A GENERAL MDS INITIALIZATION PROCEDURE
USING THE SMACOF ALGORITHM-MODEL WITH CONSTRAINTS

PROXSCAL Progress Report 1*

Willem J. Heiser
Department of Data Theory
University of Leiden

December 1985

*PROXSCAL Progress Reports provide documentation on the design and development of the PROXSCAL Multidimensional Scaling program. They contain preliminary notes, comments, ideas and decisions, without aiming at full explanations. See the Appendix for a list of options and definitions.
A general MDS initialization procedure using the SMACOF algorithm-model with constraints.

Introduction

In most Multidimensional Scaling (MDS) techniques initial values for the parameters are obtained by calling upon cognate procedures that are simpler to execute. These "rational" starts almost by definition violate or ignore some of the objectives of the analysis, such as nonmetricity, weighting, or constraints on the configuration. Moreover, they are frequently based on - what Kruskal (1977) has called - the "neglecting errors" approach, which implies that their behaviour is only predictable in perfect or nearly perfect cases. It is true that the well-known Torgerson-Gower procedure for the simple symmetric case seems to be remarkably robust, but it soon loses its simplicity if we want to incorporate weights, missing data, or individual differences models.

For the rectangular (unfolding) case things are even more complicated. Schönemann's (1970) procedure, for instance, breaks down if the row totals of the data are equal - a frequently occurring case in row-conditional unfolding. The situation does not seem to be much better for Carroll's (1980) or Heiser and de Leeuw's (1979) procedures in terms of robustness and generality. Heiser (1981) reports some results on the performance of seven additional unfolding initialization techniques, neither of which turns out to be uniformly best. And again, weights or missing data do not fit in easily here.

Many of these methods certainly merit further study; indeed, if generalized and made sophisticated (i.e., protected against unfavourable circumstances), they might be(come) serious and independent competitors for general-purpose methods like KYST, ALSCAL, MULTISCALE and PROXSCAL, which try to minimize a "rational" loss function. However, in order to give an impression of the proliferation of specialized, "rational" starting procedures needed, the four major option classes in the design of PROXSCAL are listed in Table 1. For each of the 200 cells, a reasonable provision for
Table 1. Major option combinations in PROXSCAL.

<table>
<thead>
<tr>
<th>SHAPE OF PROXIMITIES &amp; CONDITIONALITY</th>
<th>INDIVIDUAL DIFF. MODEL &amp; FURTHER RESTRICTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IDENTITY</td>
</tr>
<tr>
<td></td>
<td>NO FI RO CO EX</td>
</tr>
<tr>
<td></td>
<td>DIAGONAL</td>
</tr>
<tr>
<td></td>
<td>NO FI RO CO EX</td>
</tr>
<tr>
<td></td>
<td>REDUCED</td>
</tr>
<tr>
<td></td>
<td>NO FI RO CO EX</td>
</tr>
<tr>
<td></td>
<td>FULL</td>
</tr>
<tr>
<td></td>
<td>NO FI RO CO EX</td>
</tr>
<tr>
<td>SYMM</td>
<td>UN MA</td>
</tr>
<tr>
<td>ASYM</td>
<td>UN MA</td>
</tr>
<tr>
<td></td>
<td>RO CO</td>
</tr>
<tr>
<td>RECT</td>
<td>UN MA</td>
</tr>
<tr>
<td></td>
<td>RO CO</td>
</tr>
</tbody>
</table>

*Legend subclassifications:
UN: unconditional  NO: common space is free
MA: matrix-conditional  FI: some coordinates are fixed
RC: row-conditional  RO: centroid restrictions on the row points
CO: column-conditional  CO: centroid restrictions on the column points
EX: external variable restrictions

weights, missing values, and data transformations would be needed as well, of course. It is left as an exercise for the reader to find out how many cells are covered, and how well, by the existing procedures from the literature.

The purpose of this paper is to sketch another type of initialization procedure, especially tailored to the PROXSCAL conditions, which

a. stays within the stress minimizing framework
b. is conceptually and computationally easy
c. is well-defined under all circumstances

Here ease of computation is to be interpreted relative to the sub-procedures of the PROXSCAL program proper. Note that superiority in avoiding local minima is not included as a criterion; although it is believed that the procedure will be "good" in this respect, pursuing such an aim systematically defines quite an extensive
research program of its own, which might be started as soon as PROXSCAL is fully operative.

Proposed procedure

The point of departure is to respect all options chosen by the user except that

a. MODEL is temporarily set at IDENTITY ($X_k = X$);
b. DIMENS is temporarily set at "#objects - 1";
c. predetermined initial values are used for the pseudo-distances.

Next a single SMACOF updating step is performed starting from the simplex and restricting the (common) configuration to be of rank $p$ (the actually chosen dimensionality), and to satisfy the CONSTRAIN options.

In this compromise SHAPE, CUTOFF, WEIGHTS, AGGREGATE, and CONSTRAIN are fully in action; TRANS (with TIES, REGRES and CONDITION) and DIMENS are passive, but respected. Thus only MODEL is possibly violated; an alternative here would be to determine a $p$-dimensional initial configuration for each source separately, analogous to determining the common space (but without restrictions). Since this would increase the computational load approximately $m$ times, there should be definite advantages in order to justify it.

Making a single SMACOF step from the simplex is conceptually and computationally easy, it is well-defined under all circumstances, and it stays within the stress minimizing framework if de Leeuw and Heiser's (1980) metric projection formulation is used for the rank-$p$ and additional restrictions.

Some details of the operations required

The pseudo-distances $\hat{d}$ are initialized with the non-missing data values $\delta$ if TRANS = LINEAR ($\hat{d}_l = \delta_l - \delta_{\min}$ if REGRES = ASCENDING and $\hat{d}_l = \delta_{\max} - \delta_l$ if REGRES = DESCENDING), with the powered data values if TRANS = n or TRANS = PROPORTION (where $n = 1$), and with ascending or descending rank numbers if TRANS = MONOTONE or TRANS = SMOOTH. They are standardized such that $\sum_l w_l \hat{d}_l^2 = c$. For all cases, $l$ runs within
the partitions defined by CONDITION, and c equals the maximum value of l. Together with MODEL = IDENTITY, this simplifies the general scaling problem to:

$$\min_{X \in \Omega} \sum_{i,j,k} w_{ijk} (\hat{d}_{ijk} - d_{ij}(X))^2$$

(1)

for the most general case. Now the simplex of n points in n - 1 dimensions is simply the configuration

$$J = I - \frac{ee'}{e'e}$$

(2)

or the centered identity matrix. Because all initial distances are equal, the determination of the SMACOF subgradient is very easy as well:

$$\tilde{X}_0 = V \tilde{X}_0 = B(J)J$$

(3)

where $\tilde{X}_0$ is the initial Guttman transform, and $B(J)$ the usual B-matrix with respect to the simplex. Explicitly, the elements of $\tilde{X}_0$ are:

$$\tilde{X}_{ia}^0 = - (w_{ia} \hat{a}_{ia} + w_{ai} \hat{a}_{ai}) \text{ for } i \neq a$$

(4a)

$$\tilde{X}_{ii}^0 = \sum_b (w_{ib} \hat{a}_{ib} + w_{bi} \hat{a}_{bi})$$

(4b)

where the underscores indicate averages across sources, and the usual simplifications under SHAPE apply. Note that $\tilde{X}_0$ is square symmetric and has the form of a B-matrix (because $B(J)$ is centered, $B(J)J = B(J)$), but it has to be interpreted as a configuration matrix! This is particularly relevant in the unfolding case, where we don't have to work with the entire nxn matrix, but need only remember its n \times n \_ \_ \_ off-diagonal part, where $w_{ia} \neq 0$. For further comments, see the discussion section.

The SMACOF metric projection problem that now has to be solved is

$$\min_{H' H = I} \text{tr} (\tilde{X}_0 - XH')'V(\tilde{X}_0 - XH')$$

(5)

$$X \in \Omega$$
We use alternating least squares. For any fixed $X$, the stationary equations for $H$ are
\[
\tilde{X}_0'X = HA \quad \text{with $A$ symmetric.} \tag{6}
\]
Thus we must have
\[
H'\tilde{X}_0'X = A = A' = X'\tilde{X}_0H. \tag{7}
\]
To obtain a solution, suppose the singular value decomposition of the inner products is written as
\[
\tilde{X}_0'X = P\Phi Q', \tag{8}
\]
and choose
\[
\hat{H} = PQ'. \tag{9}
\]
This way $\hat{H}$ satisfies orthonormality, the symmetry condition (7), and the stationary equations (6) with $A = \Phi \Phi'$. For fixed $H$, we can split off a constant $q$ from problem (5) and solve the equivalent problem
\[
\min_{X \in \Omega} q + \text{tr} (\tilde{X}_0H - X)'V(\tilde{X}_0H - X), \tag{10}
\]
which is the usual form of common space restrictions, performed in PROXSCAL by the subroutine SPACER, applied to the configuration $\tilde{X}_0H$. Alternately solving (6) and (10), to any prechosen precision, yields an initial configuration satisfying the requirements. Home.

Since the initialization process as just described is iterative, we still need initial values; however, their choice is much less crucial. $H$ can be initialized by setting $p$ of its rows equal to the rows of the $p \times p$ identity matrix, and the remaining ones equal to zero. The $p$ nonzero rows are selected in such a way that $\tilde{X}_0H$ are the $p$ columns of $\tilde{X}_0$ with largest diagonal element. This immediately defines the input to SPACER, which expects subgradients rather than
Guttman transforms (in order to avoid unnecessary double calculations with V). The output from SPACER is then used to compute (8) and (9). The calculation of the inner products in (8) can very conveniently be done with UPDA as presently defined in PROXSCAL, and in order to find Q we merely need eigenvector computation of size p, not n.

Going from p to p-1 dimensions

When the analysis in p dimensions is completed, the usual initialization for a subsequent analysis in p-1 dimensions is to rotate the (common) space towards principal axes orientation, and to drop the last axis. If X is constrained, however, we have to be more careful. Following the same rationale as before, we compute

$$\min_{H' = I} \min_{X \in \Omega} \text{tr} \left( \bar{X}_p - X H' \right)' V \left( \bar{X}_p - X H' \right)$$

(11)

analogous to (5), with the last p-dimensional Guttman transform \(\bar{X}_p\) replacing \(\bar{X}_0\). If X is unrestricted, the optimal rotation-annihilation operator H can be found from

$$\bar{X}_p' V \bar{X}_p H = H \Lambda$$

(12)

i.e., from the spectral decomposition of the V-weighted cross-product matrix, not the simple cross-products. In all cases we can use the algorithm of the previous section.

Discussion

The proposed procedure combines ideas from:

- Shepard: starting from the simplex,
- Kruskal: minimizing stress,
- Guttman: using the Guttman transform,
- de Leeuw: integrating the Guttman transform with the metric projection.

This particular combination is, I believe, new. Computationally, there is some resemblance with the Guttman-Lingoes initialization
in the unweighted, unconstrained case. Their procedure is to decom- 
pose $\mathbf{X}_0$ into rank-$p$ symmetric components (which can be done because 
$\mathbf{X}_0$ is symmetric, and Gramian). Thus if 

$$
\mathbf{X}_0 = K\Psi^2K',
$$

(13)

the G-L initial configuration is $K\Psi$, while our procedure would lead 
to $K\Psi^2$ (where as usual the last $n-p$ eigenvalues have been dropped). 
Guttman and Lingoes both give a different rationale. Guttman (1968) 
refers to Guttman (1946); using binary weights only, he defines 
the related problem 

$$
\max_X \sum_i \sum_j w_{ij} d_{ij}^2 \mathbf{X} \quad (X),
$$

(14)

from which the eigenproblem (13) follows. This is indeed a whole 
independent family of related problems, which could be extended to 
individual differences models and other constraints; they are 
equivalent to Hayashi's quantification methods, and have been 
studied by de Leeuw (1973) under the name maximum sum techniques, 
and by Heiser (1981).

Lingoes' account of the G-L initialization in Lingoes and Roskam 
(1973) alludes to the simplex, but does not exploit the consequen-
ces. Lingoes and Roskam remark: "Reconsidering the matrix $C^{(0)}$ in 
(48), we see that its off-diagonal elements (...) can be looked 
upon as scalar products" (o.c., p.18), and give the further comment: 
"Although we hold the position that heuristics should be kept to 
a minimum insofar as possible when constructing an algorithm, since 
this promotes a mathematical analysis of the algorithm's properties 
and behavior, we see no conflict in maintaining this stance and, 
at the same time, recommending that some other function be minimized 
prior to minimizing stress. After all, that is done everytime we 
start with a so-called good initial configuration" (o.c., p.70- 
71). Exit the simplex.

Because the present rationale is more firmly tied up with the 
general characteristics of PROXSCAL, we may expect that it is at 
least as "good" as the G-L initialization, and perhaps superior
in the more severely restricted cases. Note that temporarily setting \( A_k = 1 \) implies staying within the feasible region except for the reduced rank model, while temporarily ignoring the restrictions on \( X \) would generally lead to a nonfeasible initial configuration in all cases. It is not difficult to introduce slight modifications to accommodate the reduced rank model as well.

References


Guttman, L. (1968). A general nonmetric technique for finding the smallest coordinate space for a configuration of points. Psychometrika, 33, 469-506.


Appendix: the PROC PROXSCAL options

Data options:

DATA=SASdataset names the SAS data set to be used by PROC PROXSCAL. If DATA= is omitted, the most recently created SAS data set is used. The data set must have the special structure indicated by the SHAPE= option.

SHAPE=SYMmetric Specifies the shape of the input data set. The default value is SHAPE=SYMmetric. A data set input to PROC PROXSCAL consists of one or more matrices. Each matrix must have the same number of rows and columns, and the number of rows must equal the number of columns, except when SHAPE=RECTangular. For symmetric and asymmetric data the diagonal is always ignored, and for symmetric data the upper triangular portion is also ignored.

For rectangular data the number of column stimuli is set equal to the number of variables in the data set, and the number of row stimuli equals the value of the ROWS= option.

ROWS=n must be used when SHAPE=RECTangular. For other values of SHAPE, ROWS has no effect. The value of ROWS= indicates the number of row stimuli in the analysis, and the number of SAS observations in each data matrix. ROWS= has no default.

TRANS=MONOTONE specifies the nature of the data transformation part of the analysis. It controls the type of regression used by PROC PROXSCALE, together with the TIES=, REGRES= and CONDITION= options. When TRANS=n is specified, a n'th power transformation is applied. Default is MONOTONE for symmetric and asymmetric data, SMOOTH for rectangular data.

TIES=KEEP indicates the treatment of ties for monotone and smooth transformations. For other values of TRANS, TIES has no effect. Default is UNTIE.

TIES=UNTIE

REGRES=ASCENDING specifies whether the regression must be ascending or descending. ASCENDING corresponds with dissimilarity data, whereas DESCENDING must be chosen for similarity data.
DESCENDING cannot be combined with TRANS=PROPORTION. Default is ASCENDING.

CONDITION=ROW
Specifies the conditionality of the data. Note that CONDITION=ROW or CONDITION=COLUMN cannot be used with symmetric data. The default is MATRIX for symmetric and asymmetric data, and ROW for rectangular data.

CONDITION=COLUMN

CONDITION=MATRIX

CONDITION=UN

CUTOFF=n
Defines the range of values treated as missing data by PROC PROXSCAL. SAS missing values are always treated as missing data, independent of the CUTOFF= value. Otherwise, values greater than or equal to CUTOFF are not missing, values less than CUTOFF are missing. By default, CUTOFF=0.0.

WEIGHTS=SASdsn
Specifies the name of the data set containing the arbitrary, non-negative data weights used in the loss function. The data set must have the structure indicated by the SHAPE= option. By default, the analysis will be unweighted.

AGGREGATE
Tells PROC PROXSCAL to find the best compromise single data matrix, respecting all other Data options. Cannot be used when CONDITION=UN. After giving a summary of the results of this operation, the analysis proceeds with the aggregated data matrix and ignores the MODEL= option.

Analysis options:

DIMENS=n1 TO n2
Scales the data in several spaces from n1 to n2 dimensions. Solutions are obtained for each dimensionality from n1 down to n2, inclusive, where n1 must be not less than n2. The lower bound of DIMENS is 1, while the upper bound depends on the size and the shape of the problem.

MODEL=IDENTITY
Specifies the type of model used to analyze the different data sources. The default is MODEL=IDENTITY.

MODEL=DIAGONAL

MODEL=REDUCED

MODEL=FULL

RANK=n
Specifies the rank of the weight matrices when MODEL=REDUCED. Default is 1.

INITIAL=SASdsn
Specifies the name of the data set containing the initial
configuration to be read in. By default, PROC PROXSCAL will
generate rational starting values, respecting all data and
analysis options as closely as possible. Note that the num-
ber of SAS variables must be equal to the maximum of DIMENS;
the number of SAS observations must be equal to the number
of stimuli for symmetric and asymmetric data. For rectangular
data the row stimulus coordinates must precede the column
stimulus coordinates, and the number of SAS observations must
equal the number of rows plus the number of columns. SAS miss-
ing values may be present.

EXTERNAL=SASdsn  Specifies the name of the data set containing external in-
formation about the stimuli. The conventions for the SAS
observations are the same as in the INITIAL= option. The type
of the variables can be multiple nominal, single nominal,
ordinal or numerical, as specified in ID. If EXTERNAL= is
used without CONSTRAN=EXTERN, the external variables will
be fitted in only after a solution for the proximities has
been obtained.

CONSTRAIN=FIX  Offers four ways to constrain the (common) configuration. FIX
can only be used in conjunction with INITIAL=. The non-missing
values of the initial configuration will remain unchanged
during the analysis. ROW and COLUMN can only be used when
SHAPE=RECTANGULAR, and cause PROC PROXSCAL to locate the row
points in the centroid of selected column points (ROW) or
vice versa (COLUMN). EXTERN can only be used in conjunction
with EXTERNAL=. The external variables will actually affect
the location of the points in the solution.

CENTROID=n  Indicates the number of distinct data values that will be
used to select the centroid restrictions when CONSTRAN=ROW
or CONSTRAN=COLUMN. If REGRES=ASCENDING this will be the n
smallest values, if REGRES=DESCENDING the n largest values.
Default is n=1.

Algorithmic options:

ACCELERATE=FIXED  Selects the acceleration device to speed up computation.

ACCELERATE=VARIA  VARIA results in less but somewhat more expensive steps. The
default is FIXED.

ITER=n  Controls the precision and cost of the solution. By default ITER=50.

CONVERGE=n  Controls the precision and cost of the solution. By default CONVERGE=0.0001.

MINSTRESS=n  Provides an additional mechanism for determining the conditions under which the iterative process terminates. By default, MINSTRESS=0.005.

Output options:

OUT=SASdsn  Specifies the name of a new SAS data set containing the results of the analysis.

PRINT  Prints the raw data and any other input data.

PLOT  Plots the multidimensional scaling results. Plots can be displayed on a terminal if it has graphics capabilities, or printed on a graphics device or on a line printer.

HEADER  Produces a header summarizing the options in effect for the analysis.

DIAGNOSTICS  Provides a number of additional diagnostic measures for assessing the quality of the solution.