

Smacof at 50: A Manual

Part 2: Metric Smacof

Jan de Leeuw - University of California Los Angeles

Started December 12 2022, Version of May 03, 2024

Abstract

TBD

Contents

1	Introduction	3
2	Bells and Whistles	4
3	Implementation	5
3.1	Type AC1	5
3.2	Type AC2	5
3.3	Type AC3	6
3.4	Type AC4	6
4	Example	11
4.1	Type AC1	11
4.2	Type AC2	12
4.3	Type AC3	14
4.4	Type AC4	15
	References	18

Note: This is a working manuscript which will be expanded/updated frequently. All suggestions for improvement are welcome. All Rmd, tex, html, pdf, R, and C files are in the public domain. Attribution will be appreciated, but is not required. All files can be found at <https://github.com/deleeuw> in the repositories smacofCode, smacofManual, and smacofExamples. A barebones R package is in the repository smacofAC.

1 Introduction

In this part of the manual we discuss metric MDS, and the program smacofAC. Metric MDS is the core of all smacof programs, because they all have the majorization basec on the Guttman transform in common.

2 Bells and Whistles

There are two options, *bounds* and *constant*, to make smacofAC more interesting and more widely applicable. Using these options the metric MDS problem becomes minimization of

$$\sigma(X, \hat{D}) = \sum \sum w_{ij} (\hat{d}_{ij} - d_{ij}(X))^2$$

over both X and \hat{D} , allowing some limited “metric” transformations of the data Δ . The four “metric” types of transformations are

1. type AC1: if bounds = 0 and constant = 0 $\hat{d}_{ij} = \delta_{ij}$.
2. type AC2: if bounds = 0 and constant = 1 $\hat{d}_{ij} = \delta_{ij} + c$ for some c ,
3. type AC3: if bounds = 1 and constant = 0 $\delta_{ij}^- \leq \hat{d}_{ij} \leq \delta_{ij}^+$,
4. type AC4: if bounds = 1 and constant = 1 $\delta_{ij}^- + c \leq \hat{d}_{ij} \leq \delta_{ij}^+ + c$ for some c ,

Here Δ^- and Δ^+ are known matrices with bounds, and c is the unknown additive constant. All four types of transformations also require that $\hat{d}_{ij} \geq 0$ for all (i, j) . There are extensions of the smacof theory (Heiser (1991)) that do not require non-negativity of the disparities, but in the implementations in this manual we always force them to be non-negative. Note that AC3 is AC4 with $c = 0$ and AC2 is AC4 with $\Delta^- = \Delta^+ = \Delta$.

Note that for types AC2 and AC4 the data Δ do not need to be non-negative. In fact, the original motivation for the additive constant in classical scaling (Messick and Abelson (1956)) was that Thurstonian analysis of paired or triadic comparisons produced dissimilarities on an interval scale, and thus could very well include negative values.

In AC3 and AC4 there is no mention of Δ , which means the bounds Δ^- and Δ^+ are actually the data. We could collect dissimilarity data by asking subjects for interval judgments. Instead of a rating scale with possible responses from one to ten we could ask for a mark on a line between zero and ten, and then interpret the marks as a choice of one of the intervals $[k, k + 1]$. These finite precision or interval type of data could even come from physical measurements of distances. Thus the bounds parameter provides one way to incorporate uncertainty into MDS, similar to interval analysis, fuzzy computing, or soft computing.

The non-negativity requirement for \hat{D} implies bounds for the additive constant c . In AC2 we need $c \geq -\min \delta_{ij}$ to maintain non-negativity. For AC4 we must have $c \geq -\min \delta_{ij}^+$, otherwise the constraints on the transformation are inconsistent. Clearly for consistency of AC3 and AC4 we require that $\delta_{ij}^- \leq \delta_{ij}^+$ for all (i, j) . It makes sense in most situations to choose Δ^- and Δ^+ to be monotone with Δ , but there is no requirement to do so.

3 Implementation

3.1 Type AC1

This is standard non-metric smacof, no bells and whistles. The user has to make some choices, coded in the parameter file.

```
nobj    9
ndim    2
init     2
width   10
precision 6
haveweights 0
havelabels 1
itmax   10000
epsi    10
verbose  1
kitmax   1
kepsi    1
kverbose  0
constant 1
bounds  1
alpha   10
```

The parameters *constant*, *bounds*, *alpha*, *kitmax*, *kepsi*, and *kverbose* are only relevant for AC2, AC3, and AC4. Nevertheless even for AC1 they should have integer values, it just doesn't matter what these values are. Parameter *ndim* is the number of dimensions, and *init* tells if an initial configuration is read from a file (*init* = 1), is computed using classical scaling (*init* = 2), or is a random configuration (*init* = 3). If *haveweights* is one a vector of weights is read from a file, of *labels* is one a vector of labels for the configuration plot is read in. Parameters *itmax*, *epsi*, and *verbose* control the iterations. The maximum number of iterations is *itmax*, the iterations stop if the decrease of stress in an iteration is less than $1E\text{-}epsi$, and if *verbose* is one intermediate iteration results are written to stdout. These intermediate iteration results are formatted with the R function `formatC()`, using *width* for the width argument and *precision* for the digits argument.

3.2 Type AC2

For AC2 we also optimize over the additive constant c , and thus the ALS algorithm has two sub-steps. The first sub-step consists of a number of Guttman iterations to update X for given \hat{D} (i.e. for given c) and the second sub-step updates c for given X . Parameters *kitmax*, *kepsi*, and *kverbose* control the inner iterations in the first sub-step in the same way as *itmax*, *epsi*, and *verbose* control the outer iterations that include both sub-steps. No inner iterations are used to update the additive constant, which only requires computing the weighted average.

$$c = -\frac{\sum \sum w_{ij}(\delta_{ij} - d_{ij}(X))}{\sum \sum w_{ij}} \quad (1)$$

AC2 should give the same results as the technique of Cooper (1972).

3.3 Type AC3

The algorithm for AC3 has the same structure as that for AC2. Instead of a second sub-step computing the additive constant, the second sub-step computes \hat{D} by squeezing the $D(X)$ into the bounds. Thus

$$\hat{d}_{ij} = \begin{cases} \delta_{ij}^- & \text{if } d_{ij}(X) < \delta_{ij}^-, \\ \delta_{ij}^+ & \text{if } d_{ij}(X) > \delta_{ij}^+, \\ d_{ij}(X) & \text{otherwise.} \end{cases} \quad (2)$$

Obviously no iterations are required in the second sub-step.

For AC3 the parameter *alpha* may also be relevant. If the *bounds* parameter is equal to one then Δ^- and Δ^+ are read from a file (as vectors with the lower diagonal elements in row-major order, as is the case for Δ and W). If *bounds* is two we set

$$\delta_{ij}^- = \delta_{ij} - \frac{\text{range}(\Delta)}{\alpha}, \quad (3a)$$

$$\delta_{ij}^+ = \delta_{ij} + \frac{\text{range}(\Delta)}{\alpha}, \quad (3b)$$

and if *bounds* is three we set

$$\delta_{ij}^- = (1 - \frac{1}{\alpha})\delta_{ij}, \quad (4a)$$

$$\delta_{ij}^+ = (1 + \frac{1}{\alpha})\delta_{ij}. \quad (4b)$$

In keeping with the general smacof conventions for parameters we use an integer parameter for α .

3.4 Type AC4

Of the four regression problems in the second ALS sub-step only the one for AC4 with both bounds and additive constant is non-trivial. We'll give it some extra attention.

It may help to give an example of what it actually requires. We use the De Gruijter example with nine Dutch political parties from 1967 (De Gruijter (1967)). For ease of reference we include the data here. Dissimilarities are averages over a group of 100 introductory psychology students.

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

We compute distances from the Torgerson solution. The Shepard plot for $c = 0$ and the Torgerson distances is in figure 1. The two blue lines are connecting the δ_{ij}^- and the δ_{ij}^+ , i.e. they give the bounds for $c = 0$. In our example the lines are parallel, because $\delta_{ij}^+ - \delta_{ij}^- = 2$ for all (i, j) , but in general this may not be the case. The points between the two lines do not contribute to the loss, and the points outside the band contribute by how much they are outside, as indicated by the black vertical lines.

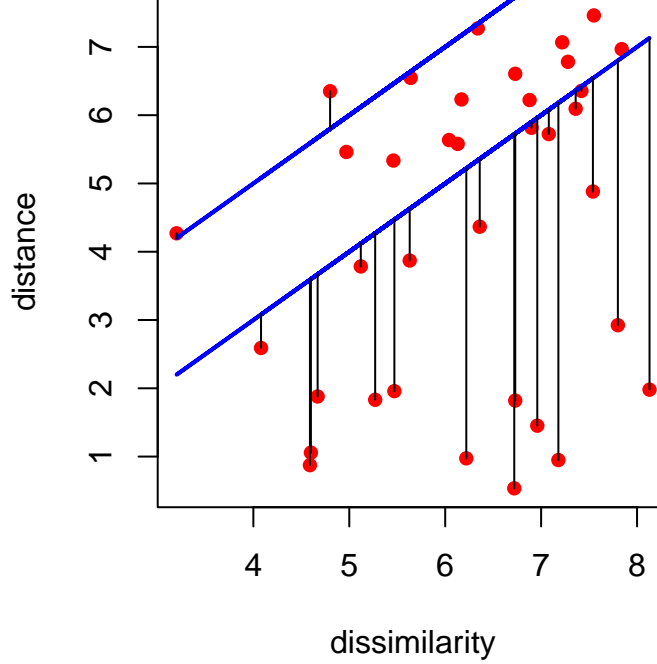


Figure 1: De Gruijter Shepard Plot

By varying c we shift the region between the two parallel lines upwards or downwards. The width of the region, or more generally the shape, always remains the same, because it is determined by the difference of δ^+ and δ^- and does not depend on c . The optimal c is that shift for which the red $(\delta_{ij}, d_{ij}(X))$ points are as much as possible within the strip between the δ^- and δ^+ lines. This is in the least squares sense, which means that we minimize the horizontal squared distances from the points outside the strip to the δ^- and δ^+ lines (i.e. the black vertical lines).

Let's formalize this. Define

$$\phi_{ij}(c) := \min_{\delta_{ij} \geq 0} \{(\delta_{ij} - d_{ij}(X))^2 \mid \delta_{ij}^- + c \leq \delta_{ij} \leq \delta_{ij}^+ + c\} \quad (5)$$

and

$$\phi(c) := \sum \sum w_{ij} \phi_{ij}(c) \quad (6)$$

The constraints are consistent if $\delta_{ij}^+ + c \geq 0$, i.e. if $c \geq c_0 := -\min \delta_{ij}^+$. The regression problem is to minimize ϕ over $c \geq c_0 := -\min \delta_{ij}^+$.

Figure 2 has an example of one of the ϕ_{ij} . The value of the $d_{ij}(X)$ we used is 4.2698418, δ_{ij} is 3.2, δ_{ij}^- is 2.2, and δ_{ij}^+ is 4.2. The two red vertical lines are at $c = d_{ij}(X) - \delta_{ij}^+$ and $c = d_{ij}(X) - \delta_{ij}^-$.

Now

$$\hat{d}_{ij} = \begin{cases} \delta_{ij}^- + c & \text{if } c \geq d_{ij}(X) - \delta_{ij}^-, \\ \delta_{ij}^+ + c & \text{if } c \leq d_{ij}(X) - \delta_{ij}^+, \\ d_{ij}(X) & \text{otherwise.} \end{cases} \quad (7)$$

and thus

$$\phi_{ij}(c) = \begin{cases} (d_{ij}(X) - (\delta_{ij}^- + c))^2 & \text{if } c \geq d_{ij}(X) - \delta_{ij}^-, \\ (d_{ij}(X) - (\delta_{ij}^+ + c))^2 & \text{if } c \leq d_{ij}(X) - \delta_{ij}^+, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

It follows that ϕ_{ij} is piecewise quadratic, convex, and continuously differentiable. The derivative is piecewise linear, continuous, and increasing. In fact

$$\mathcal{D}\phi_{ij}(c) = \begin{cases} 2(c - (d_{ij}(X) - \delta_{ij}^-)) & \text{if } c \geq d_{ij}(X) - \delta_{ij}^-, \\ 2(c - (d_{ij}(X) - \delta_{ij}^+)) & \text{if } c \leq d_{ij}(X) - \delta_{ij}^+, \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

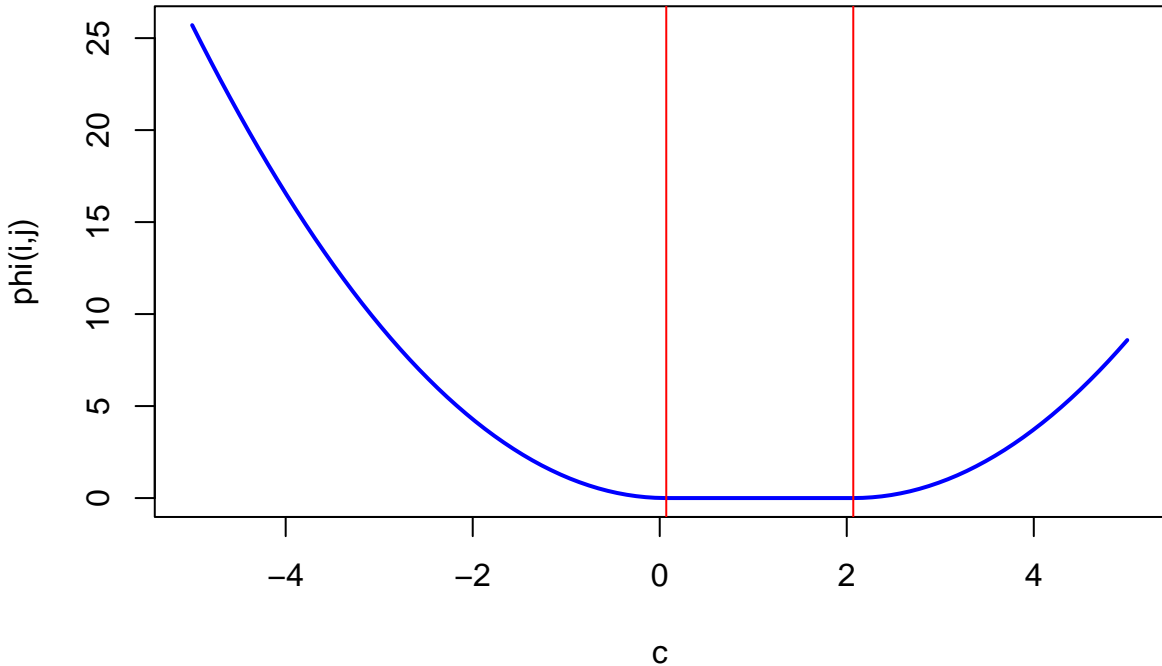


Figure 2: De Gruijter example, a single $\phi(i,j)$

Since ϕ is a positive linear combination of the ϕ_{ij} it is also piecewise quadratic, convex, and continuously differentiable, with a continuous piecewise linear increasing derivative. Note ϕ is **not** twice-differentiable and **not** strictly convex. Figure 3 has a plot of ϕ for the De Gruijter example. The red vertical lines are at $c = c_0$ and at $c_1 := \max\{d_{ij}(X) - \delta_{ij}^-\}$. From (9) we see that if $c \geq c_1$ then $\mathcal{D}\phi(c_1) \geq 0$ and thus we can look for the optimum c in the interval $[c_0, c_1]$.

We minimize ϕ by using the R function `optimize()`. The minimum of ϕ in our De Gruijter example turns out to be 75.251647, attained at c equal to -2.1403321. The Shepard plot corresponding

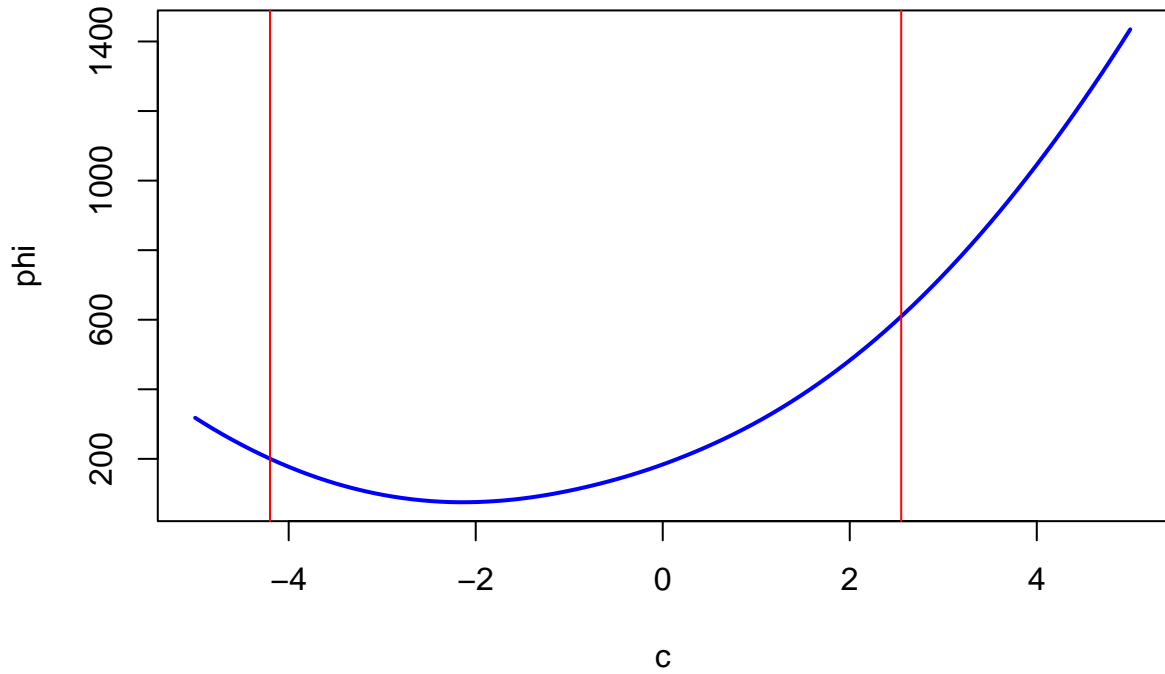


Figure 3: Function phi for De Gruijter example

with the optimum c is plotted in figure 4. Keep in mind that this is the optimal c for the initial configuration only (and for our specific choice of bounds). As we will see in a moment the fit can be much improved.

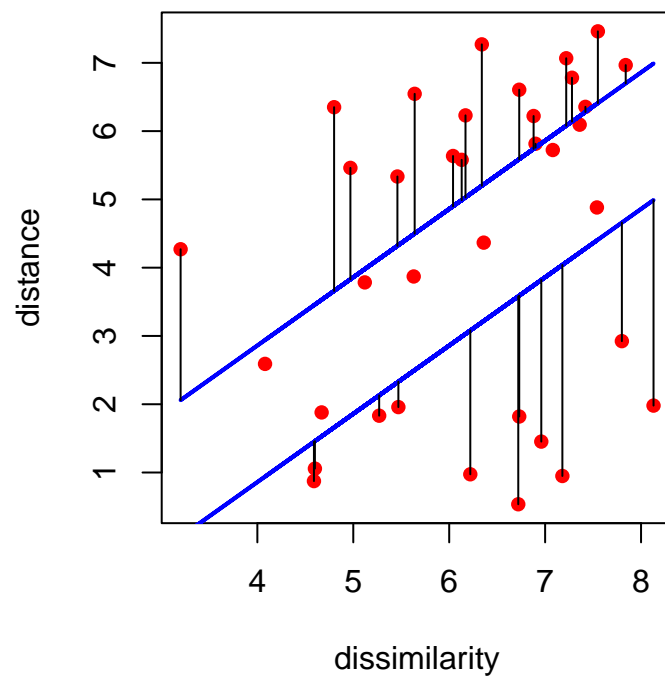


Figure 4: De Gruijter Shepard Plot

4 Example

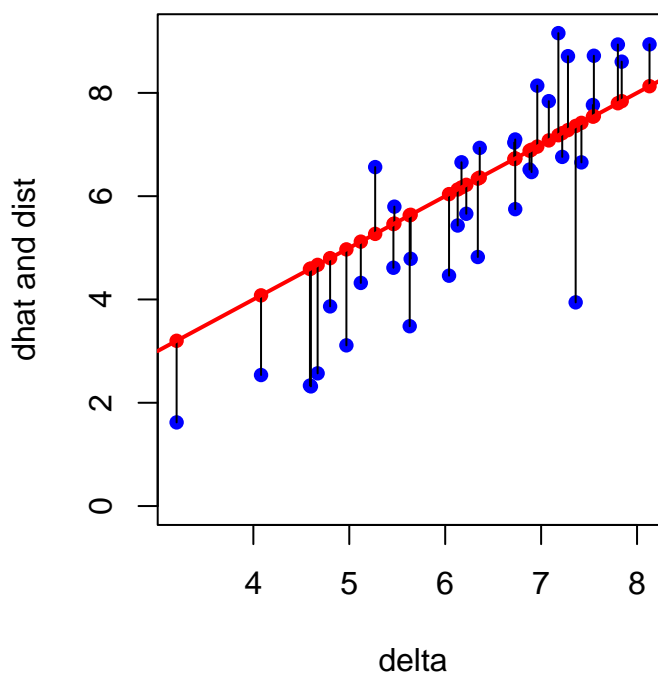
In the example we again use the De Gruijter data, but now we compute optimal solutions for all four types AC1-AC4 (two dimensions, Torgerson initial estimate, no weights). We iterate until the difference in consecutive stress values is less than $1e-10$. For each of the four runs we give the number of iterations, the final stress, and the additive constant in case of AC2 and AC4. We also make three plots: the Shepard plot with points $(\delta_{ij}, d_{ij}(X))$ in blue and with points $(\delta_{ij}, \hat{d}_{ij})$ in red, the configuration plot with a labeled X , and the dist-dhat plot with points $(d_{ij}(X), \hat{d}_{ij})$ scattered around the line $d = \hat{d}$. Line segments are drawn in the plots to show the fit of all pairs (i, j) .

4.1 Type AC1

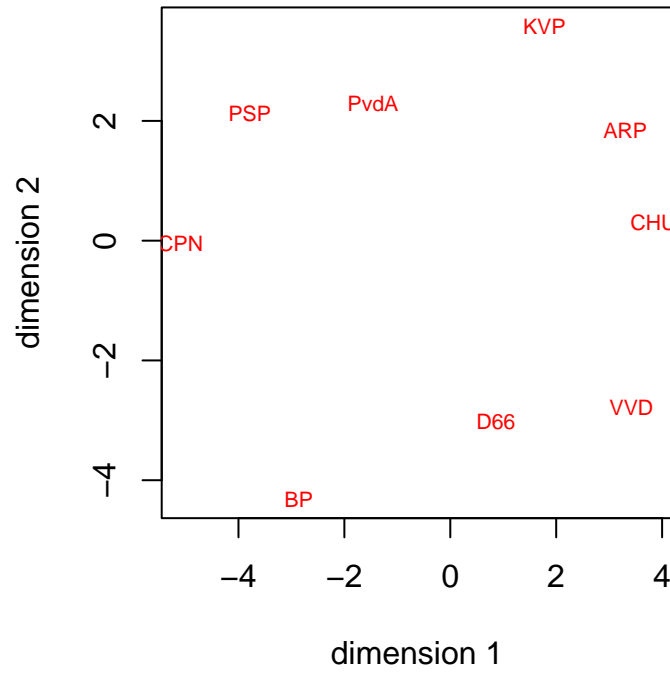
For AC1 we find a minimum stress of 32.2208145 after 552 iterations. The Shepard plot has a substantial intercept, which suggest that an additive constant may improve the fit. This is typical for average similarity judgments over heterogeneous groups of individuals. It is the reason why Ekman (1954) linearly transformed his average similarities so that the smallest became zero and the largest became one. That amounts to applying the additive constant before the MDS analysis.

To give some content to the configuration plot: CPN (communists), PSP (pacifists), and PvdA (social democrats) are leftists parties, ARP (protestants), CHU (other protestants), KVP (catholics) are religious parties, BP (farmers) is a right-wing protest party, VVD (classical liberals) is a conservative party, and D'66 (pragmatists, centrists) was brand new in 1967 and was supposedly beyond left and right.

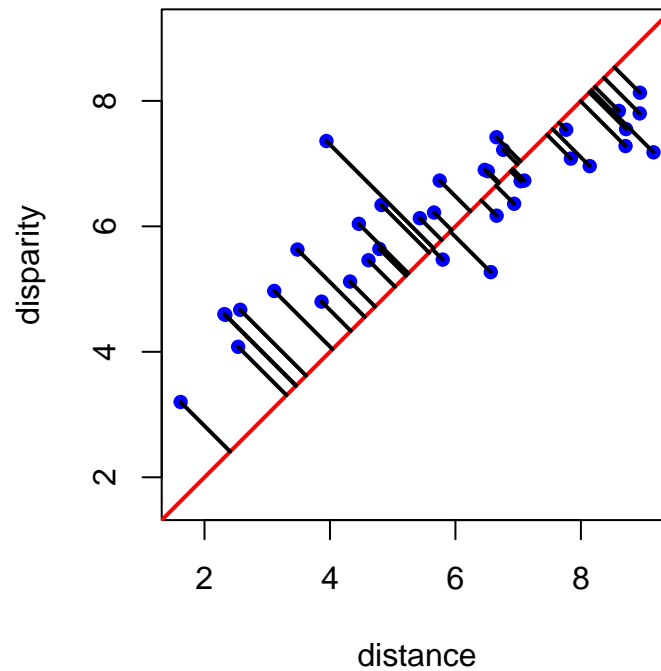
Shepard Plot AC1



Configuration Plot AC1



Dist-Dhat Plot AC1

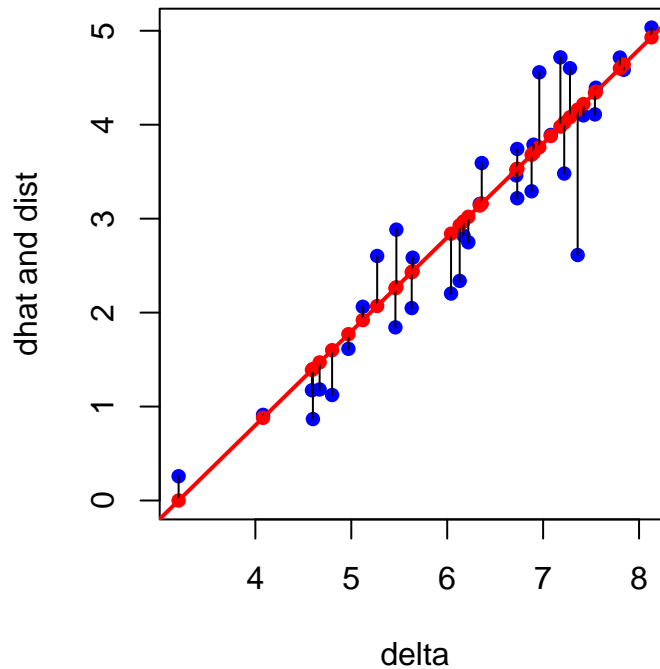


4.2 Type AC2

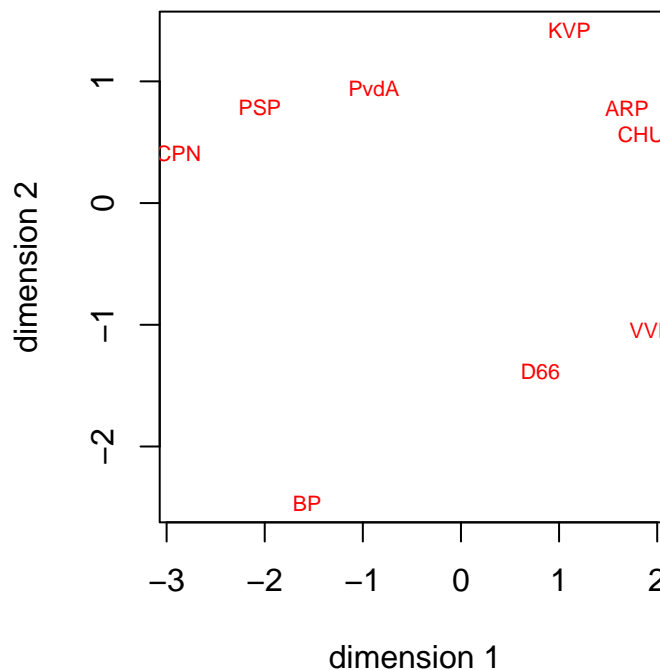
As expected, the additive constant improves the fit. We have convergence after 89 iterations to stress 3.6661492. The additive constant is -3.2, which means the smallest $\delta_{ij} + c$, between ARP

and CHU, is now zero. The configuration shows the same three groups, but they cluster a bit more tightly. This is to be expected. Without the additive constant the dissimilarities are more equal and consequently the distances are more equal to. The configuration tends more to what we see if all dissimilarities are equal, i.e. to points regularly spaced on a circle (De Leeuw and Stoop (1984)).

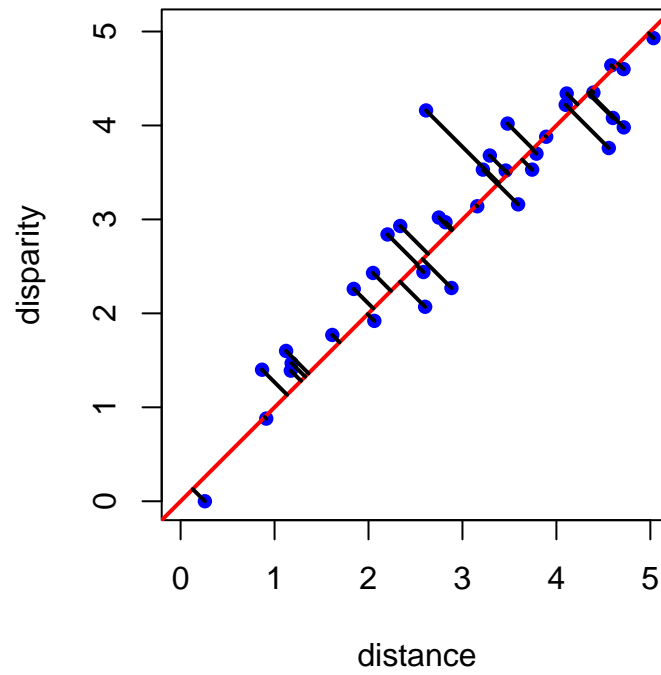
Shepard Plot AC2



Configuration Plot AC2



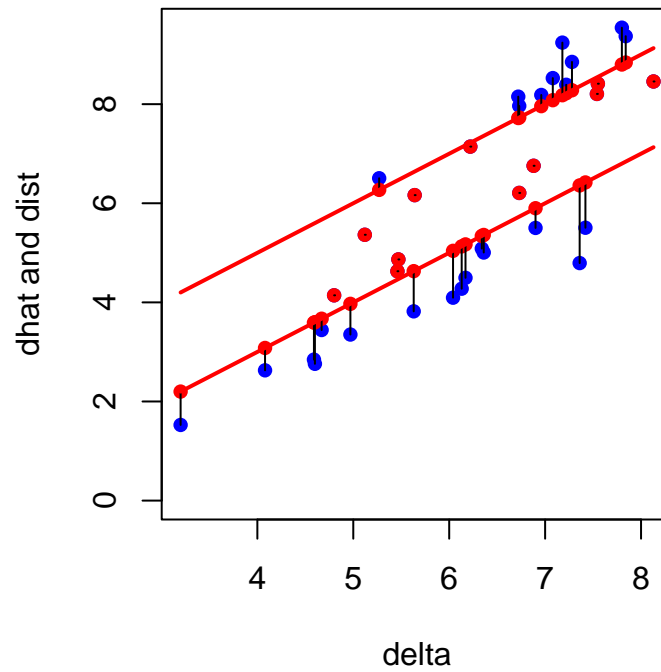
Dist-Dhat Plot AC2



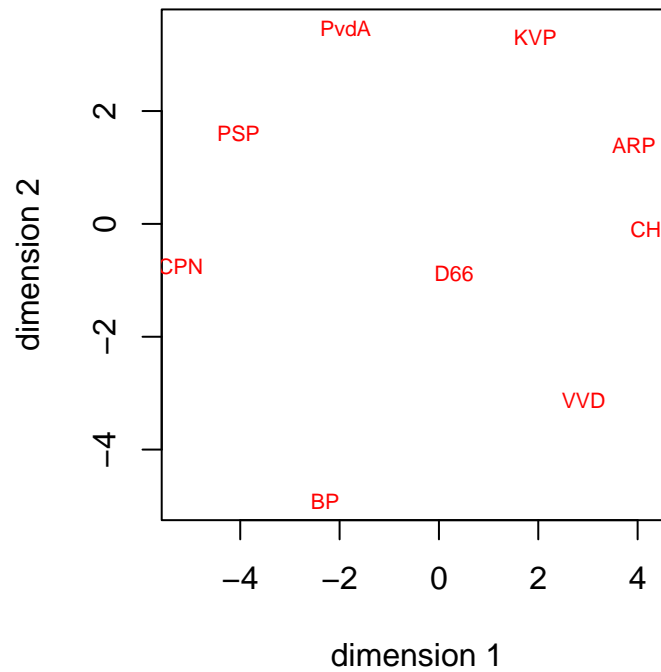
4.3 Type AC3

The bounds we use are $\delta_{ij} \pm 1$. After 271 iterations we arrive at stress 5.7972. In the configuration plot the centrists have moved to the center.

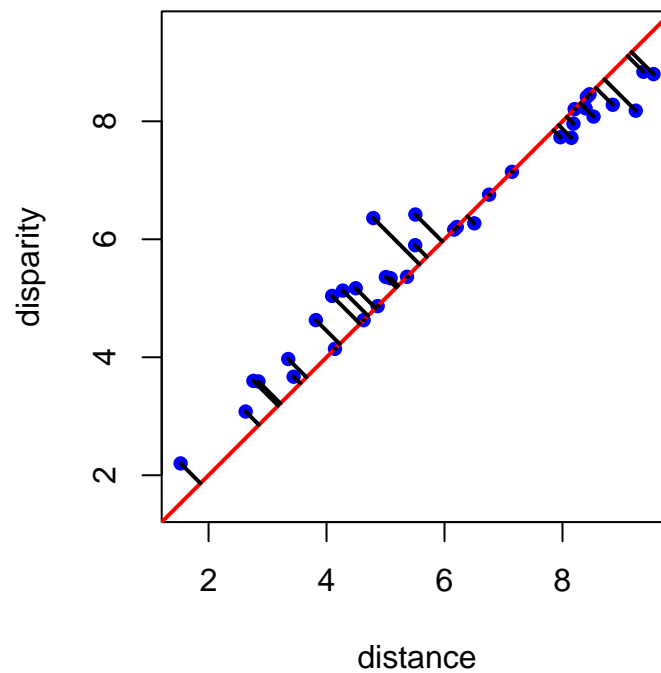
Shepard Plot AC3



Configuration Plot AC3



Dist-Dhat Plot AC3

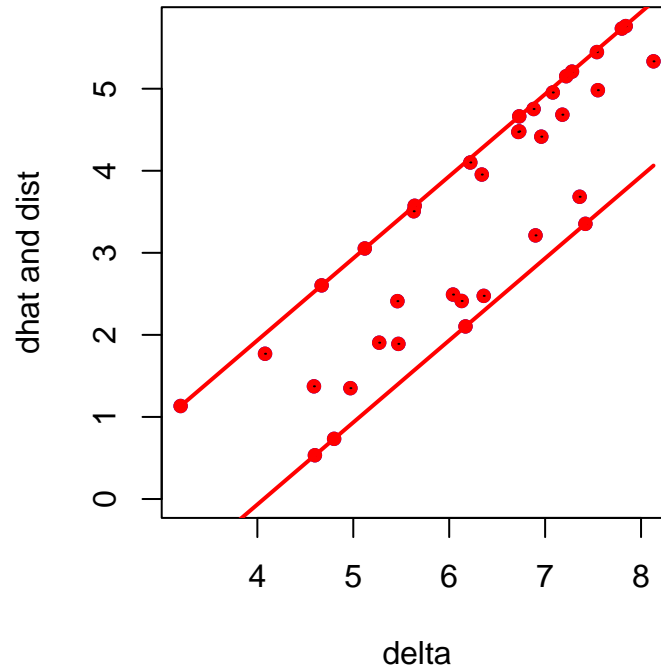


4.4 Type AC4

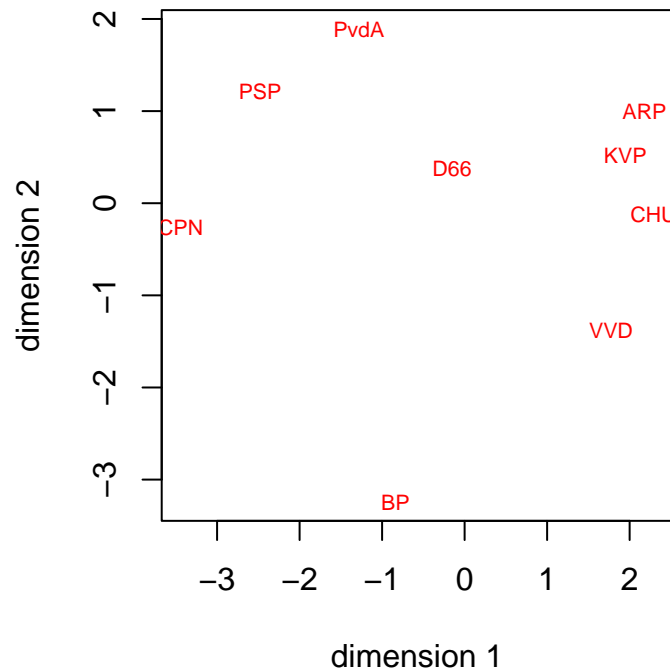
After 466 iterations stress is 1.8821595×10^{-8} , i.e. practically zero. We succeeded in moving all distances within the bounds. The additive constant is -3.0668025. The configuration is again pretty

much the same with D'66 in the center. VVD moves closer to the Christian Democrats, and BP is more isolated.

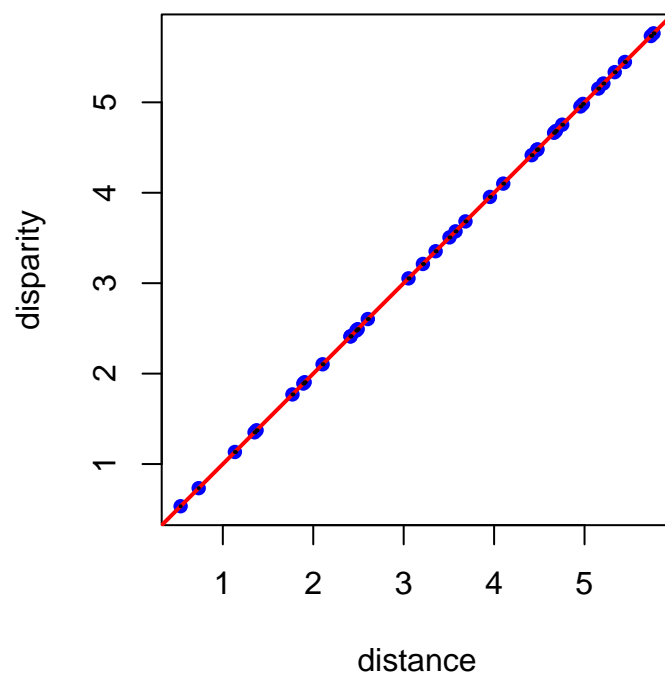
Shepard Plot AC1



Configuration Plot AC1



Dist-Dhat Plot AC1



References

- Cooper, L. G. 1972. "A New Solution to the Additive Constant Problem in Metric Multidimensional Scaling." *Psychometrika* 37 (3): 311–22.
- 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., and I. Stoop. 1984. "Upper Bounds for Kruskal's Stress." *Psychometrika* 49: 391–402.
- Ekman, G. 1954. "Dimensions of Color Vision." *Journal of Psychology* 38: 467–74.
- Heiser, W. J. 1991. "A Generalized Majorization Method for Least Squares Multidimensional Scaling of Pseudodistances that May Be Negative." *Psychometrika* 56 (1): 7–27.
- Messick, S. J., and R. P. Abelson. 1956. "The Additive Constant Problem in Multidimensional Scaling." *Psychometrika* 21 (1–17).