Robust Least Squares Multidimensional Scaling

Jan de Leeuw

September 25, 2024

We use an iteratively reweighted 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, localization theory, and sparse recovery. Code in R is included.

1 Introduction

The title of this chapter seems something paradoxical. Least squares estimation is typically not robust, it is sensitive to outliers and pays a lot of attention to fitting the larger observations. What we mean by robust least squares MDS, however, is using the smacof machinery designed to minimize loss of the form

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

to minimize robust loss functions. The prototypical robust loss function is

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

which we will call *strife*, because 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 many points where the loss function is not differentiable leads to (many) additional local minima.

In this chapter 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 as close to the absolute value function as possible. 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 here is to combines the robustness of the absolute value function with the efficiency and computational ease of least squares. If that is our goal then there is no reason to stay as close to the absolute value function as possible.

Our robust or smooth loss functions are all of the form

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

for a suitable choice of the real valued function f. We will define what we mean by "suitable" later on. For now, note that loss (1) is the special case with $f(x) = x^2$ and loss (2) is the special case with $f(x) = |x|$.

2 Majorizing Strife

The pioneering work in strife minimization using smacof is Heiser (1988), building on earlier work in 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 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 $x = y$. 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 nuisance we must deal with.

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

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

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

2.1 Algorithm

Define

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

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

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

using a standard smacof step. It then computes the new weights $w_k(X^{(k+1)})$ 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 this unfortunate event, but as far as I can see none of them have been entirely satisfactory.

$$w_k(Y) = \begin{cases} w_k \frac{1}{|\delta_k - d_k(Y)|} & \text{if } d_k(Y) \neq \delta_k, \\ 2 \ \max_k w_k & \text{otherwise.} \end{cases} \tag{9}$$

Redefine ω_1 as

$$\omega_1(X) := \frac{1}{2} \sum w_k(Y)\{(\delta_k - d_k(Y))^2 + (\delta_k - d_k(X))^2\}. \tag{10}$$

This modified ω_1 majorizes σ_1 at Y, even in the perverse case that $\delta_k = d_k(Y)$ for all k.

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

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

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

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 $w_k \delta_k > 0$ 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 $w_k > 0$.

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 $2^{\binom{n}{2}}$ polyhedra defined by the 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 the non-empty polyhedral intersections strife is a linear function of x. Thus it attains its minimum at a vertex of the intersection, which is a solution for which some distances are zero and some residuals are zero. There can be no minima, local or global, in the interior of a polyhedral intersection. Of course the number of regions, which is maximally $n!2^{\binom{n}{2}}$, is too large to actually compute or draw, except perhaps when $n = 3$, in which case there are maximally 48 of them. Thus we have shown that in one dimension strife is not differentiable at a local minimum, and that there is presumably a large number of them.

In the multidimensional case linearity goes out the window. The set of configurations $d_k(X) = \delta_k$ is an ellipsoid and $d_k(X) = 0$ is 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.

3 Generalizing Strife

The AM/GM inequality was used in the previous section to construct a quadratic majorization of strife. To fix the terminology we say that a function g *majorizes* a function f at y if $g(x) \geq f(x)$ for all x and $g(y) = f(y)$. Majorization is *strict* if $g(x) > f(x)$ for all $x \neq y$. If $\mathfrak{H}$ is a family of functions that all majorize f at y then $h \in \mathfrak{H}$ is *sharp* if $h(x) \leq g(x)$ for all $g \in \mathfrak{H}$.

We are specifically interested in this chapter 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) by De Leeuw and Lange (2009). Their Theorem 4.5 on page 2478 says

> Theorem 4.5: Suppose $f(x)$ is an even, differentiable function on $\mathbb{R}$ such that the ratio $f'(x)/x$ is non-increasing on $(0, \infty)$. Then the even quadratic
>
> $$g(x) = \frac{f'(y)}{2y}(x^2 - y^2) + f(y) \tag{11}$$
>
> is a sharp quadratic majorizer of f at the point y.
>
> Theorem 4.6. 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 only give a sufficient condition for quadratic majorization (in fact, for sharp quadratic majorization). Also note that if f is differentiable, even, and convex then it has its minimum at zero.

We now apply this theorem to functions of the form

$$\sigma_f(X) := \sum w_k\, f(\delta_k - d_k(X)), \tag{12}$$

where f satisfies the conditions in the theorem. If

$$\omega_f(X) := \sum w_k \frac{f'(\delta_k - d_k(Y))}{2(\delta_k - d_k(Y))}\{(\delta_k - d_k(X))^2 - (\delta_k - d_k(Y))^2\} + f(\delta_k - d_k(Y)), \tag{13}$$

then ω_f is a sharp quadratic majorization at Y.

Shift: Although the absolute value is not differentiable at the origin the theorem can still be applied. It just does not give a majorizer at $y = 0$. If $f(x) = |x|$ then

$$g(x) = \frac{1}{2|y|}(x^2 - y^2) + |y| = \frac{1}{2|y|}(x^2 + y^2), labeleq : abssharp \tag{14}$$

which is the same as (5). Thus the AM/GM method gives the sharp quadratic majorization.

In iteration k the robust smacof algorithm does a smacof step towards minimization of ω_f over X. We can ignore the parts of (13) that only depend on Y, and minimize

$$\sum w_k(X^{(k)})(\delta_k - d_k(X))^2, \tag{15}$$

with

$$w_k(X^{(k)}) := w_k \frac{f'(\delta_k - d_k(X^{(k)}))}{2(\delta_k - d_k(Y))}. \tag{16}$$

It then recomputes the weights $w_k(X^{(k+1)})$ and goes to the smacof step again. This can be thought of as iterativey reweighted least squares, and also as majorization within majorization, with the smacof majorization within the sharp quadratic majorization of the loss function.

A

4 Charbonnier loss

De Leeuw (2018)

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

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

which is decreasing.

$$\sigma_\epsilon(X) := \sum w_k\sqrt{(\delta_k - d_k(X))^2 + \epsilon^2}$$

Now majorization using

$$\sqrt{(\delta_k - d_k(X))^2 + \epsilon^2} \leq \frac{1}{2}\frac{1}{\sqrt{(\delta_k - d_k(Y))^2 + \epsilon^2}}\{(\delta_k - d_k(X))^2 + (\delta_k - d_k(Y))^2 + 2\epsilon^2\}$$

Alt:

$$\sigma_\epsilon(X) = \epsilon^2\left\{\sqrt{1 + \frac{x^2}{\epsilon^2}} - 1\right\}$$

5 Huber Loss

The Huber function (Huber (1964)) is

$$f(x)=\begin{cases}\frac12 x^2 & \text{if } |x|<c,\\ c|x|-\frac12 c^2 & \text{otherwise.}\end{cases}$$

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

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

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

The Huber function is even and differentiable. Moreover $f'(x)/x$ decreases from. Thus the theorem applies and the sharp quadratic majorizer at y is

$$g(x)=\{$$

$$\sigma_k(X)=\begin{cases}\frac12(\delta_k-d_k(X))^2 & \text{if } |\delta_k-d_k(X)|<c,\\ c|\delta_k-d_k(X)|-\frac12 c^2 & \text{if } |\delta_k-d_k(X)|\geq c.\end{cases}$$

$$\omega_k(x,y)=\begin{cases}\frac12\frac{c}{|y|}(x^2-y^2)-cy-\frac12 c^2 & \text{if } y\leq -c,\\ \frac12 x^2 & \text{if } |y|<c,\\ \frac12\frac{c}{|y|}(x^2-y^2)+cy-\frac12 c^2 & \text{if } y\geq +c.\end{cases}$$

Now $x=\delta_k-d_k(X)$ and $y=\delta_k-d_k(Y)$

$$\omega_k(X;Y)=\begin{cases}\frac12\frac{c}{|\delta_k-d_k(Y)|}\{(\delta_k-d_k(X))^2+(d_k(Y)-\delta_k)^2\}-c(\delta_k-d_k(Y))-\frac12 c^2 & \text{if } \delta_k-d_k(Y)\leq\\ \frac12(\delta_k-d_k(X))^2 & \text{if } |\delta_k-d_k(Y)|\\ \frac12\frac{c}{|\delta_k-d_k(Y)|}\{(\delta_k-d_k(X))^2+(d_k(Y)-\delta_k)^2\}+c(\delta_k-d_k(Y))-\frac12 c^2 & \text{if } \delta_k-d_k(Y)\geq\end{cases}$$

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

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

where

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

6 Convolution

In De Leeuw (2018) we also study the convolution smoother proposed by Voronin, Ozkaya, and Yoshida (n.d.). The idea is to use the convolution of the absolute value function and a *mollifier* as the smoothed function.

> A smooth function $\psi : \mathbb{R} \rightarrow \mathbb{R}$ is said to be a pdf if it is non-negative, and has area $\int \psi(x)dx = 1$. For any pdf ψ and any $c > 0$, define the parametric function $\psi_c : \mathbb{R} \rightarrow \mathbb{R}$ by: $\psi_c(x) := \frac{1}{c}\psi(\frac{1}{c})$, for all $x \in \mathbb{R}$. Then $\{\psi_c : c > 0\}$ is a family of pdf's, whose support decreases as $c \rightarrow 0$, but the volume under the graph always remains equal to one.

choose 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

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

The derivative is

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

It may not be immediately obvious in this case that $f'(x)/x$ is decreasing. 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.

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

Convolution with rectangular between c and -c gives the Huber function.

$$f(x) = \frac{1}{2c}\int_{-c}^{+c}|x-y|dy$$

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

$$f'(x) = \begin{cases} \frac{1}{c}x & \text{if } |x| \le c, \\ \operatorname{sign}(x) & \text{otherwise.} \end{cases}$$

$$w(x) = \begin{cases} \frac{1}{c} & \text{if } |x| \le c, \\ \frac{1}{|x|} & \text{otherwise.} \end{cases}$$

7 Barron Loss

Not surprisingly there are a large number of generalizations of Huber-like losses in the engineering community, and in their maze of conference publications. Without having any confidence of selecting a representative sample from the literature, we mention and discuss Barron (2017) and Barron (2019). These papers also give a large number of possibly useful references.

It is also 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.

8 Example

9 Discussion

Fixed weights

Tukey loss

10 Code

The function smacofRobust has a parameter "engine", which can be equal to smacofAV, smacofHuber, smacofTukey, or smacofConvolution. These four small modules compute the respective loss function values and weights for the IRLS procedure. This makes it easy to add additional robust loss functions.

```
smacofRobust <- function(delta,
                         weights = 1 - diag(nrow(delta)),
                         engine = smacofAV,
                         ndim = 2,
                         cons = 0,
                         itmax = 1000,
                         eps = 1e-15,
                         verbose = TRUE) {
  nobj <- nrow(delta)
  wmax <- max(weights)
  xold <- smacofTorgerson(delta, ndim)
```

```r
dold <- as.matrix(dist(xold))
h <- engine(nobj, weights, delta, dold, cons)
rold <- h$resi
sold <- sum(weights * rold)
wold <- h$wght
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 <- sum(weights * rnew)
  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])))
}

smacofAV <- function(nobj, wmat, delta, dmat, cons) {
  resi <- sqrt((delta - dmat) ^ 2 + cons)
  wght <- wmat / (resi + diag(nobj))
  return(list(resi = resi, wght = wght))
}

smacofConvolution <- function(nobj, wmat, delta, dmat, cons) {
  difi <- delta - dmat
  resi <- difi * (2 * pnorm(difi / cons) - 1) + 2 * cons * dnorm(difi / cons)
  wght <- wmat * (pnorm(difi / cons) - 0.5) / (difi + diag(nobj))
  return(list(resi = resi, wght = wght))
}

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

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

References

Barron, J. T. 2017. "A More General Robust Loss Function." https://arxiv.org/pdf/1701.03077v1.

———. 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.

De Leeuw, J. 1984. "Differentiability of Kruskal's Stress at a Local Minimum." *Psychometrika* 49: 111–13.

———. 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.

De Leeuw, J., and K. Lange. 2009. "Sharp Quadratic Majorization in One Dimension." *Computational Statistics and Data Analysis* 53: 2471–84.

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.

Heiser, W. J. 1987. "Correspondence Analysis with Least Absolute Residuals." *Computational Statistica 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.

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

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

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

Voronin, S., G. Ozkaya, and Y. Yoshida. n.d. "Convolution Based Smooth Approximations to the Absolute Value Function with Application to Non-smooth Regularization."