RR 93-04

MATHEMATICAL DERIVATIONS IN THE PROXIMITY SCALING(PROXSCAL) OF SYMMETRIC DATA MATRICES

Jacques J.F. Commaneur
Willem J. Heiser

Department of Data Theory
University of Leiden

Contents

1 Introduction

This paper contains most of the algebra involved in the multidimensional scaling (MDS) of *symmetric* data matrices as performed by the computer program PROXSCAL (PROXimity SCALing). The mathematical derivations underlying PROXSCAL are scattered over a number of papers, notably De Leeuw and Heiser (1980), Heiser (1985a, 1985b), Heiser and Stoop (1986), and Meulman and Heiser (1984), but also in unpublished material written by Heiser. The present paper provides a comprehensive overview of all this algebra, filling in details that are sometimes only implicitly covered in other papers. Moreover, we intend to use the present paper as a reference for more programmatical details concerning the calculations performed in PROXSCAL itself. The algebra involved in MDS of asymmetric and rectangular data matrices will be treated elsewhere.

Given the (dis)similarities δ_{ijk} between n objects ($i, j = 1, \ldots, n$) for m sources ($k = 1, \ldots, m$), PROXSCAL determines m configurations $\mathbf{X}_k$ of order (n x p), such that the Euclidean distances $d_{ij}(\mathbf{X}_k)$ between the rows of the $\mathbf{X}_k$'s (conceived of as n points in p dimensions) approximate the given (dis)similarities δ_{ijk} as well as possible for all $i, j = 1, \ldots, n$ and $k = 1, \ldots, m$. The formal problem PROXSCAL solves is the minimization of the least squares loss function

$$f(\mathbf{X}_1, \ldots, \mathbf{X}_m) \equiv \frac{1}{m} \sum_{k=1}^{m} \sum_{i<j}^{n} w_{ijk}[\delta_{ijk} - d_{ij}(\mathbf{X}_k)]^2. \tag{1.1}$$

This general loss function is called STRESS as a tribute to Kruskal (1964). In (1.1), $d_{ij}^2(\mathbf{X}_k) \equiv (\mathbf{x}_{ik} - \mathbf{x}_{jk})'(\mathbf{x}_{ik} - \mathbf{x}_{jk})$, where $\mathbf{x}_{ik}$ and $\mathbf{x}_{jk}$ are column vectors of order (p x 1) containing rows i and j of matrix $\mathbf{X}_k$, and w_{ijk} is a given nonnegative weight, possibly different for each i, j and k. The weights w_{ijk} may serve two purposes. First, they can be used to handle *missing data*, where certain (dis)similarities are not known; in that case, the corresponding weights are simply set equal to zero. A second application of the weights is to put a larger emphasis on certain (dis)similarities in the analysis than on others.

The unknown matrices $\mathbf{X}_k$ may either be free, or they may be restricted in a number of ways. In the latter case, the restrictions in PROXSCAL are always of the type $\mathbf{X}_k = \mathbf{Z}\mathbf{A}_k$, where $\mathbf{Z}$ is an unknown (n x p) matrix representing a common or group space, and the unknown (p x p) matrices $\mathbf{A}_k$ contain weights. Both $\mathbf{Z}$ and $\mathbf{A}_k$ may be unrestricted or restricted. Restrictions on the matrices $\mathbf{A}_k$ quite naturally result in the following models, of which the first three are *individual differences* models:

$\mathbf{A}_k$ an arbitrary nonsingular matrix (GENERALIZED Euclidean model, IDIOSCAL)
$\mathbf{A}_k$ an arbitrary matrix of rank $r < p$ (REDUCED rank model)
$\mathbf{A}_k$ an arbitrary diagonal matrix (WEIGHTED Euclidean model, INDSCAL)
$\mathbf{A}_k$ the identity matrix (IDENTITY model)

Moreover, in PROXSCAL the common space $\mathbf{Z}$ is allowed to be restricted in two mutually exclusive ways. First, some or all coordinates of the group space may be fixed by the user, or the user may constrain the common space to be a linear combination of *external* variables; in the latter case we require that $\mathbf{ZA}_k = \mathbf{EBA}_k$, where $\mathbf{E}$ is a given (n x s) matrix containing the external variables, and $\mathbf{B}$ is an unknown (s x p) matrix of *regression weights*. In this fitting process, PROXSCAL also determines optimal transformations for the external variables, where the type of transformation depends on the measurement level of each variable.

The user has a choice of five different transformations of the (dis)similarities. These are: monotone, smooth monotone, power, linear, and proportional transformations. Also, the user may request these transformations to be applied to each source separately (MAtrix conditional), or to all the data simultaneously (UNconditional). Table 1 contains a schematic overview of all the combinations PROXSCAL is able to handle.

Compared to other existing scaling programs, PROXSCAL has a number of important advantages. In contrast with the subgradient method used in PROXSCAL, the gradient methods used in MDSCAL and KYST as developed by Kruskal (1964, 1977), and SSA and MINISSA as developed by Lingoes and Roskam (1973) do not guarantee monotone

Table 1 Combinations of constraints (columns) and tranformations (rows) handled by PROXSCAL.

	$\mathbf{A}_k$:	GENERALIZED			REDUCED			WEIGHTED			IDENTITY		
	Z:	free	fix	ext	free	fix	ext	free	fix	ext	free	fix	ext
	mon												
	smo												
UN	pow												
	lin												
	prop												
	mon												
	smo												
MA	pow												
	lin												
	prop												

convergence of the corresponding algorithms. Other available scaling programs like, for example, ALSCAL (Takane, Young, and De Leeuw, 1977) use alternating least squares (ALS), which gives monotone convergence. However, scaling programs based on ALS start by converting the (dis)similarities to squared distances or to inner products, meaning that the analysis is not performed on given, but on derived data. In PROXSCAL, no a priori conversion of the data is needed, and the unsquared (dis)similarities are directly approximated by unsquared distances. To quote Young and Hamer (1987, p.33):

> "..., de Leeuw and Heiser (1980) have proposed SMACOF, an MDS algorithm that is as flexible as ALSCAL. This algorithm is an improvement over ALSCAL in two major ways (a) It is simpler, faster, and more elegant; and (b) the algorithm fits distances instead of squared distances, which is more desirable. Although the algorithm is as flexible as ALSCAL, at the time of this writing a single program is not available which fits all models to all types of data. However, such a program is being developed, and when completed, will become the least squares program of choice, particularly if made available in a major statistical system."

PROXSCAL combines features from several older scaling programs based on the subgradient method discussed in De Leeuw and Heiser (1980). These older programs are collectively called the SMACOF series, where SMACOF is an acronym for Scaling by MAjorizing a COmplex Function. SMACOF-I (described in Heiser and De Leeuw, 1986) covers metric MDS with the IDENTITY model including weights. SMACOF-IA (Stoop, Heiser and De Leeuw, 1981) is identical to SMACOF-I, except that it allows for transformations of the (dis)similarities by functions selected by the user. These functions include raising δ_{ijk} to a power, taking the logarithm of δ_{ijk}, etc. The important difference between SMACOF-I and SMACOF-IB (Stoop and De Leeuw, 1982) is that the latter program adds the possibility of performing nonmetric MDS, allowing monotone transformations to be applied to all (dis)similarities simultaneously, as well as to each source separately, and even to each row or column of each source separately. Work has also been invested in two other programs: SMACOF-II designed to handle MDS with restrictions on the common space, and an individual differences version of SMACOF-IB (incorporating the above mentioned IDIOSCAL, REDUCED rank, and INDSCAL models). However, these were never completely finished.

A main and recurring theme in the subgradient method for the minimization of STRESS function (1.1) over restricted configurations $\mathbf{X}_k$ is the following two-step procedure. First, an *unrestricted* solution is calculated, then the unrestricted solution is projected onto the subspace containing the constrained solutions, yielding the required constricted update.

Therefore, in section 2 we first discuss how to set up a convergent algorithm for the minimization of (1.1) over unrestricted matrices $\mathbf{X}_k$. Section 3 deals with a general and convergent procedure for obtaining restricted solutions, irrespective of the type of constraints. In section 4 the specific constraints implemented in PROXSCAL are discussed, as well as how to solve the corresponding projection problems.

A subproblem in PROXSCAL involves the projection of a matrix onto a complicated subspace in the metric of a positive semidefinite matrix; therefore, the solution to this subproblem is treated separately in section 5. In section 6, the problem of determining an update for the common space is discussed when some coordinates are required to remain fixed. Section 7 shows how to fit external variables in the PROXSCAL solution, and section 8 deals with different transformations of the (dis)similarities. Section 9 covers the general initialization procedure used in PROXSCAL, section 10 discusses ways to accelerate the convergence rate of the PROXSCAL algorithms, and section 11 finally shows how to obtain two additive components for loss function (1.1), one representing normalized STRESS, and the other Tucker's squared coefficient of congruence.

2 Unrestricted solutions

Whether the matrices $\mathbf{X}_k$ are required to be restricted or not, loss function (1.1) may always be written in the following form:

$$f(\mathbf{X}_1, \ldots, \mathbf{X}_m) = \frac{1}{m} \sum_{k=1}^{m} \sum_{i<j}^{n} w_{ijk}[\delta_{ijk} - d_{ij}(\mathbf{X}_k)]^2$$

$$= \frac{1}{m} \sum_{k=1}^{m} [\, c_k + \text{tr}\, \mathbf{X}_k'\mathbf{V}_k\mathbf{X}_k - 2\, \text{tr}\, \mathbf{X}_k'\mathbf{B}(\mathbf{X}_k)\mathbf{X}_k \,], \tag{2.1}$$

where

$$c_k = \sum_{i<j}^{n} w_{ijk}\delta_{ijk}^2, \tag{2.2}$$

$\mathbf{V}_k = \{v_{ijk}\}$ is defined by

$$v_{ijk} = \begin{cases} -w_{ijk} & \text{for } i \neq j \\ \sum_{l\neq i}^{n} w_{ilk} & \text{for } i = j \end{cases}, \tag{2.3}$$

and $\mathbf{B}(\mathbf{X}_k) = \{b_{ijk}\}$ is defined by

$$b_{ijk} = \begin{cases} \left.\begin{array}{l} -w_{ijk}\delta_{ijk}/d_{ij}(\mathbf{X}_k) \\ 0 \quad \text{if } d_{ij}(\mathbf{X}_k) = 0 \end{array}\right\} & \text{for } i \neq j \\ -\sum_{l\neq i}^{n} b_{ilk} & \text{for } i = j \end{cases}. \tag{2.4}$$

It may seem that (2.1) is a simple quadratic function, with an equally simple solution. Unfortunately, this is not the case because matrix $\mathbf{B}(\mathbf{X}_k)$ also contains the unknown parameters, as can be seen in (2.4).

However, some algebraic manipulations on the Cauchy-Schwarz inequality

$$d_{ij}(\mathbf{X}_k)d_{ij}(\mathbf{X}_k^0) \geq (\mathbf{x}_{ik} - \mathbf{x}_{jk})'(\mathbf{x}_{ik}^0 - \mathbf{x}_{jk}^0)$$

show that

$$\text{tr } \mathbf{X}_k'\mathbf{B}(\mathbf{X}_k)\mathbf{X}_k \geq \text{tr } \mathbf{X}_k'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \tag{2.5}$$

for any pair of arbitrary (n x p) matrices $\mathbf{X}_k$ and $\mathbf{X}_k^0$. Since it follows from (2.5) that

$$-\text{ tr } \mathbf{X}_k'\mathbf{B}(\mathbf{X}_k)\mathbf{X}_k \leq -\text{ tr } \mathbf{X}_k'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \tag{2.6}$$

and defining the function

$$\text{g}(\mathbf{X}_1, \ldots, \mathbf{X}_m; \mathbf{X}_1^0, \ldots, \mathbf{X}_m^0) = \frac{1}{m}\sum_{k=1}^{m} [\, \text{c}_k + \text{tr } \mathbf{X}_k'\mathbf{V}_k\mathbf{X}_k - 2 \text{ tr } \mathbf{X}_k'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \,], \tag{2.7}$$

we have that

$$\text{f}(\mathbf{X}_1, \ldots, \mathbf{X}_m) \leq \text{g}(\mathbf{X}_1, \ldots, \mathbf{X}_m; \mathbf{X}_1^0, \ldots, \mathbf{X}_m^0). \tag{2.8}$$

In fact, for each of the components we have the inequality

$$\text{f}(\mathbf{X}_k) \leq \text{g}(\mathbf{X}_k, \mathbf{X}_k^0), \tag{2.9}$$

where

$$\text{f}(\mathbf{X}_k) = \text{c}_k + \text{tr } \mathbf{X}_k'\mathbf{V}_k\mathbf{X}_k - 2 \text{ tr } \mathbf{X}_k'\mathbf{B}(\mathbf{X}_k)\mathbf{X}_k, \tag{2.10}$$

and

$$\text{g}(\mathbf{X}_k, \mathbf{X}_k^0) = \text{c}_k + \text{tr } \mathbf{X}_k'\mathbf{V}_k\mathbf{X}_k - 2 \text{ tr } \mathbf{X}_k'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0. \tag{2.11}$$

Since (2.9) states that $\text{g}(\mathbf{X}_k, \mathbf{X}_k^0)$ is always above $\text{f}(\mathbf{X}_k)$, except for the point $\mathbf{X}_k = \mathbf{X}_k^0$ where the two functions meet, the former function *majorizes* the latter. The point where the functions meet is called the *supporting* point. Also, in contrast with $\text{f}(\mathbf{X}_k)$, the majorizing function $\text{g}(\mathbf{X}_k, \mathbf{X}_k^0)$ is quadratic in $\mathbf{X}_k$, and therefore has a relatively simple solution which will now be determined.

Clearly, the minimization of (2.1) over unrestricted matrices $\mathbf{X}_k$ ($k = 1, \ldots, m$) consists of m independent subproblems. Defining the *Guttman transform* as

$$\bar{\mathbf{X}}_k = \mathbf{V}_k^-\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \tag{2.12}$$

where $\mathbf{V}_k^-$ denotes the Moore-Penrose inverse of $\mathbf{V}_k$, it follows from (2.9) that

$$\text{f}(\bar{\mathbf{X}}_k) \leq \text{g}(\bar{\mathbf{X}}_k, \mathbf{X}_k^0). \tag{2.13}$$

It further follows from (2.11) and (2.12) that

$$\begin{aligned} g(\bar{\mathbf{X}}_k, \mathbf{X}_k^0) &= c_k + \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k - 2\ \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \\ &= c_k + \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k - 2\ \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\mathbf{V}_k^-\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \\ &= c_k + \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k - 2\ \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k \\ &= c_k - \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k. \end{aligned} \tag{2.14}$$

Since

$$\begin{aligned} g(\mathbf{X}_k, \mathbf{X}_k^0) &= c_k + \text{tr}\ \mathbf{X}_k'\mathbf{V}_k\mathbf{X}_k - 2\ \text{tr}\ \mathbf{X}_k'\mathbf{V}_k\mathbf{V}_k^-\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \\ &= c_k - \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \text{tr}\ \mathbf{X}_k'\mathbf{V}_k\mathbf{X}_k - 2\ \text{tr}\ \mathbf{X}_k'\mathbf{V}_k\bar{\mathbf{X}}_k \\ &= c_k - \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \text{tr}\ (\mathbf{X}_k - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k - \bar{\mathbf{X}}_k), \end{aligned}$$

and because $\text{tr}\ (\mathbf{X}_k - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k - \bar{\mathbf{X}}_k) \geq 0$, matrix $\mathbf{V}_k$ being positive semidefinite (p.s.d.), for fixed $\mathbf{X}_k^0$ the global minimum of $g(\mathbf{X}_k, \mathbf{X}_k^0)$ is clearly attained where $\mathbf{X}_k = \bar{\mathbf{X}}_k$. From (2.10) and (2.12) we have that

$$\begin{aligned} f(\mathbf{X}_k^0) &= c_k + \text{tr}\ \mathbf{X}_k^{0\prime}\mathbf{V}_k\mathbf{X}_k^0 - 2\ \text{tr}\ \mathbf{X}_k^{0\prime}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \\ &= c_k + \text{tr}\ \mathbf{X}_k^{0\prime}\mathbf{V}_k\mathbf{X}_k^0 - 2\ \text{tr}\ \mathbf{X}_k^{0\prime}\mathbf{V}_k\mathbf{V}_k^-\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \\ &= c_k - \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \text{tr}\ \mathbf{X}_k^{0\prime}\mathbf{V}_k\mathbf{X}_k^0 - 2\ \text{tr}\ \mathbf{X}_k^{0\prime}\mathbf{V}_k\bar{\mathbf{X}}_k \\ &= c_k - \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \text{tr}\ (\mathbf{X}_k^0 - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k^0 - \bar{\mathbf{X}}_k). \end{aligned} \tag{2.15}$$

The last term on the right side of (2.15) always satisfies

$$\text{tr}\ (\mathbf{X}_k^0 - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k^0 - \bar{\mathbf{X}}_k) \geq 0, \tag{2.16}$$

and therefore it follows from (2.14), (2.15) and (2.16) that

$$g(\bar{\mathbf{X}}_k, \mathbf{X}_k^0) \leq f(\mathbf{X}_k^0). \tag{2.17}$$

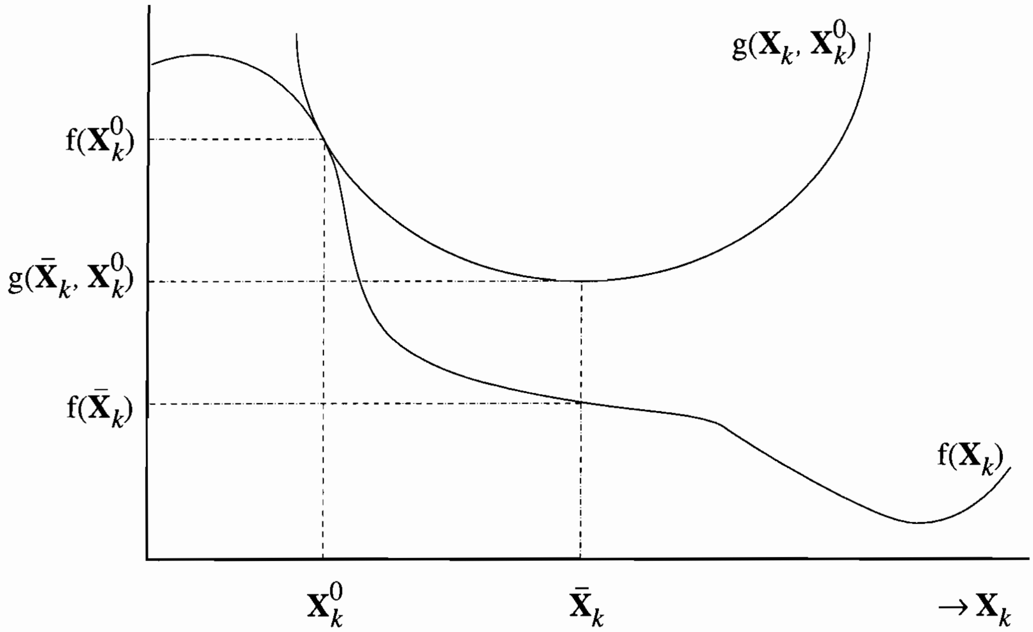

Figure 1 Illustration of the majorization method for the estimation of unrestricted matrices $\mathbf{X}_k$.

Combining (2.13) and (2.17) we obtain

$$f(\bar{\mathbf{X}}_k) \leq g(\bar{\mathbf{X}}_k, \mathbf{X}_k^0) \leq f(\mathbf{X}_k^0), \tag{2.18}$$

which immediately shows that, for a given configuration $\mathbf{X}_k^0$, the loss will never increase if we replace $\mathbf{X}_k^0$ with $\bar{\mathbf{X}}_k = \mathbf{V}_k^{-}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$, but will remain equal at the least and decrease otherwise. This is illustrated in Figure 1.

If we want to minimize (1.1) over unrestricted (n x p) matrices $\mathbf{X}_k$, we may therefore apply the following straightforward algorithm m times (i.e., for each $\mathbf{X}_k$ separately):

1. choose an initial start $\mathbf{X}_k^0$, and evaluate $f(\mathbf{X}_k^0) = \sum_{i<j}^{n} w_{ijk}[\delta_{ijk} - d_{ij}(\mathbf{X}_k^0)]^2$;

2. compute the Guttman transform $\bar{\mathbf{X}}_k = \mathbf{V}_k^{-}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$;

3. replace $\mathbf{X}_k^0$ with $\bar{\mathbf{X}}_k$ and evaluate $f(\mathbf{X}_k^0) = \sum_{i<j}^{n} w_{ijk}[\delta_{ijk} - d_{ij}(\mathbf{X}_k^0)]^2$;

4. go to step 2 if the difference in loss between the current and the previous iteration is larger than some predefined criterion; otherwise stop.

How to determine a 'smart' initial $\mathbf{X}_k^0$ in step 1 of the algorithm will be taken up in section 9.

If we are only dealing with one (n x n) matrix of (dis)similarities and weights (i.e., if k = 1), and if we only want to perform metric MDS, then this algorithm is all we need. Convergence of the algorithm is guaranteed, but not necessarily to the global minimum of the STRESS function. In fact, convergence to local minima is often observed, and in the case that $d_{ij}(\mathbf{X}_k) = 0$ for some i, j, the algorithm may even converge to saddle points that are not local minima (see, e.g., De Leeuw, 1988).

It may be noted that the calculation of the Moore-Penrose inverse of matrix $\mathbf{V}_k$ can be reduced to the calculation of a *proper* inverse by applying the following formula:

$$\mathbf{V}_k^- = (\mathbf{V}_k + \frac{\mathbf{11}'}{n})^{-1} - \frac{\mathbf{11}'}{n}.. \tag{2.19}$$

where $\mathbf{1}$ is the n-vector of ones. The latter formula may only be applied on the condition that $\mathbf{V}_k$ is *irreducible*, that is, it should be impossible to reduce the weight matrix to a block-diagonal matrix by permutation of its rows and columns. Such a situation may arise when $\mathbf{V}_k$ contains zero elements due to missing data. This is hardly a restriction, however: if it is possible to reduce $\mathbf{V}_k$ to a block-diagonal matrix (with t blocks on the diagonal, say), then we are dealing with t independent subproblems which should therefore be solved separately.

If $\mathbf{V}_k$ satisfies the condition of irreducibility, then its rank is guaranteed to be equal to (n - 1). Also, definition (2.3) of $\mathbf{V}_k$ implies that $\mathbf{V}_k\mathbf{u} = \mathbf{0}$ for any constant vector $\mathbf{u} = a\mathbf{1}$, where a is an arbitrary scalar. Therefore, the nullspace of $\mathbf{V}_k$ is the set of constant vectors, and adding a constant vector to $\mathbf{V}_k$ yields a non-singular matrix for which a proper inverse exists. Another property of matrix $\mathbf{V}_k$ that will often be used in the sequel is that $\mathbf{V}_k\mathbf{V}_k^-\mathbf{X} = \mathbf{X}$ for all centered matrices $\mathbf{X}$.

If all weights in partition k are equal to one, then it is easily verified that $\mathbf{V}_k = n\mathbf{J}$ with $\mathbf{J}$ the centering matrix of order (n x n). Since it follows from definition (2.4) that $\mathbf{1}'\mathbf{B}(\mathbf{X}_k^0) = 0$, and therefore that $\mathbf{JB}(\mathbf{X}_k^0)\mathbf{X}_k^0 = \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$, in this case the Guttman transform in step 2 of the algorithm simplifies into $\bar{\mathbf{X}}_k = 1/n\ \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$, and the computation of (2.19) is no longer required.

In the following sections the situation will be discussed where restrictions are imposed on the matrices $\mathbf{X}_k$. Surprisingly, we will see that the calculation of (2.19) is often not needed in these cases (although other, sometimes more complicated, inverses may be required).

3 A general solution for restricted matrices

In the previous section we discussed how to minimize the general loss function (1.1), that is,

$$f(\mathbf{X}_1, \ldots, \mathbf{X}_m) = \frac{1}{m} \sum_{k=1}^{m} \sum_{i<j}^{n} w_{ijk}[\delta_{ijk} - d_{ij}(\mathbf{X}_k)]^2$$

over unrestricted (n x p) matrices $\mathbf{X}_k$. Repeatedly computing the Guttman transform defined by $\bar{\mathbf{X}}_k = \mathbf{V}_k^{-}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$ yields a convergent algorithm for the minimization of (1.1) over free matrices $\mathbf{X}_k$.

However, in practice it is often required that these matrices are restricted, by $\mathbf{X}_k = \mathbf{Z}\mathbf{A}_k$ for $k = 1, \ldots, m$, for instance, or yet other types of constraints. How can we achieve that the *restricted* updates are still computed in such a way that convergence of the algorithm is guaranteed? Here we will discuss how to solve this problem in its most general form, that is, irrespective of the kind of restrictions that we want to impose on the matrices $\mathbf{X}_k$. In the following, $\mathbf{X}_k^0$ denotes an arbitrary matrix satisfying the constraints (whatever these are), $\bar{\mathbf{X}}_k = \mathbf{V}_k^{-}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$ denotes the unrestricted update given the current $\mathbf{X}_k^0$, and $\mathbf{X}_k^+$ denotes a new and better solution for $\mathbf{X}_k$ satisfying the same constraints as $\mathbf{X}_k^0$.

We start by noting that, according to (2.9),

$$\begin{aligned} f(\mathbf{X}_k^+) &\le g(\mathbf{X}_k^+, \mathbf{X}_k^0) \\ &\le c_k + \text{tr } \mathbf{X}_k^{+\prime}\mathbf{V}_k\mathbf{X}_k^+ - 2 \text{ tr } \mathbf{X}_k^{+\prime}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \\ &\le c_k - \text{tr } \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \text{tr } \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \text{tr } \mathbf{X}_k^{+\prime}\mathbf{V}_k\mathbf{X}_k^+ - 2 \text{ tr } \mathbf{X}_k^{+\prime}\mathbf{V}_k\mathbf{V}_k^{-}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \\ &\le c_k - \text{tr } \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \text{tr } \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \text{tr } \mathbf{X}_k^{+\prime}\mathbf{V}_k\mathbf{X}_k^+ - 2 \text{ tr } \mathbf{X}_k^{+\prime}\mathbf{V}_k\bar{\mathbf{X}}_k \end{aligned}$$

and therefore, that

$$f(\mathbf{X}_k^+) \le c_k - \text{tr } \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \text{tr } (\mathbf{X}_k^+ - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k^+ - \bar{\mathbf{X}}_k). \qquad (3.1)$$

From (3.1) it immediately follows that also

$$\begin{aligned} f(\mathbf{X}_1^+,\ldots,\mathbf{X}_m^+) &\le g(\mathbf{X}_1^+,\ldots,\mathbf{X}_m^+;\mathbf{X}_1^0,\ldots,\mathbf{X}_m^0) \\ &\le \frac{1}{m}\{ \sum_{k=1}^{m} c_k - \sum_{k=1}^{m} \text{tr } \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \sum_{k=1}^{m} \text{tr } (\mathbf{X}_k^+ - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k^+ - \bar{\mathbf{X}}_k)\}. \qquad (3.2) \end{aligned}$$

We further know (see (2.15)) that

$$f(\mathbf{X}_1^0,\ldots,\mathbf{X}_m^0) = \frac{1}{m}\{\sum_{k=1}^m c_k - \sum_{k=1}^m \text{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \sum_{k=1}^m \text{tr}\ (\mathbf{X}_k^0 - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k^0 - \bar{\mathbf{X}}_k)\}. \quad (3.3)$$

The key to the solution is contained in the last terms of (3.2) and (3.3). If we are able to determine those restricted updates $\mathbf{X}_k^+$ that minimize the following loss function

$$h(\mathbf{X}_1,\ldots,\mathbf{X}_m) \equiv \frac{1}{m}\sum_{k=1}^m \text{tr}\ (\mathbf{X}_k - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k - \bar{\mathbf{X}}_k), \quad (3.4)$$

then it will always be true that

$$\frac{1}{m}\sum_{k=1}^m \text{tr}\ (\mathbf{X}_k^+ - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k^+ - \bar{\mathbf{X}}_k) \le \frac{1}{m}\sum_{k=1}^m \text{tr}\ (\mathbf{X}_k^0 - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k^0 - \bar{\mathbf{X}}_k), \quad (3.5)$$

and therefore that

$$f(\mathbf{X}_1^+,\ldots,\mathbf{X}_m^+) \le g(\mathbf{X}_1^+,\ldots,\mathbf{X}_m^+;\mathbf{X}_1^0,\ldots,\mathbf{X}_m^0) \le f(\mathbf{X}_1^0,\ldots,\mathbf{X}_m^0), \quad (3.6)$$

or, in short, that

$$f(\mathbf{X}_1^+,\ldots,\mathbf{X}_m^+) \le f(\mathbf{X}_1^0,\ldots,\mathbf{X}_m^0). \quad (3.7)$$

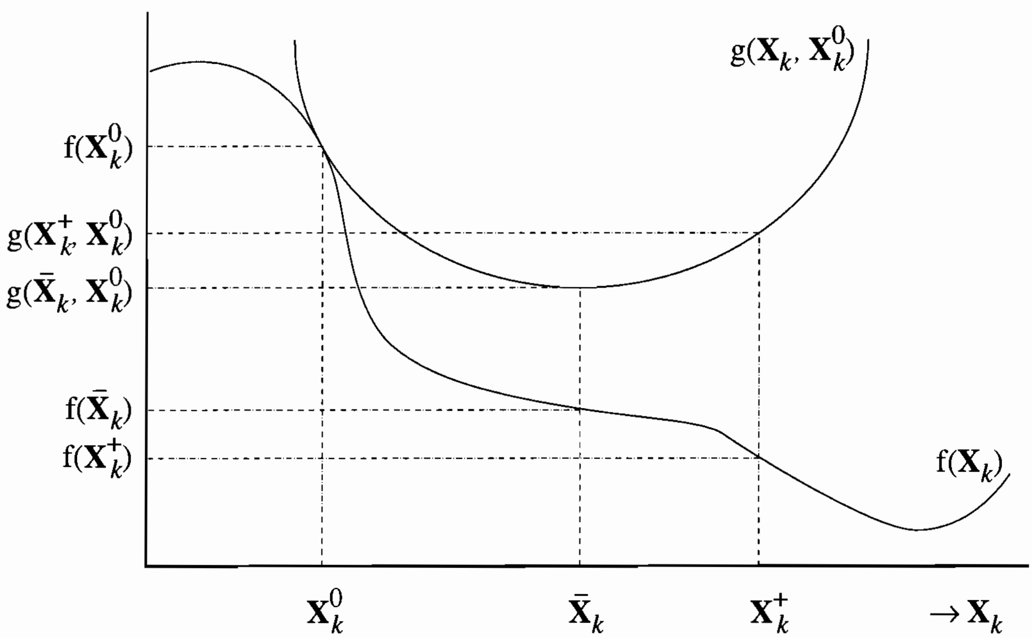

Figure 2 Illustration of majorization method for restricted matrices $\mathbf{X}_k$.

In the literature, the minimization of (3.4) is called a *metric projection problem*, since it involves the projection of the matrices $\bar{\mathbf{X}}_k$ on a restricted solution space in the metric $\mathbf{V}_k$. It follows from (3.7) that convergence of the algorithm is still guaranteed if the updates are required to be restricted, as long as a solution for (3.4) is available. This is illustrated in Figure 2.

Summarizing, the minimization of (1.1) over restricted (n x p) matrices $\mathbf{X}_k$ may be solved by applying the following algorithm:

1. choose initial matrices $\mathbf{X}_k^0$ satisfying the required constraints, and evaluate

$$f(\mathbf{X}_1^0,\ldots,\mathbf{X}_m^0) = \frac{1}{m}\sum_{k=1}^{m}\sum_{i<j}^{n} w_{ijk}[\delta_{ijk} - d_{ij}(\mathbf{X}_k^0)]^2;$$

2. compute unrestricted updates $\bar{\mathbf{X}}_k = \mathbf{V}_k^-\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$ for $k = 1, \ldots, m$;

3. solve metric projection problem (3.4), that is, find $\mathbf{X}_k^+$ minimizing

$$h(\mathbf{X}_1,\ldots,\mathbf{X}_m) = \frac{1}{m}\sum_{k=1}^{m} \operatorname{tr}\,(\mathbf{X}_k - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k - \bar{\mathbf{X}}_k);$$

4. replace $\mathbf{X}_k^0$ with $\mathbf{X}_k^+$ and evaluate

$$f(\mathbf{X}_1^0,\ldots,\mathbf{X}_m^0) = \frac{1}{m}\sum_{k=1}^{m}\sum_{i<j}^{n} w_{ijk}[\delta_{ijk} - d_{ij}(\mathbf{X}_k^0)]^2;$$

5. go to step 2 if the difference in loss between the current and the previous iteration is larger than some predefined criterion; otherwise stop.

As we already mentioned, the determination of 'smart' initial matrices $\mathbf{X}_k^0$ satisfying the constraints in step 1 of the algorithm will be taken up in section 9.

The above algorithm for the determination of restricted MDS solutions is called a *model* algorithm, blatantly vague as it is on how to solve the metric projection problem in step 3. This step has been kept vague on purpose, however, because the solution completely depends on the *kind of restrictions* that we are interested in. Therefore, the next sections will be devoted to the derivation of specific algorithms from the above model algorithm, which provide solutions to the specific restrictions needed in PROXSCAL.

4 Specific model restrictions in PROXSCAL

As discussed in the introduction, the specific restrictions used in PROXSCAL are of the type $\mathbf{X}_k = \mathbf{Z}\mathbf{A}_k$, where both the common space $\mathbf{Z}$ and the weight matrices $\mathbf{A}_k$ may be further restricted. Substituting $\mathbf{X}_k = \mathbf{Z}\mathbf{A}_k$ in the metric projection problem (3.4) we obtain

$$\mathrm{h}(\mathbf{Z};\mathbf{A}_1,\ldots,\mathbf{A}_m) = \frac{1}{m}\sum_{k=1}^{m} \mathrm{tr}\ (\mathbf{Z}\mathbf{A}_k - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{Z}\mathbf{A}_k - \bar{\mathbf{X}}_k)$$

$$= \frac{1}{m}\{\sum_{k=1}^{m} \mathrm{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + \sum_{k=1}^{m} \mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{V}_k\mathbf{Z}\mathbf{A}_k - 2\sum_{k=1}^{m} \mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{V}_k\bar{\mathbf{X}}_k\}$$

$$= \mathrm{c} + \frac{1}{m}\{\sum_{k=1}^{m} \mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{V}_k\mathbf{Z}\mathbf{A}_k - 2\sum_{k=1}^{m} \mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{V}_k\mathbf{V}_k^{-}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\}$$

$$= \mathrm{c} + \frac{1}{m}\{\sum_{k=1}^{m} \mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{V}_k\mathbf{Z}\mathbf{A}_k - 2\sum_{k=1}^{m} \mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\}, \quad (4.1)$$

where

$$\mathrm{c} = \frac{1}{m}\sum_{k=1}^{m} \mathrm{tr}\ \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k \quad (4.2)$$

is a constant with respect to $\mathbf{Z}$ and $\mathbf{A}_k$. Therefore, in PROXSCAL the metric projection problem in step 3 of the algorithm model of section 3 consists of the determination of estimates for the matrices $\mathbf{Z}$ and $\mathbf{A}_k$ which minimize (4.1). Having obtained such estimates, restricted updates for the matrices $\mathbf{X}_k$ in (2.1) are found by setting $\mathbf{X}_k^+ = \mathbf{Z}\mathbf{A}_k$ for $k = 1, \ldots, m$.

Before discussing how to solve (4.1), we first note that, except for the IDENTITY model where $\mathbf{A}_k = \mathbf{I}$ for $k = 1, \ldots, m$, the model $\mathbf{X}_k = \mathbf{Z}\mathbf{A}_k$ is not uniquely identified. Specifically, letting $\mathbf{T}$ be an arbitrary nonsingular matrix of order (p x p), nothing changes in the GENERALIZED and REDUCED models if we perform the following transformations

$$\mathbf{X}_k = \mathbf{Z}\mathbf{T}\mathbf{T}^{-1}\mathbf{A}_k = \bar{\mathbf{Z}}\bar{\mathbf{A}}_k \quad (4.3)$$

where

$$\bar{\mathbf{Z}} \equiv \mathbf{Z}\mathbf{T} \quad (4.4a)$$

$$\bar{\mathbf{A}}_k \equiv \mathbf{T}^{-1}\mathbf{A}_k. \tag{4.4b}$$

In the WEIGHTED model the same is true for an arbitrary diagonal matrix **T** of order (p x p). In PROXSCAL, the following identification condition is used

$$\frac{1}{m}\sum_{k=1}^{m} \mathbf{A}_k\mathbf{A}_k' = \mathbf{I}, \tag{4.5}$$

a condition that simplifies calculations considerably in a number of special cases to be discussed below.

For the GENERALIZED and the REDUCED models, condition (4.5) in PROXSCAL is satisfied by letting

$$\frac{1}{m}\sum_{k=1}^{m} \mathbf{A}_k\mathbf{A}_k' = \mathbf{TT}' \tag{4.6}$$

be a Cholesky factorization of the sum of squares and cross-products of the matrices $\mathbf{A}_k$, where **T** is a lower-triangular matrix, and then applying definitions (4.4a) and (4.4b). The result satifies (4.5) since

$$\frac{1}{m}\sum_{k=1}^{m} \bar{\mathbf{A}}_k\bar{\mathbf{A}}_k' = \frac{1}{m}\sum_{k=1}^{m} \mathbf{T}^{-1}\mathbf{A}_k\mathbf{A}_k'(\mathbf{T}^{-1})' = \mathbf{T}^{-1}(\frac{1}{m}\sum_{k=1}^{m} \mathbf{A}_k\mathbf{A}_k')(\mathbf{T}^{-1})' = \mathbf{T}^{-1}\mathbf{TT}'(\mathbf{T}^{-1})' = \mathbf{I}.$$

Another way to proceed is to define the eigenvalue decomposition

$$\frac{1}{m}\sum_{k=1}^{m} \mathbf{A}_k\mathbf{A}_k' = \mathbf{QL}^2\mathbf{Q}',$$

and then use $\mathbf{T} = \mathbf{QL}$ in (4.4a) and (4.4b), since the resulting matrices then also satisfy

$$\frac{1}{m}\sum_{k=1}^{m} \bar{\mathbf{A}}_k\bar{\mathbf{A}}_k' = \frac{1}{m}\sum_{k=1}^{m} \mathbf{T}^{-1}\mathbf{A}_k\mathbf{A}_k'(\mathbf{T}^{-1})' =$$

$$\mathbf{L}^{-1}\mathbf{Q}'(\frac{1}{m}\sum_{k=1}^{m} \mathbf{A}_k\mathbf{A}_k')\mathbf{QL}^{-1} = \mathbf{L}^{-1}\mathbf{Q}'\mathbf{QL}^2\mathbf{Q}'\mathbf{QL}^{-1} = \mathbf{I}.$$

The reason that a Cholesky factorization is used instead of an eigenvalue decomposition for identification condition (4.5) in the GENERALIZED and REDUCED models in PROXSCAL is that the former method is numerically more stable, and more efficient (cf., Wilkinson, 1965).

In the WEIGHTED model, identification (4.5) is simply achieved by a unit normalization of the dimension weights for each separate dimension.

It may be noted that (4.5) is not always adhered to in PROXSCAL, at least not during iterations. The situations where adherence to (4.5) is useful will be explicitly indicated in the following sections, which provide concrete solutions for the minimization of (4.1) for each model separately.

4.1 Generalized Euclidean model

In this section the situation is discussed where the matrices $\mathbf{A}_k$ in $\mathbf{X}_k = \mathbf{Z}\mathbf{A}_k$ are required to be full-rank matrices of order (p x p). This is the well-known IDIOSCAL model, as can be seen from the following observations. Letting $\mathbf{A}_k = \mathbf{P}_k\mathbf{L}_k\mathbf{Q}_k'$ be a singular value decomposition of $\mathbf{A}_k$, it follows from $\mathbf{X}_k = \mathbf{Z}\mathbf{A}_k$ that the GENERALIZED model may be written as $\mathbf{X}_k\mathbf{Q}_k = \mathbf{Z}\mathbf{P}_k\mathbf{L}_k$. Since distances are invariant under rotations and reflections it is true that $d_{ij}(\mathbf{X}_k\mathbf{Q}_k) = d_{ij}(\mathbf{X}_k)$, and $\mathbf{X}_k$ in loss function (1.1) can be replaced by $\mathbf{X}_k\mathbf{Q}_k$ without affecting the function value. Therefore, we may as well write the GENERALIZED model as $\mathbf{X}_k = \mathbf{Z}\mathbf{P}_k\mathbf{L}_k$, yielding the IDIOSCAL model where the common space $\mathbf{Z}$ is first rotated by $\mathbf{P}_k$ and the axes of $\mathbf{Z}\mathbf{P}_k$ are then weighted differentially by the diagonal matrix $\mathbf{L}_k$.

We will now discuss how to minimize (4.1) under the GENERALIZED model. Considered as a function of $\mathbf{Z}$ alone, the metric projection problem may be written as

$$h(\mathbf{z}) = c + \mathbf{z}'\mathbf{H}\mathbf{z} - 2\,\mathbf{z}'\mathbf{t} \qquad (4.1.1)$$

where

$$\mathbf{z} \equiv \text{vec}\,(\mathbf{Z}), \qquad (4.1.2a)$$

$$\mathbf{H} \equiv \frac{1}{m}\sum_{k=1}^{m} (\mathbf{A}_k\mathbf{A}_k' \otimes \mathbf{V}_k), \qquad (4.1.2b)$$

$$\mathbf{t} \equiv \text{vec}\,(\frac{1}{m}\sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'), \qquad (4.1.2c)$$

and vec ( ) denotes the vector obtained by stacking the columns of its argument beneath each other, and $\otimes$ denotes the right Kronecker product. Loss function (4.1.1) has stationary equations

$$\mathbf{Hz} = \mathbf{t}, \tag{4.1.3}$$

and a necessary and sufficient condition for a solution of (4.1.3) is that

$$\mathbf{HH}^{-}\mathbf{t} = \mathbf{t}, \tag{4.1.4}$$

where $\mathbf{H}^{-}$ denotes the MP-inverse of $\mathbf{H}$. Condition (4.1.4) is equivalent to saying that the vector $\mathbf{t}$ must be in the column space of the singular matrix $\mathbf{H}$. If (4.1.4) holds true, the general solution of (4.1.3) is

$$\mathbf{z} = \mathbf{H}^{-}\mathbf{t} + (\mathbf{I} - \mathbf{H}^{-}\mathbf{H})\mathbf{q}, \tag{4.1.5}$$

where $\mathbf{q}$ is an arbitrary vector of appropriate order. Therefore, one solution is

$$\mathbf{z} = \mathbf{H}^{-}\mathbf{t} = [\frac{1}{m}\sum_{k=1}^{m}(\mathbf{A}_k\mathbf{A}_k' \otimes \mathbf{V}_k)]^{-}[\text{vec}\,(\frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^{0\prime})\mathbf{X}_k^{0}\mathbf{A}_k')]. \tag{4.1.6}$$

Although the determination of the MP-inverse of (4.1.2b) can be reduced to the determination of a proper inverse by using

$$\{[\frac{1}{m}\sum_{k=1}^{m}(\mathbf{A}_k\mathbf{A}_k' \otimes \mathbf{V}_k)] + \mathbf{N}\}^{-1} - \mathbf{N}, \tag{4.1.7}$$

where $\mathbf{N}$ is the nullspace of (4.1.2b), this still is a problem of the order (np x np), which returns *every iteration*.

In the special case where $\mathbf{V}_1 = \mathbf{V}_2 = \ldots = \mathbf{V}_m = \mathbf{V}$, (4.1.6) may be written as

$$\text{vec}\,(\mathbf{Z}) = [(\frac{1}{m}\sum_{k=1}^{m}\mathbf{A}_k\mathbf{A}_k') \otimes \mathbf{V}]^{-}[\text{vec}\,(\frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^{0\prime})\mathbf{X}_k^{0}\mathbf{A}_k')], \tag{4.1.8}$$

which still requires the calculation of the MP-inverse of a full (np x np) matrix. However, in this special case the identification condition (4.5) comes in handy, since (4.1.8) then simplifies into

$$\text{vec}\,(\mathbf{Z}) = [\mathbf{I} \otimes \mathbf{V}]^{-}[\text{vec}\,(\frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k')],$$

and therefore into

$$\mathbf{Z} = \mathbf{V}^{-}\frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'. \qquad (4.1.9)$$

Since matrix $\mathbf{V}$ is constant its MP-inverse only has to be determined once; this MP-inverse can be computed according to (2.19).

In the situation where $w_{ijk} = 1$ for all i, j and k, (4.1.6) simplifies into

$$\mathbf{Z} = \frac{1}{n}(\frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k')(\frac{1}{m}\sum_{k=1}^{m}\mathbf{A}_k\mathbf{A}_k')^{-1}. \qquad (4.1.11)$$

If the identification condition (4.5) is used in this situation, matters simplify even further to

$$\mathbf{Z} = \frac{1}{nm}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'. \qquad (4.1.12)$$

Still, the general situation will be the one given in (4.1.6).

To avoid having to compute the inverse of an (np x np) matrix every iteration, Heiser and Stoop (1986) proposed the following *dimensionwise* approach to the minimization of (4.1) with respect to $\mathbf{Z}$. Letting $\mathbf{z}_a$ be the a-th column of $\mathbf{Z}$ ($a = 1,\ldots,p$), $\mathbf{e}_a$ be the a-th column of the identity matrix $\mathbf{I}_p$, and $\mathbf{P}_a$ be the (n x p) matrix equal to $\mathbf{Z}$ but with the a-th column containing zeroes, then it is true that

$$\mathbf{Z} \quad = \quad \mathbf{P}_a \quad + \quad \mathbf{z}_a\mathbf{e}_a'. \qquad (4.1.13)$$

$$\begin{bmatrix} & | & | & | & \\ \mathbf{z}_1 & | \; \mathbf{z}_2 & | \; \ldots & | \; \mathbf{z}_p \\ & | & | & | & \\ & | & | & | & \end{bmatrix} = \begin{bmatrix} & |\,0\,| & & | \\ \mathbf{z}_1 & |\,0\,| & \ldots & | \; \mathbf{z}_p \\ & |\,\vdots\,| & & | \\ & |\,0\,| & & | \end{bmatrix} + \begin{bmatrix} 0 \,| & & | & & |\,0 \\ 0 \,| & \mathbf{z}_a & | & \ldots & |\,0 \\ \vdots \,| & & | & & |\,\vdots \\ 0 \,| & & | & & |\,0 \end{bmatrix}$$

Substituting (4.1.13) in (4.1) and defining $\mathbf{C}_k = \mathbf{A}_k\mathbf{A}_k'$ we obtain

$$h(\mathbf{z}_a;*) = c + \frac{1}{m}\sum_{k=1}^{m} \text{tr}\,(\mathbf{P}_a + \mathbf{z}_a\mathbf{e}_a')'\mathbf{V}_k(\mathbf{P}_a + \mathbf{z}_a\mathbf{e}_a')\mathbf{C}_k$$

$$- 2\,\text{tr}\,(\mathbf{P}_a + \mathbf{z}_a\mathbf{e}_a')'\,\frac{1}{m}\sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'$$

$$= c^* + \mathbf{z}_a'(\frac{1}{m}\sum_{k=1}^{m} \mathbf{V}_k\mathbf{e}_a'\mathbf{C}_k\mathbf{e}_a)\mathbf{z}_a + 2\,\mathbf{z}_a'\frac{1}{m}\sum_{k=1}^{m} \mathbf{V}_k\mathbf{P}_a\mathbf{C}_k\mathbf{e}_a$$

$$- 2\,\mathbf{z}_a'\frac{1}{m}\sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'\mathbf{e}_a, \tag{4.1.14}$$

where the term

$$c^* \equiv c + \frac{1}{m}\sum_{k=1}^{m} \text{tr}\;\mathbf{P}_a'\mathbf{V}_k\mathbf{P}_a\mathbf{C}_k - 2\,\text{tr}\;\mathbf{P}_a'\frac{1}{m}\sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' \tag{4.1.15}$$

is independent of $\mathbf{z}_a$. Therefore, defining

$$\bar{\mathbf{x}}_a = \frac{1}{m}\sum_{k=1}^{m} [\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' - \mathbf{V}_k\mathbf{P}_a\mathbf{C}_k]\mathbf{e}_a \tag{4.1.16a}$$

and

$$\mathbf{V}_a = \frac{1}{m}\sum_{k=1}^{m} \mathbf{V}_k\mathbf{e}_a'\mathbf{C}_k\mathbf{e}_a, \tag{4.1.16b}$$

we may write (4.1.14) as

$$h(\mathbf{z}_a;*) = c^* + \mathbf{z}_a'\mathbf{V}_a\mathbf{z}_a - 2\,\mathbf{z}_a'\bar{\mathbf{x}}_a. \tag{4.1.17}$$

Using similar arguments as in (4.1.1), if

$$\mathbf{V}_a\mathbf{V}_a^-\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_a, \tag{4.1.18}$$

where $\mathbf{V}_a^-$ is the MP-inverse of $\mathbf{V}_a$, then the minimum of (4.1.17) is attained where

$$\mathbf{z}_a = \mathbf{V}_a^-\bar{\mathbf{x}}_a. \tag{4.1.19}$$

Alternatively, letting $\mathbb{x}_a$ be any vector satisfying

$$\bar{\mathbf{x}}_a = \mathbf{V}_a \mathbb{x}_a, \tag{4.1.20}$$

and substituting (4.1.20) in (4.1.17) yields

$$\begin{aligned} h(\mathbf{z}_a;*) &= c^* + \mathbf{z}_a'\mathbf{V}_a\mathbf{z}_a - 2\,\mathbf{z}_a'\bar{\mathbf{x}}_a \\ &= c^* - \mathbb{x}_a'\mathbf{V}_a\mathbb{x}_a + \mathbf{z}_a'\mathbf{V}_a\mathbf{z}_a + \mathbb{x}_a'\mathbf{V}_a\mathbb{x}_a - 2\,\mathbf{z}_a'\mathbf{V}_a\mathbb{x}_a \\ &= c^* - \mathbb{x}_a'\mathbf{V}_a\mathbb{x}_a + (\mathbf{z}_a - \mathbb{x}_a)'\mathbf{V}_a(\mathbf{z}_a - \mathbb{x}_a). \end{aligned} \tag{4.1.21}$$

Obviously, the global minimum of (4.1.21) is also obtained for (4.1.19).

Comparing (4.1.19) with (4.1.6) the determination of the inverse is reduced from a (np x np) problem to a (n x n) problem, to be solved p times. On the other hand, computing (4.1.19) for $a = 1, \ldots, p$ does not yield the conditional global minimum of (4.1) with respect to $\mathbf{Z}$; an iterative procedure has to be used until the values of $\mathbf{Z}$ stabilize. Moreover, for each dimension we still need to compute the inverse of the (n x n) matrix $\mathbf{V}_a$ in (4.1.16b), and this has to be done in every iteration. Therefore, in section 5 a procedure is discussed which disposes of the need to determine inverses, and again uses majorization to minimize (4.1.21).

For the sake of completeness, we note that in the situation where $w_{ijk} = 1$ for all i, j, and k, it is true that $\mathbf{V}_k = n\mathbf{J}$, meaning that (4.1.16a) and (4.1.16b) simplify into

$$\bar{\mathbf{x}}_a = \frac{1}{m}\sum_{k=1}^{m} [\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' - n\mathbf{J}\mathbf{P}_a\mathbf{C}_k]\mathbf{e}_a \tag{4.1.21a}$$

and

$$\mathbf{V}_a = n\mathbf{J}\;\mathbf{e}_a'[\frac{1}{m}\sum_{k=1}^{m}\mathbf{C}_k]\mathbf{e}_a, \tag{4.1.21b}$$

respectively. Therefore, letting

$$d_a = n\;\mathbf{e}_a'[\frac{1}{m}\sum_{k=1}^{m}\mathbf{C}_k]\mathbf{e}_a, \tag{4.1.21c}$$

(4.1.17) may be written in this situation as

$$h(\mathbf{z}_a;*) = c^* + \mathbf{z}_a'\mathbf{V}_a\mathbf{z}_a - 2\,\mathbf{z}_a'\bar{\mathbf{x}}_a$$

$$= c^* + d_a\,\mathbf{z}_a'\mathbf{J}\mathbf{z}_a - 2\,d_a\,\mathbf{z}_a'[\frac{1}{d_a}\bar{\mathbf{x}}_a] + \frac{1}{d_a}\bar{\mathbf{x}}_a'\bar{\mathbf{x}}_a - \frac{1}{d_a}\bar{\mathbf{x}}_a'\bar{\mathbf{x}}_a$$

$$= c^* - \frac{1}{d_a}\bar{\mathbf{x}}_a'\bar{\mathbf{x}}_a + d_a\,\{\mathbf{z}_a'\mathbf{J}\mathbf{z}_a + \frac{1}{d_a^2}\bar{\mathbf{x}}_a'\bar{\mathbf{x}}_a - 2\,\mathbf{z}_a'[\frac{1}{d_a}\bar{\mathbf{x}}_a]\}$$

$$= c\sim + d_a\,(\mathbf{z}_a - \frac{1}{d_a}\bar{\mathbf{x}}_a)'\mathbf{J}(\mathbf{z}_a - \frac{1}{d_a}\bar{\mathbf{x}}_a), \qquad (4.1.21d)$$

where c~ is another constant with respect to $\mathbf{z}_a$. Clearly, the global minimum of (4.1.21d) is attained where

$$\mathbf{z}_a = \frac{1}{d_a}\bar{\mathbf{x}}_a = \frac{\frac{1}{m}\sum_{k=1}^{m}[\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' - n\mathbf{J}\mathbf{P}_a\mathbf{C}_k]\mathbf{e}_a}{n\,\mathbf{e}_a'[\frac{1}{m}\sum_{k=1}^{m}\mathbf{C}_k]\mathbf{e}_a}, \qquad (4.1.21e)$$

and neither an inverse nor majorization are needed to obtain dimensionwise updates for the columns of the common space $\mathbf{Z}$. If identification condition (4.5) is applied to the matrices $\mathbf{A}_k$, it is true that $\frac{1}{m}\Sigma_k\mathbf{C}_k = \mathbf{I}$, and (4.1.21e) simplifies into

$$\mathbf{z}_a = \frac{1}{nm}\sum_{k=1}^{m}[\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k']\mathbf{e}_a, \qquad (4.1.21f)$$

which is nicely confirmed by (4.1.12).

We could also majorize (4.1.1) itself. Letting $\mathbb{t}$ be any vector satisfying

$$\mathbf{t} = \mathbf{H}\mathbb{t} \qquad (4.1.22)$$

and substituting (4.1.22) in (4.1.1) we obtain

$$h(\mathbf{z}) = c + \mathbf{z}'\mathbf{H}\mathbf{z} - 2\,\mathbf{z}'\mathbf{t}$$

$$= c - \mathbb{t}'\mathbf{H}\mathbb{t} + \mathbf{z}'\mathbf{H}\mathbf{z} + \mathbb{t}'\mathbf{H}\mathbb{t} - 2\,\mathbf{z}'\mathbf{H}\mathbb{t}$$

$$= c - \mathbb{t}'\mathbf{H}\mathbb{t} + (\mathbf{z} - \mathbb{t})'\mathbf{H}(\mathbf{z} - \mathbb{t}). \qquad (4.1.23)$$

Loss function (4.1.23) can be majorized using the procedure discussed in section 5. This procedure requires the calculation of the largest eigenvalue of the (*np* x *np*) matrix **H** in every iteration (or an estimation thereof). Therefore, the dimensionwise majorization approach (4.1.21) to the minimization of (4.1.1) is probably much more efficient than the majorization approach (4.1.23) applied simultaneously to all dimensions of **Z**.

Sometimes, it will be necessary to determine a *constrained* common space **Z**. How to obtain an update for **Z** in these cases will be taken up separately in sections 6 and 7.

To minimize (4.1) with respect to unrestricted $\mathbf{A}_k$ we may consider (4.1) as a function of one $\mathbf{A}_k$ only, which may then be written as

$$\mathrm{h}(\mathbf{A}_k) = \mathrm{d}_k + \frac{1}{m}\{\mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{V}_k\mathbf{Z}\mathbf{A}_k - 2\ \mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\}$$

$$= \mathrm{e}_k + \frac{1}{m} \| (\mathbf{Z}'\mathbf{V}_k\mathbf{Z})^{1/2}\mathbf{A}_k - (\mathbf{Z}'\mathbf{V}_k\mathbf{Z})^{-1/2}\mathbf{Z}'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \|^2, \qquad (4.1.25)$$

where d_k and e_k are constants with respect to $\mathbf{A}_k$. Clearly, (4.1.25) is globally minimized for

$$\mathbf{A}_k = (\mathbf{Z}'\mathbf{V}_k\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0, \qquad (4.1.26)$$

which is to be repeated *m* times (i.e., for each $\mathbf{A}_k$). In a number of cases, it is useful to apply identification condition (4.5) to the thus obtained matrices $\mathbf{A}_k$.

Summarizing the above, for the generalized Euclidean model the metric projection problem has to be solved by using alternating least squares, where we alternatingly minimize (4.1.21) for $a = 1, \ldots, p$ (either with (4.1.19), or by using majorization) keeping the matrices $\mathbf{A}_k$ fixed, and then compute (4.1.26) for $k = 1, \ldots, m$, keeping **Z** fixed. We may either continue to alternate between these two steps until some convergence criterion is met, or decide to make only one or a few steps in the right direction.

It is of some interest to note that the metric projection problem under the GENERALIZED model can be solved *analytically* in the special situation where the weights matrices $\mathbf{V}_k$ are equal to each other or where the weights w_{ijk} are all equal to one. This solution, which guarantees global minimization of (4.1), is based on the following observations. Since the weights matrices $\mathbf{V}_k$ are equal to each other, and considering (4.1) as a function of **Z** alone, the latter may be written as

$$\mathrm{h}(\mathbf{Z};*) = \mathrm{c} + \mathrm{tr}\ \mathbf{Z}'\mathbf{V}\mathbf{Z}\frac{1}{m}\sum_{k=1}^{m}\mathbf{A}_k\mathbf{A}_k' - 2\ \mathrm{tr}\ \mathbf{Z}'\frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'$$

$$= \mathrm{c} + \mathrm{tr}\,\mathbf{Z'VZC} - 2\,\mathrm{tr}\,\mathbf{Z'D}, \quad (4.1.27)$$

where $\mathbf{C} \equiv \frac{1}{m} \Sigma_k \mathbf{A}_k\mathbf{A}_k'$ and $\mathbf{D} \equiv \frac{1}{m} \Sigma_k \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'$. Letting $\mathbb{D} = \mathbf{V}^-\mathbf{D}$, and on the condition that $\mathbf{C}$ satisfies identification constraint (4.5), that is, $\mathbf{C} = \mathbf{I}$, (4.1.27) may again be written as

$$\mathrm{h}(\mathbf{Z}) = \mathrm{c} - \mathrm{tr}\,\mathbb{D}'\mathbf{V}\mathbb{D} + \mathrm{tr}\,(\mathbf{Z} - \mathbb{D})'\mathbf{V}(\mathbf{Z} - \mathbb{D}), \quad (4.1.28)$$

which is clearly globally minimized for (see also 4.1.9)

$$\mathbf{Z} = \mathbb{D} = \mathbf{V}^-\mathbf{D} = \mathbf{V}^- \frac{1}{m} \Sigma_k \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'.$$

The conditional global minimum of the majorizing function at this point is equal to

$$\mathrm{h}(\mathbf{A}_1,\ldots,\mathbf{A}_m) = \mathrm{c} - \mathrm{tr}\,\mathbb{D}'\mathbf{V}\mathbb{D}$$

$$= \mathrm{c} - \mathrm{tr}\,(\frac{1}{m} \Sigma_k \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k')'\mathbf{V}^-(\frac{1}{m} \Sigma_k \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'), \quad (4.1.29)$$

and the common space $\mathbf{Z}$ has been eliminated from the majorizing function.

The remaining problem is how to minimize (4.1.29). Defining $\mathbf{T}_k = \frac{1}{m}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$ and the partitioned matrix $\mathbf{T} = [\mathbf{T}_1 \mid \mathbf{T}_2 \mid \ldots \mid \mathbf{T}_m]$ of order $(n \times mp)$, and the partitioned matrix $\mathbf{A}' = [\mathbf{A}_1 \mid \mathbf{A}_2 \mid \ldots \mid \mathbf{A}_m]$ where $\mathbf{A}$ is of order $(mp \times p)$, the minimization of (4.1.29) under the constraint $\frac{1}{m} \Sigma_k \mathbf{A}_k\mathbf{A}_k' = \mathbf{I}$ is equivalent to the maximization of

$$\mathrm{b}(\mathbf{A}) \equiv \mathrm{tr}\,(\frac{1}{m} \Sigma_k \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k')'\mathbf{V}^-(\frac{1}{m} \Sigma_k \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k')$$

$$= \mathrm{tr}\,(\Sigma_k \mathbf{T}_k\mathbf{A}_k')'\mathbf{V}^-(\Sigma_k \mathbf{T}_k\mathbf{A}_k')$$

$$= \mathrm{tr}\,\mathbf{A'T'V^-TA} \quad (4.1.30)$$

under the restriction

$$\frac{1}{m}\mathbf{A'A} = \mathbf{I}. \quad (4.1.31)$$

The solution is standard. If $\mathbf{KLK}'$ is an eigenvalue decomposition of $\mathbf{T'V^-T}$, and letting $\mathbf{A}^* = \frac{1}{\sqrt{m}}\mathbf{A}$, then for any $\mathbf{A}$ satisfying (4.1.31), and thus for any $\mathbf{A}^*$ satisfying $\mathbf{A}^{*\prime}\mathbf{A}^* = \mathbf{I}$, it is true that

$$\mathrm{b}(\mathbf{A}) = \mathrm{tr}\,\mathbf{A'KLK'A} = m\,\mathrm{tr}\,\mathbf{K'A^*A^{*\prime}KL} = m\,\mathrm{tr}\,\mathbf{ML} \leq m\,\mathrm{tr}\,\mathbf{L}, \quad (4.1.32)$$

because $\mathbf{M} \equiv \mathbf{K'A^*A^{*\prime}K}$ is semi-orthonormal, being the product of orthonormal matrices $\mathbf{K}$ and semi-orthonormal matrices $\mathbf{A}^*$. The upper bound in (4.1.32) is attained when

$$\mathbf{M} = \begin{bmatrix} \mathbf{I}_p & 0 \\ 0 & 0 \end{bmatrix}.$$

Defining $\mathbf{K}_p$ as the $(mp \times p)$ matrix containing the p principal eigenvectors of $\mathbf{K}$, and letting

$$\mathbf{A}^* = \mathbf{K}_p\mathbf{N}, \tag{4.1.33}$$

where $\mathbf{N}$ is an arbitrary $(p \times p)$ orthonormal matrix, then

$$\mathbf{M} = \mathbf{K'A^*A^{*\prime}K} = \mathbf{K'K}_p\mathbf{NN'K}_p'\mathbf{K} = \begin{bmatrix} \mathbf{I}_p & 0 \\ 0 & 0 \end{bmatrix}.$$

This proves that $\mathbf{A} \equiv \sqrt{m}\, \mathbf{K}_p\mathbf{N}$ globally maximizes (4.1.30) under constraint (4.1.31), which it clearly satisfies. The optimal common space can always be determined using

$$\mathbf{Z} = \mathbf{V}^- \frac{1}{m} \Sigma_k\, \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' = \mathbf{V}^-\mathbf{TA}, \tag{4.1.34}$$

after the optimal $\mathbf{A}$ has been computed. This common space as well as the matrices $\mathbf{A}_k$ automatically satisfy the identification condition (4.5).

If the weights w_{ijk} are all equal to one, then $\mathbf{V} = n\mathbf{J}$ and (4.1.30) simplifies into

$$b(\mathbf{A}) = \frac{1}{n} \operatorname{tr} \mathbf{A'T'TA}, \tag{4.1.35}$$

and a completely analogous analytical solution for the minimization of majorization function (4.1) is available in this case also. The common space may then be calculated afterwards from

$$\mathbf{Z} = \mathbf{TA}. \tag{4.1.36}$$

Obviously, these analytical solutions can only be applied if no special restrictions are imposed upon the common space. Otherwise, the approaches discussed in sections 6 and 7 have to be used.

4.2 Weighted Euclidean model

If the matrices $\mathbf{A}_k$ are restricted to be diagonal we are dealing with the INDSCAL model. In that case the conditional minimum of (4.1), that is, of

$$\mathrm{h}(\mathbf{Z};\mathbf{A}_1,\ldots,\mathbf{A}_m) = \frac{1}{m}\sum_{k=1}^{m} \mathrm{tr}\ (\mathbf{Z}\mathbf{A}_k - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{Z}\mathbf{A}_k - \bar{\mathbf{X}}_k)$$

$$= \mathrm{c} + \frac{1}{m}\sum_{k=1}^{m} \mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{V}_k\mathbf{Z}\mathbf{A}_k - 2\frac{1}{m}\sum_{k=1}^{m} \mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$$

with respect to $\mathbf{Z}$ is still attained for

$$\mathrm{vec}\ (\mathbf{Z}) = \mathbf{H}^{-}\mathbf{t} = [\frac{1}{m}\sum_{k=1}^{m} (\mathbf{A}_k\mathbf{A}_k' \otimes \mathbf{V}_k)]^{-}[\mathrm{vec}\ (\frac{1}{m}\sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k')],$$

(see also (4.1.6)), but since the matrices $\mathbf{A}_k$ are diagonal this expression simplifies into

$$\mathrm{vec}\ (\mathbf{Z}) = \mathbf{H}^{-}\mathbf{t} = [\frac{1}{m}\sum_{k=1}^{m} (\mathbf{A}_k^2 \otimes \mathbf{V}_k)]^{-}[\mathrm{vec}\ (\frac{1}{m}\sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k)], \tag{4.2.1}$$

where $\mathbf{H}$ is a block-diagonal matrix, and $\mathbf{Z}$ may now therefore be solved for dimension after dimension.

We have the following special cases. If $\mathbf{V}_1 = \mathbf{V}_2 = \ldots = \mathbf{V}_m = \mathbf{V}$, (4.2.1) may be written as

$$\mathrm{vec}\ (\mathbf{Z}) = [(\frac{1}{m}\sum_{k=1}^{m} \mathbf{A}_k^2) \otimes \mathbf{V}]^{-}[\mathrm{vec}\ (\frac{1}{m}\sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k)], \tag{4.2.2}$$

meaning that if identification condition (4.5) is used (4.2.2) simplifies into

$$\mathbf{Z} = \mathbf{V}^{-}\frac{1}{m}\sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k. \tag{4.2.3}$$

If $\mathrm{w}_{ijk} = 1$ for all i, j and k then, since $\mathbf{V}^{-} = \frac{1}{n}\mathbf{J}^{-} = \frac{1}{n}\mathbf{J}$ in this case, (4.2.2) simplifies into

$$\mathbf{Z} = \frac{1}{n}(\frac{1}{m}\sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k)(\frac{1}{m}\sum_{k=1}^{m} \mathbf{A}_k^2)^{-1}, \tag{4.2.4}$$

and if, moreover, the identification condition (4.5) is used, matters simplify even further to

$$\mathbf{Z} = \frac{1}{nm} \sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k. \tag{4.2.5}$$

The computation of inverses in (4.2.1) and (4.2.3) can again be avoided by applying the dimensionwise majorization approach discussed for the GENERALIZED model in section 4.1 to the WEIGHTED situation. Specifically, we then minimize (4.1.21), that is,

$$\mathrm{h}(\mathbf{z}_a;*) = \mathrm{c}^* - \mathbf{x}_a'\mathbf{V}_a\mathbf{x}_a + (\mathbf{z}_a - \mathbf{x}_a)'\mathbf{V}_a(\mathbf{z}_a - \mathbf{x}_a)$$

dimension by dimension. In the WEIGHTED case, $\mathbf{x}_a$ is a vector satisfying $\bar{\mathbf{x}}_a = \mathbf{V}_a\mathbf{x}_a$ with

$$\bar{\mathbf{x}}_a = \frac{1}{m} \sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k\mathbf{e}_a, \tag{4.2.5a}$$

and

$$\mathbf{V}_a = \frac{1}{m} \sum_{k=1}^{m} \mathrm{a}_{ak}^2\mathbf{V}_k, \tag{4.2.5b}$$

where $\mathbf{e}_a$ is the a-th column of the identity matrix $\mathbf{I}_p$ and a_{ak} is the space weight for dimension a of source k.

Considering only one $\mathbf{A}_k$, (4.1) may be written as

$$\mathrm{h}(*;\mathbf{A}_k) = \mathrm{d}_k + \frac{1}{m}\{\mathrm{tr}\ \mathbf{A}_k^2\mathbf{Z}'\mathbf{V}_k\mathbf{Z} - 2\ \mathrm{tr}\ \mathbf{A}_k\mathbf{Z}'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\}$$

$$= \mathrm{d}_k + \frac{1}{m}\{\mathrm{tr}\ \mathbf{A}_k^2\ \mathrm{diag}(\mathbf{Z}'\mathbf{V}_k\mathbf{Z}) - 2\ \mathrm{tr}\ \mathbf{A}_k\ \mathrm{diag}(\mathbf{Z}'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0)\}$$

$$= \mathrm{e}_k + \frac{1}{m} \| \mathbf{A}_k\ \mathrm{diag}(\mathbf{Z}'\mathbf{V}_k\mathbf{Z})^{1/2} - \mathrm{diag}(\mathbf{Z}'\mathbf{V}_k\mathbf{Z})^{-1/2}\ \mathrm{diag}(\mathbf{Z}'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0) \|^2, \tag{4.2.6}$$

where d_k and e_k are constants with respect to $\mathbf{A}_k$. Therefore, the global minimum of (4.2.6) is attained for

$$\mathbf{A}_k = \mathrm{diag}(\mathbf{Z}'\mathbf{V}_k\mathbf{Z})^{-1}\ \mathrm{diag}(\mathbf{Z}'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0). \tag{4.2.7}$$

These dimension weights can be computed individually as

$$\mathrm{a}_{ak} = \frac{\mathbf{z}_a'\mathbf{b}_{ak}}{\mathbf{z}_a'\mathbf{V}_k\mathbf{z}_a}, \tag{4.2.8}$$

where $\mathbf{b}_{ak}$ and $\mathbf{z}_a$ are column a of matrices $\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$ and $\mathbf{Z}$, respectively, and a_{ak} is the weight for dimension a of source k. In contrast with the GENERALIZED model, no difficulties arise in the calculation of the dimension weights in (4.2.8) when $\mathbf{Z}$ is a singular matrix.

Summarizing, the solution of the metric projection problem for the weighted Euclidean model also requires alternating least squares, where the calculation of (4.2.1) is alternated with the computation of (4.2.7) for $k = 1, \ldots, m$.

However, just as in the GENERALIZED model an analytical solution of (4.1) under the WEIGHTED model is available when $\mathbf{V}_1 = \mathbf{V}_2 = \ldots = \mathbf{V}_m = \mathbf{V}$ and when $\mathrm{w}_{ijk} = 1$ for all i, j and k. In the situation that the weight matrices are equal, and assuming that the matrices $\mathbf{A}_k$ satisfy the identification condition (4.5), the common space $\mathbf{Z}$ can be eliminated from (4.1) (see also section 4.1), reducing the problem to the maximization of

$$\mathrm{b}(\mathbf{A}_1,\ldots,\mathbf{A}_m) \equiv \mathrm{tr}\,(\frac{1}{m}\,\Sigma_k\,\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k)'\mathbf{V}^-(\frac{1}{m}\,\Sigma_k\,\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k). \tag{4.2.9}$$

Defining $\mathbf{B}_a$ as the (n x m) matrix containing all a-th columns of the m matrices $\frac{1}{m}\,\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$, and $\mathbf{a}_a$ as the (m x 1) vector containing all dimension weights on dimension a, (4.2.9) may be written as

$$\mathrm{b}(\mathbf{a}_1,\ldots,\mathbf{a}_m) = \sum_{a=1}^{p} \mathbf{a}_a'\mathbf{B}_a'\mathbf{V}^-\mathbf{B}_a\mathbf{a}_a, \tag{4.2.10}$$

which must be maximized subject to

$$\frac{1}{m}\,\mathbf{a}_a'\mathbf{a}_a = 1 \text{ for } a = 1, \ldots, p. \tag{4.2.11}$$

This may be solved dimension after dimension: letting $\mathbf{p}_{1a}$ denote the principal eigenvector in the eigenvalue decomposition $\mathbf{B}_a'\mathbf{V}^-\mathbf{B}_a = \mathbf{P}_a\mathbf{L}_a\mathbf{P}_a'$, (4.2.10) is globally maximized subject to (4.2.11) where

$$\mathbf{a}_a = \sqrt{m}\,\mathbf{p}_{1a} \text{ for } a = 1, \ldots, p. \tag{4.2.12}$$

The proof is analogous to the one given for the GENERALIZED model (cf., section 4.1). When $\mathrm{w}_{ijk} = 1$ for all i, j and k, exactly the same procedure can be used, except that $\mathbf{V} = n\mathbf{J}$ in this case.

4.3 Reduced rank model

In this model we require that the $(p \times p)$ matrices $\mathbf{A}_k$ are of rank $r < p$. The rank must be specified in advance by the user. In the GENERALIZED model $\{p(n\text{-}1) + mp^2\}$ parameters are estimated, while the REDUCED rank model consumes $\{p(n\text{-}1) + mr^2\}$ parameters. Therefore, in comparison with the GENERALIZED model, the REDUCED rank model has the advantage that it allows for individual difference scaling in a relatively high number of dimensions (as indicated by p) without the risk of overparameterization. The number of free parameters is kept low by choosing r small enough ($r = 2$, for example).

Letting

$$\mathbf{A}_k = \mathbf{P}_k\mathbf{L}_k\mathbf{Q}_k' \tag{4.3.1}$$

be a singular value decomposition of $\mathbf{A}_k$ where $\mathbf{L}_k$ is a matrix of singular values in nonincreasing order on its diagonal, let $\mathbf{R}_k$ denote the $(p \times r)$ matrix containing the first r columns of $\mathbf{P}_k\mathbf{L}_k$ and $\mathbf{S}_k$ denote the $(p \times r)$ matrix containing the first r columns of $\mathbf{Q}_k$. Then the best rank r approximation of $\mathbf{A}_k$ is given by $\mathbf{R}_k\mathbf{S}_k'$ with $\mathbf{S}_k'\mathbf{S}_k = \mathbf{I}$, but $\mathbf{S}_k\mathbf{S}_k' \neq \mathbf{I}$. Since the corresponding model is

$$\mathbf{X}_k = \mathbf{Z}\mathbf{R}_k\mathbf{S}_k' \tag{4.3.2}$$

and therefore

$$\mathbf{X}_k\mathbf{S}_k = \mathbf{Z}\mathbf{R}_k \tag{4.3.3}$$

and because $\mathbf{X}_k\mathbf{X}_k' = \mathbf{Z}\mathbf{R}_k\mathbf{S}_k'\mathbf{S}_k\mathbf{R}_k'\mathbf{Z}' = \mathbf{Z}\mathbf{R}_k\mathbf{R}_k'\mathbf{Z}' = \mathbf{X}_k\mathbf{S}_k\mathbf{S}_k'\mathbf{X}_k'$, it follows that

$$\mathrm{d}_{ij}(\mathbf{X}_k) = \mathrm{d}_{ij}(\mathbf{X}_k\mathbf{S}_k) \tag{4.3.4}$$

for all i, j, and k. As explained in Heiser and Stoop (1986), this can be used to reduce the majority of the calculations for the reduced rank model in PROXSCAL from p to r dimensions.

The procedure is as follows. Letting

$$\underline{\mathbf{X}}_k = \mathbf{X}_k\mathbf{S}_k, \tag{4.3.5}$$

of order $(n \times r)$ and since

$$\mathrm{d}_{ij}(\underline{\mathbf{X}}_k) = \mathrm{d}_{ij}(\mathbf{X}_k), \tag{4.3.6}$$

the general loss function (1.1) may be written as

$$f(\mathbf{X}_1, \ldots, \mathbf{X}_m) = \frac{1}{m} \sum_{k=1}^{m} \sum_{i<j}^{n} w_{ijk}[\delta_{ijk} - d_{ij}(\mathbf{X}_k)]^2$$

$$= f(\underline{\mathbf{X}}_1, \ldots, \underline{\mathbf{X}}_m) = \frac{1}{m} \sum_{k=1}^{m} \sum_{i<j}^{n} w_{ijk}[\delta_{ijk} - d_{ij}(\underline{\mathbf{X}}_k)]^2$$

$$= \frac{1}{m} \sum_{k=1}^{m} [\, c_k + \mathrm{tr}\, \underline{\mathbf{X}}_k'\mathbf{V}_k\underline{\mathbf{X}}_k - 2\, \mathrm{tr}\, \underline{\mathbf{X}}_k'\mathbf{B}(\underline{\mathbf{X}}_k)\underline{\mathbf{X}}_k \,], \tag{4.3.7}$$

where c_k is defined as in (2.2), $\mathbf{V}_k$ is defined as in (2.3), and $\mathbf{B}(\underline{\mathbf{X}}_k) = \{b_{ijk}\}$ is defined as in (2.4), except that $d_{ij}(\mathbf{X}_k)$ in (2.4) is replaced everywhere with $d_{ij}(\underline{\mathbf{X}}_k)$.

Analogously to section 2, a convergent algorithm for the minimization of (4.3.7) over unrestricted matrices $\underline{\mathbf{X}}_k$ is obtained by repeatedly calculating the *r*-dimensional Guttman update defined as

$$\overline{\underline{\mathbf{X}}}_k = \mathbf{V}_k^-\mathbf{B}(\underline{\mathbf{X}}_k^0)\underline{\mathbf{X}}_k^0, \tag{4.3.8}$$

and restricted updates for $\underline{\mathbf{X}}_k$ are generally found by solving the metric projection problem

$$h(\underline{\mathbf{X}}_1^+,\ldots,\underline{\mathbf{X}}_m^+) = \frac{1}{m} \sum_{k=1}^{m} \mathrm{tr}\, (\underline{\mathbf{X}}_k^+ - \overline{\underline{\mathbf{X}}}_k)'\mathbf{V}_k(\underline{\mathbf{X}}_k^+ - \overline{\underline{\mathbf{X}}}_k). \tag{4.3.9}$$

Since the restrictions in the reduced rank model are $\underline{\mathbf{X}}_k = \mathbf{Z}\mathbf{R}_k$, this can be written as

$$h(\mathbf{Z};\mathbf{R}_1,\ldots,\mathbf{R}_m) = \frac{1}{m} \sum_{k=1}^{m} \mathrm{tr}\, (\mathbf{Z}\mathbf{R}_k - \overline{\underline{\mathbf{X}}}_k)'\mathbf{V}_k(\mathbf{Z}\mathbf{R}_k - \overline{\underline{\mathbf{X}}}_k)$$

$$= c + \frac{1}{m} \sum_{k=1}^{m} \mathrm{tr}\, \mathbf{R}_k'\mathbf{Z}'\mathbf{V}_k\mathbf{Z}\mathbf{R}_k - 2\frac{1}{m} \sum_{k=1}^{m} \mathrm{tr}\, \mathbf{R}_k'\mathbf{Z}'\mathbf{B}(\underline{\mathbf{X}}_k^0)\underline{\mathbf{X}}_k^0, \tag{4.3.10}$$

with corresponding solutions

$$\mathrm{vec}\,(\mathbf{Z}) = [\frac{1}{m} \sum_{k=1}^{m} (\mathbf{R}_k\mathbf{R}_k' \otimes \mathbf{V}_k)]^- [\mathrm{vec}\,(\frac{1}{m} \sum_{k=1}^{m} \mathbf{B}(\underline{\mathbf{X}}_k^0)\underline{\mathbf{X}}_k^0\mathbf{R}_k')], \tag{4.3.11}$$

and

$$\mathbf{R}_k = (\mathbf{Z}'\mathbf{V}_k\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{B}(\underline{\mathbf{X}}_k^0)\underline{\mathbf{X}}_k^0 \quad \text{for } k = 1, \ldots, m. \tag{4.3.12}$$

Alternatingly calculating (4.3.11) and (4.3.12) yields a convergent algorithm for the minimization of the metric projection problem.

In the same special cases as discussed for the GENERALIZED model, computations during iterations can be simplified for the REDUCED model by applying the identification condition (4.5), that is,

$$\frac{1}{m}\sum_{k=1}^{m} \mathbf{A}_k\mathbf{A}_k' = \mathbf{I}.$$

Combining the latter with (4.3.1) and the corresponding definitions of $\mathbf{R}_k$ and $\mathbf{S}_k$ in the REDUCED model we find that

$$\frac{1}{m}\sum_{k=1}^{m} \mathbf{A}_k\mathbf{A}_k' = \frac{1}{m}\sum_{k=1}^{m} \mathbf{R}_k\mathbf{S}_k'\mathbf{S}_k\mathbf{R}_k' = \frac{1}{m}\sum_{k=1}^{m} \mathbf{R}_k\mathbf{R}_k', \tag{4.3.13}$$

which shows that the matrices $\mathbf{S}_k$ are not even required for the identification condition to be satisfied. Although the products $\mathbf{R}_k\mathbf{R}_k'$ are all rank deficient, the sum of these products will be nonsingular, meaning the sum can be made to satisfy (4.5) if necessary. However, in exceptional cases the situation may arise where $\mathbf{R}_1\mathbf{R}_1' = \ldots = \mathbf{R}_m\mathbf{R}_m'$ (for example, if all weight matrices $\mathbf{V}_k$ are equal, and the (dis)similarity matrices are also identical). Obviously, PROXSCAL should check for such situations and stop with an error message.

In the end, the matrices $\mathbf{S}_k$ are completely redundant. If, having obtained an optimal $\mathbf{Z}$ and optimal $\mathbf{R}_k$'s, someone would be interested in obtaining orthonormal matrices $\mathbf{S}_k$ of order (p x r) after convergence of the algorithm, then *any arbitrary orthonormal matrix* $\mathbf{S}_k$ will do. This can be proven as follows. Let $\mathbf{S}_k$ be an arbitrary columnwise orthonormal matrix of order (p x r) with $r < p$. Such a matrix has rank r by definition. Moreover, for all matrices of order (n x p) defined as

$$\mathbf{X}_k = \underline{\mathbf{X}}_k\mathbf{S}_k' \tag{4.3.14}$$

it will be true that

$$\mathbf{X}_k\mathbf{X}_k' = \underline{\mathbf{X}}_k\mathbf{S}_k'\mathbf{S}_k\underline{\mathbf{X}}_k' = \underline{\mathbf{X}}_k\underline{\mathbf{X}}_k', \tag{4.3.15}$$

and therefore that

$$d_{ij}(\mathbf{X}_k) = d_{ij}(\underline{\mathbf{X}}_k) = d_{ij}(\underline{\mathbf{X}}_k\mathbf{S}_k') = d_{ij}(\mathbf{Z}\mathbf{R}_k\mathbf{S}_k'). \qquad (4.3.16)$$

This means that we may replace $\underline{\mathbf{X}}_k$ with $\underline{\mathbf{X}}_k\mathbf{S}_k'$, and $\mathbf{Z}\mathbf{R}_k$ with $\mathbf{Z}\mathbf{R}_k\mathbf{S}_k'$ everywhere without altering anything to the solution.

4.4 Identity model

If $\mathbf{X}_1 = \mathbf{X}_2 = \ldots = \mathbf{X}_m = \mathbf{Z}$, i.e., if $\mathbf{A}_k = \mathbf{I}$ for $k = 1, \ldots, m$, then the general loss function (1.1) simplifies into

$$\mathrm{f}(\mathbf{Z}) = \frac{1}{m} \sum_{k=1}^{m} \sum_{i<j}^{n} \mathrm{w}_{ijk}[\delta_{ijk} - \mathrm{d}_{ij}(\mathbf{Z})]^2$$

$$= \frac{1}{m} \sum_{k=1}^{m} \mathrm{c}_k + \mathrm{tr}\ \mathbf{Z}'(\frac{1}{m} \sum_{k=1}^{m} \mathbf{V}_k)\mathbf{Z} - 2\ \mathrm{tr}\ \mathbf{Z}'\{\frac{1}{m} \sum_{k=1}^{m} \mathbf{B}_k(\mathbf{Z})\}\mathbf{Z}, \qquad (4.4.1)$$

where c_k is defined as in (2.2), $\mathbf{V}_k$ is defined as in (2.3), and $\mathbf{B}_k(\mathbf{Z}) = \{\mathrm{b}_{ijk}\}$ is defined by

$$\mathrm{b}_{ijk} = \begin{cases} \left.\begin{array}{l} -\mathrm{w}_{ijk}\delta_{ijk}/\mathrm{d}_{ij}(\mathbf{Z}) \\ 0 \quad \text{if } \mathrm{d}_{ij}(\mathbf{Z}) = 0 \end{array}\right\} \text{for } i \neq j \\ -\sum_{l \neq i}^{n} \mathrm{b}_{ilk} \qquad \text{for } i = j \end{cases} . \qquad (4.4.2)$$

Analogously to section 2, it is easily proved that a convergent algorithm for the minimization of (4.4.1) over unrestricted $\mathbf{Z}$ is obtained by repeatedly calculating the Guttman transform which is now defined as

$$\bar{\mathbf{Z}} = (\frac{1}{m} \sum_{k=1}^{m} \mathbf{V}_k)^{-}\{\frac{1}{m} \sum_{k=1}^{m} \mathbf{B}_k(\mathbf{Z}^0)\}\mathbf{Z}^0. \qquad (4.4.3)$$

Thus, the IDENTITY model disposes of the metric projection problem altogether.

However, if we additionally require $\mathbf{Z}$ to be restricted in some subspace, or if wish to avoid the computation of the inverse in (4.4.3), then a new update $\mathbf{Z}^+$ can be obtained by minimizing

$$\mathrm{h}(\mathbf{Z}) = \mathrm{tr}\ (\mathbf{Z} - \bar{\mathbf{Z}})'(\frac{1}{m} \sum_{k=1}^{m} \mathbf{V}_k)(\mathbf{Z} - \bar{\mathbf{Z}}), \qquad (4.4.4)$$

(with $\bar{\mathbf{Z}}$ defined as in (4.4.3)), the solution of which again depends on the restrictions imposed on $\mathbf{Z}$ (if any), and will be discussed in sections 5, 6 and 7.

5 Projecting the common space on a complicated subspace in the metric of a positive semidefinite matrix

The general problem discussed in this section is how to minimize the function

$$g(\mathbf{Z}) \equiv \text{tr } (\mathbf{Z} - \mathbf{X})'\mathbf{V}(\mathbf{Z} - \mathbf{X}), \tag{5.1}$$

where $\mathbf{V}$ is a given $(n \text{ x } n)$ positive semidefinite matrix, $\mathbf{X}$ is a given $(n \text{ x } p)$ matrix, and $\mathbf{Z}$ is an unknown $(n \text{ x } p)$ matrix, possibly restricted to some complicated subspace. A solution to (5.1) can be obtained by applying the following majorization method.

Substitution of the trivial expression

$$\mathbf{Z} = \mathbf{Z}^0 + (\mathbf{Z} - \mathbf{Z}^0), \tag{5.2}$$

where $\mathbf{Z}^0$ is an arbitrary $(n \text{ x } p)$ matrix, in (5.1) yields

$$g(\mathbf{Z}) = \text{tr } [(\mathbf{Z} - \mathbf{Z}^0) - (\mathbf{X} - \mathbf{Z}^0)]'\mathbf{V}[(\mathbf{Z} - \mathbf{Z}^0) - (\mathbf{X} - \mathbf{Z}^0)]$$

$$= \text{d} + \text{tr } (\mathbf{Z} - \mathbf{Z}^0)'\mathbf{V}(\mathbf{Z} - \mathbf{Z}^0) - 2 \text{ tr } (\mathbf{Z} - \mathbf{Z}^0)'\mathbf{V}(\mathbf{X} - \mathbf{Z}^0), \tag{5.3}$$

where $\text{d} \equiv \text{tr } (\mathbf{X} - \mathbf{Z}^0)'\mathbf{V}(\mathbf{X} - \mathbf{Z}^0)$ is a term independent of $\mathbf{Z}$. Define

$$\mathbf{V} = \mathbf{PKP}' \tag{5.4}$$

as an eigenvalue decomposition of matrix $\mathbf{V}$, and let k_1 denote the largest eigenvalue on the diagonal of $\mathbf{K}$. Then, for any arbitrary matrix $\mathbf{D}$ of order $(n \text{ x } m)$, and letting $\mathbf{U} = \mathbf{P}'\mathbf{D}$, it is true that

$$\text{tr } \mathbf{D}'\mathbf{PKP}'\mathbf{D} = \text{tr } \mathbf{U}'\mathbf{KU} = \text{tr } \mathbf{KUU}' = \text{tr } \mathbf{K}(\text{diag } \mathbf{UU}')$$

$$= k_1 \mathbf{u}_1'\mathbf{u}_1 + k_2 \mathbf{u}_2'\mathbf{u}_2 + \ldots + k_n \mathbf{u}_n'\mathbf{u}_n$$

$$\leq k_1 \mathbf{u}_1'\mathbf{u}_1 + k_1 \mathbf{u}_2'\mathbf{u}_2 + \ldots + k_1 \mathbf{u}_n'\mathbf{u}_n$$

$$= k_1 \text{ tr } (\text{diag } \mathbf{UU}') = k_1 \text{ tr } \mathbf{U}'\mathbf{U} = k_1 \text{ tr } \mathbf{D}'\mathbf{PP}'\mathbf{D} = k_1 \text{ tr } \mathbf{D}'\mathbf{D}, \tag{5.5}$$

where $\mathbf{u}_i$ denotes the i-the row of matrix $\mathbf{U}$ written as a column vector. It follows from (5.5) that

$$\text{tr } \mathbf{D}'\mathbf{VD} \leq k_1 \text{ tr } \mathbf{D}'\mathbf{D}, \tag{5.6}$$

for arbitrary matrix **D**. Substituting $\mathbf{D} = \mathbf{Z} - \mathbf{Z}^0$ in (5.6) we obtain the inequality

$$\text{tr}\,(\mathbf{Z} - \mathbf{Z}^0)'\mathbf{V}(\mathbf{Z} - \mathbf{Z}^0) \le k_1\, \text{tr}\,(\mathbf{Z} - \mathbf{Z}^0)'(\mathbf{Z} - \mathbf{Z}^0), \tag{5.7}$$

and defining

$$h(\mathbf{Z}, \mathbf{Z}^0) = d + k_1\, \text{tr}\,(\mathbf{Z} - \mathbf{Z}^0)'(\mathbf{Z} - \mathbf{Z}^0) - 2\, \text{tr}\,(\mathbf{Z} - \mathbf{Z}^0)'\mathbf{V}(\mathbf{X} - \mathbf{Z}^0) \tag{5.8}$$

it follows from (5.3), (5.7), and (5.8) that

$$g(\mathbf{Z}) \le h(\mathbf{Z}, \mathbf{Z}^0) \tag{5.9}$$

for any pair of matrices **Z** and $\mathbf{Z}^0$ of identical order. Furthermore, since $g(\mathbf{Z}) = h(\mathbf{Z}, \mathbf{Z}^0) = d$ if $\mathbf{Z} = \mathbf{Z}^0$, the function $h(\mathbf{Z}, \mathbf{Z}^0)$ majorizes the function $g(\mathbf{Z})$.

Expanding (5.8) results in

$$\begin{aligned} h(\mathbf{Z}, \mathbf{Z}^0) &= f + k_1\, \text{tr}\, \mathbf{Z}'\mathbf{Z} - 2\, k_1\, \text{tr}\, \mathbf{Z}'\mathbf{Z}^0 - 2\, \text{tr}\, \mathbf{Z}'\mathbf{V}(\mathbf{X} - \mathbf{Z}^0) \\ &= f + k_1\, \text{tr}\, \mathbf{Z}'\mathbf{Z} - 2\, k_1\, \text{tr}\, \mathbf{Z}'\{\mathbf{Z}^0 + \frac{1}{k_1}\mathbf{V}(\mathbf{X} - \mathbf{Z}^0)\}, \end{aligned} \tag{5.10}$$

where $f \equiv d + k_1\, \text{tr}\, \mathbf{Z}^{0\prime}\mathbf{Z}^0 + 2\, \text{tr}\, \mathbf{Z}^{0\prime}\mathbf{V}(\mathbf{X} - \mathbf{Z}^0)$ is a term independent of **Z**. Substitution of

$$\bar{\mathbf{Z}} \equiv \mathbf{Z}^0 + \frac{1}{k_1}\mathbf{V}(\mathbf{X} - \mathbf{Z}^0) = (\mathbf{I} - \frac{1}{k_1}\mathbf{V})\mathbf{Z}^0 + \frac{1}{k_1}\mathbf{V}\mathbf{X} \tag{5.11}$$

in (5.10) yields

$$\begin{aligned} h(\mathbf{Z}, \mathbf{Z}^0) &= f - k_1\, \text{tr}\, \bar{\mathbf{Z}}'\bar{\mathbf{Z}} + k_1\, \text{tr}\, \mathbf{Z}'\mathbf{Z} + k_1\, \text{tr}\, \bar{\mathbf{Z}}'\bar{\mathbf{Z}} - 2\, k_1\, \text{tr}\, \mathbf{Z}'\bar{\mathbf{Z}} \\ &= f - k_1\, \text{tr}\, \bar{\mathbf{Z}}'\bar{\mathbf{Z}} + k_1\, \text{tr}\, (\mathbf{Z} - \bar{\mathbf{Z}})'(\mathbf{Z} - \bar{\mathbf{Z}}), \end{aligned} \tag{5.12}$$

showing that, for fixed $\mathbf{Z}^0$, the global minimum of $h(\mathbf{Z}, \mathbf{Z}^0)$ is attained for

$$\mathbf{Z} = \bar{\mathbf{Z}} = \mathbf{Z}^0 + \frac{1}{k_1}\mathbf{V}(\mathbf{X} - \mathbf{Z}^0).$$

Due to (5.9) we also have that

$$g(\bar{\mathbf{Z}}) \le h(\bar{\mathbf{Z}}, \mathbf{Z}^0). \tag{5.13}$$

Substitution of $\mathbf{Z} = \bar{\mathbf{Z}}$ in (5.12) yields

$$h(\bar{\mathbf{Z}}, \mathbf{Z}^0) = f - k_1 \operatorname{tr} \bar{\mathbf{Z}}'\bar{\mathbf{Z}} \tag{5.14}$$

while

$$h(\mathbf{Z}^0, \mathbf{Z}^0) = f - k_1 \operatorname{tr} \bar{\mathbf{Z}}'\bar{\mathbf{Z}} + k_1 \operatorname{tr} (\mathbf{Z}^0 - \bar{\mathbf{Z}})'(\mathbf{Z}^0 - \bar{\mathbf{Z}}). \tag{5.15}$$

But since $\operatorname{tr} (\mathbf{Z}^0 - \bar{\mathbf{Z}})'(\mathbf{Z}^0 - \bar{\mathbf{Z}}) \geq 0$, it follows from (5.13) and (5.14) that

$$h(\bar{\mathbf{Z}}, \mathbf{Z}^0) \leq h(\mathbf{Z}^0, \mathbf{Z}^0). \tag{5.16}$$

Finally, combining (5.13) and (5.16) we get the chain

$$g(\bar{\mathbf{Z}}) \leq h(\bar{\mathbf{Z}}, \mathbf{Z}^0) \leq h(\mathbf{Z}^0, \mathbf{Z}^0) = g(\mathbf{Z}^0). \tag{5.17}$$

It follows from (5.17) that the minimization of (5.1) over *unrestricted* matrices $\mathbf{Z}$ is obtained by calculating (5.11).

To determine a *restricted* common space, let $\mathbf{Z}^0$ denote an arbitrary (n x p) matrix satisfying the constraints (whatever these are), $\bar{\mathbf{Z}} = \mathbf{Z}^0 + \frac{1}{k_1}\mathbf{V}(\mathbf{X} - \mathbf{Z}^0)$ denote the unrestricted update given the current $\mathbf{Z}^0$, and $\mathbf{Z}^+$ denote a new and better solution for $\mathbf{Z}$ satisfying the same constraints as $\mathbf{Z}^0$. Then it follows from (5.9) and (5.12) that

$$\begin{aligned} g(\mathbf{Z}^+) &\leq h(\mathbf{Z}^+, \mathbf{Z}^0) \\ &\leq f - k_1 \operatorname{tr} \bar{\mathbf{Z}}'\bar{\mathbf{Z}} + k_1 \operatorname{tr} (\mathbf{Z}^+ - \bar{\mathbf{Z}})'(\mathbf{Z}^+ - \bar{\mathbf{Z}}). \end{aligned} \tag{5.18}$$

It also follows from (5.12) that

$$h(\mathbf{Z}^0, \mathbf{Z}^0) = f - k_1 \operatorname{tr} \bar{\mathbf{Z}}'\bar{\mathbf{Z}} + k_1 \operatorname{tr} (\mathbf{Z}^0 - \bar{\mathbf{Z}})'(\mathbf{Z}^0 - \bar{\mathbf{Z}}). \tag{5.19}$$

This means that, if we are able to determine a new restricted common space $\mathbf{Z}^+$ which minimizes the function

$$q(\mathbf{Z}) \equiv \operatorname{tr} (\mathbf{Z} - \bar{\mathbf{Z}})'(\mathbf{Z} - \bar{\mathbf{Z}}), \tag{5.20}$$

it will be true that

$$\operatorname{tr} (\mathbf{Z}^+ - \bar{\mathbf{Z}})'(\mathbf{Z}^+ - \bar{\mathbf{Z}}) \leq \operatorname{tr} (\mathbf{Z}^0 - \bar{\mathbf{Z}})'(\mathbf{Z}^0 - \bar{\mathbf{Z}}), \tag{5.21}$$

and, therefore, that

$$g(\mathbf{Z}^+) \le h(\mathbf{Z}^+, \mathbf{Z}^0) \le h(\mathbf{Z}^0, \mathbf{Z}^0) = g(\mathbf{Z}^0). \tag{5.22}$$

Comparing the original problem (5.1) with (5.20), we see that the latter problem is much easier to solve because the projection is no longer in the metric **V**.

Summarizing, the minimization of (5.1) can be achieved with the following convergent algorithm:

1. determine the largest eigenvalue k_1 of matrix **V**, choose an initial $\mathbf{Z}^0$ satisfying the constraints, and evaluate $g(\mathbf{Z}^0)$ as in (5.1);
2. compute the unrestricted update $\bar{\mathbf{Z}} = \mathbf{Z}^0 + \frac{1}{k_1}\mathbf{V}(\mathbf{X} - \mathbf{Z}^0)$;
3. given $\bar{\mathbf{Z}}$ obtained in step 2, solve metric projection problem (5.20) yielding $\mathbf{Z}^+$ (obviously, the solution of (5.20) depends on the constraints required);
4. replace $\mathbf{Z}^0$ with $\mathbf{Z}^+$ and evaluate $g(\mathbf{Z}^0)$ as in (5.1);
5. go to step 2 if the difference in function value between the current and the previous iteration is larger than some predefined criterion; otherwise stop.

Instead of really calculating the largest eigenvalue k_1 of matrix **V** in step 1 of this algorithm, we may also use the following upper bound for k_1:

$$k_1 \le \frac{1}{(n-t)} \operatorname{tr} \mathbf{V} + \sqrt{\frac{(n-1-t)}{(n-t)} \operatorname{tr} \mathbf{V}^2 - \frac{(n-1-t)}{(n-t)^2} (\operatorname{tr} \mathbf{V})^2}, \tag{5.23}$$

where n is the number of rows of **V**, and t is the number of zero eigenvalues of **V** (see Wolkowicz and Styan, 1980). For instance, matrix $\mathbf{V}_k$ defined in (2.3) has rank $(n-1)$, i.e. one eigenvalue is equal to zero, meaning that a good estimate of the largest eigenvalue of this matrix is obtained as

$$\tilde{k}_1 = \frac{1}{(n-1)} \operatorname{tr} \mathbf{V} + \sqrt{\frac{(n-2)}{(n-1)} \operatorname{tr} \mathbf{V}^2 - \frac{(n-2)}{(n-1)^2} (\operatorname{tr} \mathbf{V})^2}. \tag{5.24}$$

In PROXSCAL, the common space is allowed to be restricted by two mutually exclusive types of constraints. Either the user may require that the common space is a linear combination of user-provided external variables. This case will be taken up in section 7. Or the user may require that two or more or even all objects in the common space have *fixed* *coordinates*, which must then be provided by the user. As an illustration, we will now show how the above convergent iterative procedure can be used to impose the latter type of constraint.

If some coordinates in the common space are required to remain fixed, then all we have to do is to adapt step 3 in the above algorithm in the following way. The formal problem is how to minimize (5.20) in this case, that is,

$$q(\mathbf{Z}) \equiv \text{tr}\,(\mathbf{Z} - \bar{\mathbf{Z}})'(\mathbf{Z} - \bar{\mathbf{Z}}),$$

where $\mathbf{Z}$ varies over the $(n \times p)$ matrices centered on the origin with some coordinates fixed. The solution is as follows. Set the free coordinates of $\mathbf{Z}$ equal to the corresponding coordinates of $\bar{\mathbf{Z}}$, and set the fixed coordinates of $\mathbf{Z}$ equal to the coordinates provided by the user. It may be noted that, if a dimensionwise approach is used to update the common space (as discussed for the GENERALIZED and WEIGHTED models, see sections 4.1 and 4.2), this procedure for fixing some coordinates can be used for each dimension $\mathbf{z}_a$ of $\mathbf{Z}$ $(a = 1, \ldots, p)$ separately. To guarantee that the new $\mathbf{Z}^+$ is also centered on the origin, leaving the fixed elements unchanged, we proceed as follows. Let $\mathbf{M}_a$ denote the diagonal matrix of order $(n \times n)$ with ones on its diagonal if the corresponding elements of $\mathbf{z}_a$ are fixed, and zeroes elsewhere. Then, clearly, $\mathbf{z}_a = \mathbf{M}_a\mathbf{z}_a + (\mathbf{I} - \mathbf{M}_a)\mathbf{z}_a$, $\mathbf{M}_a\mathbf{z}_a$ denoting the fixed elements of $\mathbf{z}_a$, and $(\mathbf{I} - \mathbf{M}_a)\mathbf{z}_a$ its free elements. To center $\mathbf{z}_a$ on the origin, leaving $\mathbf{M}_a\mathbf{z}_a$ unchanged, we have to substract

$$\mathbf{z}_a'\mathbf{1}/\mathbf{1}'(\mathbf{I} - \mathbf{M}_a)\mathbf{1} \tag{5.25}$$

from the free elements $(\mathbf{I} - \mathbf{M}_a)\mathbf{z}_a$. We now prove that the resulting vector $\tilde{\mathbf{z}}_a$ is centered on the origin, that is, that $\tilde{\mathbf{z}}_a'\mathbf{1} = 0$:

$$\tilde{\mathbf{z}}_a'\mathbf{1} = \{\mathbf{M}_a\mathbf{z}_a + (\mathbf{I} - \mathbf{M}_a)[\mathbf{z}_a - \frac{\mathbf{z}_a'\mathbf{1}}{\mathbf{1}'(\mathbf{I} - \mathbf{M}_a)\mathbf{1}}\mathbf{1}]\}'\mathbf{1}$$

$$= \mathbf{z}_a'\mathbf{1} - \frac{(\mathbf{z}_a'\mathbf{1})(\mathbf{1}'\mathbf{1})}{\mathbf{1}'(\mathbf{I} - \mathbf{M}_a)\mathbf{1}} + \frac{(\mathbf{z}_a'\mathbf{1})(\mathbf{1}'\mathbf{M}_a\mathbf{1})}{\mathbf{1}'(\mathbf{I} - \mathbf{M}_a)\mathbf{1}} = \mathbf{z}_a'\mathbf{1} - \mathbf{z}_a'\mathbf{1}\,\frac{\mathbf{1}'(\mathbf{I} - \mathbf{M}_a)\mathbf{1}}{\mathbf{1}'(\mathbf{I} - \mathbf{M}_a)\mathbf{1}} = 0.$$

In the next section, however, we discuss a different and possibly more efficient approach for the updating of the common space when some of its coordinates are required to remain fixed.

6 Updating the common space when it is partially known

In the previous section we indicated one way to determine an update for a common space whose coordinates are only partially free. Analogously to the main algorithm described in sections 2 and 3, this approach uses majorization and consists of repeatedly projecting an unrestricted update upon the required solution space. For the GENERALIZED model, for example, the procedure requires either the calculation of the largest eigenvalue of the potentially very large (*np* x *np*) matrix **H** defined in (4.1.2b), or the calculation of the largest eigenvalue of the $\mathbf{V}_a$ matrices defined in (4.1.16b).

Another possible solution is to apply the majorization approach proposed by Kiers (1990). Using our notation, an unrestricted update for **Z** in the GENERALIZED model loss function (4.1), that is, in

$$\mathrm{h}(\mathbf{Z}) = \mathrm{c} + \frac{1}{m} \{ \sum_{k=1}^{m} \mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{V}_k\mathbf{Z}\mathbf{A}_k - 2 \sum_{k=1}^{m} \mathrm{tr}\ \mathbf{A}_k'\mathbf{Z}'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \},$$

can be obtained with the approach of Kiers by calculating

$$\bar{\mathbf{Z}} = \mathbf{Z}^0 - \frac{1}{\sum_k \alpha_k} \Big(\Sigma_k\, \mathbf{V}_k\mathbf{Z}^0\mathbf{A}_k\mathbf{A}_k' - \Sigma_k\, \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' \Big),$$

where α_k denotes the product of the largest eigenvalues of the matrices $\mathbf{V}_k$ and $\mathbf{A}_k\mathbf{A}_k'$, respectively. Then, the restricted update $\mathbf{Z}^+$ is found by minimizing the function m(**Z**) = $\| \mathbf{Z} - \bar{\mathbf{Z}} \|^2$. This procedure requires the calculation of *m* largest eigenvalues of matrices $\mathbf{A}_k\mathbf{A}_k'$ of order (*p* x *p*) in every iteration, and of the *m* largest eigenvalues of fixed matrices $\mathbf{V}_k$ of order (*n* x *n*) which only has to be done once.

In this section, we discuss yet another procedure which does not require the calculation of any eigenvalues (or estimations thereof). Moreover, updates are determined which are at least globally optimal in a conditional sense. A further breakdown of the *dimensionwise* approach discussed in section 4.1 is used to set up a procedure where the common space is updated *coordinate by coordinate*. We will first develop the algebra involved in the GENERALIZED model, and then derive results for the other models as special cases.

In section 4.1, it has been shown that an unrestricted update for column *a* of matrix **Z** can be obtained by minimizing (4.1.17), that is,

$$\mathrm{h}(\mathbf{z}_a;*) = \mathrm{c}^* + \mathbf{z}_a'\bar{\mathbf{V}}_a\mathbf{z}_a - 2\, \mathbf{z}_a'\bar{\mathbf{x}}_a,$$

where $\mathbf{z}_a$ denotes the a-th column of $\mathbf{Z}$ ($a = 1, \ldots, p$), $\mathbf{e}_a$ is the a-th column of the identity matrix $\mathbf{I}_p$, c^* is a term independent of $\mathbf{z}_a$,

$$\bar{\mathbf{x}}_a = \frac{1}{m} \sum_{k=1}^{m} [\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' - \mathbf{V}_k\mathbf{P}_a\mathbf{A}_k\mathbf{A}_k']\mathbf{e}_a,$$

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

and $\mathbf{P}_a$ is the (n x p) matrix equal to $\mathbf{Z}$ but with the a-th column containing zeroes.

Analogously, defining z_{ia} as the i-th element of vector $\mathbf{z}_a$, $\mathbf{e}_i$ as the i-th column of the identity matrix $\mathbf{I}_n$, and $\tilde{\mathbf{z}}_{ia}$ as the (n x 1) vector equal to $\mathbf{z}_a$ except that its i-th element is equal to zero, the following relation holds:

$$\mathbf{z}_a = \tilde{\mathbf{z}}_{ia} + z_{ia}\mathbf{e}_i. \tag{6.1}$$

$$\begin{bmatrix} z_{1a} \\ z_{2a} \\ \vdots \\ z_{na} \end{bmatrix} = \begin{bmatrix} z_{1a} \\ 0 \\ \vdots \\ z_{na} \end{bmatrix} + \begin{bmatrix} 0 \\ z_{2a} \\ \vdots \\ 0 \end{bmatrix}$$

Substitution of (6.1) in (4.1.17) yields

$$\begin{aligned} h(z_{ia};*) &= c^* + (\tilde{\mathbf{z}}_{ia} + z_{ia}\mathbf{e}_i)'\bar{\mathbf{V}}_a(\tilde{\mathbf{z}}_{ia} + z_{ia}\mathbf{e}_i) - 2\,(\tilde{\mathbf{z}}_{ia} + z_{ia}\mathbf{e}_i)'\bar{\mathbf{x}}_a \\ &= d + \mathbf{e}_i'\bar{\mathbf{V}}_a\mathbf{e}_i\, z_{ia}^2 - 2\,\mathbf{e}_i'(\bar{\mathbf{x}}_a - \bar{\mathbf{V}}_a\tilde{\mathbf{z}}_{ia})z_{ia} \\ &= d^* + \mathbf{e}_i'\bar{\mathbf{V}}_a\mathbf{e}_i\,\Big(z_{ia} - \frac{\mathbf{e}_i'(\bar{\mathbf{x}}_a - \bar{\mathbf{V}}_a\tilde{\mathbf{z}}_{ia})}{\mathbf{e}_i'\bar{\mathbf{V}}_a\mathbf{e}_i}\Big)^2, \end{aligned} \tag{6.2}$$

where d and d^* are constant terms with respect to z_{ia}. Clearly, the global minimum of (6.2) is attained where

$$z_{ia}^+ = \frac{1}{\mathbf{e}_i'\bar{\mathbf{V}}_a\mathbf{e}_i}\,\mathbf{e}_i'(\bar{\mathbf{x}}_a - \bar{\mathbf{V}}_a\tilde{\mathbf{z}}_{ia}). \tag{6.3}$$

Because, as is easily verified,

$$\sum_{k=1}^{m} \mathbf{V}_k\mathbf{P}_a\mathbf{A}_k\mathbf{A}_k'\mathbf{e}_a = \sum_{j\neq a}^{p} \left(\sum_{k=1}^{m} \mathbf{e}_j'\mathbf{A}_k\mathbf{A}_k'\mathbf{e}_a\mathbf{V}_k \right)\mathbf{z}_j,$$

where $\mathbf{e}_j$ is the j-th column of $\mathbf{I}_p$, expressing (6.3) in terms of the original vectors and matrices yields

$$z_{ia}^{+} = \frac{1}{\mathbf{e}_i'\bar{\mathbf{V}}_a\mathbf{e}_i}\, \mathbf{e}_i'\left[\frac{1}{m}\sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'\mathbf{e}_a - \frac{1}{m}\sum_{j\neq a}^{p} \left(\sum_{k=1}^{m} \mathbf{e}_j'\mathbf{A}_k\mathbf{A}_k'\mathbf{e}_a\mathbf{V}_k \right)\mathbf{z}_j \right]$$

$$- \frac{1}{\mathbf{e}_i'\bar{\mathbf{V}}_a\mathbf{e}_i}\, \mathbf{e}_i'\bar{\mathbf{V}}_a\tilde{\mathbf{z}}_{ia} \qquad (6.4)$$

where

$$\bar{\mathbf{V}}_a = \frac{1}{m}\sum_{k=1}^{m} \mathbf{e}_a'\mathbf{A}_k\mathbf{A}_k'\mathbf{e}_a\mathbf{V}_k. \qquad (6.5)$$

With (6.4), the common space may be updated coordinate by coordinate, leaving the fixed coordinates unaffected. Since the calculation of (6.4) is conditional upon the current values of the remaining free coordinates of **Z**, updating all free coordinates will only locally minimize loss function (4.1) with respect to **Z**. If we want global minimization we have to cycle repeatedly through all free coordinates until some convergence criterion is reached. After all free coordinates have been updated (either once or repeatedly), the complete matrix **Z** is centered on the origin.

In the special situation where the fixed elements of **Z** all consist of complete rows, an alternative scheme can be set up where the common space is updated in a rowwise fashion. Letting $\mathbf{z}_r$ be the (p x 1) vector containing the r-th row of **Z** ($r = 1, \ldots, n$), $\mathbf{e}_r$ be the r-th column of the identity matrix $\mathbf{I}_n$, and $\mathbf{R}_r$ be the (n x p) matrix equal to **Z** but with the r-th row containing zeroes, then an update for the r-th row of **Z** is obtained by calculating

$$\mathbf{z}_r^{+} = \left(\frac{1}{m}\textstyle\sum_k \mathbf{e}_r'\mathbf{V}_k\mathbf{e}_r\mathbf{A}_k\mathbf{A}_k' \right)^{-1} \left[\frac{1}{m}\left(\sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' \right)' - \frac{1}{m}\sum_{k=1}^{m} \mathbf{A}_k\mathbf{A}_k'\mathbf{R}_r'\mathbf{V}_k \right]\mathbf{e}_r. \qquad (6.6)$$

We omit the proof, since it is completely analogous to the one given in section 4.1 for the dimensionwise update of the common space.

We will now discuss a number of special cases of (6.4). If the weight matrices $\mathbf{V}_k$ are equal to each other, matters would simplify considerably if we could apply identification condition (4.5), that is, the condition that $\frac{1}{m}\sum_k \mathbf{A}_k\mathbf{A}_k' = \mathbf{I}$. Unfortunately, this condition may not be used in the present case, since it involves a transformation of the fixed elements of $\mathbf{Z}$ which does not preserve their original relations. When the weight matrices are equal to each other, the update formula becomes

$$z_{ia}^{+} = \frac{1}{\mathbf{e}_i'\overline{\mathbf{V}}_a\mathbf{e}_i}\ \mathbf{e}_i'\Big[\frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'\mathbf{e}_a - \frac{1}{m}\sum_{j\neq a}^{p}\mathbf{e}_j'\Big(\sum_{k=1}^{m}\mathbf{A}_k\mathbf{A}_k'\Big)\mathbf{e}_a\mathbf{V}\mathbf{z}_j\Big] - \frac{1}{\mathbf{e}_i'\overline{\mathbf{V}}_a\mathbf{e}_i}\ \mathbf{e}_i'\overline{\mathbf{V}}_a\tilde{\mathbf{z}}_{ia}, \tag{6.7a}$$

where

$$\overline{\mathbf{V}}_a = \mathbf{e}_a'\Big(\frac{1}{m}\sum_{k=1}^{m}\mathbf{A}_k\mathbf{A}_k'\Big)\mathbf{e}_a\mathbf{V}. \tag{6.7b}$$

Clearly, in the WEIGHTED model the term $\mathbf{e}_j'\mathbf{A}_k\mathbf{A}_k'\mathbf{e}_a = \mathbf{e}_j'\mathbf{A}_k^2\mathbf{e}_a$ in (6.4) is equal to zero. As a result, the coordinatewise update formula (6.4) may then be written as

$$z_{ia}^{+} = \frac{1}{\mathbf{e}_i'\overline{\mathbf{V}}_a\mathbf{e}_i}\ \mathbf{e}_i'\Big[\frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k\mathbf{e}_a - \overline{\mathbf{V}}_a\tilde{\mathbf{z}}_{ia}\Big], \tag{6.8}$$

where $\overline{\mathbf{V}}_a$ defined in (6.5) simplifies into $\overline{\mathbf{V}}_a = \frac{1}{m}\sum_k \mathbf{e}_a'\mathbf{A}_k^2\mathbf{e}_a\mathbf{V}_k$.

In the IDENTITY model where $\mathbf{A}_k = \mathbf{I}$, the coordinatewise update may be written as

$$z_{ia}^{+} = \frac{1}{\mathbf{e}_i'\big(\frac{1}{m}\sum_k\mathbf{V}_k\big)\mathbf{e}_i}\ \mathbf{e}_i'\Big[\Big(\frac{1}{m}\sum_k\mathbf{B}_k(\mathbf{Z}^0)\Big)\mathbf{Z}^0\mathbf{e}_a - \Big(\frac{1}{m}\sum_k\mathbf{V}_k\Big)\tilde{\mathbf{z}}_{ia}\Big], \tag{6.9}$$

where $\mathbf{B}_k(\mathbf{Z}^0)$ is defined as in (4.4.2).

When we are only dealing with one source (i.e., one (dis)similarity matrix) the update formula further reduces to

$$x_{ia}^{+} = \frac{1}{\mathbf{e}_i'\mathbf{V}\mathbf{e}_i}\ \mathbf{e}_i'\big[\mathbf{B}(\mathbf{X}^0)\mathbf{X}^0\mathbf{e}_a - \mathbf{V}\tilde{\mathbf{x}}_{ia}^0\big]. \tag{6.10}$$

We finally consider the situation where there is one source and all weights w_{ij} in the corresponding STRESS loss function

$$f(\mathbf{X}) = \sum_{i<j}^{n} w_{ij}[\delta_{ij} - d_{ij}(\mathbf{X})]^2$$

are equal to one. In that case $\mathbf{V} = n\mathbf{J}$ where $\mathbf{J}$ is the centering matrix (see also section 2), meaning that $\mathbf{e}_i'\mathbf{V}\mathbf{e}_i = n$-1, and that (6.10) therefore simplifies into

$$x_{ia}^{+} = \frac{1}{n\text{-}1}\ \mathbf{e}_i'\mathbf{B}(\mathbf{X}^0)\mathbf{X}^0\mathbf{e}_a - \frac{n}{n\text{-}1}\ \mathbf{e}_i'\mathbf{J}\tilde{\mathbf{x}}_{ia}^{0}$$

$$= \frac{1}{n\text{-}1}\ [\mathbf{e}_i'\mathbf{B}(\mathbf{X}^0)\mathbf{X}^0\mathbf{e}_a + \mathbf{1}'\tilde{\mathbf{x}}_{ia}^{0}]. \qquad (6.11)$$

If we define $\mathbf{X}_{ia}^{+}$ as the $(n \times p)$ matrix equal to $\mathbf{X}^0$ except that element x_{ia} has been updated according to (6.11), then this matrix has the following interesting property

$$\mathbf{e}_i'\mathbf{J}\mathbf{X}_{ia}^{+}\mathbf{e}_a = \mathbf{e}_i'(\frac{1}{n}\mathbf{B}(\mathbf{X}^0)\mathbf{X}^0)\mathbf{e}_a. \qquad (6.12)$$

Proof. Expanding the left hand side of (6.12) we have

$$\mathbf{e}_i'\mathbf{J}\mathbf{X}_{ia}^{+}\mathbf{e}_a = \mathbf{e}_i'(\mathbf{I} - \frac{\mathbf{11}'}{n})\mathbf{X}_{ia}^{+}\mathbf{e}_a = \mathbf{e}_i'\mathbf{X}_{ia}^{+}\mathbf{e}_a - \mathbf{e}_i'\frac{\mathbf{11}'}{n}\mathbf{X}_{ia}^{+}\mathbf{e}_a$$

$$= x_{ia}^{+} - \frac{1}{n}(x_{ia}^{+} + \mathbf{1}'\tilde{\mathbf{x}}_{ia}^{0}) = \frac{n\text{-}1}{n}x_{ia}^{+} - \frac{1}{n}\mathbf{1}'\tilde{\mathbf{x}}_{ia}^{0}. \qquad (6.13)$$

Substitution of (6.11) in (6.13) yields

$$\mathbf{e}_i'\mathbf{J}\mathbf{X}_{ia}^{+}\mathbf{e}_a = \frac{n\text{-}1}{n}(\frac{1}{n\text{-}1}\ [\mathbf{e}_i'\mathbf{B}(\mathbf{X}^0)\mathbf{X}^0\mathbf{e}_a + \mathbf{1}'\tilde{\mathbf{x}}_{ia}^{0}]) - \frac{1}{n}\mathbf{1}'\tilde{\mathbf{x}}_{ia}^{0}$$

$$= \mathbf{e}_i'(\frac{1}{n}\mathbf{B}(\mathbf{X}^0)\mathbf{X}^0)\mathbf{e}_a.$$

This completes the proof.

In section 2, we discussed that $\frac{1}{n}\mathbf{B}(\mathbf{X}^0)\mathbf{X}^0$ is the Guttman transform formula for a simultaneous update of all coordinates of unrestricted $\mathbf{X}$. Since $\mathbf{e}_i'(\frac{1}{n}\mathbf{B}(\mathbf{X}^0)\mathbf{X}^0)\mathbf{e}_a$ is element *ia* of the Guttman transform, equality (6.12) nicely illustrates the close relationship between a simultaneous and a coordinatewise update of the unknown configuration $\mathbf{X}$. Specifically, (6.12) states that the coordinatewise update for object i on the a-th dimension is always identical to element *ia* of the simultaneous Guttman update, on the condition that matrix $\mathbf{X}_{ia}^{+}$ containing the new coordinate has been centered on the origin.

In PROXSCAL, the fixed elements of the (common) space are assumed to be fixed *up to an isotropic scaling factor*. Thus, ideally we would like to estimate such a factor for the fixed elements of the (common) space and then correct these elements for size. However, such a procedure would destroy the property that we require of the (common) space that it is centered on the origin. Therefore, to keep the space centered an isotropic scaling factor is computed for *all* (that is, both free and fixed) coordinates of the (common) space simultaneously. In the GENERALIZED, REDUCED and WEIGHTED models the calculation of such an overall scaling factor is not necessary, because the space weight matrices $\mathbf{A}_k$ then already take care of correction for size of the individual spaces $\mathbf{X}_k = \mathbf{Z}\mathbf{A}_k$. However, this is not the case in the IDENTITY model. Therefore, letting

$$\mathbf{V} = \frac{1}{m} \sum_{k=1}^{m} \mathbf{V}_k$$

and

$$\bar{\mathbf{Z}} = \mathbf{V}^{-}\{\frac{1}{m} \sum_{k=1}^{m} \mathbf{B}_k(\mathbf{Z}^0)\}\mathbf{Z}^0,$$

in the IDENTITY model an explicit scaling factor α is computed minimizing

$$\begin{aligned} \mathrm{h}(\alpha) &= \mathrm{tr}\ (\alpha\mathbf{Z} - \bar{\mathbf{Z}})'\mathbf{V}(\alpha\mathbf{Z} - \bar{\mathbf{Z}}) \\ &= \mathrm{c} + \alpha^2\ \mathrm{tr}\ \mathbf{Z}'\mathbf{V}\mathbf{Z} - 2\alpha\ \mathrm{tr}\ \mathbf{Z}'\mathbf{V}\bar{\mathbf{Z}}, \end{aligned} \tag{6.14}$$

where c is a term independent of α. Since (6.14) is globally minimized for

$$\alpha = \frac{\mathrm{tr}\ \mathbf{Z}'\mathbf{V}\bar{\mathbf{Z}}}{\mathrm{tr}\ \mathbf{Z}'\mathbf{V}\mathbf{Z}}, \tag{6.15}$$

the update for $\mathbf{Z}$ is corrected for size by computing $\mathbf{Z}^* = \{\frac{\mathrm{tr}\ \mathbf{Z}'\mathbf{V}\bar{\mathbf{Z}}}{\mathrm{tr}\ \mathbf{Z}'\mathbf{V}\mathbf{Z}}\}\mathbf{Z}$ when the configuration in the IDENTITY model is partially known.

7 Restricting the common space to be a linear combination of external variables

The PROXSCAL program allows the user to provide an (n x s) matrix **E** containing *external variables*, which are then used to constrain the common space to be a linear combination of these external variables. Since these variables can be depicted as directions in the common space, the ordering of the objects along these directions yields a direct substantive interpretation of the location of these same objects in p-dimensional space. While fitting external variables, PROXSCAL allows for different *measurement levels* of the variables, that is, the latter may be treated either as single numerical, ordinal, or nominal variables, or any mixture of these measurement levels. PROXSCAL also allows the external variables to contain *missing data*.

Here, it will be discussed how PROXSCAL proceeds to incorporate this type of constraints while still guaranteeing a convergent algorithm for all models. The point of departure will be how to take care of such constraints in the most general situation, that is for the GENERALIZED model with different weight matrices $\mathbf{V}_k$, and then the algebra in more simple circumstances will be derived as special cases. The following exposition is based on the paper by Meulman and Heiser (1984), and on unpublished material written by Heiser.

Surprisingly, the main problem is not how to set up a convergent algorithm under the restrictions at hand, but rather how to choose that alternative from the available options that combines efficiency and ease in calculation with an optimal convergence rate of the main algorithm. We may usefully distinguish two classes of solutions. The first class consists of the determination of an unrestricted update for the common space which is then projected upon the restricted solution space. This majorization approach is used in Meulman and Heiser (1984). The second type of solution proceeds by a direct implementation of the required restrictions on the common space in loss function (4.1). Examples of both classes of solutions will be given.

As an illustration of the first approach, consider the following procedure. In section 4.1, it was found that the problem of determining an unrestricted update for the common space in the GENERALIZED model formally consisted of the minimization of (4.1.1), that is, of

$$h(\mathbf{z}) = c + \mathbf{z}'\mathbf{H}\mathbf{z} - 2\,\mathbf{z}'\mathbf{t},$$

where c is a constant defined as in (4.2), $\mathbf{z} = \text{vec}\,(\mathbf{Z})$ of order (np x 1), **H** is an (np x np) matrix defined as in (4.1.2b), and **t** is a (temporarily) given vector defined as in (4.1.2c).

We also showed in section 4.1 that, letting vector $\mathbb{t}$ be a vector satisfying $\mathbf{t} = \mathbf{H}\mathbb{t}$, (4.1.1) may be written as (4.1.23), that is, as

$$\text{h}(\mathbf{z}) = \text{g} + (\mathbf{z} - \mathbb{t})'\mathbf{H}(\mathbf{z} - \mathbb{t}),$$

where g is another constant with respect to $\mathbf{z}$.

Applying the algebra of section 5 to (4.1.23), it is immediately clear that a convergent algorithm for the minimization of (4.1.23) over restricted vectors $\mathbf{z}$ is obtained by repeatedly using the following two-step procedure. First, compute an unrestricted update $\bar{\mathbf{z}}$ for the common space as

$$\bar{\mathbf{z}} = \mathbf{z}^0 + \frac{1}{\text{k}_1}\mathbf{H}(\mathbb{t} - \mathbf{z}^0)$$

$$= \mathbf{z}^0 + \frac{1}{\text{k}_1}(\mathbf{t} - \mathbf{H}\mathbf{z}^0), \tag{7.1}$$

where k_1 is the largest eigenvalue in the eigenvalue decomposition

$$\mathbf{H} = \mathbf{PKP}', \tag{7.2}$$

and $\mathbf{z}^0$ is the current solution for the common space satisfying the constraints. Then, determine a new restricted $\mathbf{z}^+$ by minimizing the function

$$\text{q}(\mathbf{z}) = (\mathbf{z} - \bar{\mathbf{z}})'(\mathbf{z} - \bar{\mathbf{z}}), \tag{7.3}$$

which may also be written as

$$\text{q}(\mathbf{Z}) = \text{tr}\,(\mathbf{Z} - \bar{\mathbf{Z}})'(\mathbf{Z} - \bar{\mathbf{Z}}), \tag{7.4}$$

over restricted matrices $\mathbf{Z}$.

If we assume for a moment that $p \leq s$, then requiring the common space to be a linear combination of the external variables $\mathbf{E}$ implies that $\mathbf{Z}$ must satisfy

$$\mathbf{Z} = \mathbf{EB} \tag{7.5}$$

where $\mathbf{B}$ is an $(s \times p)$ matrix of *regression weights*. Substituting (7.5) in (7.4) we have

$$\text{q}(\mathbf{B}) = \text{tr}\,(\mathbf{EB} - \bar{\mathbf{Z}})'(\mathbf{EB} - \bar{\mathbf{Z}}), \tag{7.6}$$

which, assuming $\mathbf{E}$ fixed for the moment, is clearly globally minimized for

$$\mathbf{B} = (\mathbf{E}'\mathbf{E})^{-1}\mathbf{E}'\bar{\mathbf{Z}}, \tag{7.7}$$

yielding

$$\mathbf{Z}^+ = \mathbf{E}(\mathbf{E}'\mathbf{E})^{-1}\mathbf{E}'\bar{\mathbf{Z}} \tag{7.8}$$

as the new restricted update for the common space $\mathbf{Z}$.

An important disadvantage of this procedure is that we have to store the potentially very large $(np \times np)$ matrix $\mathbf{H}$ in computer memory and calculate its largest eigenvalue. As an alternative, the majorization method proposed by Kiers (1990) can be used to restrict the common space to be a linear combination of external variables. This method requires the calculation of the largest eigenvalues of the matrix products $\mathbf{A}_k\mathbf{A}_k'$ as well as of the matrices $\mathbf{V}_k$ (see also section 6).

However, a restricted common space may also be obtained by directly implementing the constraints in metric projection problem (4.1). Specifically, let h = maximum(s, p), and let $\mathbf{Q}$ denote the $(n \times h)$ matrix containing the quantified external variables $\mathbf{E}$; if $p > s$, that is, if there are more dimensions than external variables, the first p columns of $\mathbf{Q}$ consist of the s quantified external variables, and the last $(p - s)$ columns of $\mathbf{Q}$ are treated as unrestricted elements. Then, it is possible to directly minimize

$$\text{h}(\mathbf{Q}; \mathbf{B}; *) = \frac{1}{m} \sum_{k=1}^{m} \text{tr}\ (\mathbf{QBA}_k - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{QBA}_k - \bar{\mathbf{X}}_k), \tag{7.9}$$

where $\bar{\mathbf{X}}_k = \mathbf{V}_k^-\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0$. We start by noting that the product $\mathbf{QB}$ may be written as the sum of rank-one matrices. Defining $\mathbf{q}_j$ $(j = 1, \ldots, h)$ as the $(n \times 1)$ vector containing column j of matrix $\mathbf{Q}$, and $\mathbf{b}_j$ as the $(h \times 1)$ vector containing row j of matrix $\mathbf{B}$, it is true that

$$\mathbf{QB} = \sum_{j=1}^{h} \mathbf{q}_j\mathbf{b}_j', \tag{7.10}$$

and loss function (7.9) may be written as

$$\text{h}(\mathbf{Q}; \mathbf{B}; *) = \frac{1}{m} \sum_{k=1}^{m} \text{tr}\ (\sum_{j=1}^{h} \mathbf{q}_j\mathbf{b}_j'\mathbf{A}_k - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\sum_{j=1}^{h} \mathbf{q}_j\mathbf{b}_j'\mathbf{A}_k - \bar{\mathbf{X}}_k). \tag{7.11}$$

Considering only one $\mathbf{b}_j$ and $\mathbf{q}_j$, and letting

$$\mathbf{U}_j = \sum_{t \neq j} \mathbf{q}_t \mathbf{b}_t', \tag{7.12}$$

we may write (7.11) as

$$\mathrm{h}(\mathbf{b}_j, \mathbf{q}_j) = \frac{1}{m} \sum_{k=1}^{m} \mathrm{tr}\ [\mathbf{q}_j \mathbf{b}_j' \mathbf{A}_k - (\bar{\mathbf{X}}_k - \mathbf{U}_j \mathbf{A}_k)]' \mathbf{V}_k [\mathbf{q}_j \mathbf{b}_j' \mathbf{A}_k - (\bar{\mathbf{X}}_k - \mathbf{U}_j \mathbf{A}_k)]$$

$$= \mathrm{c}_j + \frac{1}{m} \sum_{k=1}^{m} \mathbf{q}_j' \mathbf{V}_k \mathbf{q}_j \mathbf{b}_j' \mathbf{A}_k \mathbf{A}_k' \mathbf{b}_j - 2\ \mathbf{q}_j' \mathbf{T}_j \mathbf{b}_j, \tag{7.13}$$

where

$$\mathbf{T}_j \equiv \frac{1}{m} \sum_{k=1}^{m} \mathbf{V}_k (\bar{\mathbf{X}}_k - \mathbf{U}_j \mathbf{A}_k) \mathbf{A}_k' = \frac{1}{m} \sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0) \mathbf{X}_k^0 \mathbf{A}_k' - \frac{1}{m} \sum_{k=1}^{m} \mathbf{V}_k \mathbf{U}_j \mathbf{A}_k \mathbf{A}_k', \tag{7.14}$$

and c_j is independent of $\mathbf{b}_j$ and $\mathbf{q}_j$. For fixed vector $\mathbf{q}_j$, the global minimum of (7.13) is attained where

$$\mathbf{b}_j = \left(\frac{1}{m} \sum_{k=1}^{m} \mathbf{q}_j' \mathbf{V}_k \mathbf{q}_j \mathbf{A}_k \mathbf{A}_k'\right)^{-1} \mathbf{T}_j' \mathbf{q}_j. \tag{7.15}$$

For fixed $\mathbf{b}_j$, (7.13) may be written as

$$\mathrm{h}(*, \mathbf{q}_j) = \mathrm{c}_j + \mathbf{q}_j' \left(\frac{1}{m} \sum_{k=1}^{m} \mathbf{b}_j' \mathbf{A}_k \mathbf{A}_k' \mathbf{b}_j \mathbf{V}_k\right) \mathbf{q}_j - 2\ \mathbf{q}_j' \mathbf{T}_j \mathbf{b}_j$$

$$= \mathrm{c}_j + \mathbf{q}_j' \bar{\mathbf{V}}_j \mathbf{q}_j - 2\ \mathbf{q}_j' \mathbf{T}_j \mathbf{b}_j, \tag{7.16}$$

where

$$\bar{\mathbf{V}}_j = \frac{1}{m} \sum_{k=1}^{m} \mathbf{b}_j' \mathbf{A}_k \mathbf{A}_k' \mathbf{b}_j \mathbf{V}_k. \tag{7.17}$$

Defining $\bar{\mathbf{q}}_j$ as a vector satisfying

$$\bar{\mathbf{V}}_j \bar{\mathbf{q}}_j = \mathbf{T}_j \mathbf{b}_j, \tag{7.18}$$

we may write (7.16) as

$$h(*, \mathbf{q}_j) = d_j + (\mathbf{q}_j - \bar{\mathbf{q}}_j)'\bar{\mathbf{V}}_j(\mathbf{q}_j - \bar{\mathbf{q}}_j), \tag{7.19}$$

with d_j a term independent of $\mathbf{q}_j$. Applying the algebra of section 5, (7.19) can be minimized by alternatingly computing

$$\tilde{\mathbf{q}}_j = \mathbf{q}_j^0 + \frac{1}{\phi_1}\bar{\mathbf{V}}_j(\bar{\mathbf{q}}_j - \mathbf{q}_j^0)$$

$$= \frac{1}{\phi_1}\mathbf{T}_j\mathbf{b}_j + (\mathbf{I} - \frac{1}{\phi_1}\bar{\mathbf{V}}_j)\mathbf{q}_j^0, \tag{7.20}$$

where ϕ_1 is the largest eigenvalue of $\bar{\mathbf{V}}_j$, and $\mathbf{q}_j^0$ is the current solution for $\mathbf{q}_j$ satisfying the constraints which correspond to the measurement level of variable j, and then determine a new $\mathbf{q}_j^+$ satisfying these same constraints by minimizing the function

$$q(\mathbf{q}_j) = (\mathbf{q}_j - \tilde{\mathbf{q}}_j)'(\mathbf{q}_j - \tilde{\mathbf{q}}_j), \tag{7.21}$$

which is generally achieved by performing a regression of $\tilde{\mathbf{q}}_j$ on the original external variable $\mathbf{q}_j$.

Before specifically discussing how to obtain optimal quantifications for variable $\mathbf{q}_j$ in (7.21), we first deal with the question whether and how quantifications of the external variables should be centered and normalized. In this context, it is important to remember that the original problem consists of the approximation of (dis)similarity data with *distances*. Because translations are distance preserving transformations, we have that

$$d_{ij}(\mathbf{Z}\mathbf{A}_k) = d_{ij}(\mathbf{Q}\mathbf{B}\mathbf{A}_k) = d_{ij}[\sum_{j=1}^{h} \mathbf{q}_j\mathbf{b}_j'\mathbf{A}_k]$$

$$= d_{ij}[\sum_{j=1}^{h} (\mathbf{q}_j + r_j\mathbf{1})\mathbf{b}_j'\mathbf{A}_k] \tag{7.22}$$

for any arbitrary scalar r_j. It follows from (7.22) that it is immaterial whether and how we center variable $\mathbf{q}_j$ since loss function (1.1) is unaffected by such a transformation.

Moreover, we are free at any time to multiply $\mathbf{q}_j$ with an arbitrary number a_j as long as we apply the inverse transformation to $\mathbf{b}_j$. This is true because

$$\sum_{j=1}^{h} \mathbf{q}_j\mathbf{b}_j' = \sum_{j=1}^{h} (a_j\mathbf{q}_j)(a_j^{-1}\mathbf{b}_j') = \sum_{j=1}^{h} \hat{\mathbf{q}}_j\hat{\mathbf{b}}_j' \tag{7.23}$$

with $\hat{\mathbf{q}}_j = a_j\mathbf{q}_j$ and $\hat{\mathbf{b}}_j = a_j^{-1}\mathbf{b}_j$. Therefore, this indeterminacy may be used to normalize either $\mathbf{q}_j$ or $\mathbf{b}_j$ on fixed length. In the following exposition we adopt the conventions $\mathbf{1}'\mathbf{q}_j = 0$ and $\mathbf{q}_j'\mathbf{q}_j = n$ for $j = 1, ..., h$.

If $\mathbf{q}_j$ in (7.21) refers to a *numerical* variable we require the update to be a linear transformation of the original variable subject to $\mathbf{1}'\mathbf{q}_j = 0$ and $\mathbf{q}_j'\mathbf{q}_j = n$. If $\mathbf{q}_j$ is complete, the latter constraints completely fix the linear transformation, meaning that the updating of the quantifications of a complete numerical variable is not required, and may be skipped. However, if the external variable is incomplete, then, letting $\mathbf{M}_j$ denote the $(n \times n)$ diagonal matrix with a one on its diagonal where an element of $\mathbf{q}_j$ is nonmissing and zeroes elsewhere, we have to solve

$$q(a_j, r_j) = [(a_j\mathbf{q}_j + r_j\mathbf{1}) - \tilde{\mathbf{q}}_j]'\mathbf{M}_j[(a_j\mathbf{q}_j + r_j\mathbf{1}) - \tilde{\mathbf{q}}_j] \tag{7.24a}$$

for the nonmissing part of $\mathbf{q}_j$. In PROXSCAL, we adopt the option called *missing data multiple* in the Gifi system (cf., Gifi, 1990) for the missing part of the quantification vectors, which implies that we minimize

$$q(\mathbf{q}_j) = [\mathbf{q}_j - \tilde{\mathbf{q}}_j]'(\mathbf{I} - \mathbf{M}_j)[\mathbf{q}_j - \tilde{\mathbf{q}}_j] \tag{7.24b}$$

for the missing elements of $\mathbf{q}_j$. The latter is achieved by simply setting $(\mathbf{I} - \mathbf{M}_j)\mathbf{q}_j^+$ equal to $(\mathbf{I} - \mathbf{M}_j)\tilde{\mathbf{q}}_j$. To solve (7.24a), we compute

$$a_j = \frac{(\mathbf{1}'\mathbf{M}_j\mathbf{1})(\mathbf{q}_j'\mathbf{M}_j\tilde{\mathbf{q}}_j) - (\mathbf{q}_j'\mathbf{M}_j\mathbf{1})(\tilde{\mathbf{q}}_j'\mathbf{M}_j\mathbf{1})}{(\mathbf{1}'\mathbf{M}_j\mathbf{1})(\mathbf{q}_j'\mathbf{M}_j\mathbf{q}_j) - (\mathbf{q}_j'\mathbf{M}_j\mathbf{1})^2} \tag{7.25a}$$

and

$$r_j = \frac{(\tilde{\mathbf{q}}_j - a_j\mathbf{q}_j)'\mathbf{M}_j\mathbf{1}}{\mathbf{1}'\mathbf{M}_j\mathbf{1}}, \tag{7.25b}$$

and set $\mathbf{M}_j\mathbf{q}_j^+$ equal to $\mathbf{M}_j(a_j\mathbf{q}_j + r_j\mathbf{1})$. Finally, the resulting vector $\mathbf{q}_j^+$ is centered and normalized, and care is taken to adapt the corresponding regression weights vector $\mathbf{b}_j$ accordingly (see (7.23)).

If $\mathbf{q}_j$ is an *ordinal* variable, (7.21) is minimized by performing a monotone regression of $\tilde{\mathbf{q}}_j$ on the original variable $\mathbf{q}_j$. Moreover, if some elements of $\mathbf{q}_j$ are missing, then the monotone regression is only applied to the nonmissing elements, while the missing elements of $\mathbf{q}_j$ are simply replaced by the corresponding elements of $\tilde{\mathbf{q}}_j$. Again, due to (7.22) the result may be centered on the origin, and due to (7.23) it may then be normalized on fixed length as long as the inverse transformation is applied to the corresponding regression weights.

If $\mathbf{q}_j$ is a *nominal* variable we have to solve, for the nonmissing elements of the variable,

$$\mathrm{q}(\mathbf{y}_j) = (\mathbf{G}_j\mathbf{y}_j - \tilde{\mathbf{q}}_j)'\mathbf{M}_j(\mathbf{G}_j\mathbf{y}_j - \tilde{\mathbf{q}}_j), \tag{7.26}$$

where $\mathbf{G}_j$ is the *indicator matrix* (cf., Gifi, 1990) of order (n x k_j), k_j being the number of categories for variable j, and the vector $\mathbf{y}_j$ of order (k_j x 1) contains the k_j distinct categories of variable j. Dropping missing rows and columns, and tacitly assuming that we deal with the resulting reduced vectors and matrices, the global minimum of (7.26) is obtained for

$$\bar{\mathbf{y}}_j = (\mathbf{G}_j'\mathbf{G}_j)^{-1}\mathbf{G}_j'\tilde{\mathbf{q}}_j. \tag{7.27}$$

Then, $\mathbf{q}_j = \mathbf{M}_j\mathbf{G}_j\bar{\mathbf{y}}_j$ yields an update for the nonmissing elements of nominal variable j, while the missing elements of $\mathbf{q}_j$ are simply replaced by the corresponding elements of $\tilde{\mathbf{q}}_j$. The result is centered on the origin, and normalized on fixed length, and the inverse transformation is applied to the corresponding regression weights (see (7.23)).

If $p > s$, that is, if there are more dimensions than external variables, then the last $(p - s)$ columns of $\mathbf{Q}$ are treated as *free* vectors. Thus, for $j = s + 1, ..., p$ an update for $\mathbf{q}_j$ is obtained by setting it equal to $\tilde{\mathbf{q}}_j$ defined in (7.20), and then normalize to length n.

In all cases, in the minimization of (7.19) an update for $\mathbf{q}_j$ may be calculated just once, or we may repeatedly alternate over (7.20) and (7.21) until some convergence criterion is met. The same applies to the calculation of updates for $\mathbf{b}_j$ and $\mathbf{q}_j$ for one variable j: in the minimization of (7.13) we may either stop after one iteration or continue to alternate until no further improvement is obtained.

Once all vectors $\mathbf{q}_j$ and $\mathbf{b}_j$ have been updated for $j = 1, \ldots, h$, we may stop and update the common space with

$$\mathbf{Z}^+ = \sum_{j=1}^{h} \mathbf{q}_j\mathbf{b}_j', \tag{7.28}$$

or, again, repeat the whole minimization procedure for (7.11) until some convergence criterion is reached.

Summarizing, in the GENERALIZED and REDUCED rank models the following convergent algorithm may be used to restrict the common space to be a linear combination of (possibly incomplete) external variables of mixed measurement levels:

1. compute $\mathbf{S} = \frac{1}{m} \sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k'$;

2. for $j = 1, \dots, h$, perform the following calculations:

2.1. compute $\mathbf{U}_j = \sum_{t \neq j} \mathbf{q}_t^0 \mathbf{b}_t^{0\prime}$ and $\mathbf{T}_j = \mathbf{S} - \frac{1}{m} \sum_{k=1}^{m} \mathbf{V}_k \mathbf{U}_j \mathbf{A}_k \mathbf{A}_k'$;

2.2. compute $\mathbf{b}_j^+ = (\frac{1}{m} \sum_{k=1}^{m} \mathbf{q}_j^{0\prime} \mathbf{V}_k \mathbf{q}_j^0 \mathbf{A}_k \mathbf{A}_k')^{-1} \mathbf{T}_j' \mathbf{q}_j^0$ and set $\mathbf{b}_j^0 = \mathbf{b}_j^+$;

2.3. if $\mathbf{q}_j$ is a numerical variable and complete: skip this step;
else:
compute $\bar{\mathbf{V}}_j = \frac{1}{m} \Sigma_k \mathbf{b}_j^{0\prime} \mathbf{A}_k \mathbf{A}_k' \mathbf{b}_j^0 \mathbf{V}_k$, and compute the largest eigenvalue ϕ_1 of $\bar{\mathbf{V}}_j$ (or an estimation thereof); calculate

$$\tilde{\mathbf{q}}_j = \frac{1}{\phi_1} \mathbf{T}_j \mathbf{b}_j^0 + (\mathbf{I} - \frac{1}{\phi_1} \bar{\mathbf{V}}_j)\mathbf{q}_j^0,$$

2.3.1. if $\mathbf{q}_j$ is a numerical variable: compute the direction cosine a_j and the intercept r_j in the regression of $\tilde{\mathbf{q}}_j$ on the original variable $\mathbf{q}_j$ according to (7.25a) and (7.25b); set the nonmissing elements of the update $\mathbf{q}_j^+$ equal to $(a_j\mathbf{q}_j + r_j\mathbf{1})$, and the missing elements equal to the corresponding elements of $\tilde{\mathbf{q}}_j$;

2.3.2. if $\mathbf{q}_j$ is an ordinal variable: for the nonmissing elements, perform a monotone regression of $\tilde{\mathbf{q}}_j$ on the original variable $\mathbf{q}_j$, and store the result in the corresponding elements of $\mathbf{q}_j^+$; set the missing elements equal to the corresponding elements of $\tilde{\mathbf{q}}_j$;

2.3.3. if $\mathbf{q}_j$ is a nominal variable: for the nonmissing elements of the variable calculate

$$\mathbf{G}_j(\mathbf{G}_j'\mathbf{G}_j)^{-1}\mathbf{G}_j'\tilde{\mathbf{q}}_j \ ,$$

and store in the corresponding elements of $\mathbf{q}_j^+$; set the missing elements equal to the corresponding elements of $\tilde{\mathbf{q}}_j$;

2.3.4. if $\mathbf{q}_j$ is free (i.e., $j > s$): treat the vector as completely missing, i.e., store all elements of $\tilde{\mathbf{q}}_j$ in the update $\mathbf{q}_j^+$ as is;

In all cases, center $\mathbf{q}_j^+$ on the origin, and then normalize $\mathbf{q}_j^+$ on length n;
apply the inverse normalization to the regression weight vector $\mathbf{b}_j^0$;
set $\mathbf{q}_j^0 = \mathbf{q}_j^+$;

3. calculate the restricted update $\mathbf{Z}^+ = \Sigma_j \mathbf{q}_j^0 \mathbf{b}_j^{0'}$, evaluate (4.1), and go to step 2 if the difference in function value between the current and the previous iteration is smaller than some predefined criterion; otherwise stop.

We have the following special cases. If the weight matrices are equal to each other in the GENERALIZED model, and the identification condition $\frac{1}{m}\ \Sigma_k \mathbf{A}_k \mathbf{A}_k' = \mathbf{I}$ is used, then formula (7.15) for updating the regression weights simplifies into

$$\mathbf{b}_j = \frac{1}{\mathbf{q}_j'\mathbf{V}\mathbf{q}_j} \mathbf{T}_j'\mathbf{q}_j, \tag{7.26}$$

with

$$\mathbf{T}_j = \frac{1}{m} \sum_{k=1}^{m} \mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' - \mathbf{V}\mathbf{U}_j \tag{7.27}$$

and $\mathbf{U}_j$ as defined in (7.12). At the same time, matrix $\bar{\mathbf{V}}_j$ defined in (7.17) can then be written as

$$\bar{\mathbf{V}}_j = \mathbf{b}_j'\mathbf{b}_j\mathbf{V}. \tag{7.28}$$

Since the largest eigenvalue ϕ_1 of matrix $\bar{\mathbf{V}}_j$ in (7.28) satisfies $\phi_1 = \phi_2 \mathbf{b}_j'\mathbf{b}_j$ with ϕ_2 the largest eigenvalue of matrix $\mathbf{V}$, the unrestricted update (7.20) for $\mathbf{q}_j$ in (7.19) can be written as

$$\tilde{\mathbf{q}}_j = \frac{1}{\phi_1}\mathbf{T}_j\mathbf{b}_j + (\mathbf{I} - \frac{1}{\phi_1}\bar{\mathbf{V}}_j)\mathbf{q}_j^0 = \frac{1}{\phi_2\mathbf{b}_j'\mathbf{b}_j}\mathbf{T}_j\mathbf{b}_j + (\mathbf{I} - \frac{1}{\phi_2\mathbf{b}_j'\mathbf{b}_j}\mathbf{b}_j'\mathbf{b}_j\mathbf{V})\mathbf{q}_j^0$$

$$= \frac{1}{\phi_2\mathbf{b}_j'\mathbf{b}_j}\mathbf{T}_j\mathbf{b}_j + (\mathbf{I} - \frac{1}{\phi_2}\mathbf{V})\mathbf{q}_j^0 \tag{7.29}$$

with $\mathbf{T}_j$ as defined in (7.27). Thus in this case ϕ_2, the largest eigenvalue of $\mathbf{V}$, is identical for all variables $\mathbf{q}_j$ and only has to be computed once. When applying identification condition (4.5) to the space weight matrices $\mathbf{A}_k$ in the external case, care must be taken to adjust the regression weight matrix $\mathbf{B}$ in $\mathbf{Z} = \mathbf{QB}$ accordingly.

If $w_{ijk} = 1$ for all i, j, and k, then, since $\mathbf{V} = n\mathbf{J}$ in this case, and still assuming that $\frac{1}{m}\ \Sigma_k \mathbf{A}_k \mathbf{A}_k' = \mathbf{I}$, updating formula (7.26) changes into

$$\mathbf{b}_j = \frac{1}{\mathbf{q}_j'(n\mathbf{J})\mathbf{q}_j}\left(\frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' - (n\mathbf{J})\mathbf{U}_j\right)'\mathbf{q}_j$$

$$= \frac{1}{\mathbf{q}_j'\mathbf{q}_j}\left[\frac{1}{nm}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' - \mathbf{U}_j\right]'\mathbf{q}_j, \tag{7.30}$$

on the condition that $\mathbf{1}'\mathbf{q}_j = 0$ for $j = 1, \ldots, h$. The updating of the quantifications of the external variables is much simplified in the unweighted case. Because (7.28) may now be written as

$$\bar{\mathbf{V}}_j = n\mathbf{b}_j'\mathbf{b}_j\mathbf{J}, \tag{7.31}$$

with $\mathbf{J}$ the centering matrix, and since ϕ_1, the largest eigenvalue of (7.31), equals $n\mathbf{b}_j'\mathbf{b}_j$, we now have for the unrestricted update (7.20) that

$$\tilde{\mathbf{q}}_j = \frac{1}{n\mathbf{b}_j'\mathbf{b}_j}\mathbf{T}_j\mathbf{b}_j + \left(\mathbf{I} - \frac{1}{n\mathbf{b}_j'\mathbf{b}_j}n\mathbf{b}_j'\mathbf{b}_j\mathbf{J}\right)\mathbf{q}_j^0 = \frac{1}{n\mathbf{b}_j'\mathbf{b}_j}\mathbf{T}_j\mathbf{b}_j, \tag{7.32}$$

with

$$\mathbf{T}_j = \frac{1}{m}\sum_{k=1}^{m}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0\mathbf{A}_k' - n\mathbf{U}_j. \tag{7.33}$$

This means that (7.21) becomes

$$\mathrm{q}(\mathbf{q}_j) = \left(\mathbf{q}_j - \frac{1}{n\mathbf{b}_j'\mathbf{b}_j}\mathbf{T}_j\mathbf{b}_j\right)'\left(\mathbf{q}_j - \frac{1}{n\mathbf{b}_j'\mathbf{b}_j}\mathbf{T}_j\mathbf{b}_j\right), \tag{7.34}$$

and shows that (7.19) can now be solved analytically in all cases.

In the most simple situation (one source, $w_{ij} = 1$ for all i, j), and assuming that the external variables are centered on the origin, we have that

$$\mathbf{b}_j = \frac{1}{\mathbf{q}_j'\mathbf{q}_j}\left(\frac{1}{n}\mathbf{B}(\mathbf{X}^0)\mathbf{X}^0 - \mathbf{U}_j\right)'\mathbf{q}_j, \tag{7.35}$$

and, in (7.34),

$$\mathbf{T}_j = \mathbf{B}(\mathbf{X}^0)\mathbf{X}^0 - n\mathbf{U}_j. \tag{7.36}$$

Again, no majorization is needed to obtain updates for the vectors $\mathbf{q}_j$.

8 Optimal scaling

In the case where the (dis)similarities are unique up to a transformation the algorithm requires an additional step. There are two possibilities. Either the (dis)similarities may only be compared within each source (MATRIX CONDITIONAL), or they may all be compared with each other (UNCONDITIONAL). In the first case the general loss function has to be generalized to

$$f(\phi_1, \ldots, \phi_m;\mathbf{X}_1, \ldots, \mathbf{X}_m) \equiv \frac{1}{m} \sum_{k=1}^{m} \sum_{i<j}^{n} w_{ijk}[\phi_k(\delta_{ijk}) - d_{ij}(\mathbf{X}_k)]^2, \tag{8.1}$$

where ϕ_k denotes the transformation for source k, and in the UNCONDITIONAL case the loss becomes

$$f(\phi;\mathbf{X}_1, \ldots, \mathbf{X}_m) \equiv \frac{1}{m} \sum_{k=1}^{m} \sum_{i<j}^{n} w_{ijk}[\phi(\delta_{ijk}) - d_{ij}(\mathbf{X}_k)]^2, \tag{8.2}$$

where ϕ denotes a combined transformation of all (dis)similarities involved in the analysis.

Apart from conditionality, several types of transformations may be required. PROXSCAL allows for PROPORTIONAL, POWER, LINEAR, MONOTONE and SMOOTH MONOTONE transformations of the data. PROPORTIONAL and POWER transformations can be dealt with in the initialization procedure of the algorithm (see section 9); no additional calculations during iterations are required in these cases. To explain the calculations involved in the LINEAR, MONOTONE and SMOOTH MONOTONE cases we need the following definitions. Let $\boldsymbol{\delta}_k$ denote the column vector containing the nonmissing dissimilarities δ_{ijk} in partition k, and $\mathbf{d}_k$ be the column vector containing the corresponding distances $d_{ij}(\mathbf{X}_k)$. Also define $\boldsymbol{\delta}$ as the column vector containing the m vectors $\boldsymbol{\delta}_k$ stacked under one another, and $\mathbf{d}$ as the column vector containing the m vectors $\mathbf{d}_k$. Finally, let $\mathbf{W}_k$ denote the diagonal matrix with nonzero weights w_{ijk} of partition k on the diagonal, and $\mathbf{W}$ the diagonal matrix with all nonzero weights on the diagonal. With these definitions loss functions (8.1) and (8.2) may be written as

$$f(\phi_1, \ldots, \phi_m) = \frac{1}{m} \sum_{k=1}^{m} (\phi_k(\boldsymbol{\delta}_k) - \mathbf{d}_k)'\mathbf{W}_k(\phi_k(\boldsymbol{\delta}_k) - \mathbf{d}_k) \tag{8.3}$$

and

$$f(\phi) = \frac{1}{m} (\phi(\boldsymbol{\delta}) - \mathbf{d})'\mathbf{W}(\phi(\boldsymbol{\delta}) - \mathbf{d}) \tag{8.4}$$

respectively.

If a LINEAR transformation is needed, then this transformation may either be applied to each partition separately (MATRIX CONDITIONAL), or to all data simultaneously (UNCONDITIONAL). Here, we only discuss how to apply a LINEAR transformation in the UNCONDITIONAL case; for MATRIX CONDITIONAL data, exactly the same procedure can be used for each partition separately.

Since a linear transformation of the dissimilarities is defined by

$$\phi(\boldsymbol{\delta}) = b\boldsymbol{\delta} + a\mathbf{1}, \tag{8.5}$$

where b and a are scalars and $\mathbf{1}$ is a vector of ones, the formal problem is how to minimize

$$f(b, a) = \frac{1}{m} \{(b\boldsymbol{\delta} + a\mathbf{1}) - \mathbf{d}\}'\mathbf{W}\{(b\boldsymbol{\delta} + a\mathbf{1}) - \mathbf{d}\}. \tag{8.6}$$

This is a simple regression problem in the metric $\mathbf{W}$. For fixed b, and defining $\mathbf{e} = b\boldsymbol{\delta} - \mathbf{d}$, (8.6) may be written as

$$f(a) = \frac{1}{m} \{\mathbf{1}'\mathbf{W}\mathbf{1}(a + \frac{\mathbf{e}'\mathbf{W}\mathbf{1}}{\mathbf{1}'\mathbf{W}\mathbf{1}})^2 + g\} \geq \frac{1}{m} g, \tag{8.7}$$

where $g \equiv (\mathbf{e}'\mathbf{W}\mathbf{e}) - (\mathbf{1}'\mathbf{W}\mathbf{1})^{-1}(\mathbf{e}'\mathbf{W}\mathbf{1})^2$ is a term independent of a. Therefore, the lower bound of $\frac{1}{m}$ g in (8.7) is attained for

$$a = -\frac{\mathbf{e}'\mathbf{W}\mathbf{1}}{\mathbf{1}'\mathbf{W}\mathbf{1}} = \frac{\mathbf{d}'\mathbf{W}\mathbf{1} - b\boldsymbol{\delta}'\mathbf{W}\mathbf{1}}{\mathbf{1}'\mathbf{W}\mathbf{1}}. \tag{8.8}$$

For fixed a, and letting $\mathbf{c} = a\mathbf{1} - \mathbf{d}$, it is true that

$$f(b) = \frac{1}{m} \{\boldsymbol{\delta}'\mathbf{W}\boldsymbol{\delta}(b + \frac{\boldsymbol{\delta}'\mathbf{W}\mathbf{c}}{\boldsymbol{\delta}'\mathbf{W}\boldsymbol{\delta}})^2 + h\} \geq \frac{1}{m} h, \tag{8.9}$$

where $h \equiv (\mathbf{c}'\mathbf{W}\mathbf{c}) - (\boldsymbol{\delta}'\mathbf{W}\boldsymbol{\delta})^{-1}(\boldsymbol{\delta}'\mathbf{W}\mathbf{c})^2$ is a term independent of b. Thus, (8.9) is globally minimized for

$$b = -\frac{\boldsymbol{\delta}'\mathbf{W}\mathbf{c}}{\boldsymbol{\delta}'\mathbf{W}\boldsymbol{\delta}} = \frac{\boldsymbol{\delta}'\mathbf{W}\mathbf{d} - a\boldsymbol{\delta}'\mathbf{W}\mathbf{1}}{\boldsymbol{\delta}'\mathbf{W}\boldsymbol{\delta}}. \tag{8.10}$$

Substitution of (8.8) in (8.10) yields

$$b = \frac{(\boldsymbol{\delta}'\mathbf{Wd})(\mathbf{1}'\mathbf{W1}) - (\boldsymbol{\delta}'\mathbf{W1})(\mathbf{d}'\mathbf{W1})}{(\boldsymbol{\delta}'\mathbf{W}\boldsymbol{\delta})(\mathbf{1}'\mathbf{W1}) - (\boldsymbol{\delta}'\mathbf{W1})^2} . \tag{8.11}$$

Therefore, a linear transformation of the data consists of the calculation of (8.11) followed by (8.8). It is important to see that this transformation may result in negative pseudo-distances, a situation where convergence of the main PROXSCAL algorithm is no longer guaranteed. A solution to this problem will be treated elsewhere.

The solutions of (8.3) and (8.4) in the MONOTONE situation are standard (see, for instance, Gifi, 1990).

For the moment, we refer to Heiser (1985a) for the solutions of (8.3) and (8.4) in the SMOOTHED MONOTONE situation.

All transformations discussed in the present section are followed by an *explicit* normalization of the transformed $\phi_k(\boldsymbol{\delta}_k)$ and $\phi(\boldsymbol{\delta})$ on fixed lengths. If s denotes the required length, then the pseudo-distances are normalized on $\boldsymbol{\delta}_k'\mathbf{W}_k\boldsymbol{\delta}_k = \mathrm{s}$ for $k = 1, \ldots, m$ (meaning that $\Sigma_k\ \boldsymbol{\delta}_k'\mathbf{W}_k\boldsymbol{\delta}_k = m\mathrm{s}$) in the MATRIX CONDITIONAL case, and on $\boldsymbol{\delta}'\mathbf{W}\boldsymbol{\delta} = m\mathrm{s}$ in the UNCONDITIONAL case, respectively. Letting $\hat{\boldsymbol{\delta}}_k = \phi_k(\boldsymbol{\delta}_k)$ and $\hat{\boldsymbol{\delta}} = \phi(\boldsymbol{\delta})$ (i.e., the unnormalized transformed pseudo-distances), the former normalization is achieved by the calculation of

$$\tilde{\boldsymbol{\delta}}_k = \hat{\boldsymbol{\delta}}_k \sqrt{\frac{\mathrm{s}}{\hat{\boldsymbol{\delta}}_k'\mathbf{W}_k\hat{\boldsymbol{\delta}}_k}} \qquad \text{for } k = 1,\ldots, m, \tag{8.12}$$

and by computing

$$\tilde{\boldsymbol{\delta}} = \hat{\boldsymbol{\delta}} \sqrt{\frac{m\mathrm{s}}{\hat{\boldsymbol{\delta}}'\mathbf{W}\hat{\boldsymbol{\delta}}}} \tag{8.13}$$

in the latter case.

It is of some interest to note that substitution of (8.12) in STRESS loss function (1.1) yields

$$f(\mathbf{X}_1, \ldots, \mathbf{X}_m) = \frac{1}{m} \Sigma_k\ \Sigma_{i<j} w_{ijk}[\tilde{\delta}_{ijk} - d_{ij}(\mathbf{X}_k)]^2$$

$$= \frac{1}{m} \Sigma_k\ \{\mathrm{s} + \Sigma_{i<j} w_{ijk} d_{ij}^2(\mathbf{X}_k) - 2\ \Sigma_{i<j} w_{ijk}\hat{\delta}_{ijk} d_{ij}(\mathbf{X}_k) \sqrt{\frac{\mathrm{s}}{\Sigma_{i<j} w_{ijk}\hat{\delta}_{ijk}^2}}\ \}. \tag{8.14}$$

But since it follows from projection theory that at the global minimum of (8.3)

$$(\phi_k(\boldsymbol{\delta}_k) - \mathbf{d}_k)'\mathbf{W}_k\phi_k(\boldsymbol{\delta}_k) = 0,$$

and thus

$$\phi_k(\boldsymbol{\delta}_k)'\mathbf{W}_k\phi_k(\boldsymbol{\delta}_k) = \mathbf{d}_k'\mathbf{W}_k\phi_k(\boldsymbol{\delta}_k),$$

and therefore

$$\Sigma_{i<j}\mathrm{w}_{ijk}\hat{\delta}_{ijk}^2 = \Sigma_{i<j}\mathrm{w}_{ijk}\hat{\delta}_{ijk}\mathrm{d}_{ij}(\mathbf{X}_k), \tag{8.15}$$

we may simply write (8.14) in terms of the unnormalized new pseudo-distances as

$$\mathrm{f}(\mathbf{X}_1, \ldots, \mathbf{X}_m) = \frac{1}{m}\Sigma_k \{\mathrm{s} + \Sigma_{i<j}\mathrm{w}_{ijk}\mathrm{d}_{ij}^2(\mathbf{X}_k) - 2\sqrt{\mathrm{s}\Sigma_{i<j}\mathrm{w}_{ijk}\hat{\delta}_{ijk}^2}\}. \tag{8.16}$$

This is true for MATRIX CONDITIONAL data. Analogously, in the UNCONDITIONAL case we may write (1.1) in terms of the unnormalized new pseudo-distances as

$$\mathrm{f}(\mathbf{X}_1, \ldots, \mathbf{X}_m) = \frac{1}{m}\{m\mathrm{s} + \Sigma_k\Sigma_{i<j}\mathrm{w}_{ijk}\mathrm{d}_{ij}^2(\mathbf{X}_k) - 2\sqrt{m\mathrm{s}\Sigma_k\Sigma_{i<j}\mathrm{w}_{ijk}\hat{\delta}_{ijk}^2}\}. \tag{8.17}$$

In PROXSCAL, the factor $\mathrm{s} = n(n-1)/2$ is used for the explicit normalization of the pseudo-distances. This has the advantage that the values of the δ_{ijk} involved in an analysis remain within comparable ranges irrespective of the number of proximities in each source k.

For the sake of completeness, we end this section by noting that a previous version of PROXSCAL used *implicit* normalization of the transformed (dis)similarities. In this older PROXSCAL version the general loss function (1.1) was defined as

$$\mathrm{f}(\mathbf{X}_1, \ldots, \mathbf{X}_m) \equiv \frac{1}{m}\Sigma_k \frac{\Sigma_{i<j}\mathrm{w}_{ijk}[\delta_{ijk} - \mathrm{d}_{ij}(\mathbf{X}_k)]^2}{\Sigma_{i<j}\mathrm{w}_{ijk}\delta_{ijk}^2} \tag{8.18}$$

in the MATRIX CONDITIONAL case (see also De Leeuw and Heiser, 1977, and Heiser, 1988). An important drawback of the minimization of this loss function is that it results in weight matrices $\mathbf{V}_k$ (analogous to those defined in (2.3)) which change in each iteration of the main algorithm. In contrast, when adopting an explicit normalization of the transformed (dis)similarities the $\mathbf{V}_k$ matrices remain constant throughout.

9 Initialization of the PROXSCAL algorithms

In this section we give an overview of the initialization procedure used in PROXSCAL, as discussed in Heiser (1985b). Roughly, this procedure involves two steps: a normalization of the raw (dis)similarities δ_{ijk} and weights w_{ijk}, followed by a very first estimation of the coordinates of the objects in the common space.

The normalization of the raw (dis)similarities δ_{ijk} and weights w_{ijk} (which are all assumed to be nonnegative) consists of the following steps:

1 The raw data values are first transformed into dissimilarities with the smallest dissimilarity being equal to zero. Specifically, let δ denote the vector containing all non-missing raw δ_{ijk}, δ_k denote the vector containing all non-missing raw δ_{ijk} *within partition k*, $\delta_{\max}$ and $\delta_{\min}$ denote the largest and the smallest value in δ, respectively, and $\delta_{\max(k)}$ and $\delta_{\min(k)}$ denote the largest and the smallest value in δ_k, respectively. Now, depending upon the options chosen by the user, pseudo-distances $\hat{\delta}_{ijk}$ are calculated as follows.

If the raw δ_{ijk} are similarities (called DESCENDING in PROXSCAL) and the data are UNCONDITIONAL, then $\hat{\delta} = \delta_{\max} - \delta$, and if the raw δ_{ijk} are similarities and the data are CONDITIONAL, then $\hat{\delta}_k = \delta_{\max(k)} - \delta_k$ for $k = 1, \ldots, m$. If the raw δ_{ijk} are dissimilarities already (called ASCENDING in PROXSCAL) and the data are UNCONDITIONAL, then $\hat{\delta} = \delta - \delta_{\min}$; if the raw δ_{ijk} are dissimilarities and the data are CONDITIONAL, then $\hat{\delta}_k = \delta_k - \delta_{\min(k)}$ for $k = 1, \ldots, m$.

2 In this second step, the type of transformation requested by the user is taken into account. If a plain metric (LINEAR in PROXSCAL) or a PROPORTIONAL analysis is required, nothing needs to be done. However, if the user wants a POWER analysis then the non-missing $\hat{\delta}_{ijk}$ obtained in the previous step are replaced by the pseudo-distances raised to the requested power. If a MONOTONIC (non-metric) or SMOOTHED MONOTONIC transformation has been chosen, the non-missing $\hat{\delta}_{ijk}$ are replaced with their ascending rank numbers within partitions in the CONDITIONAL case, and over partitions in the UNCONDITIONAL case.

3 Finally, the pseudo-distances $\hat{\delta}_{ijk}$ are normalized such that $\sum_{i<j} w_{ijk}\hat{\delta}_{ijk}^2 = n(n-1)/2$ in the CONDITIONAL case, and that $\sum_k\sum_{i<j} w_{ijk}\hat{\delta}_{ijk}^2 = mn(n-1)/2$ in the UNCONDITIONAL case.

Let the thus obtained pseudo-distances be denoted by $\hat{d}_{ijk}$. In PROXSCAL, these $\hat{d}_{ijk}$ are used throughout all calculations discussed in the previous sections.

The next problem is how to obtain a first estimate for the common space **Z**. In PROXSCAL, the algorithms for all models are started 'as if' the IDENTITY model had been chosen by the user. Moreover, the calculation of an initial $\mathbf{Z}$ again involves two steps. First, starting from the simplex, an unrestricted common space is determined in (n - 1) dimensions using one majorization step. Then, this (n - 1)-dimensional common space is restricted to a p-dimensional subspace by solving a metric projection problem using alternating least squares. In this last step, possible restrictions on the common space are also respected.

By now, the reader should be familiar with the two steps that we just described. Just as in the PROXSCAL algorithms themselves, the initialization procedure also uses majorization to determine an unrestricted update first, and then solves a metric projection problem to restrict the update to some subspace; thus, the initialization procedure completely stays within the PROXSCAL framework. We will now discuss each step in more detail.

Starting from the IDENTITY model, the first step consists of finding an unrestricted (n - 1)-dimensional configuration $\mathbf{Z}$ which minimizes the general loss function

$$\mathrm{f}(\mathbf{Z}) = \frac{1}{m} \sum_{k=1}^{m} \sum_{i<j}^{n} \mathrm{w}_{ijk}[\hat{\mathrm{d}}_{ijk} - \mathrm{d}_{ij}(\mathbf{Z})]^2, \tag{9.1}$$

where $\hat{\mathrm{d}}_{ijk}$ are the normalized pseudo-distances obtained in the first phase of the initialization procedure. In section 5, we found that the Guttman transform for updating the common space in the IDENTITY model was the one given in (4.4.3), that is,

$$\bar{\mathbf{Z}} = (\frac{1}{m} \sum_{k=1}^{m} \mathbf{V}_k)^{-}\{ \frac{1}{m} \sum_{k=1}^{m} \mathbf{B}_k(\mathbf{Z}^0)\}\mathbf{Z}^0.$$

As the very first start for $\mathbf{Z}^0$ the complete centering matrix of order (n x n) is used, that is,

$$\mathbf{J} = \mathbf{I}_n - \frac{\mathbf{11}'}{\mathbf{1}'\mathbf{1}}, \tag{9.2}$$

where $\mathbf{1}$ is the (n x 1) unit vector. This is a simplex in (n - 1) dimensions, with all its rows or columns (conceived of as points in (n - 1)-dimensional space) within equal distance of one another. Specifically, irrespective of the number of objects involved,

$$\mathrm{d}_{ij}(\mathbf{J}) = \sqrt{2} \text{ for all } i \neq j, \tag{9.3}$$

as is easily verified. In this very special case, (4.4.3) may therefore be written as

$$\bar{\mathbf{Z}} = \frac{1}{\sqrt{2}} \mathbf{V}^{-}\mathbf{B}(\mathbf{J}), \tag{9.4}$$

where

$$\mathbf{V} \equiv \frac{1}{m} \sum_{k=1}^{m} \mathbf{V}_k \tag{9.5}$$

and $\mathbf{V} = \{v_{ij}\}$ is defined by

$$v_{ij} = \begin{cases} -w_{ij.} & \text{for } i \neq j \\ \sum_{l \neq i}^{n} w_{il.} & \text{for } i = j \end{cases} \tag{9.6}$$

with

$$w_{ij.} = \frac{1}{m} \sum_{k=1}^{m} w_{ijk}. \tag{9.7}$$

Further,

$$\mathbf{B}(\mathbf{J}) \equiv \{ \frac{1}{m} \sum_{k=1}^{m} \mathbf{B}_k(\mathbf{J}) \} \mathbf{J}, \tag{9.8}$$

where $\mathbf{B}(\mathbf{J})$ has elements $\{b_{ij}\}$ defined by

$$b_{ij} = \begin{cases} -\hat{d}_{ij.} & \text{for } i \neq j \\ \sum_{l \neq i}^{n} \hat{d}_{il.} & \text{for } i = j \end{cases}, \tag{9.9}$$

with

$$\hat{d}_{ij.} = \frac{1}{m} \sum_{k=1}^{m} w_{ijk} \hat{d}_{ijk}. \tag{9.10}$$

Having obtained an unrestricted $(n - 1)$-dimensional update for $\mathbf{Z}$ using (9.4), we now want to restrict this update to a p-dimensional subspace. In section 5, we have seen that a convergent algorithm for the minimization of (9.1) over restricted matrix $\mathbf{Z}$ is generally obtained in the IDENTITY model by solving metric projection problem (4.4.4), that is, by solving

$$h(\mathbf{Z}) = \text{tr}\,(\mathbf{Z} - \bar{\mathbf{Z}})'\mathbf{V}(\mathbf{Z} - \bar{\mathbf{Z}}).$$

However, since we want to go from $(n - 1)$ to p dimensions in the present case, (4.4.4) should be adapted to

$$h(\mathbf{Z}, \mathbf{H}) = \text{tr}\,(\mathbf{ZH}' - \bar{\mathbf{Z}})'\mathbf{V}(\mathbf{ZH}' - \bar{\mathbf{Z}}), \tag{9.11}$$

where **H** denotes a columnwise orthonormal matrix of order $(n \times p)$ satisfying $\mathbf{H}'\mathbf{H} = \mathbf{I}_p$. In words, we have to find that p-dimensional matrix **Z** that, when rotated in $(n - 1)$-dimensional space, approximates $\bar{\mathbf{Z}}$ as well as possible in the least squares sense. Since this involves two sets of unknown parameters, an alternating least squares algorithm can be used to solve (9.11).

For fixed **Z**, we may write (9.11) as

$$h(\mathbf{H}) = d + \text{tr}\,\mathbf{HZ}'\mathbf{VZH}' - 2\,\text{tr}\,\bar{\mathbf{Z}}'\mathbf{VZH}' \tag{9.12}$$

where d is a constant with respect to **H**. Therefore, if **H** satisfies the orthonormality constraint, the minimization of (9.12) is equivalent to the maximization of

$$g(\mathbf{H}) \equiv \text{tr}\,\mathbf{H}'\bar{\mathbf{Z}}'\mathbf{VZ} = \frac{1}{\sqrt{2}}\,\text{tr}\,\mathbf{H}'\mathbf{B}(\mathbf{J})\mathbf{Z}, \tag{9.13}$$

Let

$$\mathbf{B}(\mathbf{J})\mathbf{Z} = \mathbf{PLQ}' \tag{9.14}$$

be a singular value decomposition of $\mathbf{B}(\mathbf{J})\mathbf{Z}$, then it is true that

$$g(\mathbf{H}) = \frac{1}{\sqrt{2}}\,\text{tr}\,\mathbf{H}'\mathbf{B}(\mathbf{J})\mathbf{Z} = \frac{1}{\sqrt{2}}\,\text{tr}\,\mathbf{H}'\mathbf{PLQ}' = \frac{1}{\sqrt{2}}\,\text{tr}\,\mathbf{Q}'\mathbf{H}'\mathbf{PL} \leq \frac{1}{\sqrt{2}}\,\text{tr}\,\mathbf{L}, \tag{9.15}$$

since $\mathbf{Q}'\mathbf{H}'\mathbf{P}$ is an orthonormal matrix, being the product of orthonormal matrices $\mathbf{Q}'$, $\mathbf{H}'$, and **P**. The upper limit in (9.15) is attained for

$$\mathbf{Q}'\mathbf{H}'\mathbf{P} = \mathbf{I}, \tag{9.16}$$

and thus for

$$\mathbf{H} = \mathbf{PQ}'. \tag{9.17}$$

The solution satisfies the orthonormality constraints since $\mathbf{H'H} = \mathbf{QP'PQ'} = \mathbf{I}$.

For fixed **H**, (9.11) may be written as

$$g(\mathbf{Z}) = \text{tr}\ \bar{\mathbf{Z}}'\mathbf{V}\bar{\mathbf{Z}} + \text{tr}\ \mathbf{HZ'VZH'} - 2\ \text{tr}\ \bar{\mathbf{Z}}'\mathbf{VZH'}$$

$$= \text{tr}\ \bar{\mathbf{Z}}'\mathbf{V}\bar{\mathbf{Z}} - \text{tr}\ \mathbf{H}'\bar{\mathbf{Z}}'\mathbf{V}\bar{\mathbf{Z}}\mathbf{H} + \text{tr}\ \mathbf{H}'\bar{\mathbf{Z}}'\mathbf{V}\bar{\mathbf{Z}}\mathbf{H} + \text{tr}\ \mathbf{Z'VZ} - 2\ \text{tr}\ \mathbf{H}'\bar{\mathbf{Z}}'\mathbf{VZ}$$

$$= e + \text{tr}\ (\mathbf{Z} - \bar{\mathbf{Z}}\mathbf{H})'\mathbf{V}(\mathbf{Z} - \bar{\mathbf{Z}}\mathbf{H}), \tag{9.18}$$

where e is a term independent of **Z**. Therefore, if **Z** is not restricted the global minimum of (9.18) is attained for

$$\mathbf{Z} = \bar{\mathbf{Z}}\mathbf{H} = \frac{1}{\sqrt{2}}\mathbf{V}^{-}\mathbf{B}(\mathbf{J})\mathbf{H}. \tag{9.19}$$

If the common space is not restricted we may alternate between (9.14) and (9.19) until the value of (9.11) no longer improves beyond some prechosen criterion. However, we still need to have initial values for **H** and **Z** to start iterations. Heiser (1985) proposes to initialize $\mathbf{H}^0$ by setting p of its rows equal to the rows of $\mathbf{I}_p$, and the remaining rows equal to zero, taking care that the nonzero rows of $\mathbf{H}^0$ are selected in such a way that the very first $\mathbf{Z}^0 = \mathbf{B}(\mathbf{J})\mathbf{H}^0$ contains the p columns of **B(J)** with largest diagonal elements.

It is of some interest to note that the global minimization of (9.11) over orthonormal matrices **H** and unrestricted matrices **Z** can be solved analytically, as follows. Substitution of (9.19) in (9.11) yields

$$h(\mathbf{H}) = \text{tr}\ (\bar{\mathbf{Z}}\mathbf{HH}' - \bar{\mathbf{Z}})'\mathbf{V}(\bar{\mathbf{Z}}\mathbf{HH}' - \bar{\mathbf{Z}})$$

$$= f - \text{tr}\ \mathbf{H}'\bar{\mathbf{Z}}'\mathbf{V}\bar{\mathbf{Z}}\mathbf{H}, \tag{9.20}$$

where f is independent of **H**. The minimization of (9.20) is equivalent to the maximization of

$$m(\mathbf{H}) = \text{tr}\ \mathbf{H}'\bar{\mathbf{Z}}'\mathbf{V}\bar{\mathbf{Z}}\mathbf{H}, \tag{9.21}$$

subject to $\mathbf{H'H} = \mathbf{I}_p$. The solution is standard. Let

$$\bar{\mathbf{Z}}'\mathbf{V}\bar{\mathbf{Z}} = \mathbf{WDW}' \tag{9.22}$$

be an eigenvalue decomposition of the matrix product $\bar{\mathbf{Z}}'\mathbf{V}\,\bar{\mathbf{Z}}$ of order (n x n), and let $\mathbf{W}_p$ denote the (n x p) matrix containing the p principal eigenvectors of $\mathbf{W}$, the global maximum of (9.21) is attained where

$$\hat{\mathbf{H}} = \mathbf{W}_p. \tag{9.23}$$

Then, the optimal unrestricted common space is equal to

$$\mathbf{Z}^+ = \bar{\mathbf{Z}}\hat{\mathbf{H}} = \frac{1}{\sqrt{2}}\mathbf{V}^-\mathbf{B}(\mathbf{J})\hat{\mathbf{H}}. \tag{9.24}$$

The reason that we use alternating least squares instead of (9.23) and (9.24) to minimize (9.11) is that it results in a more flexible general approach which makes it easy to incorporate restrictions on the common space.

Obviously, if restrictions are imposed on the common space, then we are forced to use alternating least squares to solve metric projection problem (9.11), and the minimization of (9.18) has to be adapted accordingly. Specifically, if some coordinates of the common space are required to be FIXED, then the corresponding elements of $\mathbf{Z}^0$ are replaced with the fixed values, and the remaining free elements are updated coordinate by coordinate. Defining z_{ia} as element ia of matrix $\mathbf{Z}$ (i = 1, ..., n; a = 1, ..., p), $\tilde{\mathbf{z}}_{ia}$ as the (n x 1) vector equal to the a-th column of $\mathbf{Z}$ except that its i-th element is equal to zero, $\mathbf{e}_i$ as the i-th column of the identity matrix $\mathbf{I}_n$, and $\mathbf{e}_a$ as the a-th column of the identity matrix $\mathbf{I}_p$, the coordinatewise update formula for the minimization of (9.18) is

$$z_{ia}^+ = \frac{1}{\mathbf{e}_i'\mathbf{V}\mathbf{e}_i}\,\mathbf{e}_i'\Big[\frac{1}{\sqrt{2}}\mathbf{B}(\mathbf{J})\mathbf{H}\mathbf{e}_a - \mathbf{V}\tilde{\mathbf{z}}_{ia}\Big]. \tag{9.25}$$

We omit the proof, since it is completely analogous to the one given in section 6. If *all* coordinates of $\mathbf{Z}$ are fixed, then, of course, the whole initialization procedure for $\mathbf{Z}$ should be skipped.

Another possible restriction is that the common space must be a linear combination of EXTERNAL variables. In that case, letting h = maximum(s, p), where s is the number of external variables, we have to solve

$$\begin{aligned} g(\mathbf{Q}, \mathbf{B}) &= \mathrm{e} + \mathrm{tr}\,(\mathbf{QB} - \bar{\mathbf{Z}}\mathbf{H})'\mathbf{V}(\mathbf{QB} - \bar{\mathbf{Z}}\mathbf{H}) \\ &= \mathrm{e} + \mathrm{tr}\,(\sum_{j=1}^{h} \mathbf{q}_j\mathbf{b}_j' - \bar{\mathbf{Z}}\mathbf{H})'\mathbf{V}(\sum_{j=1}^{h} \mathbf{q}_j\mathbf{b}_j' - \bar{\mathbf{Z}}\mathbf{H}) \end{aligned} \tag{9.26}$$

where $\mathbf{Q}$ is a matrix of order (n x h) whose first s columns contains the quantified external variables and whose last (p–s) columns, if any, are free, $\mathbf{B}$ is the (h x p) matrix of regression weights, and $\mathbf{q}_j$ and $\mathbf{b}_j$ are defined as in section 7. Again, we have to provide initial values for $\mathbf{Q}$ and $\mathbf{B}$. This may be done by filling the first s columns of $\mathbf{Q}$ with the external variables centered on the origin and normalized on fixed length, and the last (p–s) columns, if any, with a centered and normalized arbitrary full rank matrix of order (n x (p–s)), and then using as intitial estimates for $\mathbf{B}$:

$$\mathbf{B} = \frac{1}{\sqrt{2}}(\mathbf{Q}'\mathbf{V}\mathbf{Q})^{-1}\mathbf{Q}'\mathbf{B}(\mathbf{J})\mathbf{H}, \tag{9.27}$$

which globally minimizes g(*, $\mathbf{B}$) under the assumption that all variables are numerical. With these initial estimates we may proceed to minimize (9.26) variable by variable using the following update formulas. For the regression weights corresponding to variable j (j = 1, ..., h) we have

$$\mathbf{b}_j = \frac{1}{\mathbf{q}_j'\mathbf{V}\mathbf{q}_j}\left(\frac{1}{\sqrt{2}}\mathbf{B}(\mathbf{J})\mathbf{H} - \mathbf{V}\mathbf{U}_j\right)'\mathbf{q}_j, \tag{9.28}$$

where

$$\mathbf{U}_j = \sum_{t \neq j} \mathbf{q}_t\mathbf{b}_t'.$$

For fixed regression weights, and only considering one quantification vector $\mathbf{q}_j$, we have to minimize

$$g(*, \mathbf{q}_j) = c_j + \mathbf{b}_j'\mathbf{b}_j\,(\mathbf{q}_j - \mathbf{T}_j\mathbf{b}_j)'\mathbf{V}(\mathbf{q}_j - \mathbf{T}_j\mathbf{b}_j), \tag{9.29}$$

where

$$\mathbf{T}_j \equiv \frac{1}{\mathbf{b}_j'\mathbf{b}_j}(\bar{\mathbf{Z}}\mathbf{H} - \mathbf{U}_j) \tag{9.30}$$

and c_j is a term independent of $\mathbf{q}_j$. Letting ϕ_1 denote the largest eigenvalue of matrix $\mathbf{V}$, (9.29) can be minimized with majorization, that is, by alternatingly calculating

$$\tilde{\mathbf{q}}_j = \mathbf{q}_j^0 + \frac{1}{\phi_1}\mathbf{V}(\mathbf{T}_j\mathbf{b}_j - \mathbf{q}_j^0)$$

$$= \frac{1}{\phi_1 \mathbf{b}_j'\mathbf{b}_j} \left(\frac{1}{\sqrt{2}} \mathbf{B}(\mathbf{J})\mathbf{H} - \mathbf{V}\mathbf{U}_j)\mathbf{b}_j + (\mathbf{I} - \frac{1}{\phi_1} \mathbf{V})\mathbf{q}_j^0, \right. \tag{9.31}$$

and then finding a new $\mathbf{q}_j^+$ minimizing the function

$$\mathrm{q}(\mathbf{q}_j) = (\mathbf{q}_j - \tilde{\mathbf{q}}_j)'(\mathbf{q}_j - \tilde{\mathbf{q}}_j) \tag{9.32}$$

subject to the constraints appropriate for $\mathbf{q}_j$. We refer to section 7 for details concerning the computation of updates minimizing (9.32) for numerical, ordinal, and nominal variables. After all variables and all regression weights have been updated using the above procedures,

$$\mathbf{Z} = \sum_{j=1}^{h} \mathbf{q}_j \mathbf{b}_j', \tag{9.33}$$

yields a first estimation for the common space in p dimensions satisfying the external constraints.

If the common space is not restricted, we discussed above how an update for $\mathbf{Z}$ can be obtained using (9.19). This updating scheme has the disadvantage that it requires the computation of the MP-inverse of the $(n \times n)$ matrix $\mathbf{V}$ defined in (9.6). Therefore, we will now discuss an alternative method to update $\mathbf{Z}$ which does not require the calculation of this inverse, and is based on the majorization procedure described in section 5.

For fixed $\mathbf{H}$, we have to minimize (9.18), that is,

$$\mathrm{g}(\mathbf{Z}) = \mathrm{e} + \mathrm{tr}\,(\mathbf{Z} - \bar{\mathbf{Z}}\mathbf{H})'\mathbf{V}(\mathbf{Z} - \bar{\mathbf{Z}}\mathbf{H})$$

over unrestricted $(n \times p)$ matrices $\mathbf{Z}$. Letting $\mathbf{z}_a$ be the a-th column of $\mathbf{Z}$ $(a = 1,\ldots,p)$, $\mathbf{e}_a$ be the a-th column of the identity matrix $\mathbf{I}_p$, and $\mathbf{P}_a$ be the $(n \times p)$ matrix equal to $\mathbf{Z}$ but with the a-th column containing zeroes, then

$$\mathbf{Z} = \mathbf{P}_a + \mathbf{z}_a\mathbf{e}_a'. \tag{9.34}$$

Substitution of (9.34) in (9.18) yields

$$\mathrm{g}(\mathbf{z}_a) = \mathrm{e} + \mathrm{tr}\,\{(\mathbf{P}_a + \mathbf{z}_a\mathbf{e}_a') - \bar{\mathbf{Z}}\mathbf{H}\}'\mathbf{V}\{(\mathbf{P}_a + \mathbf{z}_a\mathbf{e}_a') - \bar{\mathbf{Z}}\mathbf{H}\}$$

$$= \mathrm{e}^* + \mathbf{z}_a'\mathbf{V}\mathbf{z}_a - 2\,\mathbf{z}_a'\frac{1}{\sqrt{2}}\mathbf{B}(\mathbf{J})\mathbf{H}\mathbf{e}_a, \tag{9.35}$$

where e^* is a term independent of $\mathbf{z}_a$. Letting $\mathbf{q}_a = (\sqrt{2})^{-1}\mathbf{B}(\mathbf{J})\mathbf{H}\mathbf{e}_a$ (i.e., $\mathbf{q}_a$ is the a-th column of matrix $(\sqrt{2})^{-1}\mathbf{B}(\mathbf{J})\mathbf{H}$), and $\mathbf{x}_a$ be an (n x 1) vector satisfying $\mathbf{q}_a = \mathbf{V}\mathbf{x}_a$, we may write (9.35) as

$$g(\mathbf{z}_a) = e^* + \mathbf{z}_a'\mathbf{V}\mathbf{z}_a - 2\,\mathbf{z}_a'\mathbf{q}_a$$

$$= e^* - \mathbf{x}_a'\mathbf{V}\mathbf{x}_a + \mathbf{z}_a'\mathbf{V}\mathbf{z}_a + \mathbf{x}_a'\mathbf{V}\mathbf{x}_a - 2\,\mathbf{z}_a'\mathbf{V}\mathbf{x}_a$$

$$= f + (\mathbf{z}_a - \mathbf{x}_a)'\mathbf{V}(\mathbf{z}_a - \mathbf{x}_a), \qquad (9.36)$$

where f is another constant with respect to $\mathbf{z}_a$. Letting k_1 denote the largest eigenvalue of $\mathbf{V}$, and $\mathbf{z}_a^j$ the current best estimate of $\mathbf{z}_a$, updates for dimension a of $\mathbf{Z}$ are obtained by repeatedly calculating

$$\mathbf{z}_a^{j+1} = \mathbf{z}_a^j + \frac{1}{k_1}\mathbf{V}(\mathbf{x}_a - \mathbf{z}_a^j)$$

$$= \mathbf{z}_a^j + \frac{1}{k_1}\left(\frac{1}{\sqrt{2}}\mathbf{B}(\mathbf{J})\mathbf{H}\mathbf{e}_a - \mathbf{V}\mathbf{z}_a^j\right). \qquad (9.37)$$

This can be applied to each dimension of $\mathbf{Z}$ separately. Instead of actually calculating the largest eigenvalue of $\mathbf{V}$, we may compute its upper bound using the procedure proposed by Wolkowicz and Styan (1980) (cf., section 5).

Since $\mathbf{V} = \frac{1}{m}\Sigma_k\mathbf{V}_k = \frac{1}{m}\Sigma_k n\mathbf{J} = n\mathbf{J}$ when all weights w_{ijk} are equal to 1, and because the largest eigenvalue of $n\mathbf{J}$ is equal to n, (9.37) simplifies into

$$\mathbf{z}_a = \frac{1}{n\sqrt{2}}\mathbf{B}(\mathbf{J})\mathbf{H}\mathbf{e}_a \qquad (9.38)$$

in this special situation, and no iterations are needed to obtain an update for $\mathbf{Z}$.

10 Acceleration schemes

In sections 2 and 3 we have discussed how the main PROXSCAL algorithm uses majorization to approximate the (dis)similarity data with low-dimensional distances. Unfortunately, for all its beautiful simplicity, this majorization algorithm has a very slow linear convergence rate, as has been discussed by De Leeuw and Heiser (1980), and De Leeuw (1988). Theoretically, there are two methods which can be used to speed up the convergence rate of the PROXSCAL majorization algorithm. The first method is to design another majorization function which is 'closer' to the original STRESS function. Kiers and ten Berge (1992) have shown how this may be achieved for a certain class of matrix functions. However, apart from the problem how to determine a 'better fitting' majorization function, once such a function has been found, the metric projection problems associated with restricted solutions also tend to become more complicated.

The second method to speed up convergence is to use a *stepsize parameter*. Here, we will only discuss this second method for unrestricted solutions; a method for restricted solutions will be treated elsewhere. In section 2, it was shown that the minimization of

$$\begin{aligned} f(\mathbf{X}_k) &= \sum_{i<j}^{n} w_{ijk}[\delta_{ijk} - d_{ij}(\mathbf{X}_k)]^2 \\ &= c_k + \operatorname{tr} \mathbf{X}_k'\mathbf{V}_k\mathbf{X}_k - 2 \operatorname{tr} \mathbf{X}_k'\mathbf{B}(\mathbf{X}_k)\mathbf{X}_k \end{aligned}$$

over unrestricted (n x p) matrix $\mathbf{X}_k$ can be solved iteratively by repeatedly calculating the Guttman transform defined as in (2.12):

$$\bar{\mathbf{X}}_k = \mathbf{V}_k^{-}\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0.$$

where $\mathbf{X}_k^0$ denotes the current solution for $\mathbf{X}_k$. The latter update globally minimizes majorizing function (2.11), that is,

$$g(\mathbf{X}_k, \mathbf{X}_k^0) = c_k + \operatorname{tr} \mathbf{X}_k'\mathbf{V}_k\mathbf{X}_k - 2 \operatorname{tr} \mathbf{X}_k'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0,$$

and satisfies inequality (2.18), that is,

$$f(\bar{\mathbf{X}}_k) \leq g(\bar{\mathbf{X}}_k, \mathbf{X}_k^0) \leq f(\mathbf{X}_k^0),$$

which proves convergence.

Now, define the *relaxed update* $\hat{\mathbf{X}}_k$ as

$$\hat{\mathbf{X}}_k = (1 - \alpha)\mathbf{X}_k^0 + \alpha\bar{\mathbf{X}}_k \tag{10.1}$$

where α is a stepsize parameter and $\bar{\mathbf{X}}_k$ is the Guttman transform. We have the following two special cases. For $\alpha = 0$, $\hat{\mathbf{X}}_k = \mathbf{X}_k^0$ (i.e., no change or improvement at all), and for $\alpha = 1$, $\hat{\mathbf{X}}_k = \bar{\mathbf{X}}_k$ (i.e., the Guttman transform). We will now determine for what values of α the majorization method remains convergent. By definition,

$$f(\hat{\mathbf{X}}_k) \leq g(\hat{\mathbf{X}}_k, \mathbf{X}_k^0). \tag{10.2}$$

For fixed $\mathbf{X}_k^0$, the value of the majorizing function at $\mathbf{X}_k = \hat{\mathbf{X}}_k$ is equal to

$$\begin{aligned} g(\hat{\mathbf{X}}_k, \mathbf{X}_k^0) &= c_k + \text{tr}\, \hat{\mathbf{X}}_k'\mathbf{V}_k\hat{\mathbf{X}}_k - 2\,\text{tr}\, \hat{\mathbf{X}}_k'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \\ &= c_k + \text{tr}\, [(1-\alpha)\mathbf{X}_k^0 + \alpha\bar{\mathbf{X}}_k]'\mathbf{V}_k[(1-\alpha)\mathbf{X}_k^0 + \alpha\bar{\mathbf{X}}_k] \\ &\qquad - 2\,\text{tr}\, [(1-\alpha)\mathbf{X}_k^0 + \alpha\bar{\mathbf{X}}_k]'\mathbf{B}(\mathbf{X}_k^0)\mathbf{X}_k^0 \\ &= f(\mathbf{X}_k^0) + \alpha(\alpha - 2)\,\text{tr}\, (\mathbf{X}_k^0 - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k^0 - \bar{\mathbf{X}}_k). \end{aligned} \tag{10.3}$$

Convergence is guaranteed if

$$g(\hat{\mathbf{X}}_k, \mathbf{X}_k^0) \leq f(\mathbf{X}_k^0), \tag{10.4}$$

and, therefore, if

$$\alpha(\alpha - 2)\,\text{tr}\, (\mathbf{X}_k^0 - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k^0 - \bar{\mathbf{X}}_k) \leq 0. \tag{10.5}$$

Since $\text{tr}\, (\mathbf{X}_k^0 - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k^0 - \bar{\mathbf{X}}_k) \geq 0$ for arbitrary matrices $\mathbf{X}_k^0$ and $\bar{\mathbf{X}}_k$, it follows from (10.5) that the algorithm remains convergent for all stepsizes satisfying $0 \leq \alpha \leq 2$. We have three special cases. Because (10.3) may also be written as

$$g(\hat{\mathbf{X}}_k, \mathbf{X}_k^0) = c_k - \text{tr}\, \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k + (1-\alpha)^2\text{tr}\, (\mathbf{X}_k^0 - \bar{\mathbf{X}}_k)'\mathbf{V}_k(\mathbf{X}_k^0 - \bar{\mathbf{X}}_k), \tag{10.6}$$

as is easily verified, for $\alpha = 1$, $\hat{\mathbf{X}}_k = \bar{\mathbf{X}}_k$ and (10.6) reduces to

$$g(\bar{\mathbf{X}}_k, \mathbf{X}_k^0) = c_k - \text{tr}\, \bar{\mathbf{X}}_k'\mathbf{V}_k\bar{\mathbf{X}}_k,$$

which is the global minimum of the majorization function (see also (2.14) in section 2). Moreover, it follows from (10.3) that if $\alpha = 0$ *and* if $\alpha = 2$, then $g(\hat{\mathbf{X}}_k, \mathbf{X}_k^0) = f(\mathbf{X}_k^0)$, i.e., the value of the majorizing function and of the STRESS function are identical in both cases.

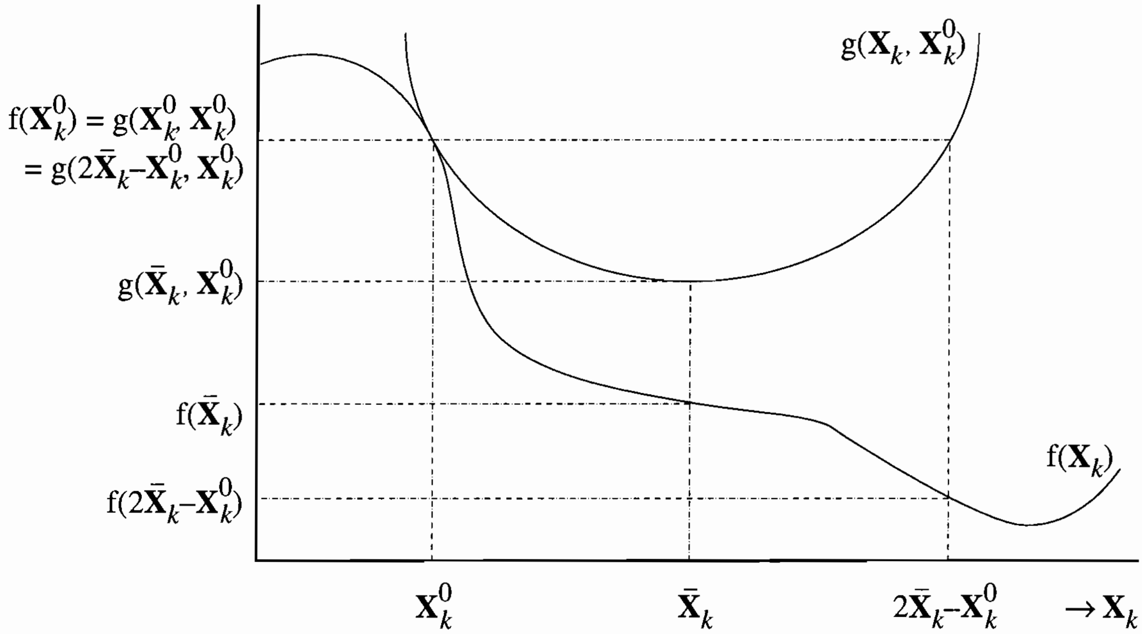

Figure 3 Illustration of majorization using relaxed updates for stepsizes $\alpha = 0$, $\alpha = 1$, and $\alpha = 2$.

At the same time, $\hat{\mathbf{X}}_k = \mathbf{X}_k^0$ for $\alpha = 0$, and $\hat{\mathbf{X}}_k = 2\bar{\mathbf{X}}_k - \mathbf{X}_k^0$ for $\alpha = 2$. These seemingly contradictory results are illustrated in Figure 3. The figure shows why the relaxed updates $\hat{\mathbf{X}}_k = \mathbf{X}_k^0$ (corresponding to $\alpha = 0$) and $\hat{\mathbf{X}}_k = 2\bar{\mathbf{X}}_k - \mathbf{X}_k^0$ (corresponding to $\alpha = 2$), although being completely different, still yield identical values of the majorizing function. Geometrically, a stepsize of $\alpha = 2$ involves crossing all the way to the 'other side' of the function $g(\mathbf{X}_k, \mathbf{X}_k^0)$. De Leeuw and Heiser (1980) discussed that the use of a stepsize of $\alpha =$ 2 (that is, of the relaxed update $\hat{\mathbf{X}}_k = 2\bar{\mathbf{X}}_k - \mathbf{X}_k^0$) approximately squares the convergence rate of the majorization algorithm, and halves the number of iterations. Intuitively, this can be deduced from Figure 3: on the whole, such a stepsize yields larger steps, and thus also larger improvements of the STRESS function value.

11 Normalized STRESS

An objective measure of fit, which allows for comparisons between different STRESS solutions, is the following. First note that the scalar β defined as

$$\beta = \frac{\Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk} d_{ij}(\mathbf{X}_k)}{\Sigma_k \Sigma_{i<j} w_{ijk} d_{ij}^2(\mathbf{X}_k)}, \qquad (11.1)$$

is the global minimizer of STRESS loss function (1.1)

$$f(\mathbf{X}_1, \ldots, \mathbf{X}_m; \beta) \equiv \frac{1}{m} \sum_{k=1}^{m} \sum_{i<j}^{n} w_{ijk}[\delta_{ijk} - \beta\, d_{ij}(\mathbf{X}_k)]^2 \qquad (11.2)$$

with respect to β. Substituting (11.1) in (11.2) we have

$$f(\mathbf{X}_1, \ldots, \mathbf{X}_m) = \frac{1}{m} \Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk}^2 - \frac{1}{m} \frac{\{\Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk} d_{ij}(\mathbf{X}_k)\}^2}{\Sigma_k \Sigma_{i<j} w_{ijk} d_{ij}^2(\mathbf{X}_k)}, \qquad (11.3)$$

and dividing (11.3) by $\frac{1}{m} \Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk}^2$ we obtain

$$\frac{f(\mathbf{X}_1, \ldots, \mathbf{X}_m)}{\frac{1}{m} \Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk}^2} + \frac{\{\Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk} d_{ij}(\mathbf{X}_k)\}^2}{\{\Sigma_k \Sigma_{i<j} w_{ijk} d_{ij}^2(\mathbf{X}_k)\}\{\Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk}^2\}} = 1, \qquad (11.4)$$

where the term

$$\frac{f(\mathbf{X}_1, \ldots, \mathbf{X}_m)}{\frac{1}{m} \Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk}^2} \qquad (11.5)$$

is *normalized* STRESS, and

$$\phi^2 \equiv \frac{\{\Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk} d_{ij}(\mathbf{X}_k)\}^2}{\{\Sigma_k \Sigma_{i<j} w_{ijk} d_{ij}^2(\mathbf{X}_k)\}\{\Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk}^2\}} \qquad (11.6)$$

is the square of Tucker's coefficient of congruence ϕ between the elements in $\{w_{ijk}^{1/2} d_{ij}(\mathbf{X}_k)\}$ and in $\{w_{ijk}^{1/2} \delta_{ijk}\}$. Moreover, since we use the explicit normalization $\frac{1}{m} \Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk}^2 = s$ in PROXSCAL (see also sections 8 and 9), (11.4) may be written in the convenient form

$$\frac{1}{s} f(\mathbf{X}_1, \ldots, \mathbf{X}_m) + \frac{\{\Sigma_k \Sigma_{i<j} w_{ijk} \delta_{ijk} d_{ij}(\mathbf{X}_k)\}^2}{m s \Sigma_k \Sigma_{i<j} w_{ijk} d_{ij}^2(\mathbf{X}_k)} = 1. \qquad (11.7)$$

The squared coefficient of congruence of Tucker (last term on the left hand side of (11.7)) and its counterpart, normalized STRESS (first term on the left hand side of (11.7)), are not sensitive to differences in scale of the configurations, to differences in weights, or to differences in the numbers of objects. Therefore, in practice we use normalized STRESS to monitor convergence of the PROXSCAL algorithms as well as to compare the fit of different PROXSCAL solutions.

References

De Leeuw, J. (1988). Convergence of the majorization method for multidimensional scaling. *Journal of Classification, 5,* 163-180.

De Leeuw, J. & Heiser, W.J. (1977). Convergence of correction-matrix algorithms for multidimensional scaling. In: Lingoes et al. (Eds). *Geometric representations of relational data.* Ann Arbor: Mathesis Press.

De Leeuw, J. & Heiser, W.J. (1980). Multidimensional scaling with restrictions on the configuration. In: Krishnaiah, P.R. (Ed.). *Multivariate analysis, Vol V.* Amsterdam: North Holland Publishing company.

Gifi, A. (1990). *Nonlinear multivariate analysis.* Chichester: Wiley.

Heiser, W.J. (1985a). *Multidimensional scaling by optimizing goodness-of-fit to a smooth hypothesis.* Research Report RR-85-07. Leiden: Department of Data Theory.

Heiser, W.J. (1985b). *A general MDS initialization procedure using the SMACOF algorithm model with constraints.* Research Report RR-85-23. Leiden: Department of Data Theory.

Heiser, W.J. (1988). PROXSCAL, multidimensional scaling of proximities. In: A. Di Ciaccio and G. Bove (Eds.), *MULTIWAY 1988, International meeting on the analysis of multiway data matrices, Software guide.* Rome: C.N.R., 77-81.

Heiser, W.J. & De Leeuw, J. (1986). *How to use SMACOF–I.* Research Report RR-86-02. Leiden: Department of Data Theory.

Heiser, W.J. & Stoop, I. (1986). *Explicit SMACOF algorithms for individual differences scaling.* Research Report RR-86-04. Leiden: Department of Data Theory.

Kiers, H. (1990). Majorization as a tool for optimizing a class of matrix functions. *Psychometrika, 55,* 417-428.

Kiers, H. & ten Berge, J.M.F. (1992). Minimization of a class of matrix trace functions by means of refined majorization. *Psychometrika, 57,* 371-382.

Kruskal, J.B. (1964). Nonmetric multidimensional scaling. *Psychometrika, 29,* 115-129.

Kruskal, J.B. (1977). Multidimensional scaling and other methods for discovering structure. In: *Mathematical methods for digital computers, Vol III.* New York: Wiley.

Lingoes, J.C. & Roskam, E.E. (1973). A mathematical and empirical analysis of two multidimensional scaling algorithms. *Psychometrika, 38.* Monograph supplement.

Meulman, J. & Heiser, W.J. (1984). Constrained multidimensional scaling: more directions than dimensions. In: T. Havránek et al. (Eds.), *COMPSTAT 1984, Proceedings in computational statistics.* Vienna: Physica Verlag, 137-142.

Stoop, I., Heiser, W.J. & De Leeuw, J. (1981). *How to use SMACOF–IA*. Leiden: Department of Data Theory.

Stoop, I. & De Leeuw, J. (1982). *How to use SMACOF–IB*. Leiden: Department of Data Theory.

Takane, Y., Young, F.W. & De Leeuw, J. (1977). Nonmetric individual differences MDS: An alternating least squares method with optimal scaling features. *Psychometrika, 42,* 7-67.

Wilkinson, J.H. (1965). *The algebraic eigenvalue problem.* Oxford: Clarendon Press.

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

Young, F.W. & Hamer, R.M. (1987). *Multidimensional scaling. History, theory, and applications.* Hillsdale, New Jersey: Lawrence Erlbaum Associates, Publishers.