PROXSCAL

PROXSCAL performs multidimensional scaling of proximity data to find a least-squares representation of the objects in a low-dimensional space. Individual differences models can be specified for multiple sources. A majorization algorithm guarantees monotone convergence for optionally transformed, metric and nonmetric data under a variety of models and constraints.

Detailled mathematical derivations concerning the algorithm can be found in Commandeur and Heiser (1993).

Notation

The following notation is used throughout this chapter, unless stated otherwise. For the dimensions of the vectors and matrices are used:

n	Number of objects
m	Number of sources
p	Number of dimensions
s	Number of independent variables
h	maximum(s, p)
l	Length of transformation vector
r	Degree of spline
t	Number of interior knots for spline

The input and input-related variables are:

Δ_k	$n \times n$ matrix with raw proximities for source k
$\mathbf{W}_k$	$n \times n$ matrix with weights for source k
$\mathbf{E}$	$n \times s$ matrix with raw independent variables
$\mathbf{F}$	$n \times p$ matrix with fixed coordinates

Output and output-related variables are:

$\hat{\mathbf{D}}_k$	$n \times n$ matrix with transformed proximities for source k
$\mathbf{Z}$	$n \times p$ matrix with common space coordinates
$\mathbf{A}_k$	$p \times p$ matrix with space weights for source k
$\mathbf{X}_k$	$n \times p$ matrix with individual space coordinates for source k
$\mathbf{Q}$	$n \times h$ matrix with transformed independent variables
$\mathbf{B}$	$h \times p$ matrix with regression weights for independent variables
$\mathbf{S}$	$l \times (r + t)$ matrix of coefficients for the spline basis

Special matrices and functions are:

$\mathbf{J}$	$\mathbf{I} - \mathbf{1}\mathbf{1}^{\mathrm{T}} / \mathbf{1}^{\mathrm{T}}\mathbf{1}$, centering matrix of appropriate size
$D(\mathbf{X}_k)$	$n \times n$ matrix with distances, with elements $\{d_{ijk}\}$, where $d_{ijk} = \sqrt{(\mathbf{x}_{ik} - \mathbf{x}_{jk})(\mathbf{x}_{ik} - \mathbf{x}_{jk})}$
$\mathbf{V}_k$	$n \times n$ matrix with elements $\{v_{ijk}\}$, where $v_{ijk} = \begin{cases} -w_{ijk} & \text{for } i \neq j \\ \sum_{l \neq i}^{n} w_{ilk} & \text{for } i = j \end{cases}$
$\mathbf{B}(\mathbf{X}_k)$	$n \times n \times m$ matrix with elements $\{b_{ijk}\}$, where $b_{ijk} = \begin{cases} \dfrac{-w_{ijk}\, \mathrm{f}(\delta_{ijk})}{\mathrm{d}_{ij}(\mathbf{X}_k)} & \text{if } \mathrm{d}_{ij}(\mathbf{X}_k) > 0 \text{ and } i \neq j \\ 0 & \text{if } \mathrm{d}_{ij}(\mathbf{X}_k) = 0 \text{ and } i \neq j \\ -\sum_{l \neq i}^{n} b_{ilk} & \text{if } i = j \end{cases}$

Introduction

The following loss function is minimized by PROXSCAL,

$$\sigma^2 \equiv \frac{1}{m} \sum_{k=1}^{m} \sum_{i<j}^{n} w_{ijk} \left[\hat{d}_{ijk} - \mathrm{d}_{ij}(\mathbf{X}_k) \right]^2, \tag{1.1}$$

which is the weighted mean squared error between the transformed proximities and the distances of n object within m sources. The transformation function for the

proximities provides nonnegative, monotonically nondecreasing values for the transformed proximities $\hat{d}_{ijk}$. The distances $\mathrm{d}_{ij}(\mathbf{X}_k)$ are simply the Euclidean distances between the object points, with the coordinates in the rows of $\mathbf{X}_k$.

The main algorithm consists of the following major steps:

1. find initial configurations $\mathbf{X}_k$, and evaluate the loss function;
2. find an update for the configurations $\mathbf{X}_k$;
3. find an update for the transformed proximities $\hat{d}_{ijk}$;
4. evaluate the loss function; if some predefined stop criterion is satisfied, stop; otherwise, go to step 2.

Preliminaries

At the start of the procedure, several preliminary computations are performed to handle missing weights or proximities, and initialize the raw proximities.

Missings

On input, missing values may occur for both weights and proximities. If a weight is missing, it is set equal to zero. If a proximity is missing, the corresponding weight is set equal to zero.

Proximities

Only the upper or lower triangular part (without the diagonal) of the proximity matrix is needed. In case both triangles are given, the weighted mean of both triangles is used. Next, the raw proximities are transformed such that similarities become dissimilarities by multiplying with -1, taking into account the conditionality, and setting the smallest dissimilarity equal to zero.

Transformations

For ordinal transformations, the nonmissing proximities are replaced by their ascending rank numbers, also taking into account the conditionality. For spline transformations, the spline basis $\mathbf{S}$ is computed.

Normalization

The proximities are normalized such that the weighted squared proximities equal the sum of the weights, again, taking into account the conditionality.

Step 1: Initial Configuration

PROXSCAL allows for several initial configurations. Before determining the initial configuration, missings are handled, and the raw proximities are initialized. Finally, after one of the starts described below, the common space $\mathbf{Z}$ is centered on the origin and optimally dilated in accordance with the normalized proximities.

Simplex Start

The simplex start consists of a rank-*p* approximation of the matrix $\mathbf{V}^{-}\mathbf{B}(\mathbf{J})$. Set $\mathbf{H}$, an $n \times p$ columnwise orthogonal matrix, satisfying $\mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}_p$ equal to $\mathbf{I}_p$, where $\mathbf{I}_p$ denotes the matrix with the first *p* columns of the identity matrix. The nonzero rows are selected in such a way that the first $\mathbf{Z} = \mathbf{B}(\mathbf{J})\mathbf{H}$ contains the p columns of $\mathbf{B}(\mathbf{J})$ with the largest diagonal elements. The following steps are computed in turn, until convergence is reached:

1. For a fixed $\mathbf{Z}$, $\mathbf{H} = \mathbf{P}\mathbf{Q}^{\mathrm{T}}$, where $\mathbf{P}\mathbf{Q}^{\mathrm{T}}$ is taken from the singular value decomposition $\mathbf{B}(\mathbf{J})\mathbf{Z} = \mathbf{P}\mathbf{L}\mathbf{Q}^{\mathrm{T}}$;
2. For a fixed $\mathbf{H}$, $\mathbf{Z} = 2^{-1/2}\mathbf{V}^{-}\mathbf{B}(\mathbf{J})\mathbf{H}$, where $\mathbf{V}^{-}$ is the pseudo-inverse of $\mathbf{V}$.

For a restricted common space $\mathbf{Z}$, the second step is adjusted in order to fullfill the restictions. This procedure was introduced in Heiser (1985).

Torgerson Start

The proximities are aggregated over sources, squared, double centered and multiplied with -0.5, after which an eigenvalue decomposition is used to determine the coordinate values, thus

$$-0.5\mathbf{J}\mathbf{D}^{*}\mathbf{J} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\mathrm{T}} \ ,$$

where elements of $\mathbf{D}^{*}$ are defined as

$$d_{ij}^{*} = \left(\sum_{k=1}^{m} w_{ijk} \hat{d}_{ijk}^{2} \right) \left(\sum_{k=1}^{m} w_{ijk} \right)^{-1}$$

followed by $\mathbf{Z} = \mathbf{Q}\boldsymbol{\Lambda}^{1/2}$, where only the first p positive ordered eigenvalues ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$) and eigenvectors are used. This technique, classical scaling, is due to Torgerson (1952, 1958) and Gower (1966) and also known under the names Torgerson scaling or Torgerson-Gower scaling.

(Multiple) Random Start

The coordinate values are randomly generated from a uniform distribution using the default random number generator from the SPSS system.

User-Provided Start

The coordinate values provided by the user are used.

Step 2: Configuration Update

Update for the Common Space

The common space $\mathbf{Z}$ is related to the individual spaces $\mathbf{X}_k$ $(k = 1, ..., m)$ through the model $\mathbf{X}_k = \mathbf{Z}\mathbf{A}_k$, where $\mathbf{A}_k$ are matrices containing space weights. Assume that weight matrix $\mathbf{A}_k$ is of full rank. Only considering $\mathbf{Z}$ defines (1.1) as

$$\sigma^2(\mathbf{z}) = c + \mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{z} - 2\mathbf{z}^{\mathrm{T}}\mathbf{t}, \tag{3.1}$$

where

$$\begin{aligned} \mathbf{z} &\equiv \text{ vec}(\mathbf{Z}), \\ \mathbf{H} &\equiv \frac{1}{m}\sum_{k=1}^{m}\left(\mathbf{A}_k\mathbf{A}_k^{\mathrm{T}} \otimes \mathbf{V}_k\right), \\ \mathbf{t} &\equiv \text{ vec}\left(\frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k)\mathbf{X}_k\mathbf{A}_k^{\mathrm{T}}\right), \end{aligned} \tag{3.2}$$

for which a solution is found as

$$\mathbf{z} = \mathbf{H}^{-}\mathbf{t} \tag{3.3}$$

Several special cases exist for which (3.3) can be simplified. First, the weights matrices $\mathbf{W}_k$ may all be equal, or even all equal to one. In these cases $\mathbf{H}$ will simplify, as will the pseudo-inverse of $\mathbf{H}$. Another simplification is concerned with the different models, reflected in restrictions for the space weights. Equation

(3.3) provides a solution for $\mathbf{Z}$, where $\mathbf{A}_k$ is of full rank. This model is the generalized Euclidean model, also known as IDIOSCAL (Carroll and Chang, 1972). The weighted Euclidean model, or INDSCAL, restricts $\mathbf{A}_k$ to be diagonal, which does simplify $\mathbf{H}$, but not the pseudo-inverse. The identity model requires $\mathbf{A}_k = \mathbf{I}$ for all *k*, and does simplify $\mathbf{H}$ and its pseudo-inverse, for the kronecker product $\otimes$ vanishes.

To avoid computing the pseudo-inverse of a large matrix, PROXSCAL uses three technical simplifications when appropriate. First, the pseudo-inverse can be replaced by a proper inverse by adding the nullspace, taking the proper inverse and then subtracting the nullspace again as

$$\mathbf{H}^{-} = \left(\mathbf{H} + \mathbf{N}\right)^{-1} - \mathbf{N}$$

where $\mathbf{N} = \left(\mathbf{1}\mathbf{1}^{\mathrm{T}}\right) / \left(\mathbf{1}^{\mathrm{T}}\mathbf{1}\right)$.

Furthermore, a dimensionwise approach (Heiser and Stoop, 1986) is used which results in a solution for dimension a of $\mathbf{Z}$ as

$$\mathbf{z}_a = \mathbf{V}_a^{-}\overline{\mathbf{z}}_a,$$

where

$$\mathbf{V}_a = \frac{1}{m}\sum_{k=1}^{m}\mathbf{V}_k\mathbf{e}_a^{\mathrm{T}}\mathbf{A}_k\mathbf{A}_k^{\mathrm{T}}\mathbf{e}_a,$$

where $\mathbf{e}_a$ is the *a*-th column of an identity matrix, and

$$\overline{\mathbf{z}}_a = \frac{1}{m}\sum_{k=1}^{m}\left[\mathbf{B}(\mathbf{X}_k)\mathbf{X}_k\mathbf{A}_k^{\mathrm{T}} - \mathbf{V}_k\mathbf{P}_a\mathbf{A}_k\mathbf{A}_k^{\mathrm{T}}\right]\mathbf{e}_a,$$

with $\mathbf{P}_a$ an $n \times p$ matrix equal to $\mathbf{Z}$, but with the *a*-th column containing zeros.

Still, the proper inverse of a $n \times n$ matrix is required. The final simplification is concerned with a majorization function in which the largest eigenvalue of $\mathbf{V}$ allows for an easy update (Heiser, 1987; Groenen, Heiser, and Meulman, 1999). Instead of the largest eigenvalue itself, an upper bound is used for this scalar (Wolkowicz and Styan, 1980).

Update for the Space Weights

An update for the space weights $\mathbf{A}_k \left(k = 1, ..., m\right)$ for the generalized Euclidean model is given by

$$\mathbf{A}_k = \left(\mathbf{Z}^{\mathrm{T}}\mathbf{V}_k\mathbf{Z}\right)^{-1}\left(\mathbf{Z}^{\mathrm{T}}\mathbf{B}\left(\mathbf{X}_k\right)\mathbf{X}_k\right). \tag{3.4}$$

Suppose, $\mathbf{P}_k\mathbf{L}_k\mathbf{Q}_k^{\mathrm{T}}$ is the singular value decomposition of $\mathbf{A}_k$, for which the diagonal matrix with singular values $\mathbf{L}_k$ is in nonincreasing order. Then, for the reduced rank model, the best $r\,(r < p)$ rank approximation of $\mathbf{A}_k$ is given by $\mathbf{R}_k\mathbf{T}_k^{\mathrm{T}}$, where $\mathbf{R}_k$ contains the first r columns of $\mathbf{P}_k\mathbf{L}_k$, and $\mathbf{T}_k$ contains the first r columns of $\mathbf{Q}_k$.

For the weighted Euclidean model, (3.4) reduces to a diagonal matrix

$$\mathbf{A}_k = \operatorname{diag}\left(\mathbf{Z}^{\mathrm{T}}\mathbf{V}_k\mathbf{Z}\right)^{-1} \operatorname{diag}\left(\mathbf{Z}^{\mathrm{T}}\mathbf{B}\left(\mathbf{X}_k\right)\mathbf{X}_k\right).$$

The space weights for the identity model need no update, since $\mathbf{A}_k = \mathbf{I}$ for all *k*.

Simplifications can be obtained if all weights $\mathbf{W}$ are equal to one and for the reduced rank model, which can be done in r dimensions, as explained in Heiser and Stoop (1986).

Restrictions

Fixed coordinates

If some of the coordinates of $\mathbf{Z}$ are fixed by the user, then only the free coordinates of $\mathbf{Z}$ need to be updated. The dimensionwise approach is taken one step further, which results in an update for object *i* on dimension *a* as

$$z_{ia}^{+} = \frac{1}{\mathbf{e}_i^{\mathrm{T}}\bar{\mathbf{V}}_a\mathbf{e}_i}\mathbf{e}_i^{\mathrm{T}}\left[\frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k)\mathbf{X}_k\mathbf{A}_k^{\mathrm{T}}\mathbf{e}_a - \frac{1}{m}\sum_{j\neq a}^{p}\left(\sum_{k=1}^{m}\mathbf{e}_j^{\mathrm{T}}\mathbf{A}_k\mathbf{A}_k^{\mathrm{T}}\mathbf{e}_a\mathbf{V}_k\right)\mathbf{z}_j\right] - \frac{1}{\mathbf{e}_i^{\mathrm{T}}\bar{\mathbf{V}}_a\mathbf{e}_i}\mathbf{e}_i^{\mathrm{T}}\bar{\mathbf{V}}_a\tilde{\mathbf{z}}_{ia}$$

where the *a*th column of $\mathbf{Z}$ is divided into $\mathbf{z}_a = \tilde{\mathbf{z}}_{ia} + z_{ia}\mathbf{e}_i$, with $\mathbf{e}_i$ the *i*th column of the identity matrix, and $\bar{\mathbf{V}}_a = \frac{1}{m}\sum_{k=1}^{m}\mathbf{e}_j^{\mathrm{T}}\mathbf{A}_k\mathbf{A}_k^{\mathrm{T}}\mathbf{e}_a\mathbf{V}_k$.

This update procedure will only locally minimize (3.1) and repeatedly cycling through all free coordinates until convergence is reached, will provide global optimization. After all free coordinates have been updated, $\mathbf{Z}$ is centered on the origin. On output, the configuration is adapted as to coincide with the initial fixed coordinates.

Independent variables

Independent variables $\mathbf{Q}$ are used to express the coordinates of the common space $\mathbf{Z}$ as a weighted sum of these independent variables as

$$\mathbf{Z} = \mathbf{QB} = \sum_{j=1}^{h} \mathbf{q}_j \mathbf{b}_j^{\mathrm{T}} .$$

An update for $\mathbf{Z}$ is found by performing the following calculations for $j = 1, ..., h$:

1. $\mathbf{U}_j = \sum_{k \neq j}^{h} \mathbf{q}_k \mathbf{b}_k^{\mathrm{T}}$

2. $\mathbf{T}_j = \mathbf{C} - \frac{1}{m} \sum_{k=1}^{m} \mathbf{V}_k \mathbf{U}_j \mathbf{A}_k \mathbf{A}_k^{\mathrm{T}}$, where $\mathbf{C} = \frac{1}{m} \sum_{k=1}^{m} \mathbf{B}\left(\mathbf{X}_k\right) \mathbf{X}_k \mathbf{A}_k^{\mathrm{T}}$

3. update $\mathbf{b}_j$ as $\mathbf{b}_j = \left(\frac{1}{m} \sum_{k=1}^{m} \mathbf{q}_j \mathbf{V}_k \mathbf{q}_j \mathbf{A}_k \mathbf{A}_k^{\mathrm{T}} \right)^{-1} \mathbf{T}_j^{\mathrm{T}} \mathbf{q}_j$

4. optionally, compute optimally transformed variables by regressing $\tilde{\mathbf{q}}_j = \frac{1}{k_1} \mathbf{T}_j \mathbf{b}_j + \left(\mathbf{I} - \frac{1}{k_1} \bar{\mathbf{V}}_j \right) \mathbf{q}_j$, where $\bar{\mathbf{V}}_j = \frac{1}{m} \sum_{k=1}^{m} \mathbf{b}_j^{\mathrm{T}} \mathbf{A}_k \mathbf{A}_k^{\mathrm{T}} \mathbf{b}_j \mathbf{V}_k$ and k_1 is greater than or equal to the largest eigenvalue of $\bar{\mathbf{V}}_j$, on the original variable $\mathbf{q}_j$. Missing elements in the original variable are replaced with the corresponding values from $\tilde{\mathbf{q}}_j$.

   Finally, set $\mathbf{Z} = \mathbf{QB} = \sum_{j=1}^{h} \mathbf{q}_j \mathbf{b}_j^{\mathrm{T}}$.

Independent variables restrictions were introduced for the MDS model in Bentler and Weeks (1978), Bloxom (1978), de Leeuw and Heiser (1980) and Meulman and Heiser (1984). If there are more dimensions (*p*) than independent variables (*s*), *p-s* dummy variables are created and treated completely free in the analysis. The transformations for the independent variables from Step 4 are identical to the transformations of the proximities, except that the nonnegativety constraint does not apply. After transformation, the variables $\mathbf{q}$ are centered on the origin, normalized on *n*, and the reverse normalization is applied to the regression weights $\mathbf{b}$.

Step 3: Transformation Update

Conditionality

Two types of conditionalities exist in PROXSCAL. Conditionality refers to the possible comparison of proximities in the transformation step. For unconditional transformations, all proximities are allowed to be compared with each other, irrespective of the source. Matrix-conditional transformations only allow for comparison of proximities within one matrix *k*, in PROXSCAL refered to as one source *k*. Here, the transformation is computed for each source seperately (thus *m* times).

Transformation Functions

All transformation functions in PROXSCAL result in nonnegative values for the transformed proximities. After the transformation, the transformed proximities are normalized and the common space is optimally dilated accordingly. The following transformations are available.

Ratio

$\hat{\mathbf{D}} = \boldsymbol{\Delta}$. No transformation is necessary, since the scale of $\hat{\mathbf{D}}$ is adjusted in the normalization step.

Interval

$\hat{\mathbf{D}} = \alpha + \beta\boldsymbol{\Delta}$. Both α and β are computed using linear regression, in such a way that both parameters are nonnegative.

Ordinal

$\hat{\mathbf{D}} = \mathrm{WMON}(\boldsymbol{\Delta}, \mathbf{W})$. Weighted monotone regression (WMON) is computed using the up-and-down-blocks minimum violators algorithm (Kruskal, 1964; Barlow et al., 1972). For the secondary approach to ties, ties are kept tied, the proximities within tieblocks are first contracted and expanded afterwards.

Spline

$\text{vec}\left(\hat{\mathbf{D}}\right) = \mathbf{Sb}$. PROXSCAL uses monotone spline transformations (Ramsay, 1988). In this case, the spline transformation gives a smooth nondecreasing piecewise polynomial transformation. It is computed as a weighted regression of $\mathbf{D}$ on the spline basis $\mathbf{S}$. Regression weights $\mathbf{b}$ are restricted to be nonnegative and computed using nonnegative alternating least squares (Groenen, van Os and Meulman, 2000).

Normalization

After transformation, the transformed proximities are normalized such that the sum-of-squares of the weighted transformed proximities are equal to $mn(n\text{-}1)/2$ in the unconditional case and equal to $n(n\text{-}1)/2$ in the matrix-conditional case.

Step 4: Termination

After evaluation of the loss function, the old function value and new function values are used to decide whether iterations should continue. If the new function value is smaller than or equal to the minimum Stress value MINSTRESS, provided by the user, iterations are terminated. Also, if the difference in consecutive Stress values is smaller than or equal to the convergence criterion DIFFSTRESS, provided by the user, iterations are terminated. Finally, iterations are terminated if the current number of iterations, exceeds the maximum number of iterations MAXITER, also provided by the user. In all other cases, iterations continue.

Remaining Issues

Acceleration

For the identity model without further restictions, the common space can be updated with acceleration as $\mathbf{Z}^{\text{new}} = 2\mathbf{Z}^{\text{update}} - \mathbf{Z}^{\text{old}}$, also refered to as the relaxed update.

Lowering dimensionality

For a restart in p-1 dimensions, the p-1 most important dimensions need to be identified. For the identity model, the first p-1 principal axes are used. For the

weighted Euclidean model, the *p*-1 most important space weights are used, and for the generalized Euclidean and reduced rank models, the *p*-1 largest singular values of the space weights determine the remaining dimensions.

Stress measures

The following statistics are used for the computation of the Stress measures:

$$\eta^2\left(\hat{\mathbf{D}}\right) = \sum_{k=1}^{m}\sum_{i<j}^{n} w_{ijk}\hat{d}_{ijk}^2$$

$$\eta^4\left(\hat{\mathbf{D}}\right) = \sum_{k=1}^{m}\sum_{i<j}^{n} w_{ijk}\hat{d}_{ijk}^4$$

$$\eta^2\left(\mathbf{X}\right) = \sum_{k=1}^{m}\sum_{i<j}^{n} w_{ijk}\mathrm{d}_{ij}^2\left(\mathbf{X}_k\right)$$

$$\eta^4\left(\mathbf{X}\right) = \sum_{k=1}^{m}\sum_{i<j}^{n} w_{ijk}\mathrm{d}_{ij}^4\left(\mathbf{X}_k\right)$$

$$\rho\left(\mathbf{X}\right) = \sum_{k=1}^{m}\sum_{i<j}^{n} w_{ijk}\hat{d}_{ijk}\mathrm{d}_{ij}\left(\mathbf{X}_k\right)$$

$$\rho^2\left(\mathbf{X}\right) = \sum_{k=1}^{m}\sum_{i<j}^{n} w_{ijk}\hat{d}_{ijk}^2\mathrm{d}_{ij}^2\left(\mathbf{X}_k\right)$$

$$\kappa^2\left(\mathbf{X}\right) = \sum_{k=1}^{m}\sum_{i<j}^{n} w_{ijk}\left(\mathrm{d}_{ij}\left(\mathbf{X}_k\right) - \overline{\mathrm{d}}\left(\mathbf{X}\right)\right)^2$$

where $\overline{\mathrm{d}}\left(\mathbf{X}\right)$ is the average distance.

The loss function minimized by PROXSCAL, normalized raw Stress, is given by:

$$\sigma^2 = \frac{\eta^2\left(\hat{\mathbf{D}}\right) + \eta^2\left(\alpha\mathbf{X}\right) - 2\rho\left(\alpha\mathbf{X}\right)}{\eta^2\left(\hat{\mathbf{D}}\right)}, \text{ with } \alpha = \frac{\rho\left(\mathbf{X}\right)}{\eta^2\left(\mathbf{X}\right)}.$$

Note that at a local minimum of $\mathbf{X}$, α is equal to one. The other Fit and Stress measures provided by PROXSCAL are given by:

Stress-I: $\dfrac{\eta^2\left(\hat{\mathbf{D}}\right) + \eta^2\left(\alpha\mathbf{X}\right) - 2\rho\left(\alpha\mathbf{X}\right)}{\eta^2\left(\alpha\mathbf{X}\right)}$, with $\alpha = \dfrac{\eta^2\left(\hat{\mathbf{D}}\right)}{\rho\left(\mathbf{X}\right)}$.

Stress-II: $\frac{\eta^2\left(\hat{\mathbf{D}}\right)+\eta^2\left(\alpha\mathbf{X}\right)-2\rho\left(\alpha\mathbf{X}\right)}{\kappa^2\left(\alpha\mathbf{X}\right)}$, with $\alpha=\frac{\eta^2\left(\hat{\mathbf{D}}\right)}{\rho\left(\mathbf{X}\right)}$.

S-Stress: $\eta^4\left(\hat{\mathbf{D}}\right)+\eta^4\left(\alpha\mathbf{X}\right)-2\rho^2\left(\alpha\mathbf{X}\right)$, with $\alpha^2=\frac{\rho^2\left(\mathbf{X}\right)}{\eta^4\left(\mathbf{X}\right)}$.

Dispersion Accounted For (DAF): $1-\sigma^2$.

Tucker's coefficient of congruence: $\sqrt{1-\sigma^2}$.

Decomposition of normalized raw Stress

Each part of normalized raw Stress, as described before, is assigned to objects and sources. Either sum over objects or sum over sources are equal to total normalized raw Stress.

Transformations on output

On output, whenever fixed coordinates or independent variables do not apply, the models are not unique. In these cases transformations of the common space and the space weights are in order.

For the identity model, the common space $\mathbf{Z}$ is rotated to principal axes. For the weighted Euclidean model, $\mathbf{Z}=\sqrt{n}\mathbf{Z}\left(\operatorname{diag}\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\right)^{-1/2}$ so that $\operatorname{diag}\left(\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\right)=n\mathbf{I}$, and reverse tranformations are applied to the space weights $\mathbf{A}_k$. Further, the sum over sources of the squared space weights are put in descending order as to specify the importance of the dimensions. For the generalized Euclidean model, the Cholesky decomposition $\mathbf{Z}^{\mathrm{T}}\mathbf{Z}=\mathbf{L}\mathbf{L}^{\mathrm{T}}$ specifies the common space on output as $\mathbf{Z}=\sqrt{n}\mathbf{Z}\left(\mathbf{L}^{\mathrm{T}}\right)^{-1}$, so that $\mathbf{Z}^{\mathrm{T}}\mathbf{Z}=n\mathbf{I}$.

References

Barlow, R. E., Bortholomew, D. J., Bremner, J. M. and Brunk, H. D. (1972). *Statistical inference under order restrictions*. New York: Wiley.

Bentler, P. M. and Weeks, D. G. (1978). Restricted multidimensional scaling models. *Journal of Mathematical Psychology, 17*, 138-151.

Bloxom, B. (1978). Constrained multidimensional scaling in *n* spaces. *Psychometrika, 43*, 397-408.

Carroll, J. D. and Chang, J. J. (1972, March). *IDIOSCAL (Individual Differences In Orientation SCALing): A generalization of INDSCAL allowing idiosyncratic reference systems as well as analytic approximation to INDSCAL*. Paper presented at the spring meeting of the Psychometric Society. Princeton, NJ.

Commandeur, J. J. F. and Heiser, W. J. (1993). *Mathematical derivations in the proximity scaling (PROXSCAL) of symmetric data matrices* (Tech. Rep. No. RR-93-03). Leiden, The Netherlands: Department of Data Theory, Leiden University.

De Leeuw, J. (1977). Applications of convex analysis to multidimensional scaling. In J. R. Barra, F. Brodeau, G. Romier, and B. van Cutsem (Eds.), *Recent developments in statistics* (pp. 133-145). Amsterdam, The Netherlands: North-Holland.

De Leeuw, J. and Heiser, W. J. (1980). Multidimensional scaling with restrictions on the configuration. In P. R. Krishnaiah (Ed.), *Multivariate analysis* (Vol. V, pp. 501-522). Amsterdam, The Netherlands: North-Holland.

Gower, J. C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika, 53*, 325-338.

Groenen, P. J. F., Heiser, W. J. and Meulman, J. J. (1999). Global optimization in least squares multidimensional scaling by distance smoothing. *Journal of Classification, 16*, 225-254.

Groenen, P. J. F., van Os, B. and Meulman, J. J. (2000). Optimal scaling by alternating length-constained nonnegative least squares, with application to distance-based analysis. *Psychometrika, 65*, 511-524.

Heiser, W. J. (1985). *A general MDS initialization procedure using the SMACOF algorithm-model with constraints* (Tech. Rep. No. RR-85-23). Leiden, The Netherlands: Department of Data Theory, Leiden University.

Heiser, W. J. (1987). Joint ordination of species and sites: The unfolding technique. In P. Legendre and L. Legendre (Eds.), *Developments in numerical ecology* (pp. 189-221). Berlin, Heidelberg: Springer-Verlag.

Heiser, W. J. and De Leeuw, J. (1986). *SMACOF-I* (Tech. Rep. No. UG-86-02). Leiden, The Netherlands: Department of Data Theory, Leiden University.

Heiser, W. J. and Stoop, I. (1986). *Explicit SMACOF algorithms for individual differences scaling* (Tech. Rep. No. RR-86-14). Leiden, The Netherlands: Department of Data Theory, Leiden University.

Kruskal, J. B. (1964). Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, *29*, 28-42.

Meulman, J. J. and Heiser, W. J. (1984). Constrained Multidimensional Scaling: more Directions than Dimensions. In T. Havranek et al (Eds.), *COMPSTAT 1984* (pp. 137-142). Wien: Physica Verlag.

Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical Science*, *3*(4), 425-461.

Stoop, I., Heiser, W. J. and De Leeuw, J. (1981). *How to use SMACOF-IA*. Leiden, The Netherlands: Department of Data Theory, Leiden University.

Stoop, I. and De Leeuw, J. (1982). *How to use SMACOF-IB*. Leiden, The Netherlands: Department of Data Theory, Leiden University.

Torgerson, W. S. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika*, *17*, 401-419.

Torgerson, W. S. (1958). *Theory and methods of scaling*. New York: Wiley.

Wolkowicz, H. and Styan, G. P. H. (1980). Bounds for eigenvalues using traces. *Linear algebra and its applications*, *29*, 471-506.