Smacof at 50: A Manual
Part 1: The Basics

Jan de Leeuw

December 8, 2024

Table of contents

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. The files can be found at https://github.com/deleeuw in the repositories smacofCode, smacofManual, and smacofExamples.

1 Loss Function

In the pioneering papers Kruskal (1964a) and Kruskal (1964b) the MDS problem was formulated for the first time as minimization of an explicit *loss function* or *badness-of-fit function*, which measures the quality of the approximation of the dissimilarities by the distances. To be historically accurate, we should mention that the non-metric MDS technique proposed by Shepard (1962a) and Shepard (1962b) can be reformulated as minimization of an explicit loss function (see, for example, De Leeuw (2017)). And the classical Young-Householder-Torgerson MDS technique (Torgerson (1952)) for metric MDS can be reformulated as minimizing an explicit least squares loss function (De Leeuw and Heiser (1982)) as well. But neither of these two predecessors was formulated originally as an explicit minimization problem for a specific loss function

1.1 Metric MDS

The loss function in least squares metric Euclidean MDS is called *raw stress* and is defined as

$$\sigma_R(X) := \frac{1}{2}\sum\sum_{1\leq j<i\leq n} w_{ij}(\delta_{ij} - d_{ij}(X))^2.(\#eq:stressdef) \tag{1}$$

The subscript R in σ_R stands for "raw", because we will discuss other least squares loss functions for which we will also use the symbol σ, but with other subscripts.

In definition @ref(eq:stressdef) the w_{ij} are known non-negative *weights*, the δ_{ij} are the known non-negative *dissimilarities* between objects o_i and o_j, and the $d_{ij}(X)$ are the *distances* between the corresponding points x_i and x_j. The summation is over all pairs (i,j) with $w_{ij} > 0$. From now on we use "metric MDS" to mean the minimization of σ_R.

The $n \times p$ matrix X, which has the coordinates x_i of the n points as its rows, is called the *configuration*, where p is the *dimension* of the Euclidean space in which we make the map. The metric MDS problem (of dimension p, for given W and Δ) is the minimization of @ref(eq:stressdef) over the $n \times p$ configurations X.

The weights w_{ij} can be used to quantify information about the precision or importance of the corresponding dissimilarities. Some of the weights may be zero, which can be used to code *missing data*. If all weights are positive we have *complete data*. If we have complete data, and all weights are equal to one, we have *unweighted* metric MDS. The pioneering papers by Shepard, Kruskal, and Guttman only consider the unweighted case. Weights were only introduced in MDS in De Leeuw (1977).

We assume throughout that the weights are *irreducible* (De Leeuw (1977)). This means there is no partitioning of the index set $I_n := \{1, 2, \cdots, n\}$ into subsets for which all between-subset weights are zero. A reducible metric MDS problems decomposes into a number of smaller independent metric MDS problems, so the irreducibility assumption causes no real loss of generality.

The fact that the summation in @ref(eq:stressdef) is over all $j < i$ indicates that the diagonal elements of Δ are not used (they are assumed to be zero) and the elements above the diagonal are not used either (they are assumed to be equal to the corresponding elements below the diagonal). The somewhat mysterious factor $\frac{1}{2}$ in definition @ref(eq:stressdef) is there because it simplifies some of the formulas in later sections of this paper.

1.2 Non-linear MDS

Kruskal was not really interested in metric MDS and the "raw" loss function @ref(eq:stressdef). His papers are really about non-metric MDS, by which we mean least squares non-metric Euclidean MDS. Non-metric MDS differs from metric MDS because we have incomplete information about the dissimilarities. As we have seen, that if some dissimilarities are missing metric MDS can handle this by using zero weights. In some situations, however, we only know the rank order of the non-missing dissimilarities. We do not know, or we refuse to use, their actual numeric values. Or, to put it differently, even if we have numerical dissimilarities we are looking for a *transformation* of the non-missing dissimilarities, where the transformation is chosen from a set of admissible transformations (for instance from all linear or monotone transformations). If the dissimilarities are non-numerical, for example rank orders or partitionings, we choose from the set of admissible *quantifications*.

In non-metric MDS raw stress becomes

$$\sigma_R(X,\Delta) := \frac{1}{2}\sum w_{ij}(\delta_{ij} - d_{ij}(X))^2, (\#eq:rawstressdef) \tag{2}$$

where Δ varies over the quantified or transformed dissimilarities. In MDS parlance they are also called *pseudo-distances* or *disparities*. Loss function @ref(eq:rawstressdef) must be minimized over both configurations and disparities, with the condition that the disparities Δ are an admissible transformation or quantification of the data. In Kruskal's non-metric MDS this means requiring monotonicity. In this paper we will consider various other choices for the set of admissible transformations. We will use the symbol $\mathfrak{D}$ for the set of admissible transformations

The most familiar examples of $\mathfrak{D}$ (linear, polynomial, splines, monotone) define convex cones with apex at the origin. This means that if $\Delta \in \mathfrak{D}$ then so is $\lambda\Delta$ for all $\lambda \geq 0$. But consequently minimizing @ref(eq:rawstressdef) over all $\Delta \in \mathfrak{D}$ and over all configurations has the trivial solution $\Delta = 0$ and $X = 0$, corresponding with the global minimum $\sigma(X,\Delta) = 0$. We need additional constraints to rule out this trivial solution, and in non-metric MDS this is done by choosing a *normalization* that keeps the solution away from zero.

Kruskal's original solution is to define *normalized stress* as

$$\sigma(X,\Delta):=\frac{\sum w_{ij}(\delta_{ij}-d_{ij}(X))^2}{\sum w_{ij}d_{ij}^2(X)}.(\#eq:nstressdef) \tag{3}$$

To be precise, in Kruskal's formulation there are no weights, and he actually takes the square root of @ref(eq:nstressdef) to define *Kruskal's stress*. The non-metric Euclidean MDS problem now is to minimize loss function @ref(eq:nstressdef) over all $n\times p$ configurations X and all admissible disparities Δ.

1.3 Non-metric MDS

1.4 Normalization

Equation @ref(eq:nstressdef) is only one way to normalize raw stress. Some obvious alternatives are discussed in detail in Kruskal and Carroll (1969) and De Leeuw (1975). In the terminology of De Leeuw (1975) there are *explicit* and *implicit* normalizations.

In implicit normalization we minimize either

$$\sigma(X,\hat{D}):=\frac{\sum w_{ij}(\hat{d}_{ij}-d_{ij}(X))^2}{\sum w_{ij}\hat{d}_{ij}^2}(\#eq:implicit1) \tag{4}$$

or

$$\sigma(X,\hat{D}):=\frac{\sum w_{ij}(\hat{d}_{ij}-d_{ij}(X))^2}{\sum w_{ij}d_{ij}^2(X)}(\#eq:implicit2) \tag{5}$$

over X and $\Delta\in\mathfrak{D}$.

As we have seen, Kruskal (1964a) chooses definition @ref(eq:implicit2) and calls the explicitly normalized loss function *normalized stress*. Note that we overload the symbol σ to denote any one of the least squares loss functions. It will always be clear from the text which σ we are talking about.

In explicit normalization we minimize the raw stress $\sigma_R(X,\hat{D})$ from @ref(eq:rawstressdef), but we add the explicit constraint

$$\sum w_{ij}d_{ij}^2(X)=1,(\#eq:explicit1) \tag{6}$$

or the constraint

$$\sum w_{ij}\hat{d}_{ij}^2 = 1.(\#eq : explicit2) \tag{7}$$

Kruskal and Carroll (1969) and De Leeuw (2019) show that these four normalizations all lead to essentially the same solution for X and $\widehat{D}$, up to scale factors dictated by the choice of the particular normalization. It is also possible to normalize both X and $\widehat{D}$, either explicitly or implicitly, and again this will give the same solutions, suitably normalized. These invariance results assume the admissible transformations form a closed cone with apex at the origin, i.e. if $\widehat{D}$ is admissible and $\lambda \geq 0$ then $\lambda\widehat{D}$ is admissible as well. The matrices of Euclidean distances $D(X)$ form a similar closed cone as well. The non-metric MDS problem is to find an element of the $\widehat{D}$ cone $\mathcal{D}$ and an element of the $D(X)$ cone where the angle between the two is a small as possible.

In the R version of smacof (De Leeuw and Mair (2009), Mair, Groenen, and De Leeuw (2022)) we use explicit normalization @ref(eq:explicit2). This is supported by the result, also due to De Leeuw (1975), that projection on the intersection of the cone of disparities and the sphere defined by @ref(eq:explicit2) is equivalent to first projecting on the cone and then normalizing the projection (see also Bauschke, Bui, and Wang (2018)).

In the version of non-metric MDS discussed in this manual we need more flexibility. For algorithmic reasons that may become clear later on, we will go with the original @ref(eq:nstressdef), i.e. with the implicitly normalized Kruskal's stress. For the final results the choice between normalizations should not make a difference, but the iterative computations will be different for the different choices.

1.5 Some thoughts on ALS

The formulation in equations @ref(eq:gmdsdef1) and @ref(eq:gmdsdef2) neatly separates the metric MDS part @ref(eq:gmdsdef1) and the transformation/quantification part @ref(eq:gmdsdef2). This second part is also often called the *optimal scaling* part.

Equations @ref(eq:gmdsdef1) and @ref(eq:gmdsdef2) corresponds with the way most iterative non-linear and non-metric MDS techniques are implemented. The algorithms use *Alternating Least Squares* (ALS). There have been quite a few ALS algorithms avant-la-lettre, but as far as I know both the name and ALS as a general approach to algorithm construction were first introduced in De Leeuw (1968), and then widely disseminated in a series of papers by De Leeuw, Young, and Takane in the 1970's (work summarized in Young, De Leeuw, and Takane (1980) and Young (1981)).

In the ALS implementation of MDS two sub-algorithms are used in each iteration: one to improve the fit of the distances to the current disparities Δ and one to improve the fit of the disparities to the current distances. The two sub-algorithms define one major iteration of the

MDS technique. In formulas (using superscript (k) for major iteration number) we start with $(X^{(0)},\Delta^{(0)})$ and then alternate the mimization problems

$$X^{(k+1)}\ni\{\sigma(X^{(k+1)},\Delta^{(k)})=\min_{X\in\mathfrak{X}}\sigma(X,\Delta^{(k)})\},\tag{8a}$$

$$\Delta^{(k+1)}\ni\{\sigma(X^{(k+1)},\Delta^{(k+1)})=\min_{\Delta\in\mathfrak{D}}\sigma(X^{(k+1)},\Delta)\},\tag{8b}$$

where $\ni$ is short for "such that". In MDS it is more realistic not to minimize loss in the sub-steps but merely to decrease it. Minimization in one or both of the two subproblems may itself require an infinite iterative method, which we have to truncate anyway. Thus

$$X^{(k+1)}\in\mathfrak{X}\ni\{\sigma(X^{(k+1)},\Delta^{(k)})<\sigma(X^{(k)},\Delta^{(k)})\},\tag{9a}$$

$$\Delta^{(k+1)}\in\mathfrak{D}\ni\{\sigma(X^{(k+1)},\Delta^{(k+1)})<\sigma(X^{(k+1)},\Delta^{(k)})\}.\tag{9b}$$

1.5.1 The Single-Phase approach

In Kruskal (1964a) defines

$$\sigma(X):=\min_{\hat{D}\in\mathfrak{D}}\ \sigma(\hat{D},X)=\sigma(X,\hat{D}(X)),(\#eq:project)\tag{10}$$

where $\sigma(\hat{D},X)$ is defined by @ref(eq:implicit2). The minimum in @ref(eq:project) is over admissible transformations. In definition @ref(eq:project)

$$\hat{D}(X):=\mathop{\text{argmin}}_{\hat{D}\in\mathfrak{D}}\sigma(X,\hat{D}).(\#eq:optscal)\tag{11}$$

Normalized stress defined by @ref(eq:project) is now a function of X only. Under some conditions, which are true in Kruskal's definition of non-metric MDS, there is a simple relation between the partials of @ref(eq:implicit2) and those of @ref(eq:project).

$$\mathcal{D}\sigma(X)=\mathcal{D}_1\sigma(X,\hat{D}(X)),(\#eq:partials)\tag{12}$$

where $\mathcal{D}\sigma(X)$ are the derivatives of σ from @ref(eq:project) and $\mathcal{D}_1\sigma(X,\hat{D}(X))$ are the partial derivatives of σ from @ref(eq:implicit2) with respect to X. Thus the partials of σ from @ref(eq:project) can be computed by evaluating the partials of σ from @ref(eq:implicit2) with respect to X at $(X,\hat{D}(X))$. This has created much confusion in the past. The non-metric MDS problem in Kruskal's original formulation is now to minimize σ from @ref(eq:project), which is a function of X alone.

Guttman (1968) calls this the *single-phase approach*. A variation of Kruskal's single-phase approach defines

$$\sigma(X)=\sum w_{ij}(d_{ij}^{\#}(X)-d_{ij}(X))^2,(\#eq:rankimage)\tag{13}$$

where the $d_{ij}^{\#}(X)$ are *Guttman's rank images*, i.e. the permutation of the $d_{ij}(X)$ that makes them monotone with the δ_{ij} (Guttman (1968)). Or, alternatively, define

$$\sigma(X) := \sum w_{ij}(d_{ij}^{\%}(X) - d_{ij}(X))^2, (\#eq:shepard) \tag{14}$$

where the $\widehat{d_{ij}^{\%}}(X)$ are *Shepard's rank images*, i.e. the permutation of the δ_{ij} that makes them monotone with the $d_{ij}(X)$ (Shepard (1962a), Shepard (1962b), De Leeuw (2017)).

Minimizing the Shepard or Guttman single-phase loss functions is computationally more complicated than Kruskal's *monotone regression* approach, mostly because the rank-image transformations are not differentiable, and there is no analog of @ref(eq:partials) and of the equivalence of the different implicit and explicit normalizations.

1.5.2 The Two-Phase Approach

The *two-phase approach* or *alternating least squares (ALS)* approach alternates minimization of $\sigma(\widehat{D}, X)$ over X for our current best estimate of $\widehat{D}$ with minimization of $\sigma(\widehat{D}, X)$ over $\Delta \in \mathfrak{D}$ for our current best value of X. Thus an update from iteration k to iteration $k+1$ looks like

$$\widehat{D}^{(k)} = \underset{\widehat{D}\in\mathfrak{D}}{\text{argmin}}\, \sigma(\widehat{D}, X^{(k)}), (\#eq:step1) \tag{15a}$$

$$X^{(k+1)} = \underset{X}{\text{argmin}}\, \sigma(\widehat{D}^{(k)}, X).(\#eq:step2) \tag{15b}$$

This ALS approach to MDS was in the air since the early (unsuccessful) attempts around 1968 of Young and De Leeuw to combine Torgerson's classic metric MDS method with Kruskal's monotone regression transformation. All previous implementations of non-metric smacof use the two-phase approach, and we will do the same in this paper.

As formulated, however, there are some problems with the ALS algorithm. Step @ref(eq:step1) is easy to carry out, using monotone regression. Step @ref(eq:step2) means solving a metric scaling problem, which is an iterative proces that requires an infinite number of iterations. Thus, in the usual implementations, step @ref(eq:step1) is combined with one of more iterations of a convergent iterative procedure for metric MDS, such as smacof. If we take only one of these *inner iterations* the algorithm becomes indistinguishable from Kruskal's single-phase method. This has also created much confusion in the past.

In the usual implementations of the ALS approach we solve the first subproblem @ref(eq:step1) exactly, while we take only a single step towards the solution for given $\widehat{D}$ in the second phase @ref(eq:step2). If we have an infinite iterative procedure to compute the optimal $\widehat{D} \in \mathfrak{D}$ for given X, then a more balanced approach would be to take several inner iterations in the first phase and several inner iterations in the second phase. How many of each, nobody knows. In our current implementation of smacof we take several inner iteration steps in the first phase and a single inner iteration step in the second phase.

2 Smacof Notation and Terminology

We discuss some the MDS notation used in smacof, which was first introduced in De Leeuw (1977) and De Leeuw and Heiser (1977). More detailed De Leeuw and Heiser (1980), De Leeuw (1988), Borg and Groenen (2005), Groenen and Van de Velden (2016)

This notation is useful for the second phase of the ALS algorithm, in which solve the metric MDS problem of we minimizing unnormalized $\sigma(X, \widehat{D})$ over X for fixed $\widehat{D}$. We will discuss the first ALS phase later in the paper.

Start with the unit vectors e_i of length n. They have a non-zero element equal to one in position i, all other elements are zero. Think of the e_i as the columns of the identity matrix.

Using the e_i we define for all $i \neq j$ the matrices

$$A_{ij} := (e_i - e_j)(e_i - e_j)'. \tag{16}$$

The A_{ij} are of order n, symmetric, doubly-centered, and of rank one. They have four non-zero elements. Elements (i, i) and (j, j) are equal to $+1$, elements (i, j) and (j, i) are -1.

The importance of A_{ij} in MDS comes from the equation

$$d_{ij}^2(X) = \text{tr } X' A_{ij} X.(\#eq : dfroma) \tag{17}$$

In addition we use the fact that the A_{ij} form a basis for the $binomn2$-dimensional linear space of all doubly-centered symmetric matrices.

Expanding the square in the definition of stress gives

$$\sigma(X) = \frac{1}{2}\{\sum w_k \delta_k^2 - 2 \sum w_k \delta_k d_k(X) + \sum w_k d_k^2(X)\}.(\#eq : expand) \tag{18}$$

It is convenient to have notation for the three separate components of stress from equation @ref(eq:expand). Define

$$\eta_{\widehat{D}}^2 = \sum w_{ij} \widehat{d}_{ij}^2, (\#eq : condef) \tag{19}$$

$$\rho(X) = \sum w_{ij} \widehat{d}_{ij} d_{ij}(X), (\#eq : rhodef) \tag{20}$$

$$\eta^2(X) = \sum w_{ij} d_{ij}(X)^2.(\#eq : etadef) \tag{21}$$

which lead to

$$\sigma(X) = \frac{1}{2}\left\{\eta_{\widehat{D}}^2 - 2\rho(X) + \eta^2(X)\right\}.(\#eq : stressshort) \tag{22}$$

We also need

$$\lambda(X)=\frac{\rho(X)}{\eta(X)}.(\#eq:lambdadef)\tag{23}$$

Using the A_{ij} makes it possible to give matrix expressions for ρ and η^2. First

$$\eta^2(X)=\text{tr }X'VX,(\#eq:etamat)\tag{24}$$

with

$$V:=\sum w_{ij}A_{ij}.(\#eq:vdef)\tag{25}$$

In the same way

$$\rho(X)=\text{tr }X'B(X)X,(\#eq:rhomat)\tag{26}$$

with

$$B(X):=\sum w_{ij}r_{ij}(X)A_{ij},(\#eq:bdef)\tag{27}$$

with

$$r_{ij}(X):=\begin{cases}\frac{\delta_{ij}}{d_{ij}(X)} & \text{if } d_{ij}(X)>0,\\ 0 & \text{if } d_{ij}(X)=0.\end{cases}\tag{28}$$

Note that B is a function from the set of $n\times p$ configurations into the set of symmetric doubly-dentered matrices of order n. All matrices of the form $\sum x_{ij}A_{ij}$, where summation is over all pairs (i,j) with $j<i$, are symmetric and doubly-centered. They have $-x_{ij}$ as off-diagonal elements while the diagonal elements (i,i) are $\sum_{j=1}^n x_{ij}$.

Because $B(X)$ and V are non-negative linear combinations of the A_{ij} they are both positive semi-definite. Because W is assumed to be irreducible the matrix V has rank $n-1$, with only vectors proportional to the vector e with all elements equal to one in its null-space (De Leeuw (1977)).

Summarizing the results so far we have

$$\sigma(X)=\frac{1}{2}\{\eta^2_{\hat{D}}-\text{tr }X'B(X)X+\text{tr }X'VX\}.(\#eq:sigmat)\tag{29}$$

Next we define the *Guttman transform* of a configuration X, for given W and Δ, as

$$G(X)=V^+B(X)X,(\#eq:gudef)\tag{30}$$

with V^+ the Moore-Penrose inverse of V. In our computations we use

$$V^+=(V+\frac{1}{n}ee')^{-1}-\frac{1}{n}ee'\tag{31}$$

Also note that in the unweighted case with complete data $V=nJ$, where J is the centering matrix $I-\frac{1}{n}ee'$, and thus $V^+=\frac{1}{n}J$. The Guttman transform is then simply $G(X)=n^{-1}B(X)X$.

3 Intermezzo: Explicit Normalization

$$\sigma(X,\hat{D})=\frac{1}{2}\frac{\sum w_{ij}(\hat{d}_{ij}-d_{ij}(X))^2}{\sum w_{ij}d_{ij}^2(X)}$$

Majorize

$$\sigma(X,\hat{D})\leq\frac{1}{2}\frac{\eta^2(\hat{D})-2\text{tr }X'V\overline{Y}+\text{ tr }X'VX}{\text{tr }X'VX}=\frac{\omega(X,Y)}{\eta^2(X)}$$

Stationary equations

$$\eta^2(X)(VX-VG(Y))-\omega(X,Y)VX=V\{(\eta^2(X)-\omega(X,Y))X-\eta^2(X)\overline{Y}\}$$

So at a minimum X is proportional to $\overline{Y}$ or $X=\alpha\overline{Y}$ for some α. For ... to be zero we must have

$$\alpha(\alpha^2\eta^2(\overline{Y})-(\eta^2(\hat{D})-2\alpha\eta^2(\overline{Y})+\alpha^2\eta^2(\overline{Y}))=\alpha^2\eta^2(\overline{Y})$$

which works out to be

$$\alpha=\frac{\eta^2(\hat{D})}{\eta^2(\overline{Y})}$$

$$\hat{X}=\frac{\eta^2(\hat{D})}{\eta^2(\overline{Y})}\,\overline{Y}$$

The minimum is equal to

$$\frac{-\frac{(\eta^2(\overline{Y}))^2}{\eta^2(\hat{D})}+\eta^2(\overline{Y})}{\eta^2(\overline{Y})}=1-\frac{\eta^2(\overline{Y})}{\eta^2(\hat{D})}$$

Use homogeneity of the Guttman transform.

More generally suppose we update with

$$X=\overline{Y}+\alpha(Y-\overline{Y})$$

Write

$$\omega(X,Y)=\eta^2(\hat{D})+\text{tr }(X-\overline{Y})'V(X-\overline{Y})-\eta^2(\overline{Y})$$

Thus if $X(\alpha)=\overline{Y}+\alpha(Y-\overline{Y})$ we have

$$\omega(\alpha)=\eta^2(\hat{D})+\alpha^2\text{tr }(Y-\overline{Y})'V(Y-\overline{Y})-\eta^2(\overline{Y})$$

and

$$\eta^2(\alpha) = \eta^2(\overline{Y}) + 2\alpha\text{tr }(Y-\overline{Y})'V\overline{Y}+\alpha^2\text{tr }(Y-\overline{Y})'V(Y-\overline{Y})$$
$$\omega(Y,Y) = \eta^2(\hat{D}) + \text{tr }(Y-\overline{Y})'V(Y-\overline{Y}) - \eta^2(\overline{Y})$$
$$\frac{\omega(\alpha)}{\eta^2(\alpha)} \leq \sigma(Y)$$

4 Smacof Algorithm

4.1 Introduction to Majorization

Majorization, these days better known as MM (Lange (2016)), is a general approach for the construction of minimization algorithms. There is also minorization, which leads to maximization algorithms, which explains the MM acronym: minorization for maximization and majorization for minimization.

Before the MM principle was formulated as a general approach to algorithm construction there were some important predecessors. Major classes of MM algorithms avant la lettre were the *EM Algorithm* for maximum likelihood estimation of Dempster, Laird, and Rubin (1977), the *Smacof Algorithm* for MDS of De Leeuw (1977), the *Generalized Weiszfeldt Method* of Vosz and Eckhardt (1980), and the *Quadratic Approximation Method* of Böhning and Lindsay (1988). The first formulation of the general majorization principle seems to be De Leeuw (1994).

Let's start with a brief introduction to majorization. Minimize a real valued function σ over $x \in \mathbb{S}$, where $\mathbb{S}$ is some subset of $\mathbb{R}^n$. There are obvious extensions of majorization to functions defined on more general spaces, with values in any partially ordered set, but we do not need that level of generality in this manual. Also majorization applied to σ is minorization applied to $-\sigma$, so concentrating on majorization-minimization and ignoring minorization-maximization causes no loss of generality

Suppose there is a real-valued function η on $\mathbb{S}\otimes\mathbb{S}$ such that

$$\sigma(x) \leq \eta(x,y) \qquad \forall x,y \in \mathbb{S}, (\#eq:maj1) \tag{32}$$
$$\sigma(x) = \eta(x,x) \qquad \forall x \in \mathbb{S}.(\#eq:maj2) \tag{33}$$

The function η is called a *majorization scheme* for σ on S. A majorization scheme is *strict* if $\sigma(x) < \eta(x,y)$ for all $x,y \in S$ withj $x \neq y$.

Define

$$x^{(k+1)}\in\mathop{\text{argmin}}_{x\in\mathbb{S}}\eta(x,x^{(k)}),(\#eq:majalg) \tag{34}$$

assuming that $\eta(\bullet,y)$ attains its (not necessarily unique) minimum over $x\in\mathbb{S}$ for each y. If $x^{(k)}\in\text{argmin}_{x\in\mathbb{S}}\,\eta(x,x^{(k)})$ then we stop.

By majorization property @ref(eq:maj1)

$$\sigma(x^{(k+1)})\leq\eta(x^{(k+1)},x^{(k)}). \tag{35}$$

Because we did not stop update rule @ref(eq:majalg) implies

$$\eta(x^{(k+1)},x^{(k)})<\eta(x^{(k)},x^{(k)}). \tag{36}$$

and finally by majorization property @ref(eq:maj1)

$$\eta(x^{(k)},x^{(k)})=\sigma(x^{(k)}). \tag{37}$$

If the minimum in @ref(eq:majalg) is attained for a unique x then $\eta(x^{(k+1)},x^{(k)})<\eta(x^{(k)},x^{(k)})$. If the majorization scheme is strict then $\sigma(x^{(k+1)})<\eta(x^{(k+1)},x^{(k)})$. Under either of these two additional conditions $\sigma(x^{(k+1)})<\sigma(x^{(k)})$, which means that the majorization algorithm is a monotone descent algorithm, and if σ is bounded below on $\mathbb{S}$ the sequence $\sigma(x^{(k)})$ converges.

Note that we only use the order relation to prove convergence of the sequence of function values. To prove convergence of the $x^{(k)}$ we need stronger compactness and continuity assumptions to apply the general theory of Zangwill (1969). For such a proof the argmin in update formula @ref(eq:majalg) can be generalized to $x^{(k+1)}=\phi(x^{(k)})$, where ϕ maps $\mathbb{S}$ into $\mathbb{S}$ such that $\eta(\phi(x),x)\leq\sigma(x)$ for all x.

We give a small illustration in which we minimize σ with $\sigma(x)=\sqrt{x}-\log x$ over $x>0$. Obviously we do not need majorization here, because solving $\mathcal{D}\sigma(x)=0$ immediately gives $x=4$ as the solution we are looking for.

To arrive at this solution using majorization we start with

$$\sqrt{x}\leq\sqrt{y}+\frac{1}{2}\frac{x-y}{\sqrt{y}},(\#eq:sqrtmaj) \tag{38}$$

which is true because a differentiable concave function such as the square root is majorized by its tangent everywhere. Inequality @ref(eq:sqrtmaj) implies

$$\sigma(x)\leq\eta(x,y):=\sqrt{y}+\frac{1}{2}\frac{x-y}{\sqrt{y}}-\log x.(\#eq:examplemaj) \tag{39}$$

Note that $\eta(\bullet, y)$ is convex in its first argument for each y. We have $\mathcal{D}_1\eta(x, y) = 0$ if and only if $x = 2\sqrt{y}$ and thus the majorization algorithm is

$$x^{(k+1)} = 2\sqrt{x^{(k)}}(\#eq : examplealg) \tag{40}$$

The sequence $x^{(k)}$ converges monotonically to the fixed point $x = 2\sqrt{x}$, i.e. to $x = 4$. If $x^{(0)} < 4$ the sequence is increasing, if $x^{(0)} < 4$ it is decreasing. Also, by l'Hôpital,

$$\lim_{x\rightarrow 4}\frac{2\sqrt{x}-4}{x-4} = \frac{1}{2}(\#eq : hopi1) \tag{41}$$

and thus convergence to the minimizer is linear with asymptotic convergence rate $\frac{1}{2}$. By another application of l'Hôpital

$$\lim_{x\rightarrow 4}\frac{\sigma(2\sqrt{x})) - \sigma(4)}{\sigma(x) - \sigma(4)} = \frac{1}{4}, (\#eq : hopi2) \tag{42}$$

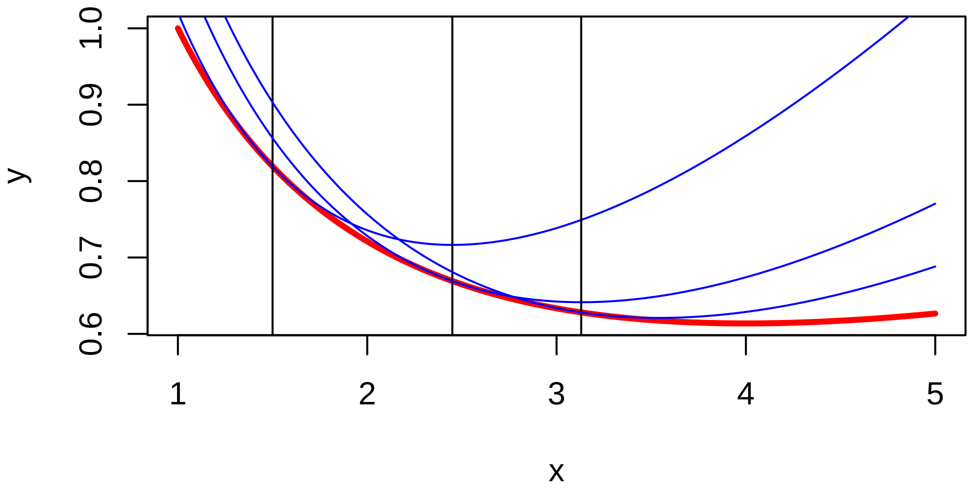
and convergence to the minimum is linear with asymptotic convergence rate $\frac{1}{4}$. Linear convergence to the minimizer is typical for majorization algorithms, as is the twice-as-fast linear convergence to the minimum value.

This small example is also of interest, because we minimize a *DC function*, the difference of two convex functions. In our example the convex functions are minus the square root and minus the logarithm. Algorithms for minimizing DC functions define other important subclasses of MM algorithms, the *DC Algorithm* of Tao Pham Dinh (see Le Thi and Tao (2018) for a recent overview), the *Concave-Convex Procedure* of Yuille and Rangarajan (2003), and the *Half-Quadratic Method* of Donald Geman (see Niikolova and Ng (2005) for a recent overview). For each of these methods there is a huge literature, with surprisingly little non-overlapping literatures. The first phase of the smacof algorithm, in which we improve the configuration for given disparities, is DC, concave-convex, and half-quadratic.

In the table below we show convergence of @ref(eq:examplealg) starting at $x = 1.5$. The first column show how far $x^{(k)}$ deviates from the minimizer (i.e. from 4), the second shows how far$\sigma(x^{(k)})$ deviates from the minimum (i.e. from $2 - \log 4$). We clearly see the convergence rates $\frac{1}{2}$ and $\frac{1}{4}$ in action.

```
itel   1 2.5000000000 0.2055741244
itel   2 1.5505102572 0.0554992066
itel   3 0.8698308399 0.0144357214
itel   4 0.4615431837 0.0036822877
itel   5 0.2378427379 0.0009299530
itel   6 0.1207437506 0.0002336744
itel   7 0.0608344795 0.0000585677
itel   8 0.0305337787 0.0000146606
itel   9 0.0152961358 0.0000036675
itel  10 0.0076553935 0.0000009172
itel  11 0.0038295299 0.0000002293
itel  12 0.0019152235 0.0000000573
itel  13 0.0009577264 0.0000000143
itel  14 0.0004788919 0.0000000036
itel  15 0.0002394531 0.0000000009
```

The first three iterations are shown in the figure below. The vertical lines indicate the value of x, function is in red, and the first three majorizations are in blue.



4.1.1 Majorizing Stress

$$\sigma(X)=\frac{1}{2}\sum w_{ij}(\delta_{ij}-d_{ij}(X))^2=1-\rho(X)+\frac{1}{2}\eta^2(X)$$

Lemma 4.1 (Cauchy-Schwartz)**.** *For all X and Y*

$$\rho(X)\geq tr\ X'B(Y)Y=tr\ X'V\bar{Y}$$

with equality if $X=Y$.

Proof.

$$d_{ij}(X)=\sqrt{\text{tr }X'A_{ij}X}$$

$$d_{ij}(X)d_{ij}(Y)\geq\text{tr }X'A_{ij}Y$$

If $d_{ij}(Y)>0$ this

$$d_{ij}(X)\geq\frac{1}{d_{ij}(Y)}\text{tr }X'A_{ij}Y$$

If $d_{ij}(Y)=0$ then

$$d_{ij}(X)\geq b_{ij}\text{ tr }X'A_{ij}Y=0$$

□

$$\rho(X)=\text{tr }X'V\overline{X}\leq\eta(X)\eta(\overline{X})$$

$$\sigma(X)=1+\frac{1}{2}\eta^2(X-\overline{X})-\frac{1}{2}\eta^2(\overline{X}),\tag{43}$$

$$\sigma(X)\leq1+\frac{1}{2}\eta^2(X-\overline{X})-\frac{1}{2}\eta^2(\overline{X}).\tag{44}$$

4.1.2 Accelerating

$$X^+=\frac{1}{2}(1+\beta)\overline{X}+\frac{1}{2}(1-\beta)X$$

Regular smacof $\beta=1$. Accelerated $\beta=3$

4.2 Stress formula two

Minorization result

5 smacof Datastructure

with or without weights

5.0.1 Metric

$(i, j, \text{dissimilarity}, \text{weight})$

5.1 Interval

$(i, j, \text{lower bound}, \text{upper bound}, \text{weight})$

5.2 Ordinal

$(i, j, \text{tied}, textweight)$

5.3 Paired Comparisons

$(i, j, k, l, \text{tied}, \text{weight})$

5.4 Complete triads

$$(i, j, k, \text{smallest}, \text{largest}, \text{weight})$$

Indicator

$$(i, l, \text{weight})$$

6 Code

The programs for the techniques discussed in this manual are (currently) written in R (R Core Team (2024)). There are plans to translate them, or at least their computational cores, to C, but I am not sure I'll ever get to that.

The functions in the R files that I wrote are all called smacofFoo, using Camel Case, where Foo is something more or less descriptive of what the function is doing. Of course functions that come with R, or with packages written by others, keep their original names. Plots are made in ggplot2 ((**?**)), the manual is written in quarto ().

Each chapter of the manual has one main function implementing the technique discussed in the chapter. Since the programs share a lot of code there are many subroutines or modules implementing common operations. For my private use the code for each chapter is compiled into a barebones R package.

Almost all programs contain what I call a "partial iterator". It is a piece of code that performs iterations and that looks like

```
smacofFoo <- function(xold, itmax, eps, verbose, ...) {
  itel <- 1
  fold <- evaluation xold
  repeat {
    xnew <- update xold
    fnew <- evaluation xnew
    if (verbose) {
      cat(
        "itel ",
        formatC(itel, format = "d"),
        "fold ",
        formatC(fold, format = "f", digits = some number),
        "fnew ",
        formatC(fnew, format = "f", digits = some number),
        "\n"
      )
    }
    if ((test(fold, fnew) || (itel == itmax))) {
      break
    }
    itel <- itel + 1
    xold <- xnew
    fold <- fnew
  }
  return(list(x = xnew, f = fnew, other results))
}
```

Partial iterators can, and often are, nested, so there are outer, inner, innermost and so on iterations. Iterators test for convergence, but in inner iterations they are often called with a small value of itmax, so they only perform a small number of iterations. They merely improve their objective, they do not go all the way to the optimum or fixed point. Many of the iterators depend on alternating least squares (De Leeuw (1994)). majorization (De Leeuw (1994)), or MM (Lange (2016)) to compute these improvements

References

Bauschke, H. H., M. N. Bui, and X. Wang. 2018. "Projecting onto the Intersection of a Cone and a Sphere." *SIAM Journal on Optimization* 28: 2158–88.

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

Borg, I., and P. J. F. Groenen. 2005. *Modern Multidimensional Scaling*. Second Edition. Springer.

De Leeuw, J. 1968. "Nonmetric Discriminant Analysis." Research Note 06-68. Department of Data Theory, University of Leiden.

———. 1975. "A Normalized Cone Regression Approach to Alternating Least Squares Algorithms." Department of Data Theory FSW/RUL.

———. 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.

———. 1988. "Convergence of the Majorization Method for Multidimensional Scaling." *Journal of Classification* 5: 163–80.

———. 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.

———. 2017. "Shepard Non-metric Multidimensional Scaling." 2017. https://jansweb.netlify.app/publication/deleeuw-e-17-e/deleeuw-e-17-e.pdf.

———. 2019. "Normalized Cone Regression." 2019. https://jansweb.netlify.app/publication/deleeuw-e-19-d/deleeuw-e-19-d.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.

———. 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 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.

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.

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

Kruskal, J. B. 1964a. "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis." *Psychometrika* 29: 1–27.

———. 1964b. "Nonmetric Multidimensional Scaling: a Numerical Method." *Psychometrika* 29: 115–29.

Kruskal, J. B., and J. D. Carroll. 1969. "Geometrical Models and Badness of Fit Functions." In *Multivariate Analysis, Volume II*, edited by P. R. Krishnaiah, 639–71. North Holland Publishing Company.

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

Le Thi, H. A., and P. D. Tao. 2018. "DC Programming and DCA: Thirty Years of Developments." *Mathematical Programming, Series B*.

Mair, P., P. J. F. Groenen, and J. De Leeuw. 2022. "More on Multidimensional Scaling in R: smacof Version 2." *Journal of Statistical Software* 102 (10): 1–47. https://www.jstatsoft.org/article/view/v102i10.

Niikolova, M., and M. Ng. 2005. "Analysis of Half-Quadratic Minimization Methods for Signal and Image Recovery." *SIAM Journal Scientific Computing* 27 (3): 937–66.

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

Shepard, R. N. 1962a. "The Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function. I." *Psychometrika* 27: 125–40.

———. 1962b. "The Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function. II." *Psychometrika* 27: 219–46.

Torgerson, W. S. 1952. "Multidimensional Scaling: I. Theory and Method." *Psychometrika* 17 (4): 401–19.

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

Young, F. W. 1981. "Quantitative Analysis of Qualitative Data." *Psychometrika* 46: 357–88.

Young, F. W., J. De Leeuw, and Y. Takane. 1980. "Quantifying Qualitative Data." In *Similarity and Choice. Papers in Honor of Clyde Coombs*, edited by E. D. Lantermann and H. Feger. Bern: Hans Huber.

Yuille, A. L., and A. Rangarajan. 2003. "The Concave-Convex Procedure." *Neural Computation* 15: 915–36.

Zangwill, W. I. 1969. *Nonlinear Programming: a Unified Approach*. Englewood-Cliffs, N.J.: Prentice-Hall.