even if one knows the support points y and z, one still has to prove majorization. This is precisely how Heiser (1988), Verboon and Heiser (1994), and Groenen, Giaquinto, and Kiers (2003) establish their majorizations. Van Ruitenburg (2005) takes us a step further down that road.

Lemma 3.1. *If different quadratics g and h majorize f at y then either g strictly majorizes h or h strictly majorizes g.*

Proof. We have

$$g(x) = f(y) + f'(y)(x - y) + \frac{1}{2}a_1(x - y)^2, \tag{16a}$$

$$h(x) = f(y) + f'(y)(x - y) + \frac{1}{2}a_2(x - y)^2.. \tag{16b}$$

Thus $g(x) - h(x) = \frac{1}{2}(a_1 - a_2)(x - y)^2$, which is either strictly positive or strictly negative for $x \neq y$ and zero for $x = y$. □

Lemma 3.2. *Suppose quadratics g and $h \neq g$ majorize f at y. Suppose, in addition, that g majorizes f at $z \neq y$. Then h strictly majorizes g at y.*

Proof. If g strictly majorizes h at y then $h(z) < g(z) = f(z)$ and thus h does not majorize f. By Lemma 3.1 h strictly majorizes g at y. □

Theorem 3.5. *If the quadratic g majorizes f at y and at $z \neq y$, then g is a sharp majorizer of f at y.*

Proof. Directly from Lemma 3.2. □

3.1.4 Even Functions

Theorem 3.6. *If f is even and the quadratic g majorizes f at y and $-y$, where $y \neq 0$, then g is the even quadratic given by*

$$g(x) = f(y) + \frac{1}{2}\frac{f'(y)}{y}(x^2 - y^2). \tag{17}$$

Moreover g is the sharp quadratic majorization of f at y and $-y$.

Proof. If f is even then f' is odd. Consequently Equation 15 becomes

$$a = \frac{f'(y)}{y}, \tag{18}$$

which is even. Moreover because g majorizes f at y

$$g(x) = f(y) + f'(y)(x - y) + \frac{1}{2}\frac{f'(y)}{y}(x - y)^2, \tag{19a}$$

and because g majorizes f at $-y$

$$g(x) = f(y) - f'(y)(x + y) + \frac{1}{2}\frac{f'(y)}{y}(x + y)^2. \tag{19b}$$

Averaging the two equations (19a) and (19b) for g, and simplifying, gives the required result. That the majorization is sharp follows from Theorem 3.5. □

If f is even then $x = -y$ makes both sides of (13a) equal to zero. Thus α has a stationary point at $-y$.

If in addition

$$f''(y) < \frac{f'(y)}{y} \tag{20}$$

then $-y$ is a local maximum of α.

If $y > 0$ then Equation 20 can be written as $yf''(y) - f'(y) < 0$.

Now the sign of the derivative of $f'(x)/x$ is the sign of $xf''(x) - f'(x)$ and thus … is equivalent to $f'(x)/x$ is decreasing for $x > 0$.

On the positive real line $f'(x)/x$ is decreasing if and only if

$$f''(x) \leq \frac{f'(x)}{x}.$$

$$\left(\frac{f'(x)}{x}\right)' = \frac{xf''(x) - f'(x)}{x^2}$$

If $f'(x)/x$ is decreasing then $xf''(x) - f'(x) < 0$. Thus if $x > 0$ $f'(x)/x$ is decreasing is equivalent with

$$f''(x) \leq \frac{f'(x)}{x}$$

3.1.5 Mis

Theorem 3.7. *Suppose*

1. *f is the primitive of a regulated function h on $\mathbb{R}$*
2. *the ratio $h(x)/x$ is non-increasing on $(0, \infty)$*
3. *f is even*

Then the even quadratic

$$g(x) = \frac{f'(y)}{2y}(x^2 - y^2) + f(y)$$

is a sharp quadratic majorizer of f at the point y.

Proof. Suppose $0 < y < x$. Then

$$f(x) - f(y) = \int_y^x h(z)dz = \tag{21}$$
$$= \int_y^x \frac{h(z)}{z} z dz \leq \frac{h(y)}{y} \int_y^x z dz =$$
$$= \frac{1}{2}\frac{h(y)}{y}(x^2 - y^2).$$

If $0 < x < y$ then

$$f(y) - f(x) = \int_x^y h(z)dz = \tag{22}$$
$$= \int_x^y \frac{h(z)}{z} z dz \geq \frac{h(y)}{y} \int_y^x z dz = \tag{23}$$
$$= \frac{1}{2}\frac{h(y)}{y}(y^2 - x^2). \tag{24}$$

If $x < 0 < y$ then

$$f(y) - f(x) = \int_x^y h(z)dz = \int_x^0 h(z)dz + \int_0^y h(z)dz = \int_0^{-x} h(z)dz + \int_0^y h(z)dz \leq \frac{h(-x)}{-x}$$

□

Theorem 3.8. *The ratio $f'(x)/x$ is decreasing on $(0, \infty)$ if and only $f(\sqrt{x})$ is concave. The set of functions satisfying this condition is closed under the formation of (a) positive multiples, (b) convex combinations, (c) limits, and (d) composition with a concave increasing function $g(x)$.*

Note that these theorems give a sufficient condition for quadratic majorization (in fact, for sharp quadratic majorization) and not a necessary one. Quadratic majorization, and even sharp quadratic majorization, may still be possible if the conditions in the theorem are violated.

We now apply Theorem 3.7 to functions of the form

$$\sigma_f(X) := \sum \omega_k \, f(\delta_k - d_k(X)), \tag{25}$$

where f satisfies the conditions in the theorem. If

$$\omega_f(X) := \sum \omega_k \frac{f'(\delta_k - d_k(Y))}{2(\delta_k - d_k(Y))}\{(\delta_k - d_k(X))^2 - (\delta_k - d_k(Y))^2\} + f(\delta_k - d_k(Y)), \tag{26}$$

then ω_f is a sharp quadratic majorization at Y.

In iteration k the robust smacof algorithm does a smacof step towards minimization of ω_f over X. We can ignore the parts of (26) that only depend on Y, and minimize

$$\sum \omega_k(X^{(k)})(\delta_k - d_k(X))^2, \tag{27}$$

with

$$\omega_k(X^{(k)}) := \omega_k \frac{f'(\delta_k - d_k(X^{(k)}))}{2(\delta_k - d_k(Y))}. \tag{28}$$

It then recomputes the weights $\omega_k(X^{(k+1)})$ and goes to the smacof step again. This can be thought of as iteratively reweighted least squares (IRLS), and also as nested majorization, with the smacof majorization based on the Cauchy-Schwartz inequality within the sharp quadratic majorization of the loss function based on the AM/GM inequality.

4 Power Smoothers

We first discuss a class of smoothers of the absolute value function that maintain most of its structure. They have a shift parameter c that takes care of the non-differentiability. Although different smoothers have different scales and interpretations for c, we will use the same symbol throughout. Also some smoothers have a power parameter q that determines the shape of the loss function bowl.

4.1 Charbonnier

The first, and perhaps most obvious, choice for smoothing the absolute value function is

$$f_c(x) = \sqrt{x^2 + c^2}. \tag{29}$$

For $c > 0$ we have $f_c(x) > |x|$. If $c \to 0$ then $f_c(x)$ decreases monotonically to $|x|$. Also $\max_x |f_c(x) - |x|| = c$ attained at $x = 0$, which implies uniform convergence of f_c to $|x|$.

In the engineering literature Equation 33 is known as Charbonnier loss, after Charbonnier et al. (1994), who were possibly the first researchers to use it in image restauration. Ramirez et al. (2014) count the number of elementary computer operations and argue Equation 33 is also the "most computationally efficient smooth approximation to $|x|$".

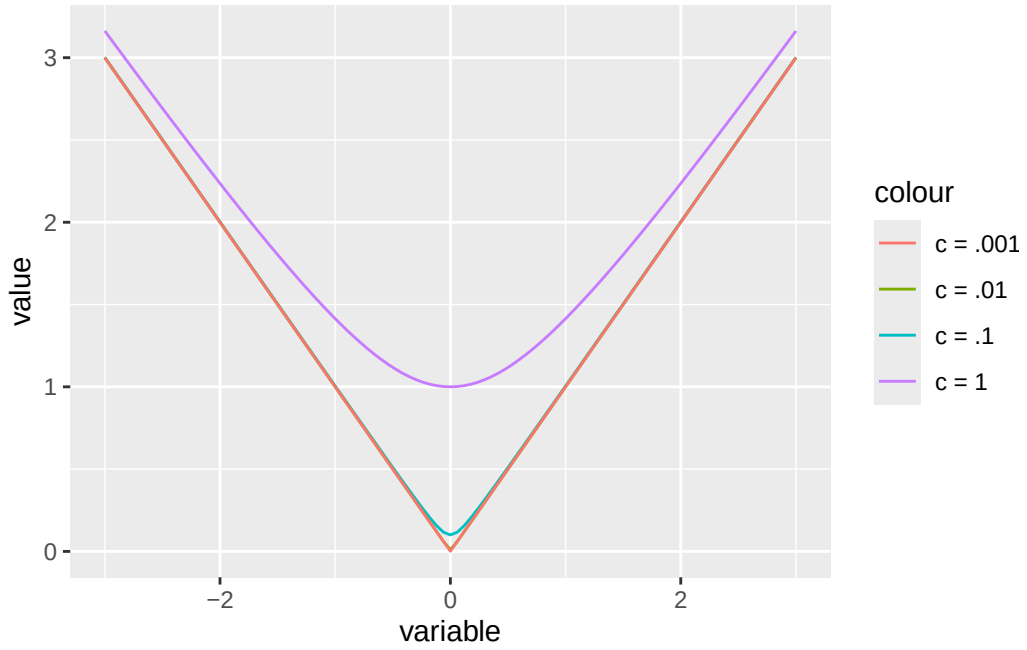

Figure 1: Charbonnier Loss

By l'Hôpital

$$\lim_{x\to 0} \frac{\sqrt{x^2+c^2}-c}{\frac{1}{2}x^2} = 1. \tag{30a}$$

Of course also

$$\lim_{x\to \infty} \frac{\sqrt{x^2+c^2}}{|x|} = 1 \tag{30b}$$

and

$$\lim_{x\to \pm\infty} \sqrt{x^2+c^2} - |x| = 0 \tag{30c}$$

Thus if x is much smaller than c then loss is approximately a quadratic in x, and if x is much larger than c then loss is approximately the absolute value.

Loss function Equation 29 is infinitely many times differentiable. Its first derivative is

$$f'_c(x) = \frac{1}{\sqrt{x^2+c^2}}x, \tag{31}$$

which converges, again in the sup-norm and uniformly, to the sign function if $c \to 0$. The IRLS weights are

$$w_c(x) = \frac{1}{\sqrt{x^2+c^2}} \tag{32}$$

which is clearly a decreasing function of x on $\mathbb{R}^+$.

4.2 Generalized Charbonnier

The loss function $(x^2+c^2)^{\frac{1}{2}}$ smoothes $|x|$. In the same way generalized Charbonnier loss smoothes ℓ_p loss $|x|^q$. We have a two-parameter family of loss functions in this case.

$$f_{c,q}(x) := (x^2+c^2)^{\frac{1}{2}q} \tag{33}$$

$$w_{c,q}(x) = q(x^2+c^2)^{\frac{1}{2}q-1} \tag{34}$$

which is non-increasing for $q \leq 2$. Note that we do not assume that $q > 0$, and consequently generalized Charbonnier loss provides us with more flexibility than Charbonnier loss from Equation 29. Of course if $q < 0$ "loss" becomes "gain", with a maximum at zero instead of a minimum. To get a proper loss function, take the negative. **?@fig-gcharfig** plots generalized Charbonnier loss for some negative values of q.

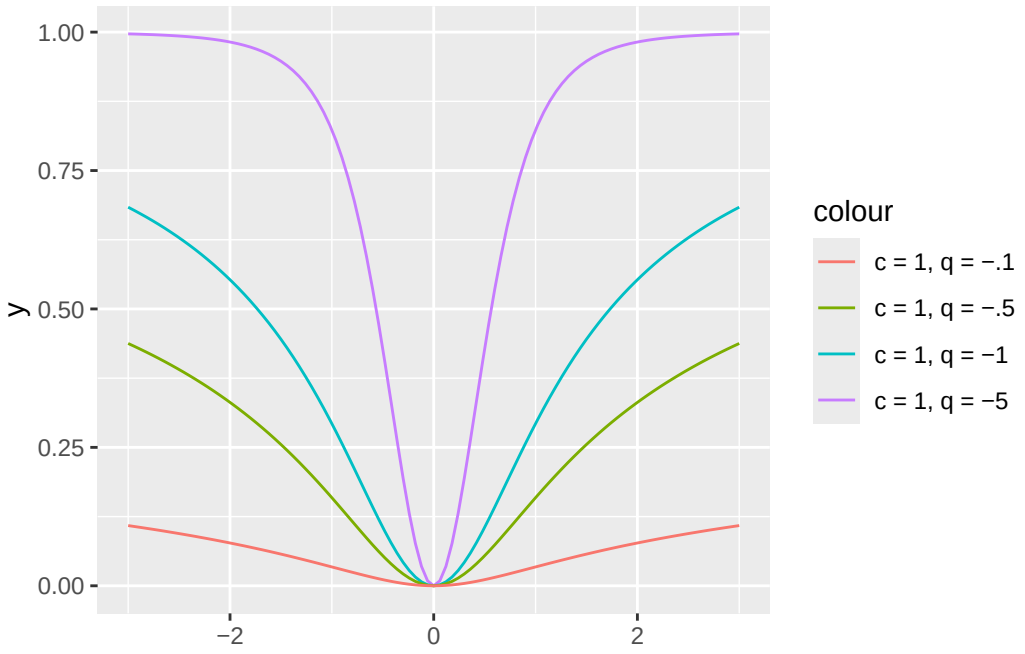

Figure 2: Generalized Charbonnier Loss

We see that for $\alpha \to -\infty$ generalized Charbonnier loss aproximates ℓ_0 loss.

4.3 Barron

There are a fair number of generalizations of the power smoother loss functions in the engineering literature. We will discuss one nice generalization from Barron (2019).

$$f_{\alpha,c}(x) = \frac{|\alpha - 2|}{\alpha}\left(\left(\frac{(x/c)^2}{|\alpha - 2|} + 1\right)^{\alpha/2} - 1\right). \tag{35}$$

To quote Barron

> Here $\alpha \in \mathbb{R}$ is a shape parameter that controls the robustness of the loss and $c > 0$ is a scale parameter that controls the size of the loss's quadratic bowl near $x = 0$.

A number of interesting special cases of Equation 35 are obtained by selecting various values of the α parameters. For $\alpha = 1$ it becomes Charbonnier loss, and for $\alpha = -2$ it is Geman-McClure loss. There are also some limiting cases. For $\alpha \to 2$ Barron loss becomes squared

error loss, for $\alpha \to 0$ it becomes Cauchy loss, and for $\alpha \to -\infty$ it becomes Welsch loss. Accordingly

$$f'_{\alpha,c}(x) = \begin{cases} \frac{x}{c^2} & \text{if } \alpha = 2, \\ \frac{2x}{x^2+2c^2} & \text{if } \alpha = 0, \\ \frac{x}{c^2} \exp\left(-\frac{1}{2}(x/c)^2\right) & \text{if } \alpha \to -\infty, \\ \frac{x}{c^2}\left(\frac{(x/c)^2}{|\alpha-2|}+1\right)^{\left(\frac{1}{2}\alpha-1\right)} & \text{otherwise.} \end{cases} \tag{36}$$

and thus

$$w_{\alpha,c}(x) = \begin{cases} \frac{1}{c^2} & \text{if } \alpha = 2, \\ \frac{2}{x^2+2c^2} & \text{if } \alpha = 0, \\ \frac{1}{c^2} \exp\left(-\frac{1}{2}(x/c)^2\right) & \text{if } \alpha \to -\infty, \\ \frac{1}{c^2}\left(\frac{(x/c)^2}{|\alpha-2|}+1\right)^{\left(\frac{1}{2}\alpha-1\right)} & \text{otherwise.} \end{cases} \tag{37}$$

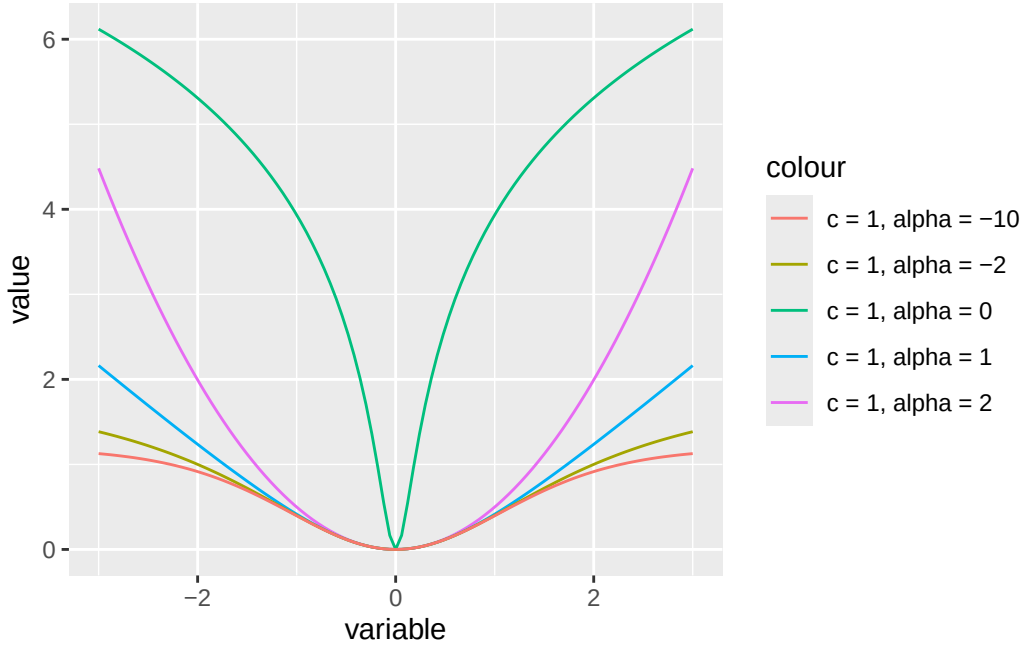

Figure 3: Barron Loss

5 Convolution Smoothers

Suppose π is a probability density, symmetric around zero, with finite or infinite support, expectation zero, and variance one. Define the convolution

$$f_c(x) := \frac{1}{c}\int_{-\infty}^{+\infty} |x-y|\,\pi(\frac{y}{c})dy.$$

Now $c^{-1}\pi(y/c)$ is still a symmetric probability density integrating to one, with expectation zero, but it now has variance c^2. Thus if $c \to 0$ it becomes more and more like the Dirac delta function and $f_c(x)$ converges to the absolute value function.

It is clear that we can use any scale family of probability densities to define convolution smoothers. There is an infinite number of possible choices, with finite or infinite support, smooth or nonsmooth, using splines or wavelets, and so on. We give two quite different examples.

5.1 Huber

Take

$$\pi(x) = \begin{cases} \frac{1}{2} & \text{if } |x| \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$f_c(x) = \frac{1}{2c}\int_{-c}^{+c} |x-y|dy = \begin{cases} \frac{1}{2c}(x^2+c^2) & \text{if } |x| \leq c, \\ |x| & \text{otherwise.} \end{cases} \tag{38}$$

The Huber function (Huber (1964)) is traditionally transformed linearly so that it is zero for $x = 0$. This gives

$$f_c(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| < c, \\ c|x| - \frac{1}{2}c^2 & \text{otherwise.} \end{cases} \tag{39}$$

For robust estimation and IRLS it does not matter if we use Equation 38 or Equation 39. Our discussion in the introduction suggests that if we just want a smoother of the absolute value function, then Equation 38 is the natural choice, if we want a robust loss function that combines the advantages of least squares and least absolute value then that leads us to Equation 39.

Because Charbonnier loss behaves the same way as Huber loss, as absolute value loss for large x and as squared loss for small x, it is also known as Pseudo-Huber loss.

The Huber function is differentiable, although not twice diffentiable. Its derivative is

$$f'(x) = \begin{cases} c & \text{if } x \geq c, \\ x & \text{if } |x| \leq c, \\ -c & \text{if } x \leq -c. \end{cases}$$

$$\omega(x) = \begin{cases} \frac{c}{x} & \text{if } x \geq c, \\ 1 & \text{if } |x| \leq c, \\ -\frac{c}{x} & \text{if } x \leq -c. \end{cases}$$

The Huber function is even and differentiable. Moreover $f'(x)/x$ decreases from. Thus Theorem 3.7 applies.

The MDS majorization algorithm for the Huber loss is to update Y by minimizing (or by performing one smacof step to decrease)

$$\sum \omega_k(Y)(\delta_k - d_k(X))^2$$

where

$$\omega_k(Y) = \begin{cases} \omega_k & \text{if } |\delta_k - d_k(Y)| < c, \\ \frac{c\omega_k}{|\delta_k - d_k(Y)|} & \text{otherwise.} \end{cases}$$

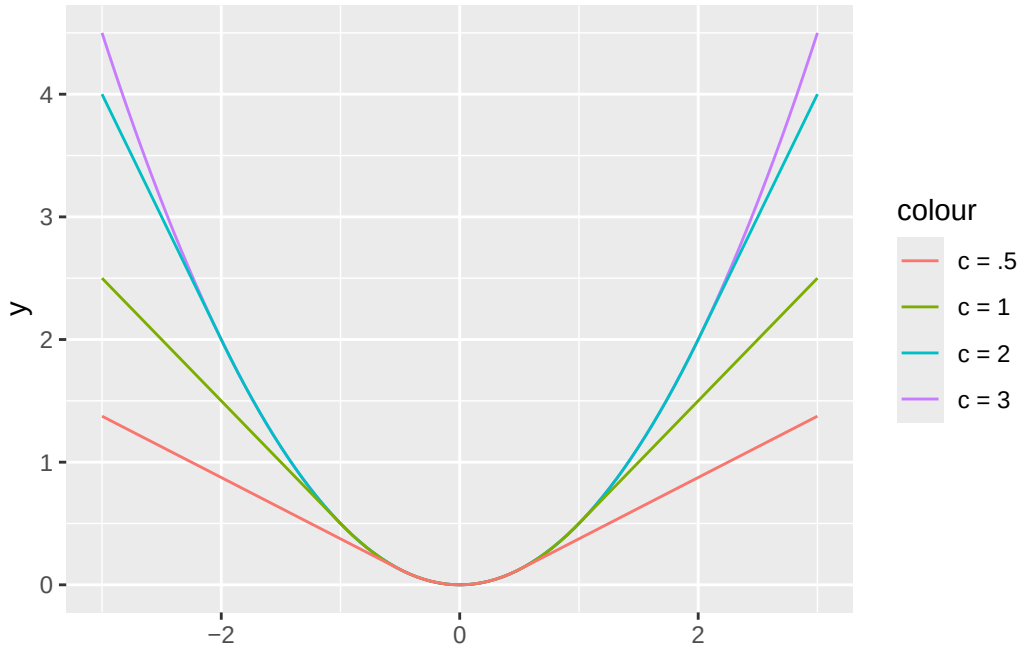

Figure 4: Huber Loss

5.2 Gaussian

In De Leeuw (2018) we also discussed the convolution smoother proposed by Voronin, Ozkaya, and Yoshida (2014). The idea is to use the convolution of the absolute value function and a Gaussian pdf.

$$f(x) = \frac{1}{c\sqrt{2\pi}} \int_{-\infty}^{+\infty} |x-y| \exp\left\{-\frac{1}{2}(\frac{y}{c})^2\right\} dy$$

Carrying out the integration gives

$$f_c(x) = x\{2\Phi(x/c) - 1\} + 2c\phi(x/c).$$

The derivative is

$$f_c'(x) = 2\Phi(x/c) - 1$$

It may not be immediately obvious in this case that the weight function $f'(x)/x$ is non-increasing on $\mathbb{R}^+$. We prove that its derivative is negative on $(0, +\infty)$. The derivative of $f'(x)/x$ has the sign of $xf''(x) - f'(x)$, which is $z\phi(z) - \Phi(z) + 1/2$, with $z = x/c$. It remains to show that $\Phi(z) - z\phi(z) \geq \frac{1}{2}$, or equivalently that $\int_0^z \phi(x)dx - z\phi(z) \geq 0$. Now if $0 \leq x \leq z$ then $\phi(x) \geq \phi(z)$ and thus $\int_0^z \phi(x)dx \geq \phi(z) \int_0^z dx = z\phi(z)$, which completes the proof.

$$\omega_k(Y) = \frac{\Phi((\delta_k - d_k(Y))/c) - \frac{1}{2}}{\delta_k - d_k(Y)}$$

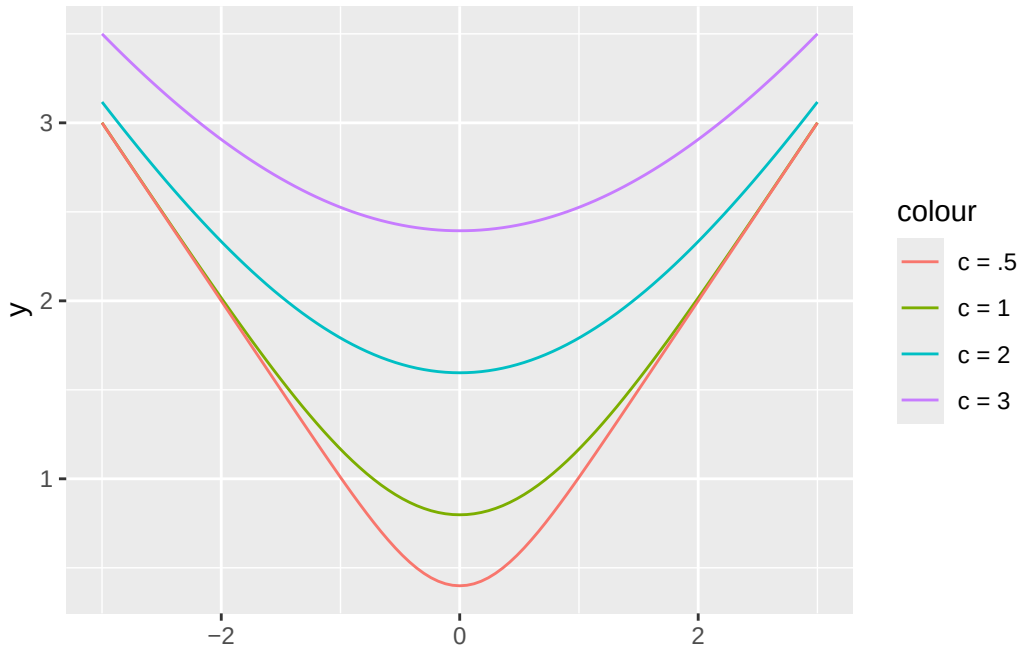

Figure 5: Gaussian Convolution Loss

6 A Bouquet of Loss Functions

In the early seventies, after the pioneering mostly theoretical work in robust statistics of Huber, Hampel, and Tukey, the mainframe computer allowed statisticians to make large-scale comparisons of many robust loss functions. The most impressive of such comparisons was the Princeton Robustness Study (Andrews et al. (1972)).

In Holland and Welsch (1977) the computer package ROSEPACK was introduced that made it relatively easy to compute robust estimators using several different loss functions. Eight different weight functions were implemented as options. Somewhat later Coleman et al. (1980) made an more modern computer implementation available, using the same eight weight functions, which was not limited to mainframes.

We have implemented the same eight weight functions in smacofRobust. Below we give formulas for the loss function, the influence function, and the weight function. One of the eight is Huber loss, which we already discussed in the convolution section. We graph the remaining seven loss functions for selected values of the "tuning constants" c.

Holland and Welsch (1977), following Andrews et al. (1972), distnguish between "hard redescenders" that have an influence function f' equal to zero if x is large enough (Andrews, Tukey, and Hinich loss), "soft redescenders" with influence functions asymptotic to zero for large x (Cauchy, Welsch loss), and loss functions with a monotone influence function (Huber, Logistic, Fair loss)

6.1 Andrews

The first loss function in this section is taken from Andrews et al. (1972).

$$f(x) = \begin{cases} c^2(1-\cos(x/c)) & \text{if } |x| \leq \pi c, \\ 2c^2 & \text{otherwise.} \end{cases} \tag{40}$$

$$f'(x) = \begin{cases} c\sin(x/c) & \text{if } |x| \leq \pi c, \\ 0 & \text{otherwise.} \end{cases} \tag{41}$$

$$\omega(x) = \begin{cases} (x/c)^{-1}\sin(x/c) & \text{if } |x| \leq \pi c, \\ 0 & \text{otherwise.} \end{cases} \tag{42}$$

Because cos is even and $\sin(x)/x$ decreases on $[0, \pi]$ Theorem 3.7 applies.

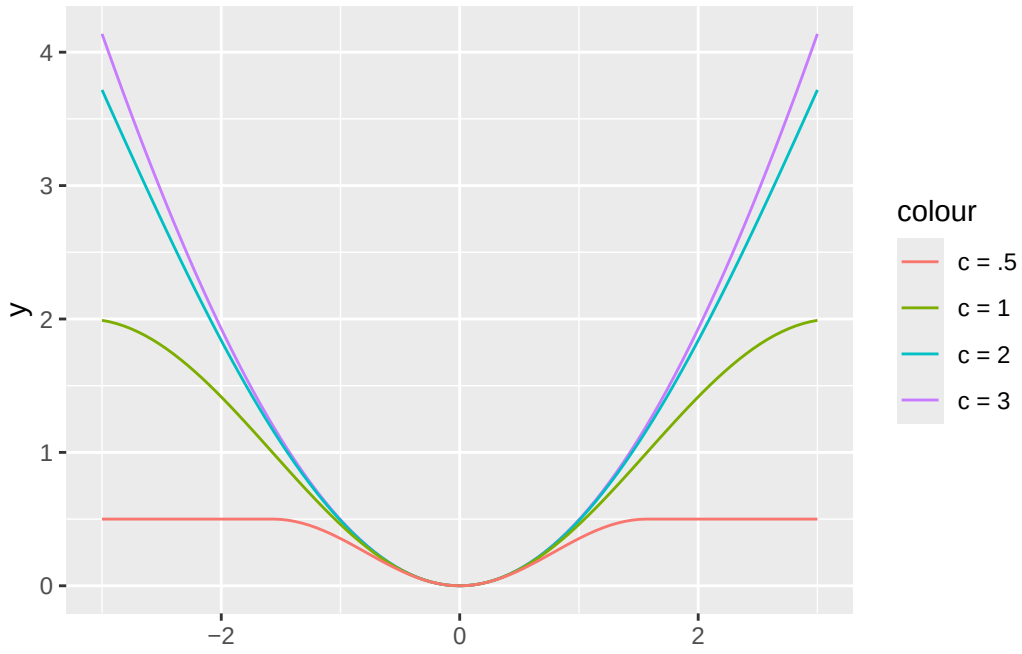

Figure 6: Andrews Loss

6.2 Tukey

The usual reference for Tukey loss is Beaton and Tukey (1974), although closely related hard redescenders are also in Andrews et al. (1972).

$$f(x) = \begin{cases} \frac{c^2}{6}\left(1-\left(1-(x/c)^2\right)^3\right) & \text{if } |x| \leq c, \\ \frac{c^2}{6} & \text{otherwise.} \end{cases} \tag{43}$$

$$f'(x) = \begin{cases} x\left(1-\left(1-(x/c)^2\right)^2\right) & \text{if } |x| \leq c, \\ 0 & \text{otherwise.} \end{cases} \tag{44}$$

$$\omega(x) = \begin{cases} \left(1-\left(1-(x/c)^2\right)^2\right) & \text{if } |x| \leq c, \\ 0 & \text{otherwise.} \end{cases} \tag{45}$$

The conditions of Theorem 3.7 are clearly satisfied.

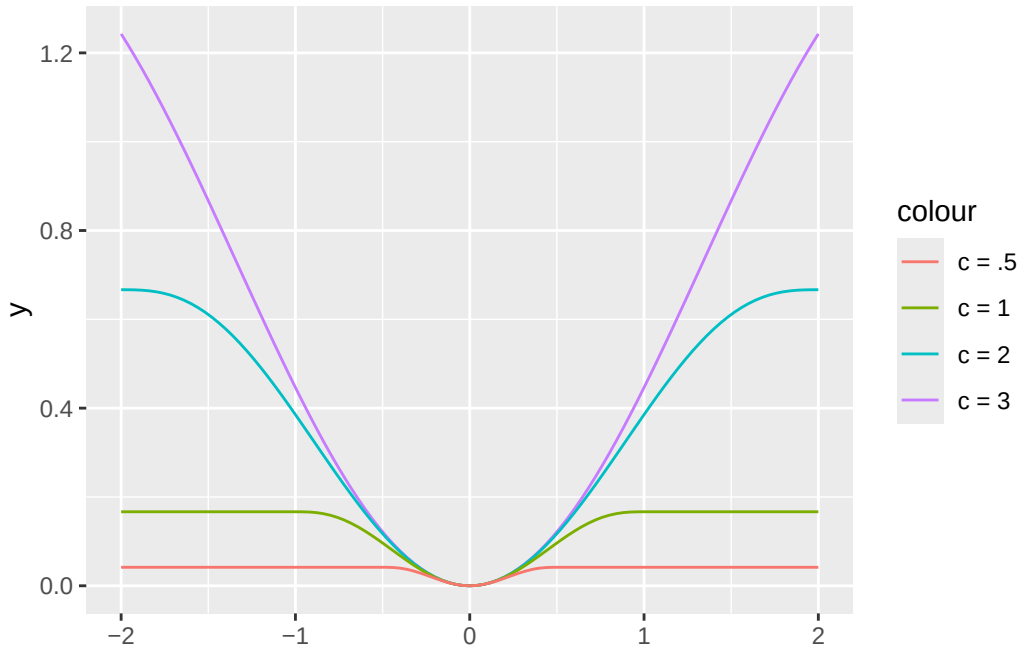

Figure 7: Tukey Loss

6.3 Hinich

Hinich loss, from Hinich and Talwar (1975), is somewhat special because it is even but not differentiable at c. For $x \neq c$ and $x > 0$ the function $f'(x)/x$ is non-increasing.

$$f(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| \leq c, \\ \frac{1}{2}c^2 & \text{otherwise.} \end{cases} \tag{46}$$

$$f'(x) = \begin{cases} 1 & \text{if } |x| \leq c, \\ 0 & \text{otherwise.} \end{cases} \tag{47}$$

$$\omega(x) = \begin{cases} 1/x & \text{if } |x| \leq c, \\ 0 & \text{otherwise.} \end{cases} \tag{48}$$

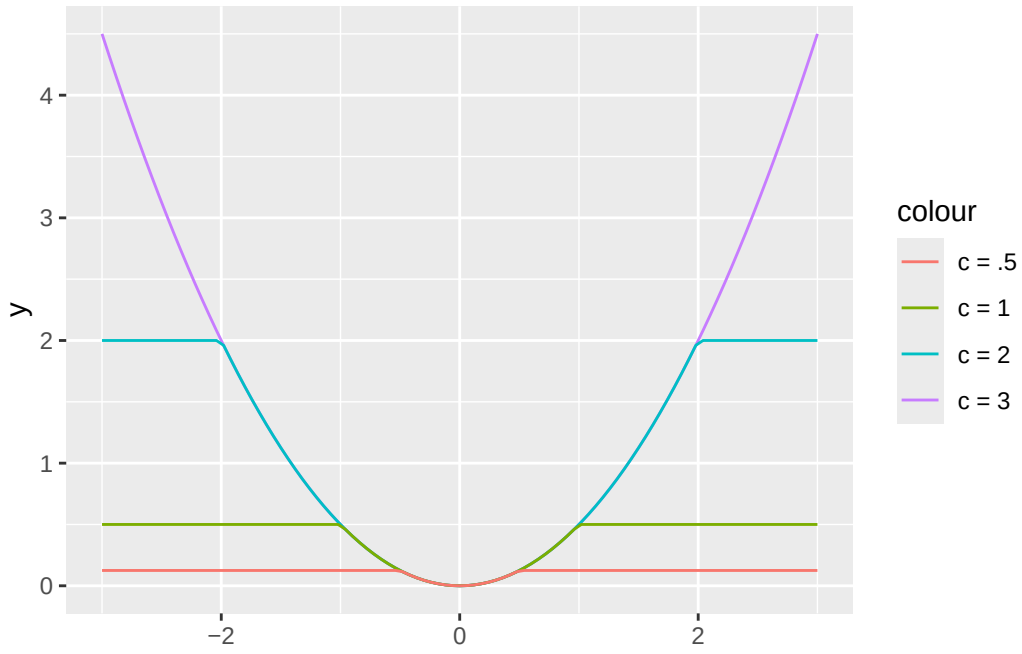

Figure 8: Hinich Loss

6.4 Cauchy

Cauchy loss seems to have many names. Black and Anandan (1996) call it Lorentzian loss, and Holland and Welsch (1977) call it t-likelihood loss. It is related to the Cauchy distribution, which is Student's t distribution with one degree of freedom.

Mlotshwa, Van Deventer, and Sergeevna Bosman (2023)

$$f(x) = \frac{1}{2}c^2 \log(1 + \{\frac{x}{c}\}^2), \tag{49}$$

$$f'(x) = x\frac{1}{\{1 + \frac{x}{c}\}^2}, \tag{50}$$

$$\omega(x) = \frac{1}{\{1 + \frac{x}{c}\}^2} \tag{51}$$

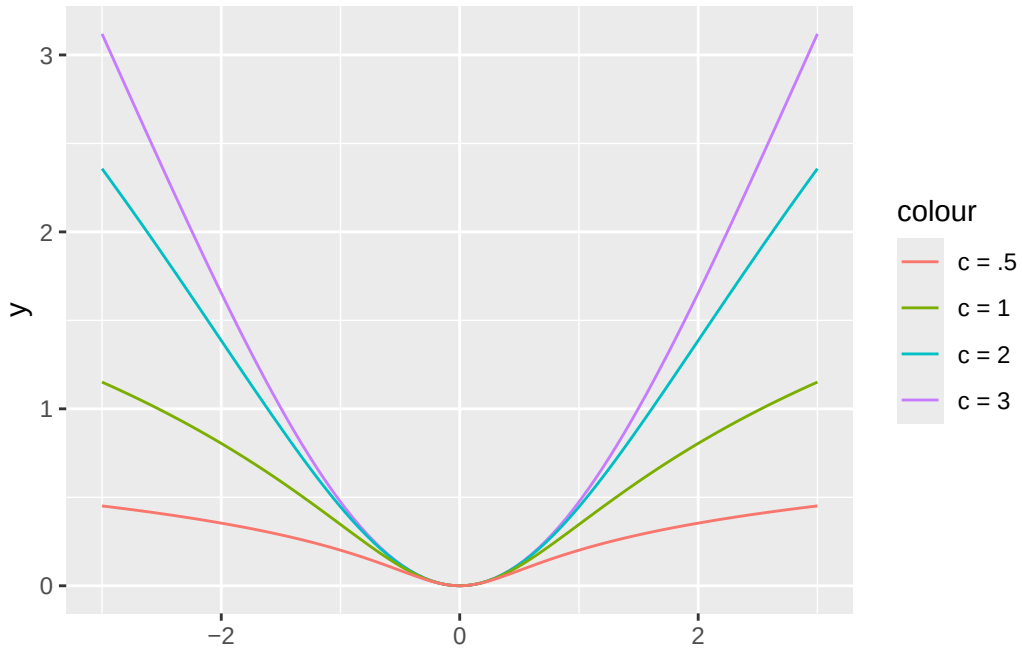

Figure 9: Cauchy Loss

6.5 Welsch

Dennis Jr and Welsch (1978)

Leclerc loss

$$f(x) = \frac{1}{2}c^2[1 - \exp(-\{\frac{x}{c}\}^2)], \tag{52}$$

$$f'(x) = x\exp(-\{\frac{x}{c}\}^2, \tag{53}$$

$$\omega(x) = \exp(-\{\frac{x}{c}\}^2), \tag{54}$$

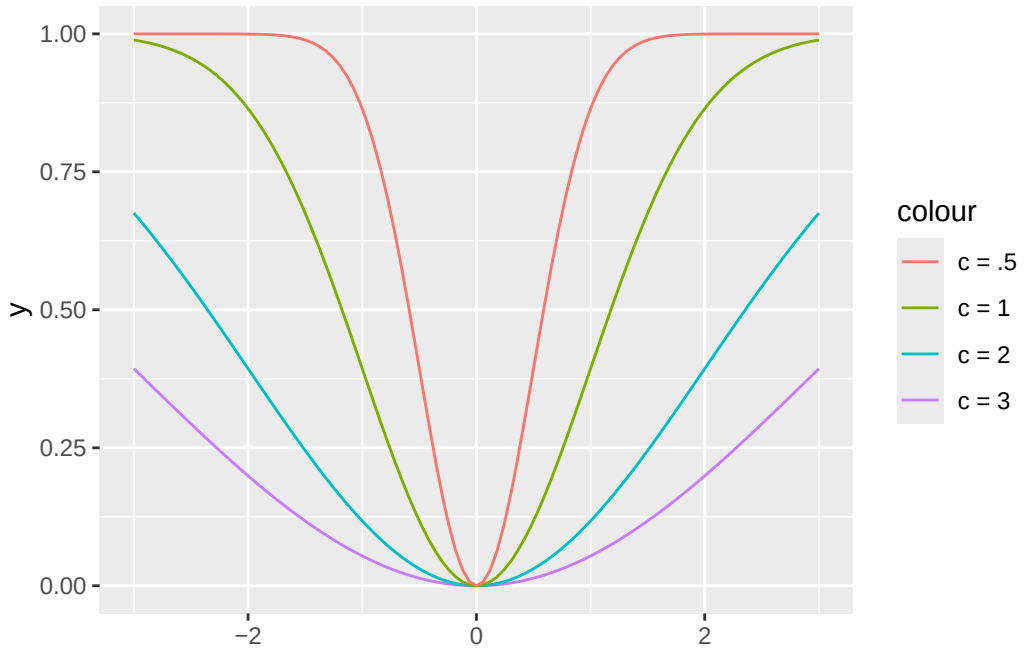

Figure 10: Welsch Loss

6.6 Logistic

$$f(x) = c^2[\log(\cosh(x/c))], \tag{55}$$
$$f'(x) = c\tanh(x/c), \tag{56}$$
$$\omega(x) = (x/c)^{-1}\tanh(x/c). \tag{57}$$

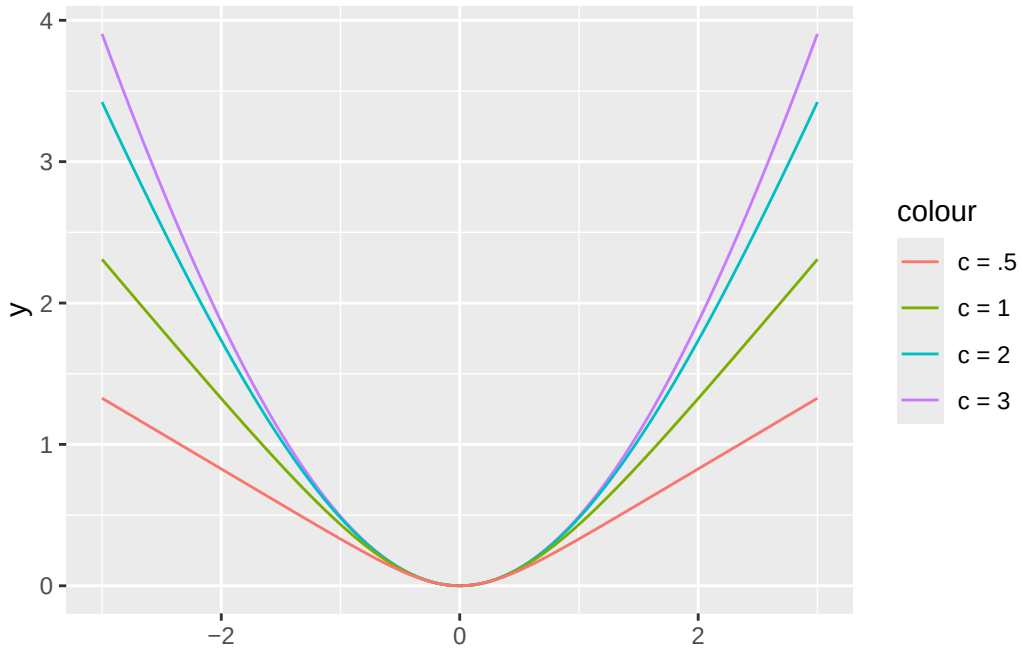

Figure 11: Logistic Loss

6.7 Fair

$$f(x) = c^2\{|x|/c - \log(1 + |x|/c)\}, \tag{58}$$
$$f'(x) = x(1 + (|x|/c))^{-1}, \tag{59}$$
$$\omega(x) = (1 + (|x|/c))^{-1}. \tag{60}$$

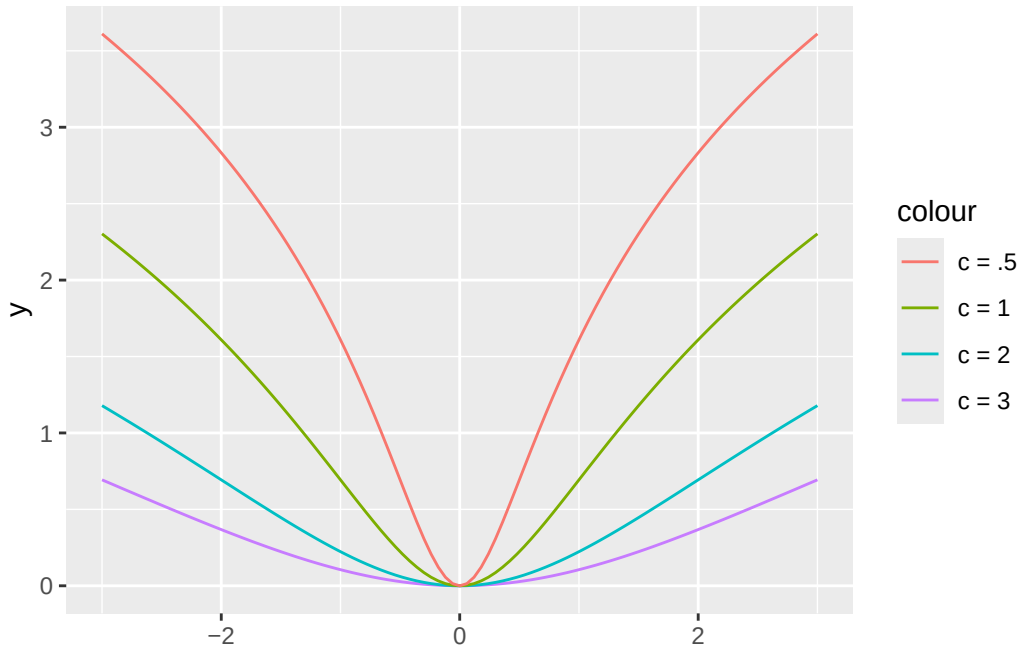

Figure 12: Fair Loss

7 Examples

7.1 Gruijter

The example we use are dissimilarities between nine Dutch political parties, collected by De Gruijter (1967). They are averages over a politically heterogenous group of 100 introductory psychology students, and consequently they regress to the mean. Any reasonable MDS analysis of these data would at least allow for an additive constant.

Some background on Dutch politics around that time may be useful.

- CPN - Communists.
- PSP - Pacifists, left-wing.
- PvdA - Labour, Democratic Socialists.
- D'66 - Pragmatists, nether left-wing nor right-wing, brand new in 1967.
- KVP - Christian Democrats, catholic.
- ARP - Christian Democrats, protestant.
- CHU - Christian Democrats, protestant.
- VVD - Liberals, European flavour, conservative.
- BP - Farmers, protest party, right-wing.

The dissimilarities are in the table below.

	KVP	PvdA	VVD	ARP	CHU	CPN	PSP	BP	D66
KVP	0.00	5.63	5.27	4.60	4.80	7.54	6.73	7.18	6.17
PvdA	5.63	0.00	6.72	5.64	6.22	5.12	4.59	7.22	5.47
VVD	5.27	6.72	0.00	5.46	4.97	8.13	7.55	6.90	4.67
ARP	4.60	5.64	5.46	0.00	3.20	7.84	6.73	7.28	6.13
CHU	4.80	6.22	4.97	3.20	0.00	7.80	7.08	6.96	6.04
CPN	7.54	5.12	8.13	7.84	7.80	0.00	4.08	6.34	7.42
PSP	6.73	4.59	7.55	6.73	7.08	4.08	0.00	6.88	6.36
BP	7.18	7.22	6.90	7.28	6.96	6.34	6.88	0.00	7.36
D66	6.17	5.47	4.67	6.13	6.04	7.42	6.36	7.36	0.00

The reason we have chosen this example is partly because CPN and BP are outliers, and we can expect the robust loss functions to handle outlying dissimilarities differently from the bulk of the data.

Unless otherwise indicated we run smacofRobust() with a maximum of 10,000 iterations, and we decide that we have convergence if the difference between consecutive stress values is less than 10^{-15}. We perform one single smacof iteration between the updates of the weights. For

each analysis we show the configuration plot, the Shepard plot, and a histogram of the absolute values of the residuals. In the Shepard plot points corresponding to the eight CPN-dissimilarities are labeled "C", while BP-dissimilarities are "B".

7.1.1 Least Squares

We start with a least squares analysis, actually with Huber loss with $c = 10$, which for these data is equivalent to least squares. The process converges in 859 iterations.

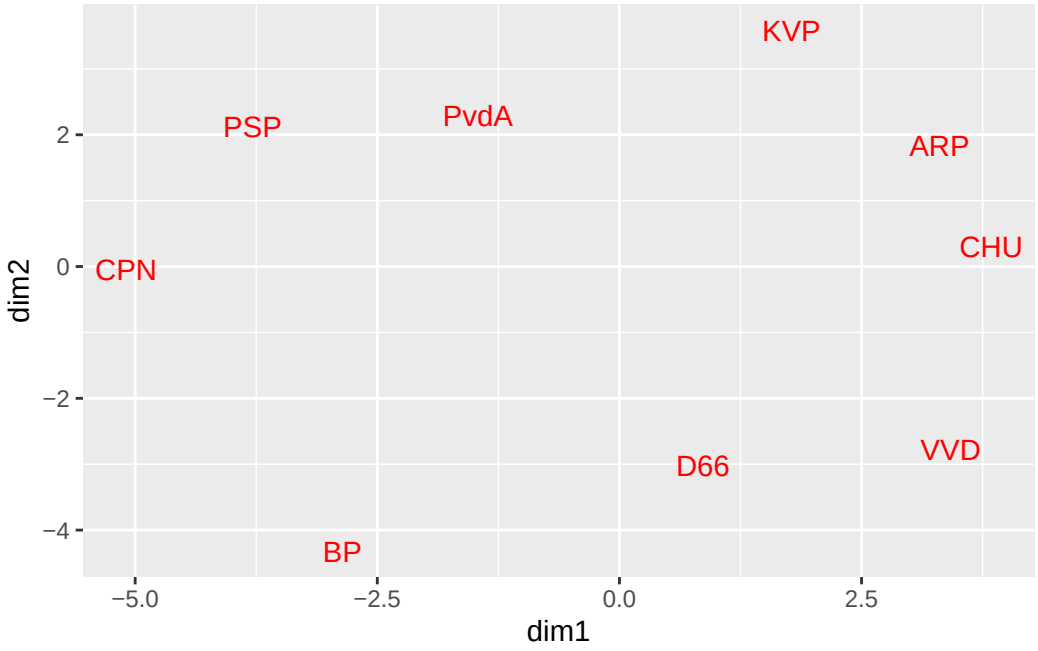

Figure 13: Gruijter Configuration Least Squares

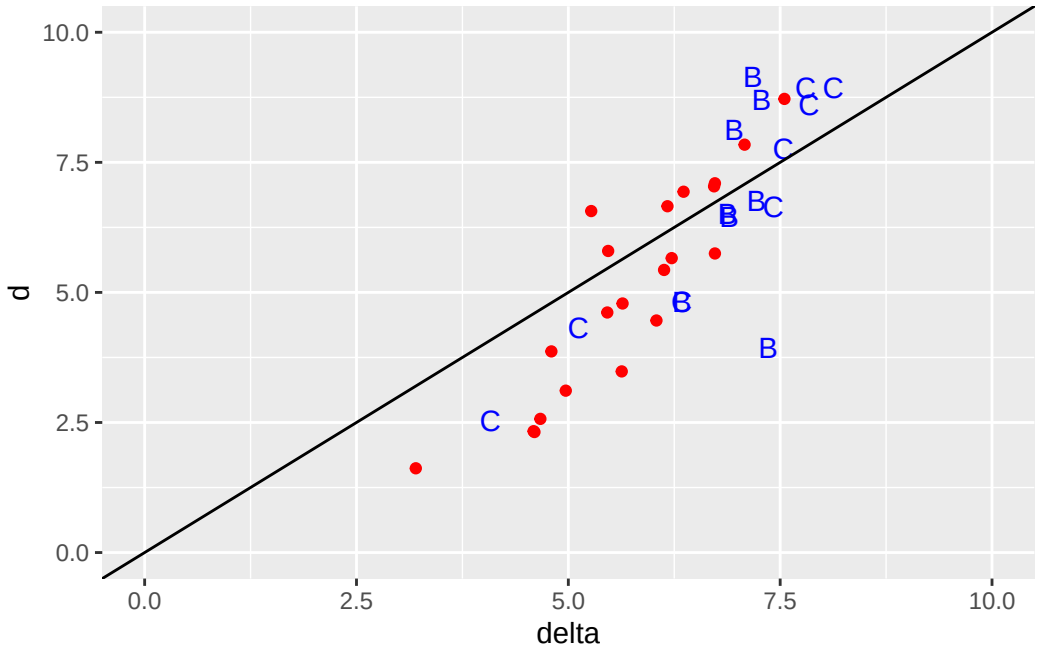

Figure 14: Grujter Shepard Plot Least Squares

The Shepard plot clearly shows why an additive constant would be very beneficial in this case.

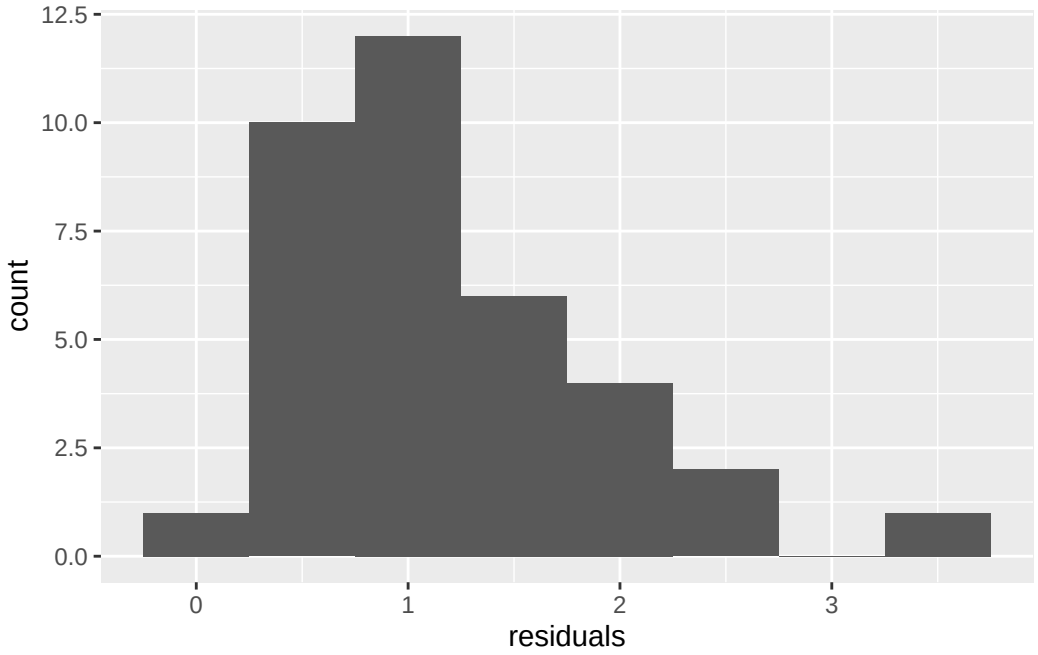

Figure 15: Gruijter Histogram Least Squares Residuals

7.1.2 Least Absolute Value

For our LAV smacof we use engine smacofCharbonnier with $c = .001$. We have convergence in 637 iterations.

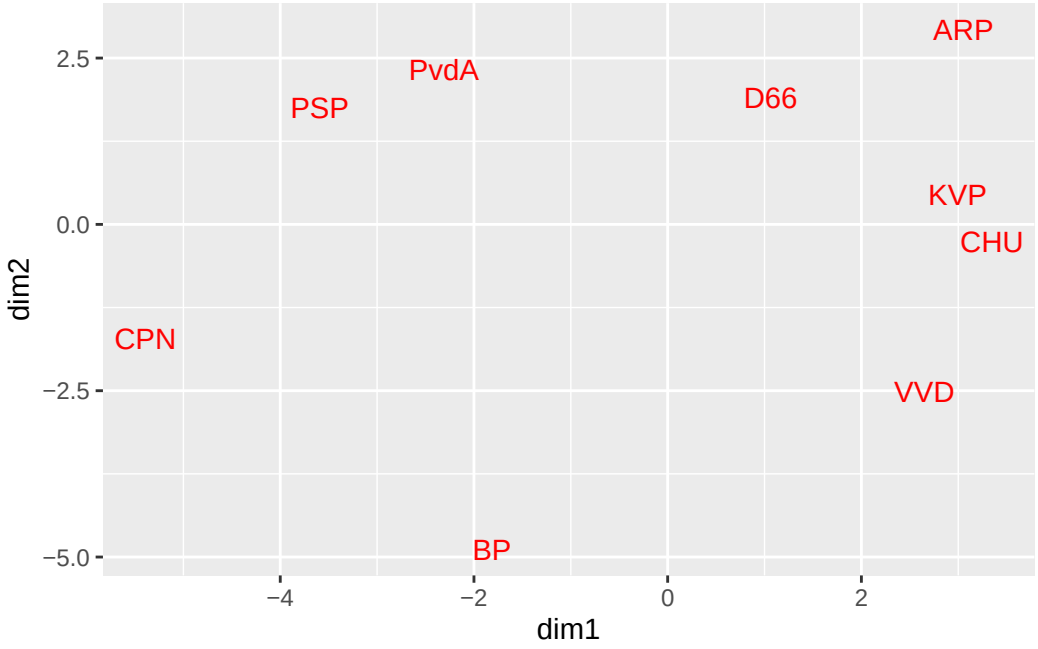

Figure 16: Gruijter Configuration Least Absolute Value

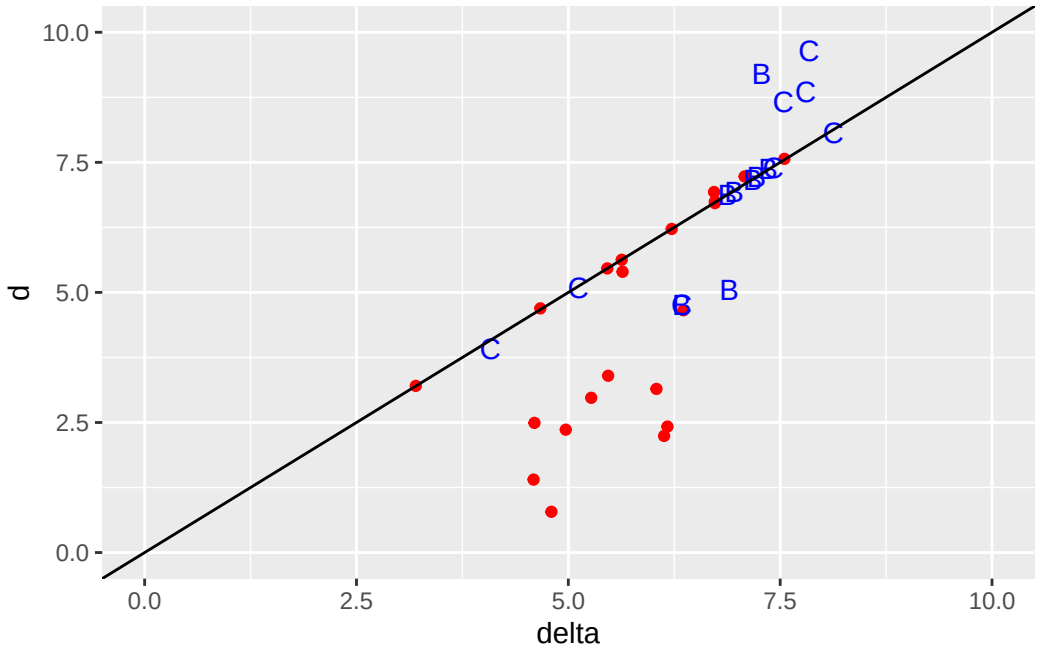

Figure 17: Gruijter Shepard Plot Least Absolute Value

In the Shepard plot we see that there are a number of dissimilarities which are fitted exactly. If we count them there are about 15-20. Note that configurations in two dimensions have $(n-1)+(n-2)=2n-3$ degrees of freedom, which is 15 in this case. Thus if we take the 15 dissimilarities which are fitted exactly, give them weight one, give all other 21 dissimilarities weight zero, and do a regular non-robust smacof analysis using these weights, then we will have perfect fit in two dimensions, and the solution will be the LAV solution. All this is easier said than done, because it presumes that we use Charbonnier loss with $c=0$ and that we are able to decide which residuals are exactly equal to zero. The LAV analysis also suggests the possibility of a huge number of local minima, because there are so many ways to pick 15 out of 36 dissimilarities.

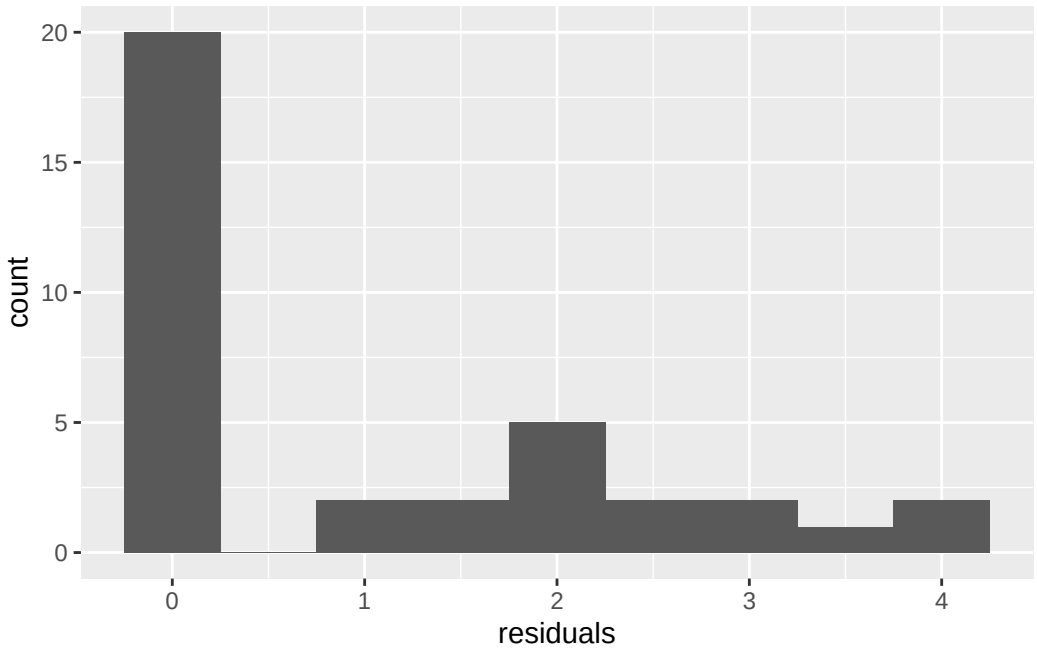

Figure 18: Gruijter Histogram Least Absolute Value Residuals

7.1.3 Huber

smacofHuber with $c = 1$ converges in 165 iterations.

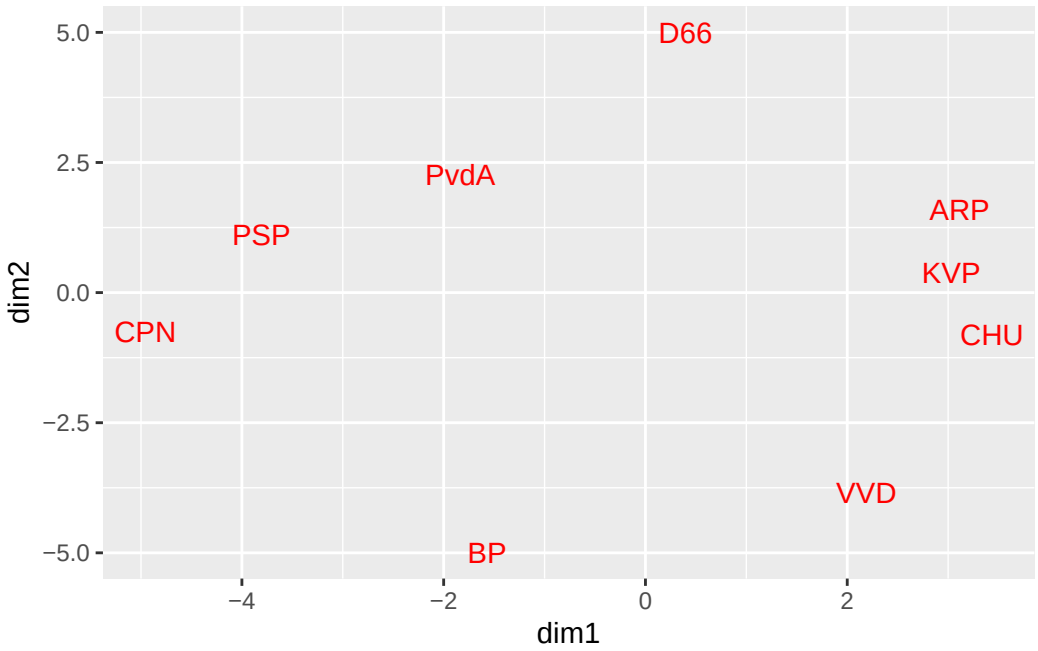

Figure 19: Gruijter Configuration Huber c = 1

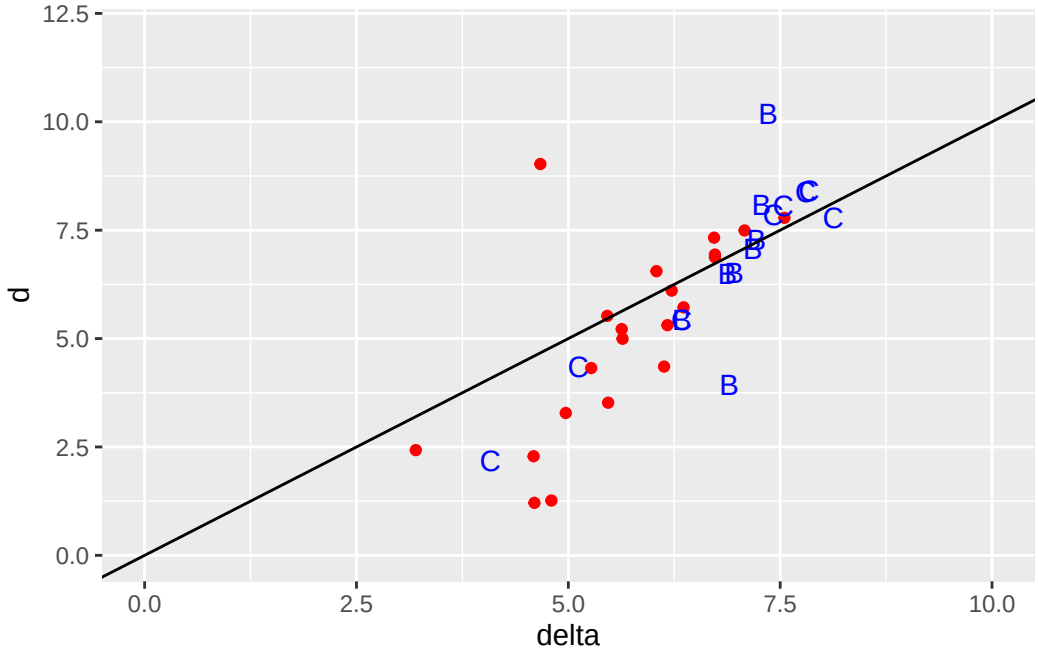

Figure 20: Gruijter Shepard Plot Huber c = 1

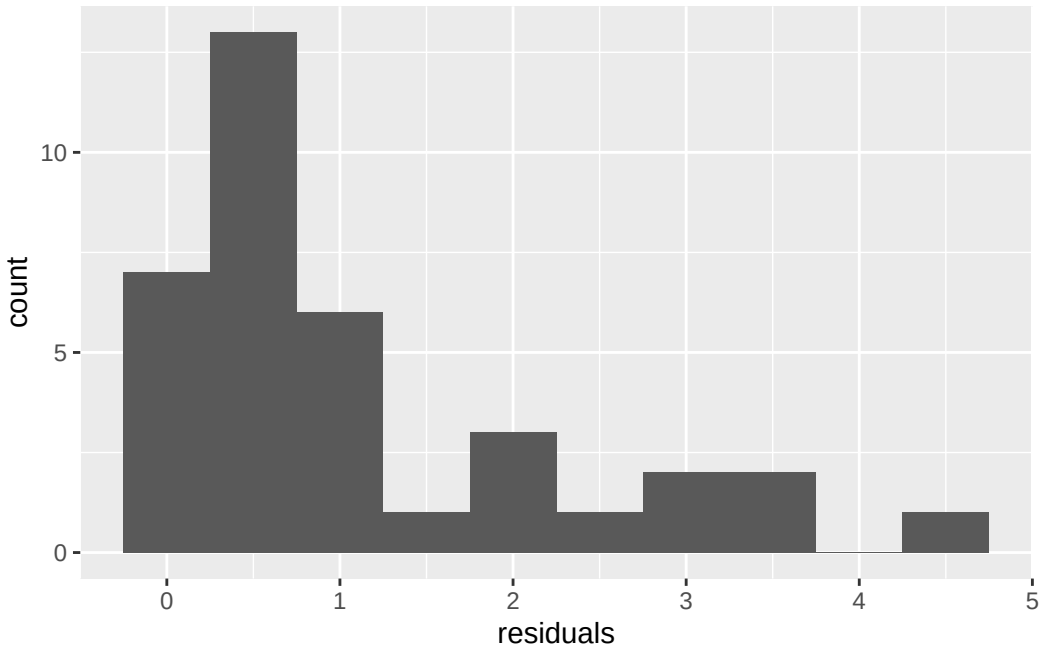

Figure 21: Gruijter Histogram Huber Residuals

7.1.4 Tukey

smacofTukey with $c = 2$ converges in 180 iterations.

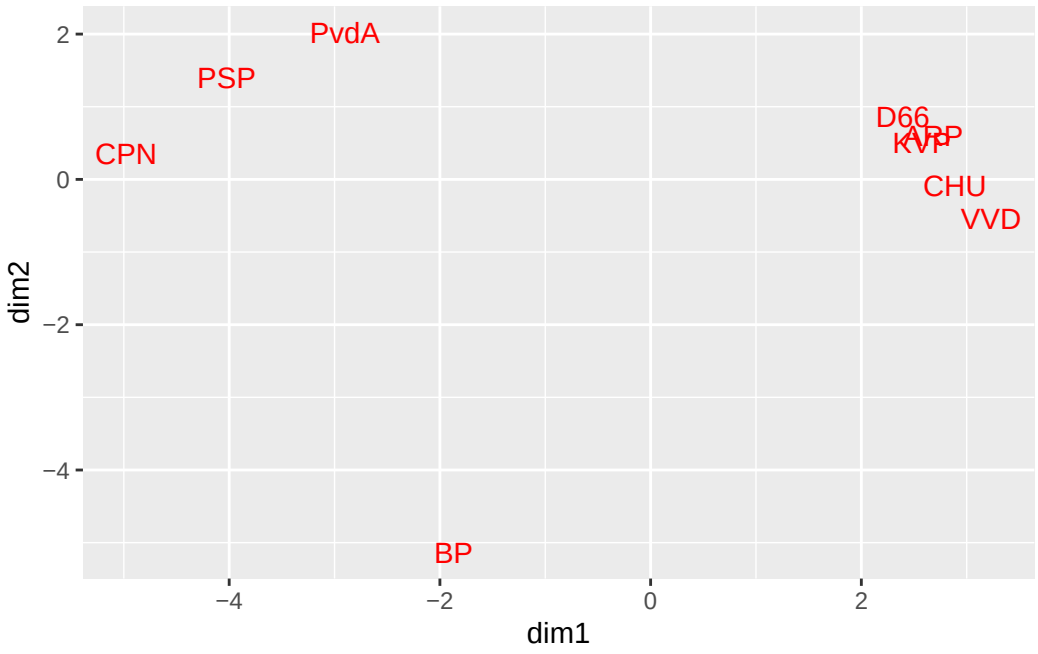

Figure 22: Gruijter Configuration Tukey c = 2

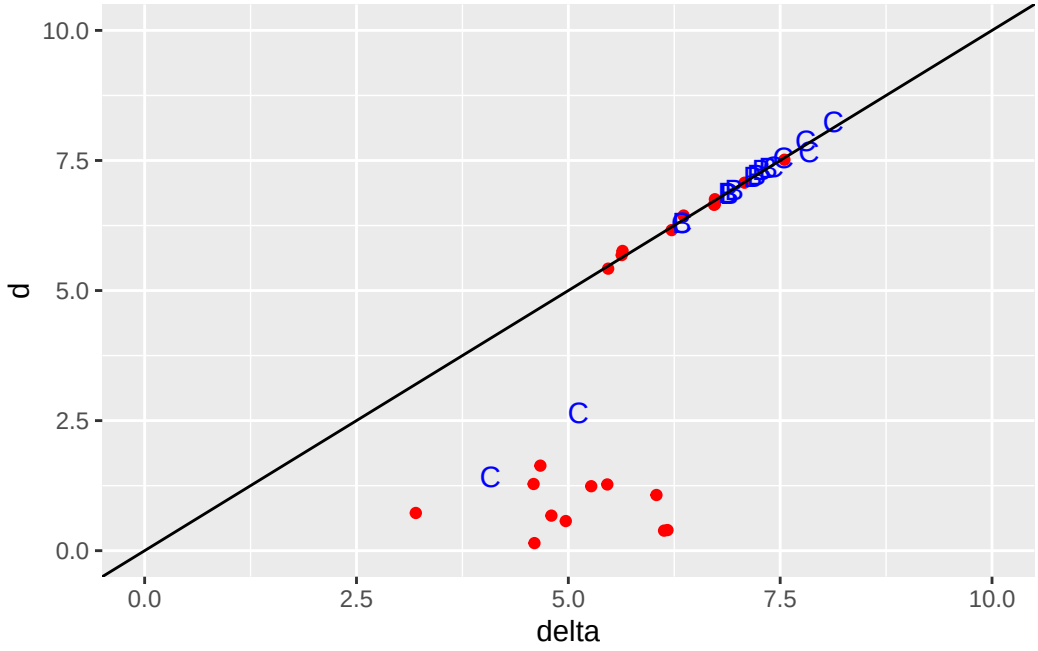

Figure 23: Gruijter Shepard Plot Tukey c = 2

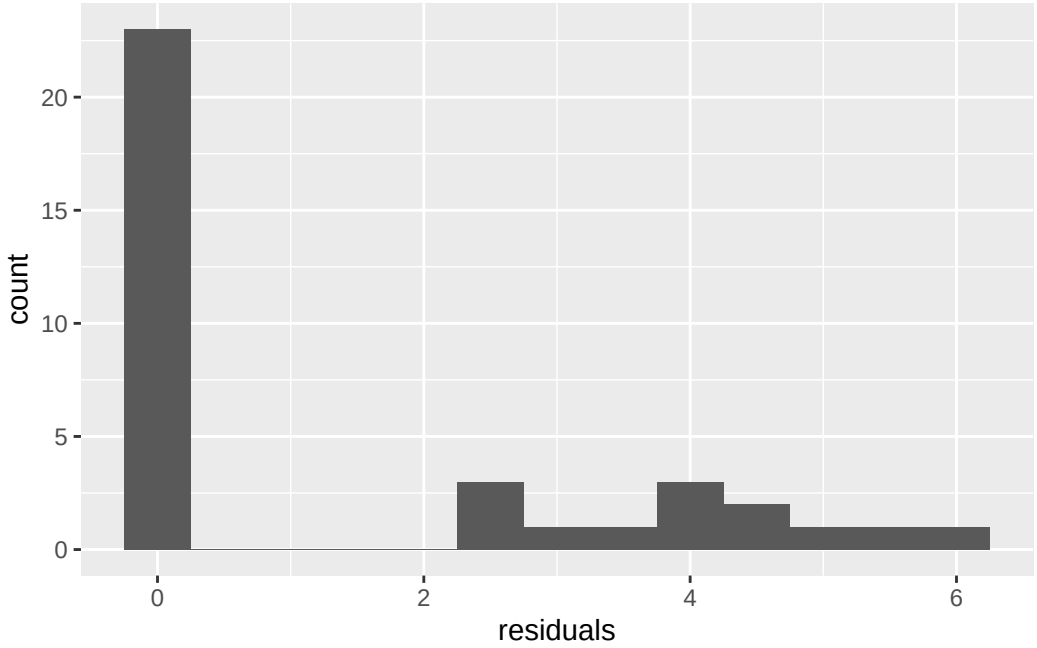

Figure 24: Gruijter Histogram Tukey Residuals

7.2 Rothkopf

Our second example are the Rothkopf Morse data (Rothkopf (1957)), which have a better fit and have fewer outliers than the Gruijter data. We used the asymetric confusion matrix from the smacof package (De Leeuw and Mair (2009)) and defined dissimilarities by the Shepard-Luce formula

$$\delta_{ij} = -\log \frac{p_{ij}p_{ji}}{p_{ii}p_{jj}}.$$

7.2.1 Least Squares

For least squares we use the smacofHuber engine with $c = 25$, well outside the range of the residuals. We have convergence in 213 iterations.

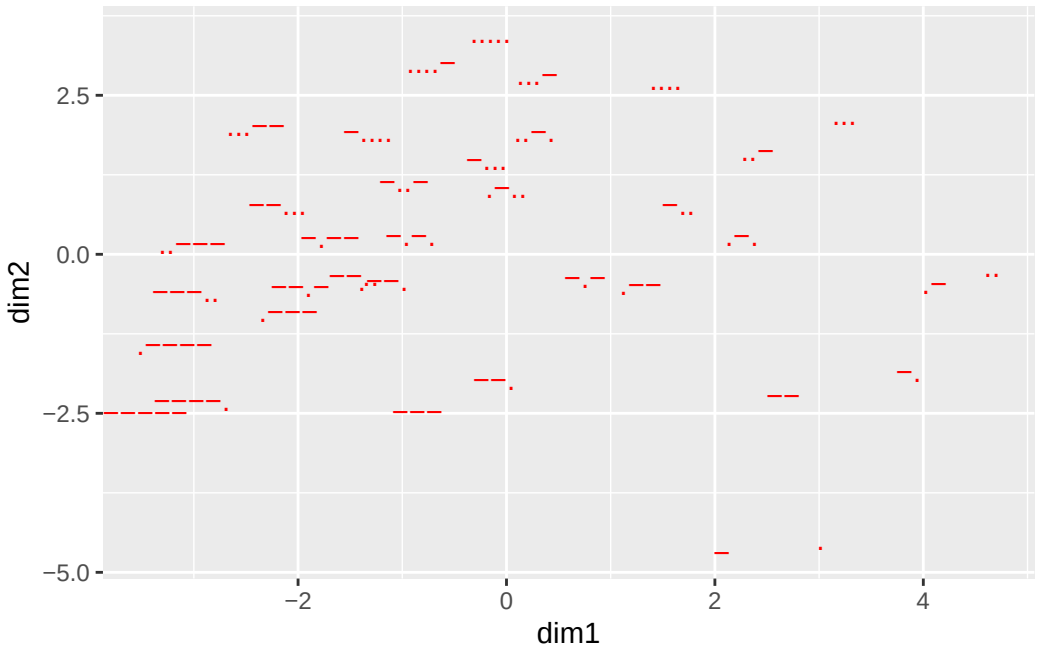

Figure 25: Rothkopf Configuration Least Squares

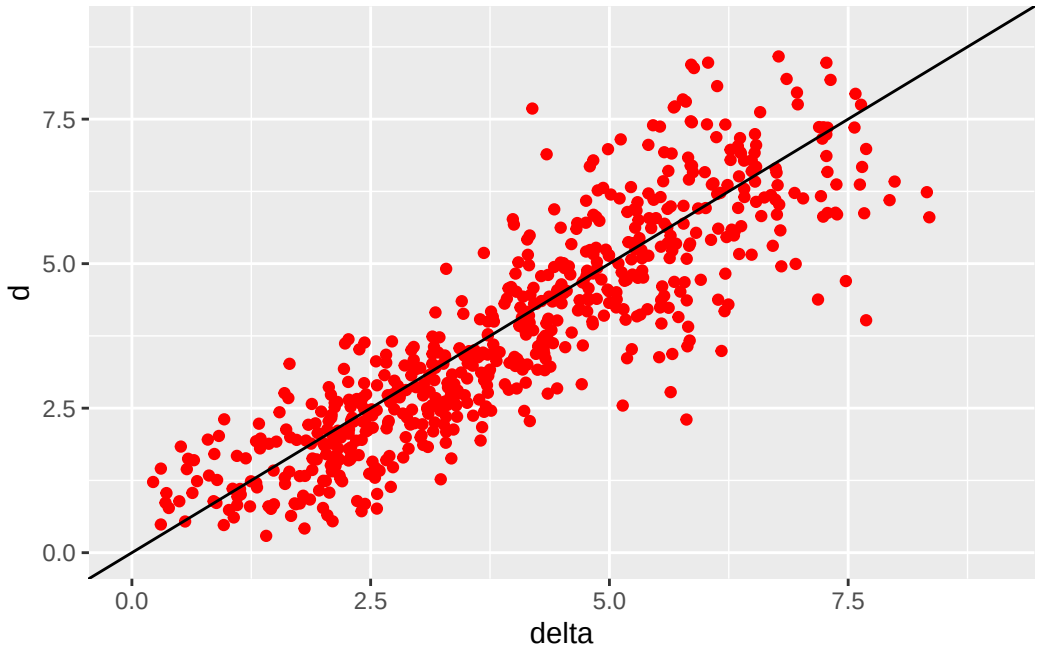

Figure 26: Rothkopf Shepard Plot Least Squares

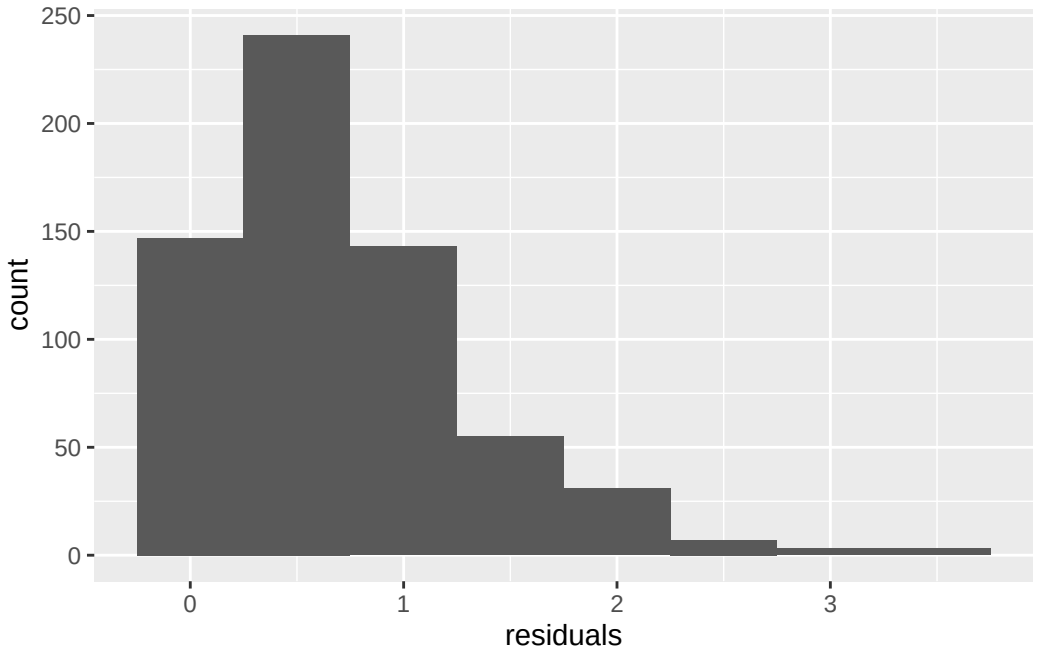

Figure 27: Rothkopf Histogram Least Squares Residuals

7.2.2 Least Absolute Value

For least absolute value we use Chardonnier loss with $c = .001$. We have convergence in 2291 iterations.

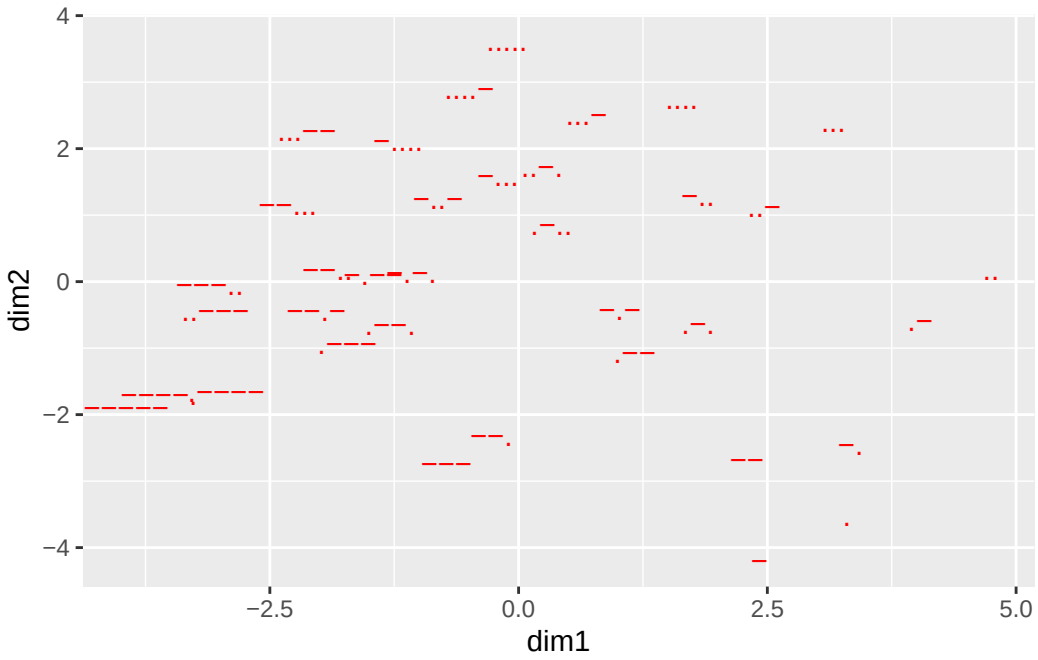

Figure 28: Rothkopf Configuration Least Absolute Value

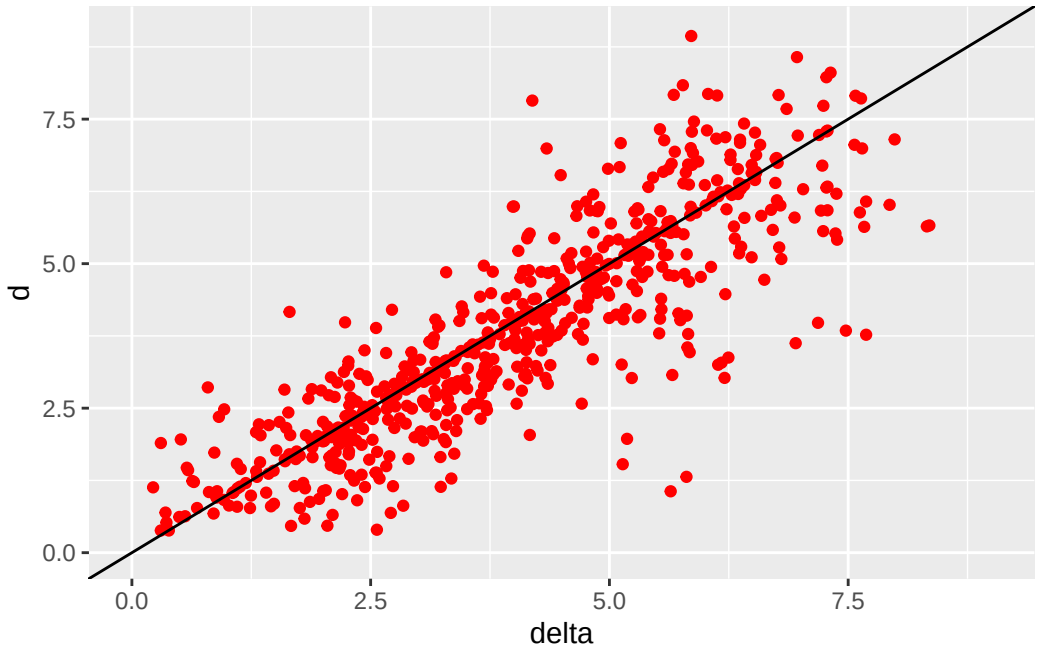

Figure 29: Rothkopf Shepard Plot Least Absolute Value

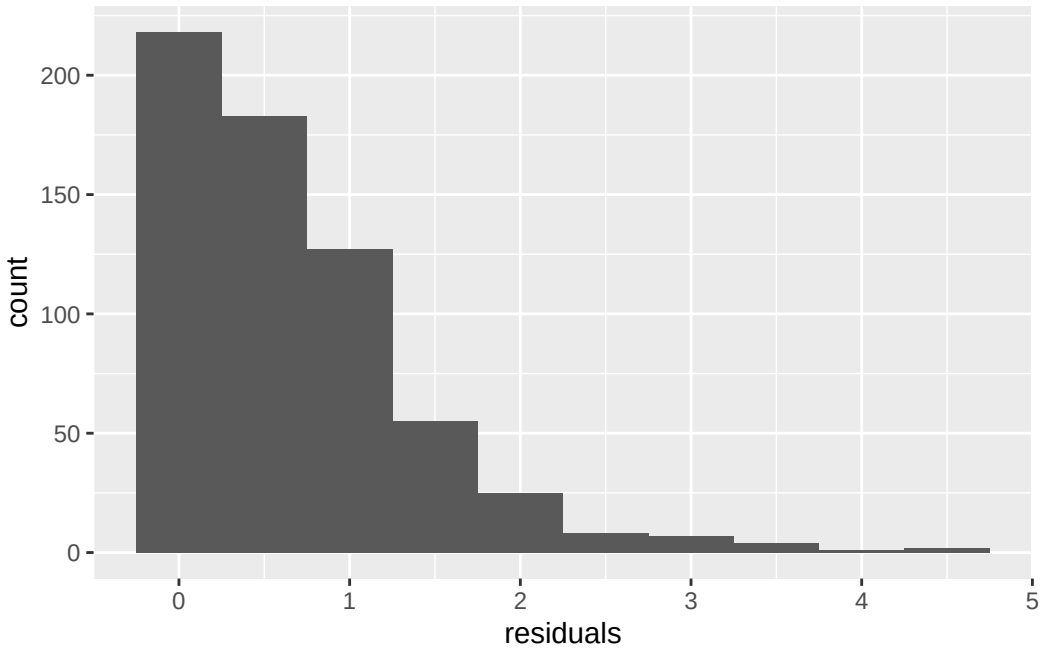

Figure 30: Rothkopf Histogram Least Absolute Value Residuals

7.2.3 Huber

smacofHuber with $c = 1$ converges in 680 iterations.

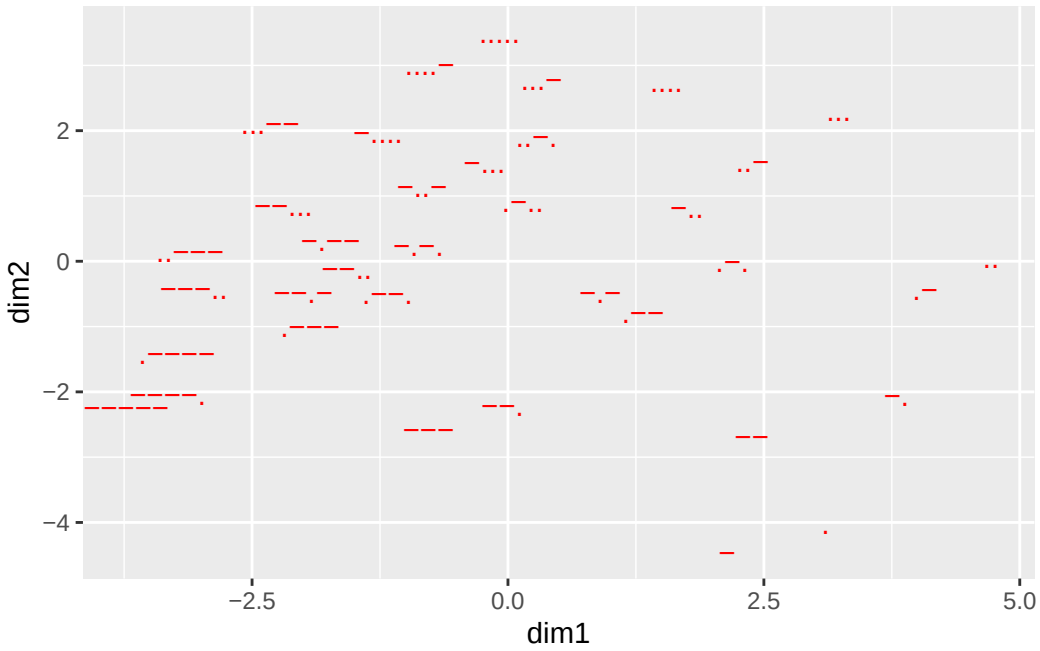

Figure 31: Rothkopf Configuration Huber c = 1

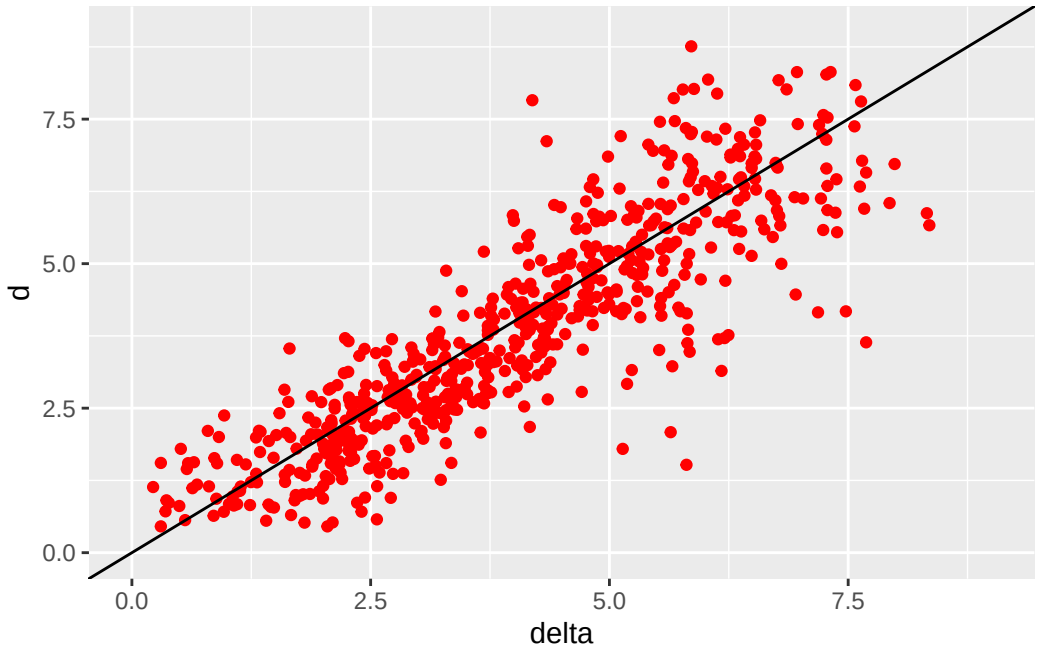

Figure 32: Rothkopf Shepard Plot Huber c = 1

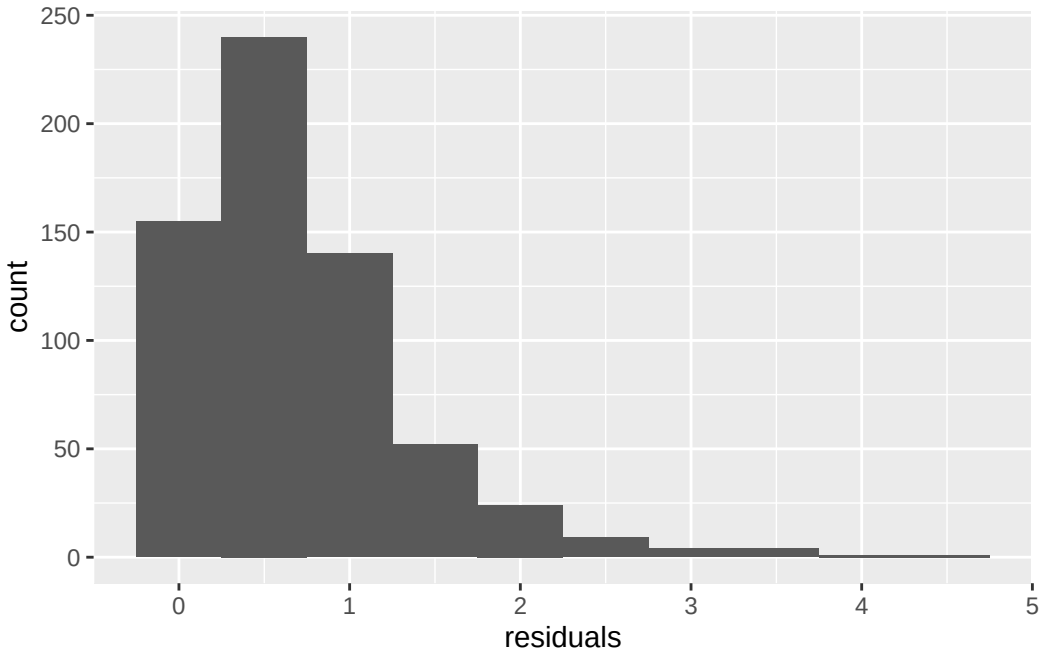

Figure 33: Rothkopf Histogram Huber Residuals

7.2.4 Tukey

Tukey with $c = 1$ converges in 812 iterations.

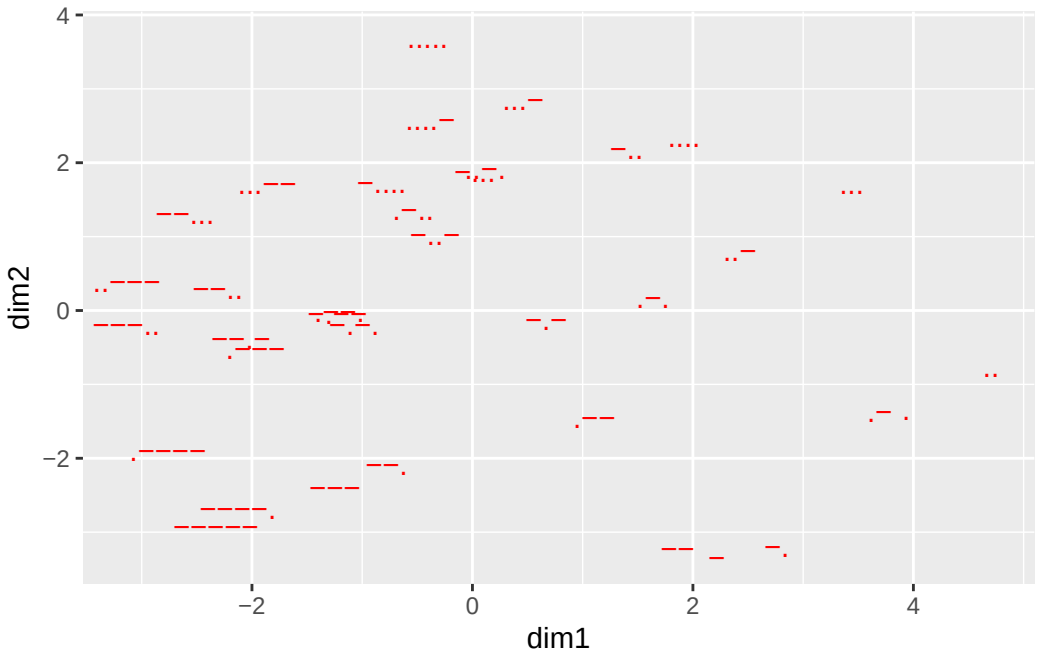

Figure 34: Rothkopf Configuration Tukey c = 1

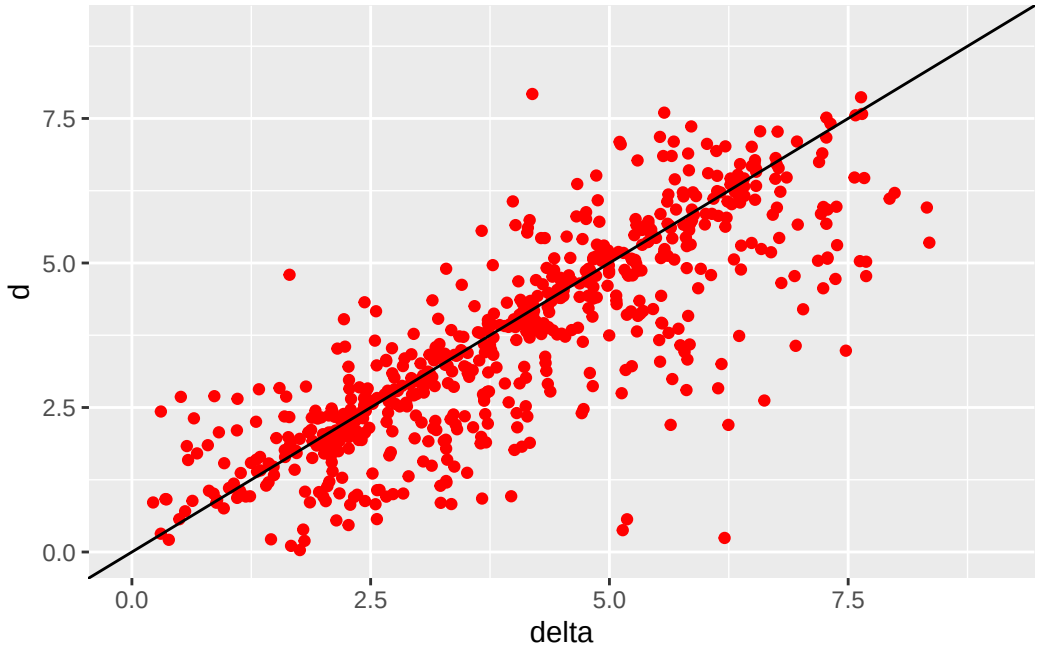

Figure 35: Rothkopf Shepard Plot Tukey c = 1

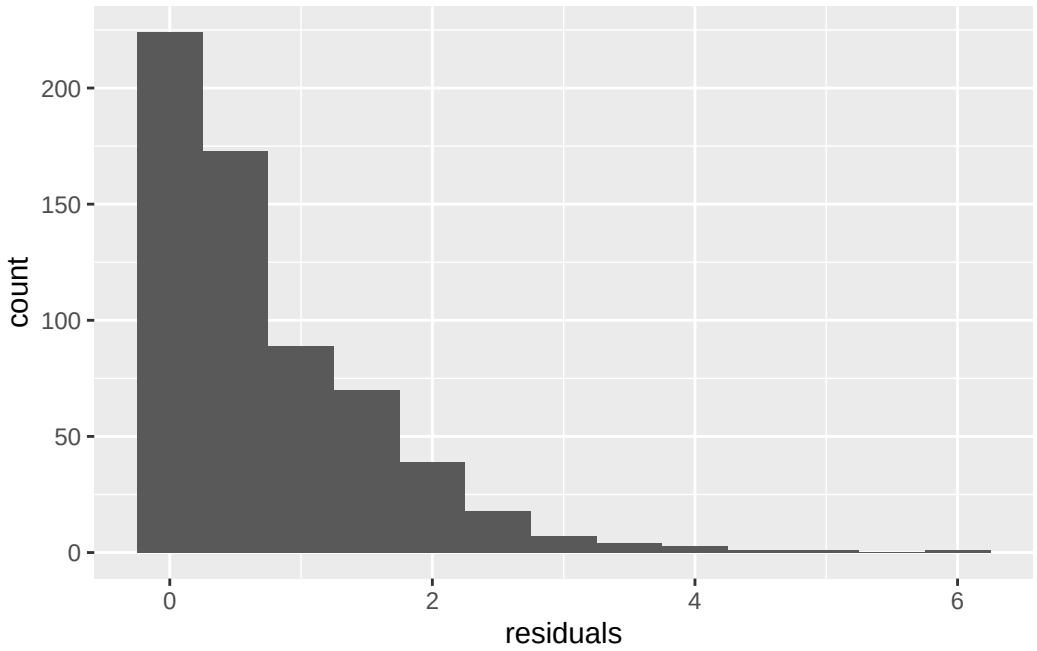

Figure 36: Rothkopf Histogram Tukey Residuals

8 Literature

The literature on results like Theorem 3.7 and Theorem 3.8 is difficult to review. There are various reasons for that. Relevant results have been published in robust statistics, computational statistics, optimization, location analysis, image restoration, sparse recovery. As is often the case, there are not many references between fields, almost everything is within. Even the names of the loss functions differ between fields. Much of it is hard to find in conference proceedings. Also, in most cases, the authors have specific applications in mind, which they then embed in a likelihood, Bayesian, linear regression, logistic regression, facility location, or EM framework and language.

De Leeuw and Lange (2009) give some references to previous work on results like Theorem 3.7, notably Groenen, Giaquinto, and Kiers (2003), Jaakkola and Jordan (2000), and Hunter and Li (2005). In these earlier papers we do not find Theorem 3.7 in its full generality. In Groenen, Giaquinto, and Kiers (2003) majorization of the log logistic function is considered. Besides requiring equality of the function and the majorizing quadratic at the support point y they also require equality at $-y$ and then check that the resulting quadratic is indeed a majorizer. In Jaakkola and Jordan (2000) also consider a symmetrized version of the log logistic function. They note that the resulting function is a convex funcion of x^2, and use a linear majorizer at x^2 to obtain a quadratic majorization. Hunter and Li (2005) come closest to Theorem 3.7. In their proposition 3.1 they approximate the general penalty function they use for varable selection at y by a quadratic with coefficient $f'(y)/2y$, and then show that it provides a quadratic majorization. In neither of the three papers there is a notion of sharp quadratic majorization.

I will discuss some of the literature under the headings "robust statistics", "location analysis", and "sparse recovery". Since I most definitely am not an expert in either of these three fields the literature reviews will be biased and incomplete. A final section, where I am somewhat more sure-footed, is "multivariate analysis".

8.1 Robust Statistics

In robust statistics it has been known for a long time that iterative reweighted least squares (IRLS) with weights $f'(x)/x$ gives a quadratic majorization algorithm. This result, and the corresponding IRLS algorithm, is often attributed to Beaton and Tukey (1974).

8.2 Location Analysis

In location analysis the first majorization/IRLS method is generally attributed to a 16-year old Hungarian mathematics prodigy (Weiszfeld (1937)). His algorithm avant-la-lettre was

intended to minimize a function of the form

$$\sigma(x) = \sum_k \|x - y_k\|, \tag{61}$$

over x in the plane or in three-space. The y_i are known locations, called *anchors* in the literature, and the norm is Euclidean. There is an English translation of Weiszfeld's paper, with bibliography and comments, in Weiszfeld and Plastria (2009). The minimization of Equation 61 is known under various names, usually consisting of one of the seven different non-empty selections from the triple (Torricelli, Fermat, Weber). The history of the problem is discussed, for example, in Plastria (2011).

Over the years the location problem has been generalized in numerous directions, to multiple locations, to using different norms, to unknown anchors, to nonlinear manifolds, and to obnoxious anchors you want to be far from. A good recent overview is Beck and Sabach (2015). An paper close in spirit to our paper is Aftab, Hartley, and Trumpf (2015), which has generalizations to ℓ_q norms and to Riemannian manifolds of non-negative curvature.

There is a huge literature on the convergence of the Weiszfeld algorithm. As in our Section 2.1 the basis is majorization based on the AM/GM inequality. Thus

$$\|x - y_\nu\| \leq \frac{1}{2}\frac{1}{\|x^{(k)} - y_\nu\|}(\|x - y_\nu\|^2 + \|x^{(k)} - y_\nu\|^2) \tag{62}$$

Thus the update is

$$x^{(k+1)} = \frac{\sum \omega_\nu(x^{(k)})y_\nu}{\sum \omega_\nu(x^{(k)})} \tag{63}$$

with weights

$$\omega_\nu(x) = \frac{1}{\|x - y_\nu\|}. \tag{64}$$

Unlike robust smacof the Toricelli-Fermat-Weber problem is convex, and consequently has no problems with non-global local minima. A most elegant proof of convergence using the tools of modern convex analysis is in Mordukhovich and Nam (2019). Older proofs sometimes have difficulty dealing with cases in which the iterates coincide with one of the anchors or in which the solution is actually one of the anchors. This creates problems similar to the problems in our Section 2.2, but in this simple case the problem is can be completely resolved using convexity and has no serious algorithmic consequences.

8.3 Sparse Recovery

This is a field which is difficult to delineate. A somewhat ad-hoc definition is recovering complete information from incomplete information, often in the context of specific engineering

problems. There is overlap with signal detection, image analysis, matrix completion, … But "sparse recovery" scientific activities "sparse recovery" could be extended far beyond these boundaries. Since classical statistics infers properties of the population from those of a sample it is a form of sparse recovery. Since science infers properties of the real world from outcomes of experiments it is sparse recovery too.

8.4 Multivariate Analysis

The smacof majorization method for multidimensional scaling was first presented at the *US-Japan Seminar on Theory, Methods and Applications of Multidimensional Scaling and Related Techniques* at UCSD in La Jolla, August 1975. Shortly after that I read the basic EM paper by Dempster, Laird, and Rubin (1977), and shirtky after that I realized that smacof and EM were both special cases of a general minimization strategy, which I called majorization at the time. In June 1978 both Nan Laird and I attended the *Fifth International Symposium on Multivariate Analysis* at the University of Pittsburgh. I remember mentioning majorization, excitedly, to Nan on the conference bus.

The smacof majorization method was fully discussed in De Leeuw (1977), De Leeuw and Heiser (1977), and De Leeuw and Heiser (1980). The familiar picture illustrating two steps of the general majorization algorithm appears in De Leeuw (1988a). But unlike EM the general idea of majorization remained unpublished, until De Leeuw (1994) and Heiser (1995).

Majorization was used regularly in the Gifi project. The book Gifi (1990), which is a version of 1981 lecture notes, mentions majorization only once, but since then a stream of papers and dissertations using majorization appeared. Heiser (1995) briefly mentions most of them. In De Leeuw (1988b) another large majorization subfield, the *aspect approach* to multivariate analysis, was developed. In section 7 of that paper the general majorization/minorization approach to optimization is outlined, maybe for the first time in print.

Robust versions of low rank matrix approximation, a.k.a. principal component analysis, were first considered by Gabriel and Odoroff (1984). They start by discussing the alternating least squares algorithm for least squares weighted matrix approximation of Gabriel and Zamir (1979). The alternating is to compute new row scores for currently fixed column scores by linear regression, and then computing new column scores corresponding with the new row scores, again by linear regression. Gabriel and Odoroff (1984) suggest to replace the linear least squares weighted averages in each of the two stages by medians or trimmed means to get a robust PCA. There is no sign of a convergence proof, but there is the suggestion to use alternating least absolute value methods to minimize the sum of absolute residuals of the matrix approximation. This suggestion was taken up by Verboon and Heiser (1994) using the majorization approach and the Huber and Tukey robust loss functions. Their robust PCA method is very similar to our robust MDS method, but the presentation of their method has

some magical elements. The Huber and Tukey majorization functions are presented without any discussion where they came from, and it is then verified that they are indeed majorizations. There is clearly nothing wrong with this, but using our Theorem 3.7 gives a more general and more direct approach.

Heiser (1986) was the first to connect the Weiszfeld problem with correspondence analysis and multidimensional scaling, emphasizing the majorization aspects. As we have seen in Heiser (1987) and Heiser (1988) he constructed majorization algorithms for multidimensional scaling and correspondence analysis.

The IRLS approach to robustifying multivariate matrix approximation techniques could easily lead to a large and varied number of publications. There are some excellent examples making their way through the usual publication channels. I will just give two recent examples, with good bibliographies. They are Huber Principal Component Analysis (He et al. (2023)) and Cauchy Factor Analysis (Li (2024)).

9 Discussion

9.1 Bounding the Second Derivative

In some cases our basic theorems may not apply, but there may be an alternative way to majorize loss. In fact, this is classic quadratic bounding as in Vosz and Eckhardt (1980) or Böhning and Lindsay (1988). As before, we want to minimize $\sum \omega_k f(\delta_k - d_k(X))$, but now we suppose that there is a $K > 0$ such that $f''(x) \leq K$. We then have the majorization

$$f(\delta_k - d_k(X)) \leq f(\delta_k - d_k(Y)) + f'(\delta_k - d_k(Y))(d_k(Y) - d_k(X)) + \frac{1}{2}K(d_k(Y) - d_k(X))^2 \tag{65}$$

and in iteration k we minimize, or at least decrease,

$$\sum \omega_k \left[d_k(X) - \{d_k(X^{(k)}) - K^{-1}f'(\delta_k - d_k(X^{(k)}))\}\right]^2 \tag{66}$$

Note that in this algorithm the weights do not change. Instead of fitting a fixed target with moving weights, we fit a moving target with fixed weights.

We can apply bounding the second derivative, for example, to Charbonnier loss, using the inequality

$$f_c''(x) = (x^2 + c^2)^{-\frac{1}{2}} - x^2(x^2 + c^2)^{-\frac{3}{2}} \leq (x^2 + c^2)^{-\frac{1}{2}} \leq c^{-1}, \tag{67}$$

Of course this method requires that the second derivative exists at x. Although I have not done any comparisons it will probably require more iterations and take longer than the method in Section 4.1.

The paper by Vosz and Eckhardt (1980) deserves some special mention here.

$$\mathcal{D}^2\sigma(x) = \sum \frac{1}{d_i(x)}\left\{I - \frac{(x - y_i)(x - y_i)'}{d_i^2(x)}\right\}$$

9.2 Fixed Weights

One could also consider using the fixed weights in regular non-robust smacof to achieve some form of robustness. Redefine stress as

$$\sigma(X) := \sum_k \omega_k f(\delta_k)(\delta_k - d_k(X))^2 \tag{68}$$

For example, we can choose a negative power for f, so that it downweights the large dissimilarities. If the dissimilarities is large, then it should have less influence on the fit, and thus on the solution X. This type of fixed power-weighting is used in various places (De Leeuw and Heiser (1980), Groenen and Van de Velden (2016)) to approximate loss functions such the one with logarithmic residuals in Ramsay (1977).

But we have to keep in mind that downweighting large dissimilarities is not the same thing as downweighting large residuals. The residuals depend on X, and it is perfectly possible that some small dissimilarities have large residuals. On the other hand emphasizing small dissimilarities in the loss function means that we want small dissimilarities to be fitted relatively well, which means that on average we want small dissimilarities to have small residuals. The Shepard plot will tend to fan out at the high end.

Despite these reservations, it will be useful to study if and how fixed weights can be used to improve robustness of smacof. If only because fixed weights correspond with a simpler and presumably more efficient algorithm.

9.3 Residual Definition

In our examples and in our code we use the residuals $\delta_k - d_k(X)$ are arguments of our loss functions. From the statistical point of view we have to remember, however, that most of these loss functions were designed for the robust estimation of a location parameter or a linear regression function. The error distributions were explicitly or implicitly assumed to be symmetric around zero, and defined on the whole real line, which was reflected in the fact that loss functions were even and had infinite support. In MDS, however, distances and dissimilarities are non-negative and reasonable error functions are not symmetric. One could follow the example of Ramsay (1977) and measure residuals as $\log \delta_{ij} - \log d_{ij}(X)$. This does not have any effect on the majorization of the loss functions, but it means that in the smacof step to find $X^{(k+1)}$ we have to minimize

$$\sigma(X) = \sum \omega_k(X^{(k)})(\log \delta_{ij} - \log d_{ij}(X))^2,$$

which is considerably more complicated (De Leeuw, Groenen, and Mair (2016)).

9.4 Robust Nonmetric MDS

Our discussion and our software is all about metric MDS. It seems easy to extend the discussion to non-linear and non-metric MDS by adding an alternating least squares step optimally scaling the dissimilarities. This would take place between two majorizations of the robust loss function,

so one or more transformation and smacof steps can be taken between updating the weights. But this paradigm does not work for robust smacof.

Consider any hard redescender, such as Tukey or Hinich. At iteration ν, for current weights, first first improve the configuration, then compute the optimal transformation of the dissimilarities, and then compute new weights. This is a recipe for disaster. At some point we minimize

$$\sigma(\hat{d}) = \sum \omega_k(X^{(\nu)})(\hat{d}_k - d_k(X^{(\nu+1)}))^2$$

{#eq:zero} over the disparities $\hat{d}$, which must be monotone with the dissimilarities. Because of the hard redescending some of the weights, for current absolute residuals larger than c, will be zero. The monotone regression is done for the observations with non-zero weights, and the disparities corresponding with zero weights are only determined by the order they are required to have. Thus they can be freely chosen in an interval between two disparities obtained from the monotone regression. That interval can be large, in fact if one of the zero weights corresponds with the largest dissimilarity it can be infinite. What we choose in the interval will determine the new residual and thus the next set of weights.

In the unfortunate situation that the current absolute residuals are all larger than c, even after choosing the optimal $\hat{d}$, the next weights will all be zero and the algorithm stops with zero stress.

9.5 Practicalities

Recommending one particular loss function from the many we have discussed is not easy. In some cases, for example for Cauchy loss, one can justify the choice of a loss function by assuming a particular error distribution and using the maximum likelihood principle. But in general perhaps the best way to proceed for a given MDS problem is to take what we could call a *trajectory approach*. Choose one particular parametric family, for example the Huber one, and compute the robust smacof solution $X(c)$ for a number of increasing positive c values. For small c we start with a close approximation of the LAV solution, increasing c will eventually take us to the LS solution. The starting point for computing each solution will be the solution for the previous c. We can plot the trajectory of the points in the configurations $X(c)$, and even make an animation. It seems that the Huber family is a good candidate for such a study, with the generalized Charbonnier a good second. If the main concern is to suppress the influence of outliers then trying some of the hard redescenders, such as the Tukey family, makes sense. Studying trajectories for some of robust loss functions is clearly interesting, but it is not something we can or will explore in this paper.

10 Code

The function smacofRobust has a parameter "engine", which can be equal to smacofCharbonnier, smacofGeneralizedCharbonnier, smacofBarron, smacofHuber, smacofTukey, smacofHinnich, smacofCauchy, smacofFair, smacofAndrews, smacofLogistic, smacofWelsch, or smacofGaussian. These thirteen small modules compute the respective loss function values and weights for the IRLS procedure. This makes it easy for interested parties to add additional robust loss functions.

```
smacofRobust <- function(delta,
                         weights = 1 - diag(nrow(delta)),
                         ndim = 2,
                         xold = smacofTorgerson(delta, ndim),
                         engine = smacofAV,
                         cons = 0,
                         itmax = 1000,
                         eps = 1e-15,
                         verbose = TRUE) {
  nobj <- nrow(delta)
  wmax <- max(weights)
  dold <- as.matrix(dist(xold))
  h <- engine(nobj, weights, delta, dold, cons)
  rold <- h$resi
  wold <- h$wght
  sold <- h$strs
  itel <- 1
  repeat {
    vmat <- -wold
    diag(vmat) <- -rowSums(vmat)
    vinv <- solve(vmat + (1 / nobj)) - (1 / nobj)
    bmat <- -wold * delta / (dold + diag(nobj))
    diag(bmat) <- -rowSums(bmat)
    xnew <- vinv %*% (bmat %*% xold)
    dnew <- as.matrix(dist(xnew))
    h <- engine(nobj, weights, delta, dnew, cons)
    rnew <- h$resi
    wnew <- h$wght
    snew <- h$strs
    if (verbose) {
      cat(
```

```
        "itel ",
        formatC(itel, width = 4, format = "d"),
        "sold ",
        formatC(sold, digits = 10, format = "f"),
        "snew ",
        formatC(snew, digits = 10, format = "f"),
        "\n"
      )
    }
    if ((itel == itmax) || ((sold - snew) < eps)) {
      break
    }
    xold <- xnew
    dold <- dnew
    sold <- snew
    wold <- wnew
    rold <- rnew
    itel <- itel + 1
  }
  return(list(
    x = xnew,
    s = snew,
    d = dnew,
    r = rnew,
    itel = itel
  ))
}

smacofTorgerson <- function(delta, ndim) {
  dd <- delta^2
  rd <- apply(dd, 1, mean)
  md <- mean(dd)
  sd <- -.5 * (dd - outer(rd, rd, "+") + md)
  ed <- eigen(sd)
  return(ed$vectors[, 1:ndim] %*% diag(sqrt(ed$values[1:ndim])))
}

smacofCharbonnier <- function(nobj, wmat, delta, dmat, cons) {
  resi <- sqrt((delta - dmat)^2 + cons)
  resi <- ifelse(resi < 1e-10, 2 * max(wmat), resi)
```

```
  rmin <- sqrt(cons)
  wght <- wmat / (resi + diag(nobj))
  strs <- sum(wmat * resi) - rmin * sum(wmat)
  return(list(
    resi = resi,
    wght = wght,
    strs = strs
  ))
}

smacofGeneralizedCharbonnier <- function(nobj, wmat, delta, dmat, cons) {
  resi <- ((delta - dmat) ^ 2 + cons[1]) ^ cons[2]
  rmin <- cons[1] ^ cons[2]
  wght <- wmat * ((delta - dmat) ^ 2 + cons[1] + diag(nobj)) ^ (cons[2] - 1)
  strs <- sum(wmat * resi) - rmin * sum(wmat)
  return(list(
    resi = resi,
    wght = wght,
    strs = strs
  ))
}

smacofBarron <- function(nobj, wmat, delta, dmat, cons) {
  f1 <- abs(cons[2] - 2) / cons[2]
  f2 <- ((((delta - dmat) / cons[1]) ^ 2) / abs(cons[2] - 2) + 1)
  resi <- f1 * (f2 ^ (cons[2] / 2) - 1)
  wght <- wmat * f2 ^ (cons[2] / 2 - 1)
  strs <- sum(wmat * resi)
  return(list(
    resi = resi,
    wght = wght,
    strs = strs
  ))
}

smacofGauss <- function(nobj, wmat, delta, dmat, cons) {
  difi <- delta - dmat
  resi <- difi * (2 * pnorm(difi / cons) - 1) + 2 * cons * dnorm(difi / cons)
  rmin <- 2 * cons * dnorm(0)
  wght <- wmat * (pnorm(difi / cons) - 0.5) / (difi + diag(nobj))
```

```
  strs <- sum(wmat * resi) - rmin * sum(wmat)
  return(list(
    resi = resi,
    wght = wght,
    strs = strs
  ))
}

smacofHuber <- function(nobj, wmat, delta, dmat, cons) {
  difi <- delta - dmat
  resi <- ifelse(abs(difi) < cons, (difi ^ 2) / 2, cons * abs(difi) - ((cons ^ 2) / 2)
  wght <- ifelse(abs(difi) < cons,
                 wmat,
                 wmat * sign(difi - cons) * cons / (difi + diag(nobj)))
  strs <- sum(wmat * resi)
  return(list(
    resi = resi,
    wght = wght,
    strs = strs
  ))
}

smacofTukey <- function(nobj, wmat, delta, dmat, cons) {
  cans <- (cons ^ 2) / 6
  difi <- delta - dmat
  resi <- ifelse(abs(difi) < cons, cans * (1 - (1 - (difi / cons)^2)^3), cans)
  wght <- wmat * ifelse(abs(difi) < cons, (1 - (difi / cons)^2)^2, 0)
  strs <- sum(wmat * resi)
  return(list(
    resi = resi,
    wght = wght,
    strs = strs
  ))
}

smacofCauchy <- function(nobj, wmat, delta, dmat, cons) {
  difi <- delta - dmat
  resi <- log((difi / cons)^2 + 1)
  wght <- wmat * (1 / ((difi / cons)^2 + 1))
  strs <- sum(wmat * resi)
```

```
  return(list(
    resi = resi,
    wght = wght,
    strs = strs
  ))
}

smacofWelsch <- function(nobj, wmat, delta, dmat, cons) {
  difi <- delta - dmat
  resi <- 1 - exp(-(difi / cons)^2)
  wght <- wmat * exp(-(difi / cons)^2)
  strs <- sum(wmat * resi)
  return(list(
    resi = resi,
    wght = wght,
    strs = strs
  ))
}

smacofAndrews <- function(nobj, wmat, delta, dmat, cons) {
  difi <- delta - dmat
  resi <- ifelse(abs(difi) < pi * cons,
                 (cons ^ 2) * (1 - cos(x / cons)),
                 2 * (cons^2))
  wght <- wmat * ifelse(abs(difi) < pi * cons, sin(x / cons) / (x / cons), 0)
  strs <- sum(wmat * resi)
  return(list(
    resi = resi,
    wght = wght,
    strs = strs
  ))
}

smacofHinich <- function(nobj, wmat, delta, dmat, cons) {
  difi <- delta - dmat
  resi <- ifelse(abs(difi) < cons, (difi^2) / 2, (cons^2) / 2)
  wght <- wmat * ifelse(abs(difi) < cons, 1, 0)
  strs <- sum(wmat * resi)
  return(list(
    resi = resi,
```

```
    wght = wght,
    strs = strs
  ))
}

smacofLogistic <- function(nobj, wmat, delta, dmat, cons) {
  difi <- delta - dmat
  resi <- (cons ^ 2) * log(cosh(x / cons))
  wght <- wmat * tanh(x / cons) / (x / cons)
  strs <- sum(wmat * resi)
  return(list(
    resi = resi,
    wght = wght,
    strs = strs
  ))
}

smacofFair <- function(nobj, wmat, delta, dmat, cons) {
  difi <- delta - dmat
  resi <- log((difi / cons)^2 + 1)
  wght <- wmat * (1 / ((difi / cons) ^ 2 + 1))
  strs <- sum(wmat * resi)
  return(list(
    resi = resi,
    wght = wght,
    strs = strs
  ))
}
```

References

Aftab, K., and R. Hartley. 2015. "Convergence of Iteratively Re-Weighted Least Squares to Robust m-Estimators." In *2015 IEEE Winter Conference on Applications of Computer Vision,* 480–87. https://doi.org/10.1109/WACV.2015.70.

Aftab, K., R. Hartley, and J. Trumpf. 2015. "Generalized Weiszfeld Algorithms for Lq Optimization." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (4): 728–44. https://doi.org/10.1109/TPAMI.2014.2353625.

Andrews, D. F., P. J. Bickel, F. R. Hampel, P. J. Huber, W. H. Rogers, and J. W. Tukey. 1972. *Robust Estimators of Location: Survey and Advances*. Princeton University Press.

Barron, J. T. 2019. "A General and Adaptive Robust Loss Function." In *Proceedings 2019 IEEE/CVF Conferehce on Computer Vision and Pattern Recognition*, 4331–39. https://openaccess.thecvf.com/content_CVPR_2019/papers/Barron_A_General_and_Adaptive_Robust_Loss_Function_CVPR_2019_paper.pdf.

Beaton, A. E., and W. Tukey J. 1974. "The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data." *Technometrics* 16 (147–185).

Beck, A., and S. Sabach. 2015. "Weiszfeld's Method: Old and New Results." *Journal of Optimization Theory and Applications* 164: 1–40.

Black, M. J., and P. Anandan. 1996. "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields." *Computer Vision and Image Understanding* 63 (1): 75–104.

Böhning, D., and B. G. Lindsay. 1988. "Monotonicity of Quadratic-approximation Algorithms." *Annals of the Institute of Statistical Mathematics* 40 (4): 641–63.

Candes, E. J., and T. Tao. 2005. "Decoding by Linear Programming." *IEEE Transactions on Information Theory* 51 (12): 4203–15.

Candes, E. J., M. B. Wakin, and S. P. Boyd. 2008. "Enhancing Sparsity by Reweighted l_1 Minimization." *Journal of Fourier Analysis and Applications* 14: 877–905. https://doi.org/10.1007/s00041-008-9045-x.

Charbonnier, P., L. Blanc-Feraud, G. Aubert, and M. Barlaud. 1994. "Two deterministic half-quadratic regularization algorithms for computed imaging." *Proceedings of 1st International Conference on Image Processing* 2: 168–72. https://doi.org/10.1109/icip.1994.413553.

Coleman, D., P. Holland, N. Kaden, V. Klema, and S. C. Peters. 1980. "A System of Subroutines for Iteratively Reweighted Least Squares Computations." *ACM Transactions on Mathematical Software* 6 (3): 327–36.

De Gruijter, D. N. M. 1967. "The Cognitive Structure of Dutch Political Parties in 1966." Report E019-67. Psychological Institute, University of Leiden.

De Leeuw, J. 1977. "Applications of Convex Analysis to Multidimensional Scaling." In *Recent Developments in Statistics*, edited by J. R. Barra, F. Brodeau, G. Romier, and B. Van Cutsem, 133–45. Amsterdam, The Netherlands: North Holland Publishing Company.

———. 1984. “Differentiability of Kruskal’s Stress at a Local Minimum.” *Psychometrika* 49: 111–13.
———. 1988a. “Convergence of the Majorization Method for Multidimensional Scaling.” *Journal of Classification* 5: 163–80.
———. 1988b. “Multivariate Analysis with Optimal Scaling.” In *Proceedings of the International Conference on Advances in Multivariate Statistical Analysis*, edited by S. Das Gupta and J. K. Ghosh, 127–60. Calcutta, India: Indian Statistical Institute.
———. 1994. “Block Relaxation Algorithms in Statistics.” In *Information Systems and Data Analysis*, edited by H. H. Bock, W. Lenski, and M. M. Richter, 308–24. Berlin: Springer Verlag. https://jansweb.netlify.app/publication/deleeuw-c-94-c/deleeuw-c-94-c.pdf.
———. 2018. “MM Algorithms for Smoothed Absolute Values.” 2018. https://jansweb.netlify.app/publication/deleeuw-e-18-f/deleeuw-e-18-f.pdf.
De Leeuw, J., P. Groenen, and P. Mair. 2016. “Minimizing rStress Using Majorization.” 2016. https://jansweb.netlify.app/publication/deleeuw-groenen-mair-e-16-a/deleeuw-groenen-mair-e-16-a.pdf.
De Leeuw, J., and W. J. Heiser. 1977. “Convergence of Correction Matrix Algorithms for Multidimensional Scaling.” In *Geometric Representations of Relational Data*, edited by J. C. Lingoes, 735–53. Ann Arbor, Michigan: Mathesis Press.
———. 1980. “Multidimensional Scaling with Restrictions on the Configuration.” In *Multivariate Analysis, Volume V*, edited by P. R. Krishnaiah, 501–22. Amsterdam, The Netherlands: North Holland Publishing Company.
De Leeuw, J., and K. Lange. 2009. “Sharp Quadratic Majorization in One Dimension.” *Computational Statistics and Data Analysis* 53: 2471–84.
De Leeuw, J., and P. Mair. 2009. “Multidimensional Scaling Using Majorization: SMACOF in R.” *Journal of Statistical Software* 31 (3): 1–30. https://www.jstatsoft.org/article/view/v031i03.
Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. “Maximum Likelihood for Incomplete Data via the EM Algorithm.” *Journal of the Royal Statistical Society* B39: 1–38.
Dennis Jr, J. E., and R. E. Welsch. 1978. “Techniques for Nonlinear Least Squares and Robust Regression.” *Communications in Statistics - Simulation and Computation* 7 (4): 345–59.
Dieudonné, J. 1969. *Foundations of Modern Analysis*. Academic Press.
Donoho, D. L., and M. Elad. 2003. “Optimally Sparse Representation in General (Nonorthogonal) Dictionaries via ℓ_1 Minimization.” *Proceedings of the National Academy of Sciences* 100 (5): 2197–2202.
Gabriel, K. R., and Ch. L. Odoroff. 1984. “Resistant Lower Rank Approximation of Matrices.” In *Data Analysis and Informatics*, edited by E. Diday, M. Jambu, L. Lebart, J. Pages, and R. Tomassone, 3:23–30. North Holland Publishing Company.
Gabriel, K. R., and S. Zamir. 1979. “Lower Rank Approximation of Matrices by Least Squares with Any Choize of Weights.” *Technometrics* 21 (4): 489–98.
Gifi, A. 1990. *Nonlinear Multivariate Analysis*. New York, N.Y.: Wiley.
Groenen, P. J. F., P. Giaquinto, and H. A. L Kiers. 2003. “Weighted Majorization Algorithms

for Weighted Least Squares Decomposition Models." Econometric Institute Report EI 2003-09. Econometric Institute, Erasmus University Rotterdam. https://repub.eur.nl/pub/1700.

Groenen, P. J. F., W. J. Heiser, and J. J. Meulman. 1999. "Global Optimization in Least-Squares Multidimensional Scaling by Distance Smoothing." *Journal of Classification* 16: 225–54.

Groenen, P. J. F., and M. Van de Velden. 2016. "Multidimensional Scaling by Majorization: A Review." *Journal of Statistical Software* 73 (8): 1–26. https://www.jstatsoft.org/index.php/jss/article/view/v073i08.

He, Y., L. Li, D. Liu, and W.-Z. Zhou. 2023. "Huber Principal Component Analysis for Large-Dimensional Factor Models." https://arxiv.org/abs/2303.02817.

Heiser, W. J. 1986. "A Majorization Algorithm for the Reciprocal Location Problem." RR-86-12. Department of Data Theory, University of Leiden.

———. 1987. "Correspondence Analysis with Least Absolute Residuals." *Computational Statistics and Data Analysis* 5: 337–56.

———. 1988. "Multidimensional Scaling with Least Absolute Residuals." In *Classification and Related Methods of Data Analysis*, edited by H. H. Bock, 455–62. North-Holland Publishing Co.

———. 1995. "Convergent Computing by Iterative Majorization: Theory and Applications in Multidimensional Data Analysis." In *Recent Advantages in Descriptive Multivariate Analysis*, edited by W. J. Krzanowski, 157–89. Oxford: Clarendon Press.

Hinich, M. J., and P. P. Talwar. 1975. "A Simple Method for Robust Regression." *Journal of the American Statistical Association* 70: 113–19.

Holland, P. W., and R. E. Welsch. 1977. "Robust Regression Using Iteratively Reweighted Least-Squares." *Communications in Statistics - Theory and Methods* 6 (9): 813–27. https://doi.org/10.1080/03610927708827533.

Huber, P. J. 1964. "Robust Estimation of a Location Parameter." *Annals of Mathematical Statistics* 35 (1): 73–101.

Hunter, D. R., and R. Li. 2005. "Variable Selection Using MM Algorithms." *The Annals of Statistics* 33: 1617–42.

Jaakkola, T. S., and M. I. Jordan. 2000. "Bayesian Parameter Estimation via Variational Methods." *Statistics and Computing* 10: 25–37.

Lange, K. 2016. *MM Optimization Algorithms*. SIAM.

Li, J. 2024. "Robust Matrix Factor Analysis Method with Adaptive Parameter Adjustment Using Cauchy Weighting." *Computational Statistics*. https://doi.org/10.1007/s00180-024-01548-4.

Mlotshwa, T., H. Van Deventer, and Sergeevna Bosman A. 2023. "Cauchy Loss Function: Robustness Under Gaussian and Cauchy Noise." https://arxiv.org/abs/2302.07238.

Mordukhovich, B. S., and N. M. Nam. 2019. "The Fermat-Torricelli Problem and Weiszfeld's Algorithm in the Light of Convex Analysis." *Journal of Applied Numerical Optimization* 1 (3): 205–19. https://doi.org/https://doi.org/10.23952/jano.1.2019.3.02.

Phillips, R. F. 2002. "Least Absolute Deviations Estimation via the EM Algorithm." *Statistics and Computing* 12: 281–85.

Plastria, F. 2011. "The Weiszfeld Algorithm: Proof, Amendments and Extensions." In *Foundations of Location Analysis*, edited by H. A. Eiselt and V. Marianov, 155:357–89. International Series inOperations Research and Management Science. Springer.

Pliner, V. 1996. "Metric Unidimensional Scaling and Global Optimization." *Journal of Classification* 13: 3–18.

R Core Team. 2024. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Ramirez, C., R. Sanchez, V. Kreinovich, and M. Argaez. 2014. "$\sqrt{x^2+\mu}$ is the Most Computationally Efficient Smooth Approximation to |x|." *Journal of Uncertain Systems* 8: 205–10.

Ramsay, J. O. 1977. "Maximum Likelihood Estimation in Multidimensional Scaling." *Psychometrika* 42: 241–66.

Rothkopf, E. Z. 1957. "A Measure of Stimulus Similarity and Errors in some Paired-associate Learning." *Journal of Experimental Psychology* 53: 94–101.

Schlossmacher, E. J. 1973. "An Iterative Technique for Absolute Deviations Curve Ftting." *Journal of the American Statistical Association* 68: 857–59.

Van Ruitenburg, J. 2005. "Algorithms for Parameter Estimation in the Rasch Model." Measurement and Research Department Reports 2005-04. Arnhem, Netherlands: CITO. https://www.researchgate.net/publication/355568984_Algorithms_for_parameter_estimation_in_the_Rasch_model_Measurement_and_Research_Report_05-04#fullTextFileContent.

Verboon, P., and W. J. Heiser. 1994. "Resistant Lower Rank Approximation of Matrices by Iterative Majorization." *Computational Statistics and Data Analysis* 18: 457–67.

Voronin, S., G. Ozkaya, and Y. Yoshida. 2014. "Convolution Based Smooth Approximations to the Absolute Value Function with Application to Non-smooth Regularization." 2014. https://arxiv.org/abs/1408.6795.

Vosz, H., and U. Eckhardt. 1980. "Linear Convergence of Generalized Weiszfeld's Method." *Computing* 25: 243–51.

Weiszfeld, E. 1937. "Sur le Point par lequel la Somme des Distances de n Points Donnés est Minimum." *Tohoku Mathematics Journal* 43: 355–86.

Weiszfeld, E., and F. Plastria. 2009. "On the Point for Which the Sum of the Distances to n Given Points Is Minimum." *Annals of Operations Research* 167: 7–41.