Smacof at 50: A Manual
Part zz: smacofSS: Strain and Sstress

Jan de Leeuw

December 7, 2024

Table of contents

1 Introduction

The best way to avoid unwanted (i.e. non-global) local minima in smacof is to start the iterations with an excellent initial configuration. Even if the iterations converge to a non-global local minimum, at least we can be sure than the solution has a lower stress than the initial configuration, which was supposedly already excellent.

In this chapter we implement two fairly elaborate initial estimates for metric smacof, which can of course also be used in the alternating least squares algorithm for non-linear and non-metric smacof. Both initial estimates basically iteratively minimize alternative loss functions, respectively called sstress and strain (see De Leeuw and Heiser (1982) for a definition and comparison of these loss functions).

The R functions implementing the minimization of these loss functions can be interpreted (and can actually be used) as alternative MDS methods. It may seem somewhat peculiar to start an iterative MDS technique with an initial estimate computed with another iterative MDS technique. But it is not exactly unprecedented, since traditionally MDS programs compute their initial configurations by using classical MDS (which uses iterative eigenvalue-eigenvector methods).

Our initial configuration techniques have a great deal of flexibility, because we do not necessarily iterate to convergence in the initial phase. We use the option in monotone iterative algorithms to merely improve instead of completely solve.

Also keep in mind that our objective is to find excellent local minima of stress, in fact our ultimate objective is to find the global minimum. This makes it desirable to find as good an initial approximation as possible, even if that requires a considerable amount of computation.

2 Strain

In the standard R versions of smacof the initial configuration is chosen using classical multidimensional scaling, i.e. the method proposed in Torgerson (1958) and Gower (1966).

Classical scaling is based on the Euclidean embedding theorem of Schoenberg (1935), revealed to psychometricians by G. Young and Householder (1938). A real hollow symmetric matrix Δ of order n, with non-negative entries, is a Euclidean distance matrix if and only if the matrix

$$B := -\frac{1}{2}J\Delta^2 J \tag{1}$$

is positive semi-definite. Here Δ^2 is the elementwise square of Δ and $J = I - n^{-1}ee'$ is the centering matrix. Moreover if B is positive semi-definite then the embedding dimension is the rank of B. Classical scaling computes B and its eigen-decomposition. It then constructs the initial estimate X for p-dimensional MDS by using the p largest eigenvalues Λ of B, and the corresponding eigenvectors in K, to give $X = K\Lambda^{\frac{1}{2}}$.

Classical scaling, as an MDS technique, has one major advantage over other forms of MDS, but also some disadvantages. The advantage is that it finds the global minimum of the strain loss function, defined as

$$\sigma(X) := \frac{1}{4}\text{tr } J(\Delta^2 - D^2(X))J(\Delta^2 - D^2(X))) \tag{2}$$

with $D^2(X)$ the squared Euclidean distances. So not only does classical scaling minimize strain, but it actually computes its global minimumm. If the ordered eigenvalues of B from Equation 1 satisfy $\lambda_p(B) > \lambda_{p+1}(B)$ then the global minimum over p-dimensional configurations is unique (up to rotation and translation).

To see that minimizing strain is an eigenvalue problem in disguise, note that if X is column centered then $-\frac{1}{2}JD^2(X)J = XX'$ and thus $-\frac{1}{2}J(\Delta^2 - D^2(X))J = B - XX'$. Taking the sum of squares on both sides of the equation, and using the fact that J is idempotent, gives the alternative expression for strain

$$\sigma(X) = \text{tr } (B - XX')^2. \tag{3}$$

Minimizing Equation 3 over p-dimensional configurations is indeed classical scaling. The matrix form in Equation 2 strain was first given by De Leeuw and Heiser (1982), although equivalent versions, using different notation, are already in Gower (1966) and Mardia (1978).

The first disadvantage of classical scaling is that the B-matrix may only have $q < p$ positive eigenvalues. If that is the case the global minimizer X_p of strain in $p < q$ dimensions does not exist. The global minimizer in $r \leq p$ dimensions has only q non-zero dimensions (and

$p - q$ zero dimensions). In practice this disadvantage is often not a very serious one, because having fewer than p positive eigenvalues suggests a bad fit so that maybe MDS is not a good technique choice in the first place. Because n is usually much larger than p matrix B is bound to have more than p positive eigenvalues. If not, one can always compute an additive constant to force positive semi-definiteness of B (Cailliez (1983)).

The second disadvantage is that classical scaling, unlike smacof, has no provision to include data weights w_{ij} for each of the dissimilarities δ_{ij}. Or, if there are weights it is unclear if they should be applied to Equation 2 or Equation 3, and no matter where they are used the two definitions of strain are no longer equivalent and the global minimum advantage of classical scaling is lost.

In particular this disadvantage also means that there cannot be missing dissimilarities, or, more precisely, something additional has to be done if there are missing data. One obvious possibility is to use alternating least squares to minimize strain, alternating a step to impute the missing dissimilarities and a step to compute the optimal configuration. The current smacof program in R imputes the missing dissimilarities by replacing them with the average non-missing dissimilarity, which is computationally convenient but not very satisfactory. This second disadvantage also means there is no straightforward version of classical scaling for multidimensional unfolding, in which all within-set dissimilarities are missing.

Part of the problem is the double centering operator J, because it requires complete data. This problem can be alleviated if we have one object, say the first one, for which there are no missing data. We then put that object in the origin of the configuration and compute $-\frac{1}{2}\{\delta_{ij}^2-\delta_{1i}^2-\delta_{1j}^2\}$, which is equal to $x_i'x_j$ for Euclidean dissimilarities. We can then define a version of strain on the non-missing elements of B, but we still need a low-rank approximation of a symmetric matrix with missing data. That is, we need low-rank symmetric matrix completion, for which there is a gigantic literature (Nguyen, Kim, and Shim (2019)), although that literature mostly addresses the rectangular case.

Another disadvantage, or maybe I should say peculiarity, is emphasized by De Leeuw and Meulman (1986). If Δ is Euclidean then Gower (1966) shows that

$$d_{ij}^2(X_1) \leq d_{ij}^2(X_2) \leq \cdots \leq d_{ij}^2(X_r) = \delta_{ij}^2, \tag{4}$$

where X_p is the p-dimensional classical scaling solution and r is the rank of B. Thus squared dissimilarities are approximated from below, which may not be the most obvious way to approximate. If B has negative eigenvalues then Equation 4 is no longer true and we have

$$d_{ij}^2(X_1) \leq d_{ij}^2(X_2) \leq \cdots \leq d_{ij}^2(X_s) \geq \delta_{ij}^2, \tag{5}$$

where s is the number of positive eigenvalues.

Another consideration, discussed in the excellent paper by Bailey and Gower (1990), is that strain is the sum of squares over all n^2 residuals, which means each off-diagonal element is

used twice, each diagonal element only once. In minimizing stress or sstress the diagonal does not play a role, minimizing over all elements below the diagonal gives the same result as minimizing over all elements.

And finally the squaring and double-centering of the dissimilarities may not be such a good idea from a statistical point of view. Squaring will emphasize large errors, and can easily lead to outliers. Double-centering introduces dependencies between different observations, because the pseudo scalar products in B are linear combinations of multiple (squared) dissimilarities.

Nevertheless the smacof project, and the manual and code for this chapter, includes a generalization of classical MDS that can handle missing data and weights. A similar non-metric version of strain was discussed by Trosset (1998).

2.1 Loss Function

We introduce two generalizations of strain. The first one introduces weights. Suppose V is a symmetric and doubly-centered positive semi-definite matrix of rank $n-1$. Define strain as

$$\sigma(X) := \frac{1}{4}\text{tr}\, V(\Delta^2 - D^2(X))V(\Delta^2 - D^2(X)). \tag{6}$$

with

$$V = \sum_{1\leq i<j\leq n}\sum w_{ij}A_{ij}. \tag{7}$$

If all weights are equal to one we are back to Equation 2.

With Equation 6 the global optimization advantage of classical scaling is still maintained. Suppose $s_i = x_i'x_i$. Then $D^2(X) = se' + es' - 2XX'$. Thus with $B := -\frac{1}{2}V\Delta^2V$ we have residuals $-\frac{1}{2}V(\Delta^2 - D^2(X)V = B - VXX'V$ and Equation 6 is the sum of squares of these residuals. In the Euclidean case we can recover VX from B, and set $X = V^+K\Lambda L'$, with K and Λ from the eigen-analysis of B and L an arbitrary rotation. For non-Euclidean Δ^2 different V will give different solutions.

Having weights in a doubly-centered matrix, and not in a diagonal matrix, makes it difficult to interpret and analyze the influence of weighting. We have chosen to use double-centered V because it preserves the global minimum property of classical scaling. And we have insisted that V has rank $n-1$ so that we can uniquely find X from VX. Computationally there is no problem if we drop these two constraints on V.

The second generalization makes Δ^2, and thus B, a function of a number of optimal scaling parameters. Trosset (1998) uses this to implement a non-metric scaling method based on strain, but we will use it as a way to handle missing data. Strain becomes

$$\sigma(X,\theta) := \text{tr}\, V^2(\Delta^2(\theta) - D^2(X))V^2(\Delta^2(\theta) - D^2(X)) = \text{tr}\,(B(\theta) - VXX'V)^2, \tag{8}$$

where

$$\Delta^2(\theta) := \Delta_0^2 + \sum_{1\leq i<j\leq n}\sum \theta_{ij}E_{ij}. \tag{9}$$

Here Δ_0^2 is the matrix with squared dissimilarities, with elements equal to zero for missing data. The $E_{ij} := e_ie_j' + e_je_i'$ indicate missing data. We require that $\theta_{ij} \geq 0$ if (i, j) is missing, and $\theta_{ij} = 0$ otherwise. Define

$$B(\theta) := -\frac{1}{2}V\Delta^2(\theta)V = B_0 - \sum \theta_k T_k, \tag{10}$$

where the T_{ij} are of the form

$$T_{ij} := \frac{1}{2}VE_{ij}V = \frac{1}{2}(v_iv_j' + v_jv_i'). \tag{11}$$

2.2 Algorithm

Strain from Equation 8 must be minimized over configurations X and over $\theta \geq 0$. We use alternating least squares. Minimizing

$$\sigma(X, \theta) = \text{tr } (B(\theta) - VXX'V)^2 \tag{12}$$

over VX for fixed θ is classical MDS. Minimizing

$$\sigma(X, \theta) = \text{tr } ((B_0 - VXX'V) - \sum_{1\leq i<j\leq n}\sum \theta_{ij}T_{ij})^2 \tag{13}$$

over θ for fixed X is a non-negative linear least squares problem.

For the first subproblem we use the eigs_sym function from the RSpectra package (Qiu and Mei (2024)), for the second the nnls package and function (Mullen and van Stokkum (2023)). We alternate the two subproblems in an outer iteration. Both eigs-sym and nnls have a cold start and both iterate until convergence, which may be wasteful in later outer iterations. We may eventually replace them with the hot start majorization methods in smacofEigenRoutines.R and smacofNNLS.R (both in the smacofUtilities directory in smacofCode). If there are no missing data and $V = J$ the program just performs classical MDS.

3 Maximum Sum

In some early reports (De Leeuw (1968a), De Leeuw (1968b)) I proposed using what I call the "maximum sum principle", not just for metric MDS, but for various other metric and nonmetric techniques as well. Around the same time Guttman (1968) used a similar initial configuration for his MDS algorithms.

In metric MDS the maximum sum principle maximizes

$$\rho(X) := \sum_{1<i\leq j<n}\sum w_{ij}\delta_{ij}^2 d_{ij}^2(X) \tag{14}$$

over X in some compact set of configurations. General considerations seem to suggest that ρ will tend to be large if Δ and $D(X)$ are numerically similar, or at least similarly ordered. The actual normalization constraint on X is not specified, and different constraints will lead to different solutions.

Now $\rho(X) = \text{tr } X'BX$ where B is defined as

$$B := \sum_{1<i\leq j<n}\sum w_{ij}\delta_{ij}^2 A_{ij}, \tag{15}$$

i.e. B has off-diagonal elements $-w_{ij}\delta_{ij}^2$, with the diagonal filled in such a way that rows and columns add up to zero. It follows that B is symmetric, doubly-centered, and positive semi-definite.

But no matter how simple and attractive ρ is, we still have to decide how to bound X and how to introduce multi-dimensionality. A naive choice would be requiring tr $X'X = 1$, but that gives a solution X of rank one with all columns equal to the eigenvector corresponding with the largest eigenvalue of B. De Leeuw (1970) suggests maximizing ρ over $X'X = I$, which leads to choosing X as the eigenvectors corresponding with the p largest eigenvalues of B. But that result is of limited usefulness, because the MDS problem does not specify anywhere that the configuration X must be orthonormal. But since the maximum sum solution is only to be used as an initial configuration this may not be that serious.

Guttman (1968) says his initial configuration maximizes

$$\lambda := \sum_{s=1}^{p}\frac{x_s'Bx_s}{x_s'x_s} \tag{16}$$

But that cannot be correct. In the first place it would mean that the x_s can be scaled independently and arbitrarily, in the second place the maximum of λ is just p times the largest eigenvalue of B and all x_s are proportional to the corresponding eigenvector. Thus Guttman's derivation is wrong.

Both De Leeuw and Guttman seem to arrive, in somewhat mysterious ways, at the "solution" $X = K\Lambda^{\frac{1}{2}}$, where K and Λ are the p dominant eigenvectors and eigenvalues of B. An ad hoc justification of this normalization interprets it as a two step optimization over $X = K\Psi$ with $K'K = I$ and Ψ diagonal. First maximize $\rho = \text{tr } K'BK$ over $K'K = I$. Then maximize $\rho = \text{tr } \Psi K'BK\Psi = \text{tr } \Lambda\Psi^2$ over tr $\Psi^4 = 1$, with Λ again the p largest eigenvalues. This gives $\Psi = \Lambda^{\frac{1}{2}}$. But note that if $p = n - 1$ then $K\Lambda K'$ reproduces B, and

$$d_{ij}^2(X) = (e_i - e_j)'B(e_i - e_j) = \sum_{k=1}^{n} w_{ik}\delta_{ik}^2 + \sum_{k=1}^{n} w_{jk}\delta_{jk}^2 + 2w_{ij}\delta_{ij}^2, \tag{17}$$

which does not reproduce the squared dissimilarities. Thus the maximum sum method gives at best a quick and dirty initial configuration

4 Using Sstress

4.1 Loss Function

The maximum sum method is related to the problem of minimizing the MDS loss function sstress, defined by Takane, Young, and De Leeuw (1977) as

$$\sigma(X) = \sum_{1<i\leq j<n}\sum w_{ij}(\delta_{ij}^2 - d_{ij}^2(X))^2. \tag{18}$$

For notational convenience only, assume the sum of squares of the squared dissimilarities is equal to one. Of course normalizing dissimilarities does not change the minimization problem.

If we expand Equation 18 we find

$$\sigma(X) = 1 - 2\rho(X) + \eta^2(X), \tag{19}$$

with ρ defines as in Equation 14, and with

$$\eta^2(X) := \sum_{1<i\leq j<n}\sum w_{ij}d_{ij}^4(X). \tag{20}$$

Now, by homogeneity of the distance function,

$$\min_X \sigma(X) = \min_{\eta^2(X)=1} \min_\lambda\{1 - 2\lambda\rho(X) + \lambda^2\} = 1 - \max_{\eta^2(X)=1} \rho^2(X). \tag{21}$$

Thus minimizing sstress is equivalent to maximizing ρ over all X in the compact set $\eta^2(X) = 1$. In that sense minimizing sstress is a maximum sum method. But, unlike the maximum sum approach of Guttman and de Leeuw, this explicit normalization of X does not lead to a simple eigenvalue-eigenvector problem. It can be solved, however, by majorization, which means, in thus case, iteratively solving a sequence of related eigenvalue-eigenvector problems.

If we use sstress to approximate stress there is a more or less rational way to choose the weights in sstress. This seems a good thing since we use sstress solutions as initial configurations for minimizing stress.

$$\begin{aligned}\sigma(X) &= \sum_{1<i\leq j<n}\sum w_{ij}(\delta_{ij} - d_{ij}(X))^2\\ &= \sum_{1<i\leq j<n}\sum \frac{w_{ij}}{(\delta_{ij} + d_{ij}(X))^2}(\delta_{ij}^2 - d_{ij}^2(X))^2\\ &\approx \frac{1}{4}\sum_{1<i\leq j<n}\sum \frac{w_{ij}}{\delta_{ij}^2}(\delta_{ij}^2 - d_{ij}^2(X))^2.\end{aligned} \tag{22}$$

There are some interesting connections between sstress, a least squares loss function defined on the squared distances, and strain, a least squares loss function defined on the scalar products. From Equation 6 we see that

$$\sigma(X) = \frac{1}{4}(\delta^2 - d^2(X))'(V^2 \otimes V^2)(\delta^2 - d^2(X)),$$

where $\delta^2 = \text{vec}(\Delta^2)$ and $d^2(X) = \text{vec}(D^2(X))$. It follows that

$$\sigma(X) \leq \frac{1}{4}\lambda_+^2 \sum\sum(\delta_{ij}^2 - d_{ij}^2(X))^2,$$

where λ_+ is the largest eigenvalue of V^2. If $V = J$ then $V^2 = J$. The

Sstress If $B = -\frac{1}{2}J\Delta^2 J$ the tr $A_{ij}B = \delta_{ij}^2$. Thus

$$\sigma(X) = \sum_{1\leq i<j\leq n}\sum w_{ij}(\delta_{ij}^2 - d_{ij}^2(X))^2 = \sum_{1\leq i<j\leq n}\sum w_{ij}\{\text{tr } A_{ij}(B - XX')\}^2$$

and thus, with $C = XX'$, we have

$$\sigma(C) = \sum_{1\leq i<j\leq n}\sum w_{ij} \text{ tr } A_{ij}(B - C)A_{ij}(B - C),$$

which expresses sstress as a function of the scalar products. If $b = \text{vec}(B)$ and $c = \vec{C}$ then

$$\sigma(c) = (b - c)'H(b - c)$$

with

$$H := \sum_{1\leq i<j\leq n}\sum w_{ij}(A_{ij} \otimes A_{ij})$$

If $H \lesssim V$, with V diagonal and non-negative, then

$$\sigma(C) \leq \sum_{1\leq i<j\leq n}\sum v_{ij}(b_{ij} - c_{ij})^2,$$

which is a weighted version of strain.

4.2 Algorithms

4.2.1 ELEGANT

The complicated history of this majorization algorithm, which early on was called ELEGANT, starts with De Leeuw (1975), Takane (1977), and Browne (1987). You may notice that in the

references De Leeuw (1975) is not linked to a pdf, because the paper seems to be irretrievably lost. It hangs on to its existence because it is discussed, referenced, and used, by Takane and Browne. See De Leeuw, Groenen, and Pietersz (2016) and Takane (2016) for more on the history. The original derivation in De Leeuw (1975) used augmentation (De Leeuw (1994)), which leads to unsharp majorization and to the slow convergence mentioned by De Leeuw, Browne, and Takane as a major disadvantage of ELEGANT. In De Leeuw, Groenen, and Pietersz (2016) a sharper majorization is used to improve the asymptotic convergence rate of the original algorithm by a large factor n. The modified ELEGANT is much faster than the original version (De Leeuw (2016)).

Now for the actual algorithm. Since it handles general non-negative weights, including zeroes, there is no fitting step as in our missing data generalization of strain.

Define A_{ij} as usual, $C := XX'$, $a_{ij} := \text{vec}(A_{ij})$, and $c = \text{vec}(C)$. Also use A for the $n^2 \times \frac{1}{2}n(n-1)$ matrix that has the a_{ij} with $i < j$ as columns, and W for the diagonal matrix with the w_{ij} for $i < j$. Then $d_{ij}^2(X) = \text{tr } A_{ij}C = a_{ij}'c$, and thus

$$\sigma(C) = 1 - 2b'c + c'Hc, \tag{23}$$

with $b = \text{vec}(B)$, with B from Equation 15, and with

$$H := AWA'. \tag{24}$$

There are two alternative expressions for the $n^2 \times n^2$ matrix H which are worth mentioning. The first one uses Kronecker products. It is

$$H = \sum_{1<i\leq j<n}\sum w_{ij}(A_{ij} \otimes A_{ij}). \tag{25}$$

The second expression represents H as an $n \times n$ block-matrix with elements (blocks) that are themselves $n \times n$ matrices. The (i, j) off-diagonal block of H is

$$\{H\}_{ij} = -w_{ij}A_{ij}, \tag{26}$$

while for the diagonal blocks

$$\{H_{ii}\} = \sum_{j\neq i} w_{ij}A_{ij}. \tag{27}$$

We now apply the standard majorization method for quadratic optimization. Let μ be an upper bound for the largest eigenvalue λ_+ of H. Write $C = \tilde{C} + (C - \tilde{C})$. Then

$$\begin{aligned}\sigma(c) \leq \sigma(\tilde{c}) - 2(b - H\tilde{c})'(c - \tilde{c}) + \mu(c - \tilde{c})'(c - \tilde{c}) = \\ \sigma(\tilde{c}) + \mu\|c - (\tilde{c} + \mu^{-1}H\tilde{c} - b))\|^2 - \mu^{-1}\|H\tilde{c} - b\|^2.\end{aligned} \tag{28}$$

{#eq-elegantmaj} Thus we find $c^{(\nu+1)}$ by minimizing

$$\omega(c) := \|c - (c^{(\nu)} + \mu^{-1}(Hc^{(\nu)} - b))\|^2 \tag{29}$$

over c. Now

$$Hc = \sum_{1\leq i<j\leq n}\sum w_{ij}a_{ij}a'_{ij}c = \sum_{1\leq i<j\leq n}\sum w_{ij}d^2_{ij}(c)a_{ij}. \tag{30}$$

Using Equation 30 we can translate back to matrices

$$\omega(C) = \text{tr}\ (C - (C^{(\nu)} + \mu^{-1}R(C^{(\nu)}))^2 \tag{31}$$

where

$$R(C^{(\nu)}) := \sum_{1\leq i<j\leq n}\sum w_{ij}(d^2_{ij}(C^{(\nu)}) - \delta^2_{ij}) \tag{32}$$

This is a low-rank symmetric matrix approximation problem, which we have encountered before, and which we can solve by partial eigen-decomposition.

Various choices of $\mu \geq \lambda_+$ are possible. The simplest one is the trace of H, which is equal to four times the sum of the weights.The original version of ELEGANT used this trace bound. A sharper bound, which involves only slightly more work, is the maximum row sum of the absolute values of the elements of H. This is equal to four times the maximum row sum of the weights W. The best choice for μ, of course, is $\mu = \lambda_+$, which requires more computation in the general case. For really large examples matrices of order $O(n^2)$ may be prohibitively large, and the maximum absolute value bound can be computed without actually computing and storing H.

If all weights are equal to one matters simplify. Table 1 gives the three bounds for various values of n.

Table 1: Eigenvalue Bounds

n	trc	abs	eig
2	4	4	4
4	24	12	8
8	112	28	16
16	480	60	32
32	1984	124	64
64	8064	252	128
128	32512	508	256

The trace bound is $2n(n-1)$, the maximum row sum bound is $4(n-1)$. De Leeuw, Groenen, and Pietersz (2016) show that if all weights are one the maximum eigenvalue is $2n$. In the

unfolding situation, with a rectangular $n \times m$ matrix of dissmilarities and all nm weights equal to one, the trace bound is $4nm$. The absolute value bound is $4\max(n, m)$ and the maximum eigenvalue is $n + m + 2$.

The largest eigenvalue of H, which has order n^2, is also the largest eigenvalue of the matrix with elements $W^{\frac{1}{2}}A'AW^{\frac{1}{2}}$, which is a non-negative matrix of order $\frac{1}{2}n(n-1)$. For general weights we use the algorithm of Markham (1968) to compute the largest eigenvalue (the Perron-Frobenius root) of this matrix. In balanced situations, where the weights are not too different, the absolute value bound is about twice the largest eigenvalue, and it is unclear if the additional iterative computation of the largest eigenvalue is warranted. Moreover since the iterations approximate the eigenvalue from below we need precise computations, otherwise we do not have an appropriate majorization.

4.2.2 ALSCAL

In ALSCAL we use cyclic coordinate descent to minimize sstress. The alternating least squares algorithm changes one coordinate at a time, keeping the other $np - 1$ coordinates fixed at their current values. Each cycle changes all coordinates in this way, thus generating a decreasing and convergent sequence of sstress values.

The original ALSCAL algorithm (Takane, Young, and De Leeuw (1977)) used alternating least squares, but with n blocks of size p, using a safeguarded version of Newton's method in each block subproblem. Even at the time (almost 50 years ago now) I pushed for using np blocks of size one, but I lst out initially. But pretty soon afterwards Doug Carroll and others pointed out that the original algorithm has the problem that the Newton method may converge to a block local minimum (Takane, Young, and De Leeuw (1977), page 61-63). So in subsequent versions of ALSCAL in SAS and SPSS (F. W. Young, Takane, and Lewyckyj (1978)) coordinate descent was used.

Formally changing a single coordinate is

$$X = Y + \theta e_k e_s', \tag{33}$$

which makes $d_{ij}^2(X)$ the following quadratic function of θ.

$$d_{ij}^2(\theta) = d_{ij}^2(Y) + 2\theta(y_{is} - y_{js})(\delta^{ik} - \delta^{jk}) + \theta^2(\delta^{ik} + \delta^{jk}), \tag{34}$$

Note there are superscripted Kronecker deltas such as δ^{ik} and subscripted dissimilarity deltas such as δ_{ij}.

Define the residual

$$r_{ij}(Y) := d_{ij}^2(Y) - \delta_{ij}^2. \tag{35}$$

Then

$$\sigma(\theta) = \sum_{i=1}^n \sum_{j=1}^n w_{ij}(r_{ij}(Y) + 2\theta(y_{is} - y_{js})(\delta^{ik} - \delta^{jk}) + \theta^2(\delta^{ik} - \delta^{jk})^2)^2 \tag{36}$$

Now expand Equation 36. This was also done undoubtedly by F. W. Young, Takane, and Lewyckyj (1978), and explicitly for R version in De Leeuw (2006), with a mistake in his formula for $a_3(X)$ on page 3. We use the fact that if $i \neq j$ then

$$(\delta^{ik} - \delta^{jk})^2 = \delta^{ik} + \delta^{jk}, \tag{37a}$$

$$(\delta^{ik} - \delta^{jk})(\delta^{ik} + \delta^{jk}) = \delta^{ik} - \delta^{jk}, \tag{37b}$$

$$(\delta^{ik} + \delta^{jk})^2 = \delta^{ik} + \delta^{jk}. \tag{37c}$$

Expanding gives

$$\sigma(\theta) = \sigma(Y) + \sigma_1(Y)\theta + \sigma_2(Y)\theta^2 + \sigma_3(Y)\theta^3 + \sigma_4(Y)\theta^4, \tag{38}$$

with

$$\sigma_1(Y) = 8\sum_{j=1}^n w_{kj}r_{kj}(Y)(y_{ks} - y_{js}), \tag{39a}$$

$$\sigma_2(Y) = 4\sum_{j=1}^n w_{kj}r_{kj}(Y) + 8\sum_{j=1}^n w_{kj}(y_{ks} - y_{js})^2, \tag{39b}$$

$$\sigma_3(Y) = 8\sum_{j=1}^n w_{kj}(y_{ks} - y_{js}), \tag{39c}$$

$$\sigma_4(Y) = 2\sum_{j=1}^n w_{kj}. \tag{39d}$$

This non=negative bowl-shaped quartic is minimized to find the optimum coordinate replacement of x_{ks} in Equation 33. Then go to the next coordinate, and so on. Convergence is tested after each cycle.

5 Example

We use the data from Ekman (1954), which has a very good MDS fit in two dimensions, with apparently very few local minima.

Comparisons of the various ELEGANT options are done with the microbenchmark package (Mersmann (2023)), using 100 runs and the default iteration options. First we look at the unweighted case. varies the three bounds on the largest eigenvalue and the initial configuration (zero is random, one is maximum sum).

```
Warning in microbenchmark(smacofElegant(ekmat, bnd = 0, xold = 0, itmax =
1e+05, : less accurate nanosecond times to avoid potential integer overflows
```

```
Unit: milliseconds
                                                                        expr
 smacofElegant(ekmat, bnd = 0, xold = 0, itmax = 1e+05, verbose = FALSE)
 smacofElegant(ekmat, bnd = 0, xold = 1, itmax = 1e+05, verbose = FALSE)
 smacofElegant(ekmat, bnd = 1, xold = 0, itmax = 1e+05, verbose = FALSE)
 smacofElegant(ekmat, bnd = 1, xold = 1, itmax = 1e+05, verbose = FALSE)
 smacofElegant(ekmat, bnd = 2, xold = 0, itmax = 1e+05, verbose = FALSE)
 smacofElegant(ekmat, bnd = 2, xold = 1, itmax = 1e+05, verbose = FALSE)
       min         lq       mean     median         uq       max neval
116.634791 133.557972 145.824174 141.250043 152.658662 250.22972   100
 78.363095  79.523046  81.617043  80.954479  82.400857 129.80288   100
 17.690393  20.324827  23.208637  22.091005  24.222493  75.14710   100
 12.236040  12.339586  13.598644  12.627447  12.898293  62.65095   100
  9.800107  10.653624  12.117433  11.414154  12.359921  30.47005   100
  6.761966   6.886811   7.238513   7.007064   7.159789  10.11605   100
```

As expected, bound 2 (the eigenvalue) is about twice as fast as bound 1 (maximum absolute row sum). The original ELEGANT, with bound 0, is indeed very slow. The maximum sum initial configuration gives another speedup of around two.

Next, we do the same comparison with weights $w_{ij} = \delta_{ij}^{-1}$.

```
Unit: milliseconds
```

```
 smacofElegant(ekmat, wght = wmat, bnd = 0, xold = 0, itmax = 1e+05,      verbose = FA
 smacofElegant(ekmat, wght = wmat, bnd = 0, xold = 1, itmax = 1e+05,      verbose = FA
 smacofElegant(ekmat, wght = wmat, bnd = 1, xold = 0, itmax = 1e+05,      verbose = FA
```

```
smacofElegant(ekmat, wght = wmat, bnd = 1, xold = 1, itmax = 1e+05,      verbose = FA
smacofElegant(ekmat, wght = wmat, bnd = 2, xold = 0, itmax = 1e+05,      verbose = FA
smacofElegant(ekmat, wght = wmat, bnd = 2, xold = 1, itmax = 1e+05,      verbose = FA
      min        lq      mean    median        uq       max neval
159.19447 175.64949 190.82172 185.23331 201.98250 255.89223   100
125.98418 127.64848 131.47819 129.10111 131.77470 187.07398   100
 28.74465  34.13943  36.61538  36.18350  38.38000  51.28842   100
 23.67024  24.17038  25.41483  24.78495  26.55957  29.12730   100
 32.26495  34.87546  36.64014  36.06501  37.31621  85.66450   100
 28.63924  29.97383  31.30094  31.42914  32.38555  36.70074   100
```

Using these reciprocal weights slows down everything, but the ranking of the various options remains the same, except for the fact that the maximum sum initial configuration does not give much of an improvement. Although bound equal to two now involves one-time computation of the largest eigenvalue of H it still is twice as fast as bound equal to one. But remember that for larger examples

```
Unit: milliseconds
                                                                    expr
smacofElegant(grmat, bnd = 0, xold = 0, itmax = 1e+05, verbose = FALSE)
smacofElegant(grmat, bnd = 0, xold = 1, itmax = 1e+05, verbose = FALSE)
smacofElegant(grmat, bnd = 1, xold = 0, itmax = 1e+05, verbose = FALSE)
smacofElegant(grmat, bnd = 1, xold = 1, itmax = 1e+05, verbose = FALSE)
smacofElegant(grmat, bnd = 2, xold = 0, itmax = 1e+05, verbose = FALSE)
smacofElegant(grmat, bnd = 2, xold = 1, itmax = 1e+05, verbose = FALSE)
      min        lq      mean    median        uq       max neval
 90.74719 117.26238 268.80010 279.00670 356.40039 545.97293   100
497.51405 510.18377 523.79944 515.33179 528.03843 631.07840   100
 20.51710  25.70503  61.10595  62.81598  80.76572 132.82893   100
110.13834 111.65690 114.37898 113.69878 115.54733 163.04228   100
 11.03236  15.26182  34.58285  34.66230  48.56604  70.76744   100
 59.48018  60.03392  61.89118  60.86960  62.27252 107.62242   100
```

We also used the Ekman data for a comparison of unweighted ELEGANT, with the eigenvalue bound, and unweighted ALSCAL. If we used a random initial x then microbenchmark gives a median time of 19.56022 milliseconds for ALSCAL versus 11.65208 for ELEGANT. Using the maximum sum initial x changes this to 15.611836 and 7.121864.

```
Unit: milliseconds
```

```
                                                                 expr
smacofElegant(ekmat, bnd = 2, xold = 0, itmax = 1e+05, verbose = FALSE)
smacofElegant(ekmat, bnd = 2, xold = 1, itmax = 1e+05, verbose = FALSE)
             smacofALSCAL(ekmat, x = 0, itmax = 1e+05, verbose = FALSE)
             smacofALSCAL(ekmat, x = 1, itmax = 1e+05, verbose = FALSE)
      min        lq      mean    median        uq      max neval
 9.323646 11.179244 13.031234 12.057875 13.504518 65.91205   100
 6.807312  7.054194  7.508257  7.237874  7.505911 10.21642   100
15.934199 18.044264 20.160187 19.304153 20.868672 81.47807   100
14.696942 15.361306 16.369027 15.927270 17.475409 20.84952   100
```

References

Bailey, R. A., and J. C. Gower. 1990. "Approximating a Symmetric Matrix." *Psychometrika* 55 (4): 665–75.

Browne, M. W. 1987. "The Young-Householder Algorithm and the Least Squares Multidimensional Scaling of Squared Distances." *Journal of Classification* 4: 175–90.

Cailliez, F. 1983. "The Analytical Solution to the Additive Constant Problem." *Psychometrika* 48 (2): 305–8.

De Leeuw, J. 1968a. "Canonical Discriminant Analysis of Relational Data." Research Note 007-68. Department of Data Theory FSW/RUL. https://jansweb.netlify.app/publication/deleeuw-r-68-e/deleeuw-r-68-e.pdf.

———. 1968b. "Nonmetric Multidimensional Scaling." Research Note 010-68. Department of Data Theory FSW/RUL. https://jansweb.netlify.app/publication/deleeuw-r-68-g/deleeuw-r-68-g.pdf.

———. 1970. "The Euclidean Distance Model." Research Note 002-70. Department of Data Theory FSW/RUL. https://jansweb.netlify.app/publication/deleeuw-r-70-b/deleeuw-r-70-b.pdf.

———. 1975. "An Alternating Least Squares Approach to Squared Distance Scaling." Department of Data Theory FSW/RUL.

———. 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.

———. 2006. "ALSCAL in R." UCLA Department of Statistics. https://jansweb.netlify.app/publication/deleeuw-u-06-i/deleeuw-u-06-i.pdf.

———. 2016. "Convergence Rate of ELEGANT Algorithms." 2016. https://jansweb.netlify.app/publication/deleeuw-e-16-o/deleeuw-e-16-o.pdf.

De Leeuw, J., P. Groenen, and R. Pietersz. 2016. "An Alternating Least Squares Approach to Squared Distance Scaling." https://jansweb.netlify.app/publication/deleeuw-groenen-pietersz-e-16-m/deleeuw-groenen-pietersz-e-16-m.pdf.

De Leeuw, J., and W. J. Heiser. 1982. "Theory of Multidimensional Scaling." In *Handbook of Statistics, Volume II*, edited by P. R. Krishnaiah and L. Kanal. Amsterdam, The Netherlands: North Holland Publishing Company.

De Leeuw, J., and J. J. Meulman. 1986. "Principal Component Analysis and Restricted Multidimensional Scaling." In *Classification as a Tool of Research*, edited by W. Gaul and M. Schader, 83–96. Amsterdam, London, New York, Tokyo: North-Holland.

Ekman, G. 1954. "Dimensions of Color Vision." *Journal of Psychology* 38: 467–74.

Gower, J. C. 1966. "Some Distance Properties of Latent Root and Vector Methods Used in Multivariate Analysis." *Biometrika* 53: 325–38.

Guttman, L. 1968. "A General Nonmetric Technique for Fitting the Smallest Coordinate Space for a Configuration of Points." *Psychometrika* 33: 469–506.

Mardia, K. V. 1978. “Some Properties of Classical Multidimensional Scaling.” *Communications in Statistics - Theory and Methods* 7 (13): 1233–41.

Markham, T. L. 1968. “An Iterative Procedure for Computing the Maximal Root of a Positive Matrix.” *Mathematics of Computation* 22: 869–71.

Mersmann, O. 2023. *microbenchmark: Accurate Timing Functions*. https://CRAN.R-project.org/package=microbenchmark.

Mullen, K. M., and I. H. M. van Stokkum. 2023. *nnls: The Lawson-Hanson algorithm for non-negative least squares (NNLS)*. %7Bhttps://CRAN.R-project.org/package=nnls%7D.

Nguyen, L. T., J. Kim, and B. Shim. 2019. “Low-Rank Matrix Completion: A Contemporary Survey.” https://arxiv.org/abs/1907.11705.

Qiu, Y., and J. Mei. 2024. *RSpectra: Solvers for Large-Scale Eigenvalue and SVD Problems*. https://CRAN.R-project.org/package=RSpectra.

Schoenberg, I. J. 1935. “Remarks to Maurice Frechet’s article: Sur la Definition Axiomatique d’une Classe d’Espaces Vectoriels Distancies Applicables Vectoriellement sur l’Espace de Hllbert.” *Annals of Mathematics* 36: 724–32.

Takane, Y. 1977. “On the Relations among Four Methods of Multidimensional Scaling.” *Behaviormetrika* 4: 29–42.

———. 2016. “My Early Interactions with Jan and Some of His Lost Papers.” *Journal of Statistical Software* 73 (7): 1–14. https://www.jstatsoft.org/article/view/v073i07.

Takane, Y., F. W. Young, and J. De Leeuw. 1977. “Nonmetric Individual Differences in Multidimensional Scaling: An Alternating Least Squares Method with Optimal Scaling Features.” *Psychometrika* 42: 7–67.

Torgerson, W. S. 1958. *Theory and Methods of Scaling*. New York: Wiley.

Trosset, M. W. 1998. “A New Formulation of the Nonmetric Strain Problem in Multidimensional Scaling.” *Journal of Classification* 15: 15–35.

Young, F. W., Y. Takane, and R. Lewyckyj. 1978. “Three Notes on ALSCAL.” *Psychometrika* 43 (3): 433–35.

Young, G., and A. S. Householder. 1938. “Discussion of a Set of Points in Terms of Their Mutual Distances.” *Psychometrika* 3 (19-22).