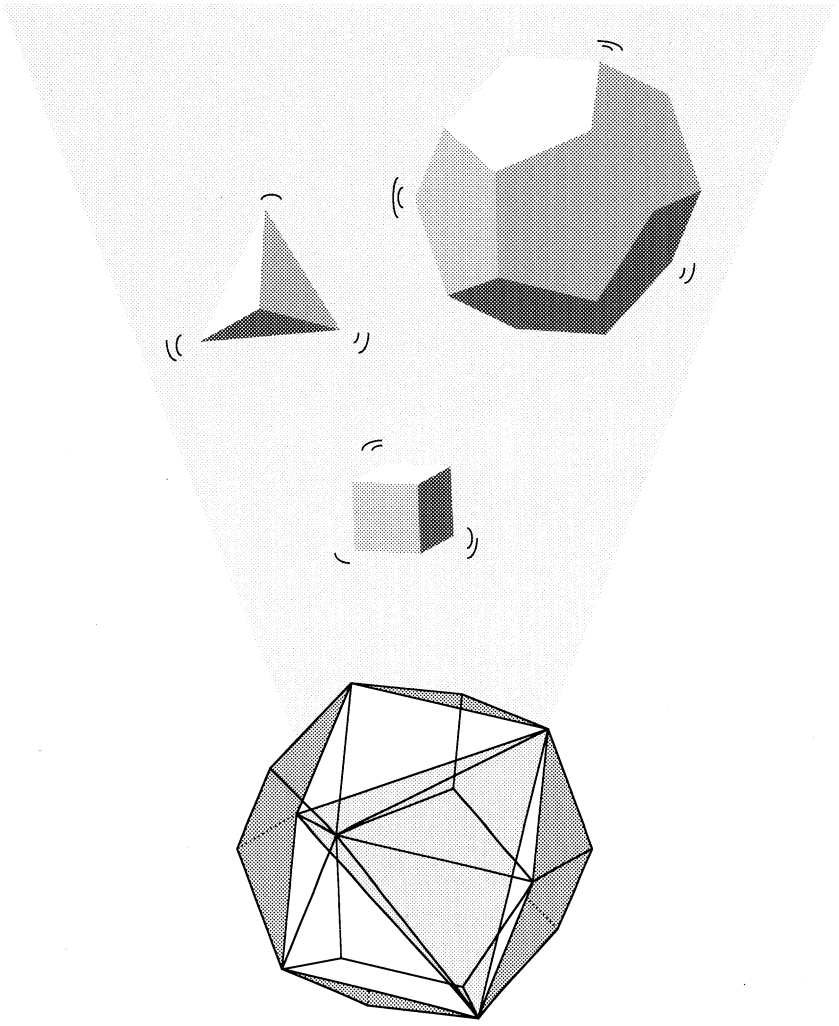
MATCHING CONFIGURATIONS

Jacques J.F. Commandeur



DSWO PRESS

MATCHING CONFIGURATIONS

M&T SERIES 19

Available from DSWO Press
Leiden University
P.O.Box 9555
2300 RB Leiden
The Netherlands
Tel. (071) 273795 / 273794

Earlier publications in this series:

Jacqueline Meulman, Homogeneity analysis of incomplete data.
M&T series 1, 1982

Pieter M. Kroonenberg, Tree-mode principle component analysis.
M&T series 2, 1983, reprint 1989

Jan de Leeuw, Canonical analysis of categorical data.
M&T series 3, 1984

Ronald A. Visser, Analysis of longitudinal data in behavioural and social research
M&T series 4, 1985

John P. van de Geer, Introduction to linear multivariate data analysis.
M&T series 5, volume 1 & 2, 1986

Jacqueline Meulman, A distance approach to nonlinear multivariate analysis.
M&T series 6, 1986

Jan de Leeuw, Willem Heiser, Jacqueline Meulman, Frank Critchley (editors), Multidimensional data analysis.
M&T series 7, 1987

Peter G.M. van der Heijden, Correspondence analysis of longitudinal categorical data.
M&T series 8, 1987

Jan van Rijckevorsel, The application of fuzzy coding and horseshoes in multiple correspondence analysis.
M&T series 9, 1987

Abby Israëls, Eigenvalue techniques for qualitative data.
M&T series 10, 1987

Eeke van der Burg, Nonlinear canonical correlation and some related techniques.
M&T series 11, 1988

Kees van Montfort, Estimating in structural models with non-normal distributed variables: some alternative approaches.
M&T series 12, 1989

Jan T. A. Koster, Mathematical aspects of multiple correspondence analysis for ordinal variables.
M&T series 13, 1989

Catrien C. J. H. Bijleveld, Exploratory linear dynamic systems analysis.
M&T series 14, 1989

Henk A. L. Kiers, Three-way methods for the analysis of qualitative and quantitative two-way data.
M&T series 15, 1989

Stef van Buuren, Optimal Scaling of Time Series.
M&T Series 16, 1990

Rien van der Leeden, Reduced rank regression with structured residuals.
M&T Series 17, 1990.

Rian van Blokland-Vogelesang, Unfolding and group consensus ranking for individual preferences.
M&T Series 18, 1991.

Jacques J.F. Commandeur, Matching Configurations.
M&T Series 19, 1991

MATCHING CONFIGURATIONS

JACQUES J.F. COMMANDEUR

DEPARTMENT OF PSYCHOMETRICS AND RESEARCH METHODOLOGY
LEIDEN UNIVERSITY

1991 DSWO Press, Leiden University

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Commandeur, Jacques J.F.

Matching Configurations / Jacques J.F. Commandeur. — Leiden: DSWO Press, Leiden University. — Ill. — M&T series; 19)
Also published as thesis Leiden, 1991. — With ref. —With summary in Dutch.
ISBN 90-6695-052-8
SISO 517 UDC 51-7
Subject headings: matching / alternating least squares.

This book was written on a Macintosh Plus and SE/30, using the Laser Author and MacWrite II wordprocessors. Proofs and final prints were produced on a LaserWriter II NTX. The text was set in Times.
Special symbols were taken from the Symbol and Bartholomew fonts.
Graphics were created with MacDraw, MacDraw II and Superpaint.
Programming and computations were done in APL.68000.

Cover drawing: Jacques J.F. Commandeur
Printed by 'Reprodienst Faculteit Sociale Wetenschappen',
Rijksuniversiteit Leiden and by printing office 'Karstens drukkers b.v.'
Leiden'

ISBN 90-6695-052-8

To Spip

Preface

This book is intended for anyone interested in the subject of comparing two or more configurations. The book starts out from the PINDIS framework, and discusses five models for comparing configurations containing the coordinates of the same objects in a number of dimensions. With these models one can investigate the similarity of configurations, as well as the existence of a number of systematic patterns in the differences between configurations. The book introduces improved estimation procedures of the corresponding model parameters, and provides a generalized framework allowing one to compare configurations even when they contain information about different numbers of objects.

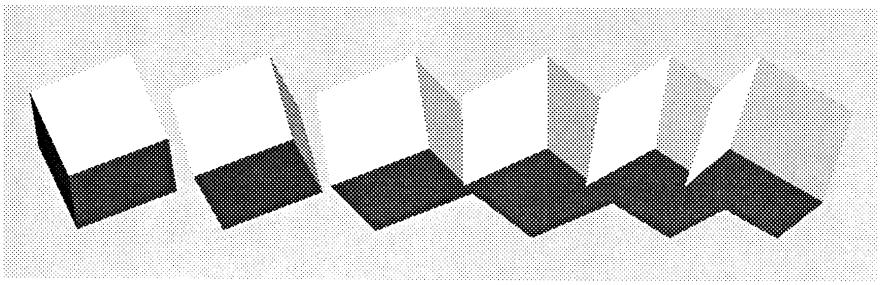
Many people helped in the realization of this book. I first would like to mention all the people referenced at the end of the book. They, and many others, have erected the edifice upon which I could continue to build. On a more personal note, I am greatly indebted to Wim van der Kloot. He taught me the basics of data analysis, aroused my interest in and enthousiasm for this branch of science, and put me on the track of the subject of matching configurations. Both Wim van der Kloot and Leo van der Kamp gave me the opportunity and time to develop the algorithms presented in this book, supported me, scientifically and otherwise, throughout the project, and gave many helpful comments on all earlier drafts of the manuscript. John van de Geer was instrumental in shedding a new light on my view of data analysis. Due to his geometrical approach, as well as to his eagle-eye, I became aware of an inherent order in the previously bewildering chaos of data analysis methods. I thank him for his comments on a previous version of the book. The contribution of Jos ten Berge to the present work has been invaluable. He suggested many important improvements and was always quick as lightning in answering any questions I had. I learned a lot from his extensive knowledge of the algorithmic aspects of data analysis. Most importantly, he taught me the lazy man's way to optimization, as is witnessed by the complete absence of calculus in the present study. Ab Mooijaart, Willem Heiser, and Pieter Kroonenberg gave helpful comments on earlier versions of the manuscript. Countless times Wim den Brinker solved problems I had with computer software, A Programming Language in particular. Job van de Raad and Ali Flohr were always

prepared to help me, and Lutgart Delvaux gave very useful advice concerning the lay-out of this book.

Renée Verdegaal, Peter Verboon, and Frits Meijerink stood by me through thick and thin. I thank them for their kindness and friendship. With Rien van der Leeden I continuously shared a røøm for no less than eight years. Among lots of other things, I thank him for the fact that, although he had to put up with my growing demands for peace and quiet in the røøm, this did not affect our friendship in the least.

Jacques Commandeur
Leiden, december 1990

Contents



Chapter 1

Introduction

1.1 Preliminaries

In the past few decades a large class of data analysis techniques have emerged resulting in a spatial representation of data. The general aim of such a spatial representation is to aid the researcher in understanding the relationships between the observational units, or objects, about which information has been gathered, especially when the underlying dimensions are not known. The objects, which can be of any kind, are represented as a geometric configuration of points, usually (but not necessarily) in a Euclidean space of limited dimensionality. Each object corresponds to one point, and similarity between objects is reflected in the distance between points. Not only does this result in quantified objects (since the coordinates of a point quantifies the corresponding object), but the ordering of the objects on each dimension can also be used to give an interpretation to those dimensions.

An important subclass of techniques aimed at spatial representation of data is known as multidimensional scaling (MDS). For recent textbooks on MDS see, for instance, Coxon (1982), Davison (1983), Young and Hamer (1987), and Green, Carmone, and Smith (1989). Some of the techniques known as multivariate data analysis (MVA) also result in a configuration of points. Examples of such configurations are the object scores in principal components analysis and homogeneity analysis, and the factor scores in factor analysis. For recent textbooks on MVA we refer to Van de Geer (1986), Tatsuoka (1988), and Gifi (1990).

It is not our intention to discuss the numerous ways in which data can be spatially represented. We merely want to emphasize that configurations have become an increasingly important tool in helping to understand the structure of data. The present study is concerned with the situation where several configurations are already available, and where the interest lies in the *relationships between these configurations*. Suppose, for example, that a number of researchers have collected data on the same topic and submitted their data to one or another form of multidimensional scaling or multivariate analysis. Each researcher thereby produces an m-dimensional 'map' or configuration of the stimuli that he or she is studying. The question immediately arises whether their results, that is, their 'maps', are the same or not. Another example is the situation where someone has collected n (dis)similarity matrices and decides to analyse each (dis)similarity matrix separately with MDS as a first step, instead of performing a simultaneous analysis of the complete three-way data matrix using, for instance,

INDSCAL (cf., Horan, 1969; Carroll and Chang, 1970). Only afterwards she or he starts investigating the possible relations between the n configurations. This procedure may seem a rather roundabout way to analyse (dis)similarity data, but it has the advantage that also other transformations (more simple as well as more complex ones) than in INDSCAL can be investigated, as will be discussed below. Another example is the situation where a researcher has analysed one and the same data matrix using different measurement levels and/or different approaches to ties in the data and/or even different analysis techniques, and is interested to find out whether the resulting configurations are the same, and, if not, whether there exists a systematic pattern in the differences.

The problem in these situations can be expressed more formally as follows: given a set of n ($n \geq 2$) configurations $\mathbf{X}_j$ ($j = 1, \ldots, n$) of order (p x m), each configuration containing the coordinates of the same p stimuli in m dimensions, what is the similarity or *match* between these configurations? This immediately confronts us with two other questions: what are the uniqueness properties of the configurations, that is, which transformations are they allowed to undergo, and what criterion should be used to evaluate the match? To provide an answer to these questions, the *PINDIS models* of Lingoes and Borg (1978) (see also Borg, 1979; Borg and Lingoes, 1987) seem particularly appropriate. PINDIS is an acronym for Procrustean INdividual DIfferences Scaling, and refers both to a series of six increasingly complex models, as well as to the algorithms used to estimate the model parameters, and the computer program in which these algorithms were collectively implemented. In all six PINDIS models certain linear transformations are applied to the given configurations $\mathbf{X}_j$, and the well-known least squares criterion is used to evaluate the match between the optimally transformed configurations. We will briefly discuss each PINDIS model in turn; for a more detailed description of these models and of the transformations involved we refer to Lingoes and Borg (1978), and to the next three chapters of this book.

1.2 The PINDIS models of Lingoes and Borg

The first and most simple PINDIS model was called 'the similarity transformation model' by Lingoes and Borg, but since it is much better known in the literature under the name Generalized Procrustes Analysis (GPA) as originally developed by Gower (1975), we will refer to it as the *GPA model* throughout this monograph. Let $\mathbf{R}_j$ be an unknown orthonormal matrix of order (m x m), s_j an unknown central dilation, and $\mathbf{u}_j$ an unknown translation vector of order (m x 1). Further defining $\mathbf{Z}$ as a group or centroid configuration of order (p x m), $\mathbf{E}_j$ as a (p x m) matrix of residuals, and $\mathbf{1}$ as a (p x 1) vector containing ones, the GPA model can be written as

$$s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j = \mathbf{Z} + \mathbf{E}_j, \qquad \text{for } j = 1, \ldots, n. \tag{1.1}$$

Orthonormal transformations, translations, and central dilations do not affect the relative distances between the stimulus points of a configuration. Since the configurations obtained from MDS and MVA are often generated according to the Euclidean distance model, and because such a configuration has the property of containing information that is unchanged by distance preserving transformations, it is only natural to start the analysis of n configurations with the GPA model.

If a generalized Procrustes analysis does not result in a satisfactory match, this may be due to the fact that the relations between the configurations are of a more complex nature. In PINDIS this can be investigated by evaluating the effect of more complex transformations which are still linear but no longer preserve relative distances between stimulus points.

In the second and third PINDIS model a transformation called *dimension weighting* is introduced, where dimensions are allowed to be stretched or shrunk (i.e., weighted) differently when matching the n configurations. Following the name of the transformation, the second and third model of Lingoes and Borg are called the *dimension weighting models*. Let $\mathbf{Q}_j$, $\mathbf{S}$, and $\mathbf{S}_j$ denote unknown orthonormal matrices of order (m x m), $\mathbf{g}_j$ and $\mathbf{h}_j$ unknown translation vectors of order (m x 1), and $\mathbf{W}_j$ an unknown diagonal matrix of order (m x m). Also, let $\mathbf{Z}$ be the centroid configuration obtained in model (1.1), then the second PINDIS model can be written as

$$(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j = (\mathbf{Z} - \mathbf{1}\mathbf{h}_j')\mathbf{S}\mathbf{W}_j + \mathbf{E}_j, \quad \text{for } j = 1, \ldots, n, \tag{1.2}$$

while the third PINDIS model takes the following form:

$$(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j = (\mathbf{Z} - \mathbf{1}\mathbf{h}_j')\mathbf{S}_j\mathbf{W}_j + \mathbf{E}_j, \quad \text{for } j = 1, \ldots, n. \tag{1.3}$$

The diagonal matrices $\mathbf{W}_j$ in (1.2) and (1.3) have the effect of differentially weighting the axes of the (optimally rotated) group configuration $\mathbf{Z}$ obtained from GPA. The essential difference between the two models is that $\mathbf{Z}$ is only rotated once by $\mathbf{S}$ in (1.2), whereas it is rotated differently by $\mathbf{S}_j$ for each configuration j in (1.3). Assuming that the individual configurations represent n subjects, the psychological implication of model (1.2) is that the subjects agree on the underlying dimensions that structure the stimuli under investigation (that is: the axes of $\mathbf{ZS}$), but differ in the importance they attach to the dimensions on which the stimuli are ordered. Model (1.3) can be interpreted as meaning that the subjects not only differ in the importance they attach to the dimensions, but also have their own views on *which* dimensions of $\mathbf{Z}$ are to be considered as important. These idiosyncratic dimensions are represented by the axes of $\mathbf{ZS}_j$. In terms of interpretation, therefore, the two dimension weighting models in PINDIS are related to the INDSCAL and IDIOSCAL models of Carroll and Chang (1970, 1972). These PINDIS models are less parsimonious than the GPA model since they require the estimation of more parameters than in GPA model (1.1). This means that the dimension weighting models always fit the data equally well as or better than the GPA model.

The fourth and fifth PINDIS model proposed by Lingoes and Borg are characterized by a transformation that we will call *stimulus weighting* in this monograph. In stimulus weighting the stimulus points of a configuration are connected with the origin of m-dimensional space, and the thus generated vectors are allowed to be shortened or lenghtened differently. Stimulus weighting gives rise to two different models: the *stimulus weighting models*. Let $\mathbf{g}_j$, $\mathbf{h}$, and $\mathbf{h}_j$ be unknown translation vectors of order (m x 1), $\mathbf{T}_j$ an unknown orthonormal matrix of order (m x m), and $\mathbf{V}_j$ an unknown diagonal matrix of order (p x p), then the fourth PINDIS model can be written as

$$(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j = \mathbf{V}_j(\mathbf{Z} - \mathbf{1}\mathbf{h}') + \mathbf{E}_j, \quad \text{for } j = 1, \ldots, n, \tag{1.4}$$

and the fifth PINDIS model is

$$(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j = \mathbf{V}_j(\mathbf{Z} - \mathbf{1}\mathbf{h}_j') + \mathbf{E}_j, \quad \text{for } j = 1, \ldots, n. \tag{1.5}$$

In both models the centroid configuration **Z** is assumed to be fixed, and is equal to the optimal group configuration obtained in the GPA model. The diagonal matrices $\mathbf{V}_j$ in (1.4) and (1.5) have the effect of differentially weighting the vectors obtained by connecting the stimulus points in $(\mathbf{Z} - \mathbf{1h}')$ or in $(\mathbf{Z} - \mathbf{1h}_j')$ with the origin of m-dimensional space.

A little thought should convince the reader that the *location* of a configuration is crucial in stimulus weighting. Imagine an arbitrary two-dimensional configuration of points, all points being connected with the origin. Changing the location of this configuration, which is achieved by a translation, will also change its position relative to the coordinate axes. Therefore, the set of vectors obtained by connecting the stimulus points of the *relocated* configuration with the origin will be quite different from the original set: both the lengthes of the vectors as well as the angles between the vectors will have been changed. This explains the essential difference between the two stimulus weighting models: **Z** is only translated once in (1.4) while it is translated differently for each configuration j in (1.5).

Assuming that the $\mathbf{X}_j$ represent subjects, model (1.4) can be interpreted as follows. All subjects share one common point of view in the group configuration **Z** (represented by the origin from which the stimulus vectors in $(\mathbf{Z} - \mathbf{1h}')$ emanate). But, since weighting the vectors representing the stimulus points in $(\mathbf{Z} - \mathbf{1h}')$ has the effect of either pushing the stimulus points further away towards the periphery, or pulling them closer towards the centre or origin of m-dimensional space, the subjects differ in the 'centrality' they assign to the stimuli. In model (1.5) the subjects not only differ in the centrality they assign to the stimuli, but they also literally take different points of view in the group configuration because **Z** is translated differently for each subject. Since the configurations usually contain more rows than columns (i.e., more stimuli than dimensions), and because one optimal weight is estimated for each row of each configuration $\mathbf{X}_j$, the stimulus weighting models are the most 'expensive', and thus also the least parsimonious of the PINDIS models.

Lingoes and Borg (1978) even discussed a sixth model where the combined effects of dimension weights *and* stimulus weights are investigated. This 'double weighting' model has such dazzling implications for a possible psychological interpretation, however, that we will not consider it in the present monograph.

The relations between the PINDIS models (1.1) through (1.5) are shown in Figure 1.1. As the figure illustrates the five models in fact form two hierarchies: the first

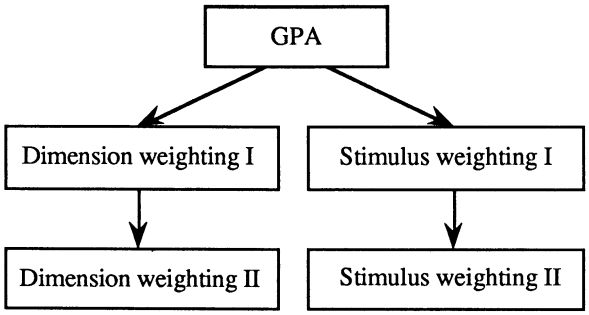


Figure 1.1 Scheme of hierarchy in the PINDIS models.

enabling one to assess the gain in fit obtained by dimension weighting compared to GPA, the second allowing one to establish the gain in fit by stimulus weighting compared to GPA. In both hierarchies the GPA model plays the role of a 'yardstick' or reference model for the more complex models. Within each hierarchy the first weighting model (whether it is dimension or stimulus weighting) further acts as a reference model for the second weighting model.

A nice feature of these two hierarchies is that they give one the opportunity to find out whether more complex transformations have anything new to add above more simple transformations. We already noted that if someone has collected n (dis)similarity matrices there may be reasons to analyse the matrices separately with MDS, and only then to investigate the relations between the resulting n configurations using the models of Lingoes and Borg. The advantage of this procedure is that, if the GPA model fits the configurations well, a more complex model involving dimension weighting, for example, becomes superfluous. By directly submitting the n (dis)similarity matrices to an INDSCAL analysis, in general one would not find out that distance preserving transformations are sufficient to uncover a common structure in the data.

Moreover, if weights have anything new to add above transformations that do not distort relative distances, they are not restricted to be applied in just one way (by which we imply: the popular choice for INDSCAL). Instead, one can investigate and compare the results of four different weighting models.

Hence, the PINDIS models offer an attractive way to investigate the relationships between n configurations containing information about the same p stimuli.

Unfortunately, quite a number of objections can be raised against the algorithms developed by Lingoes and Borg for the estimation of the unknown parameters in these models, as well as against the PINDIS program itself.

1.3 Problems with PINDIS

Lingoes and Borg obtained estimates of the unknown parameters in the PINDIS models by minimizing the sum of the squared residuals corresponding to models (1.1) through (1.5), that is, by minimizing Σ tr $\mathbf{E}_j'\mathbf{E}_j$. They found that none of the models can be solved analytically, and resorted to estimate the parameters by using iterative algorithms. However, only for model (1.1) did their estimation procedures yield a convergent algorithm. The PINDIS algorithms corresponding to model (1.2) through (1.5) are not guaranteed to converge.

Specifically, the procedure used in PINDIS for the estimation of the unknown rotation matrices $\mathbf{Q}_j$ in models (1.2) and (1.3), and of the unknown rotation matrices $\mathbf{T}_j$ in models (1.4) and (1.5) can be easily improved. Lingoes and Borg based their estimation procedures for these matrices on results derived for the two-dimensional case. Consequently, in higher dimensional spaces they were forced to use an iterative procedure by performing planar rotations for all $\binom{m}{2}$ combinations of the m dimensions. They could not prove that this procedure must converge. Lingoes and Borg remarked on this subject (1978, p.506):

> It would be possible, however, to use an alternating method analogous to the one used in INDSCAL and ALSCAL, for example. If the spaces were higher than two-dimensional, this would have the advantage that the rotation matrices would be computed directly rather than in the plane-wise manner, which cannot be proved to converge necessarily. ... In evaluation of the entire matrix approach, one could say that it has clear advantages over the scalar solutions previously discussed whenever the transformational analyses are carried out in higher dimensional spaces.

Since the estimation of the rotation matrices $\mathbf{Q}_j$ and $\mathbf{T}_j$ is the well-known orthonormal Procrustes problem for which an analytical solution is available in any number of dimensions (see, e.g., Schönemann, 1966), the latter procedure is to be preferred.

A second objection that can be raised against PINDIS is that the results of the dimension and stimulus weighting models all depend upon the centroid configuration $\mathbf{Z}$ obtained in GPA model (1.1). Although $\mathbf{Z}$ is optimal in GPA, this is not necessarily the case in models (1.2) through (1.5).

Another problem specific for the dimension weighting algorithms implemented in PINDIS is that the rotation matrices $\mathbf{S}_j$ in model (1.3) are estimated in an ad hoc way. For each $\mathbf{S}_j$, and in each plane that can be formed by the $\binom{m}{2}$ combinations of the m dimensions, the centroid configuration $\mathbf{Z}$ is rotated in steps of 5° over an interval of ±45°. Then a ±5° sector around the 'best' rotation angle in the previous interval is searched in intervals of 1°. The best angle in the latter search process is picked as the optimal rotation of $\mathbf{Z}$. This procedure is repeated for all pairwise combinations of the m dimensions. Then, for want of a better solution, the 'optimal' $\mathbf{S}$ in model (1.2) is determined as follows: $\bar{\mathbf{Z}}$ is defined as the mean of the n matrices $\mathbf{ZS}_j$ obtained in model (1.3), and $\mathbf{S}$ is identified as the orthonormal matrix bringing $\mathbf{Z}$ in maximal agreement with $\bar{\mathbf{Z}}$ (cf. Lingoes and Borg, 1978, p.500). A specific problem of stimulus weighting model (1.4) in PINDIS is that the 'optimal' translation vector $\mathbf{h}$ in model (1.5) is simply taken as the average of the optimal translation vectors $\mathbf{h}_j$ obtained in model (1.5) (cf. Lingoes and Borg, 1978, p.502). Clearly, all these procedures can only provide rough approximations of the parameters at issue.

Also, the formulas numbered (36) and (51) in the Lingoes and Borg article are not correct. Using our notation these formulas should be

$$\mathbf{h}_j = \frac{[\mathbf{V}_j\mathbf{Z} - (\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j]'\mathbf{V}_j\mathbf{1}}{\mathbf{1}'\mathbf{V}_j^2\mathbf{1}}, \tag{1.6}$$

and

$$\mathbf{S}_j'\mathbf{Z}'(\mathbf{X}_j\mathbf{Q}_j - \mathbf{Z}\mathbf{S}_j\mathbf{W}_j)\mathbf{W}_j = \text{symmetric}, \tag{1.7}$$

respectively. Anyone interested in programming the algorithm of Lingoes and Borg corresponding to model (1.5) will find that his or her algorithm behaves erratically if using formula (36) in the 1978 article to update $\mathbf{h}_j$, while (1.6) gives the expected results. Moreover, formula (51) in the Lingoes and Borg article suggests that the estimation of $\mathbf{S}_j$ in model (1.3) is a simple orthonormal Procrustes problem. But the problem is much more complicated, as is witnessed by (1.7).

We finally note that the output of the PINDIS program available in the MDS(X) series by Davies and Coxon (1983) is very obscure and difficult to understand, and that a number of options which, according to the accompanying manual, should give the user control of the program do not work properly, or do not work at all.

1.4 MATCHALS as an alternative

Clearly, the PINDIS program contains a number of serious shortcomings. The purpose of the present study is to provide better and more general alternating least squares algorithms than those implemented in PINDIS. In the following chapters we will develop convergent algorithms, generalizing all estimation procedures to the m-dimensional case. Moreover, a number of features will be added that are not available in the PINDIS models. First of all, the five models and the corresponding algorithms will be generalized to the case of incomplete configurations, that is, the case where information about some stimuli in some configurations is missing in any arbitrary pattern. Next, new optimal group configurations will be estimated in the dimension and stimulus weighting models instead of borrowing the group configuration **Z** from the GPA model. Finally, all models will be supplemented with an analysis of variation allowing one to assess the relative contributions of configurations, stimuli and dimensions to the total fit of each model. In this respect we have been inspired by the analysis of variation proposed by Gower (1975) for the GPA model.

Specifically, the incorporation of missing data in GPA yields the following model:

$$\mathbf{M}_j s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j = \mathbf{M}_j\mathbf{Z} + \mathbf{M}_j\mathbf{E}_j, \qquad \text{for } j = 1, \ldots, n, \tag{1.8}$$

where $\mathbf{M}_j$ is a given diagonal matrix of order (p x p). This matrix contains ones on the diagonal if the corresponding rows of configuration $\mathbf{X}_j$ are not missing and zeroes elsewhere. PINDIS model (1.2) will be replaced by the following model

$$\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j = \mathbf{M}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{W}_j + \mathbf{M}_j\mathbf{E}_j, \quad \text{for } j = 1, \ldots, n, \tag{1.9}$$

and we will use

$$\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j = \mathbf{M}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{S}_j\mathbf{W}_j + \mathbf{M}_j\mathbf{E}_j, \quad \text{for } j = 1, \ldots, n, \tag{1.10}$$

instead of model (1.3). The matrix **Y** of order (p x m) in (1.9) and (1.10) represents an unknown group configuration. We refer to Chapter 3 for an explanation why matrix **S** contained in model (1.2) no longer needs to be considered in (1.9). The stimulus weighting models (1.4) and (1.5) will be replaced by

$$\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j = \mathbf{M}_j\mathbf{V}_j\mathbf{Y} + \mathbf{M}_j\mathbf{E}_j, \quad \text{for } j = 1, \ldots, n, \tag{1.11}$$

and

$$\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j = \mathbf{M}_j\mathbf{V}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j') + \mathbf{M}_j\mathbf{E}_j, \quad \text{for } j = 1, \ldots, n, \tag{1.12}$$

respectively. Again, matrix $\mathbf{Y}$ in the latter two models is assumed to be an unknown group configuration. For the reason why translation vector $\mathbf{h}$ in model (1.4) has disappeared in (1.11) we refer to Chapter 4 of this monograph.

We have baptized the computer program used to estimate the parameters in (1.8) through (1.12) the MATCHALS program, an acronym for MATCHing configurations by Alternating Least Squares.

1.5 Restrictions

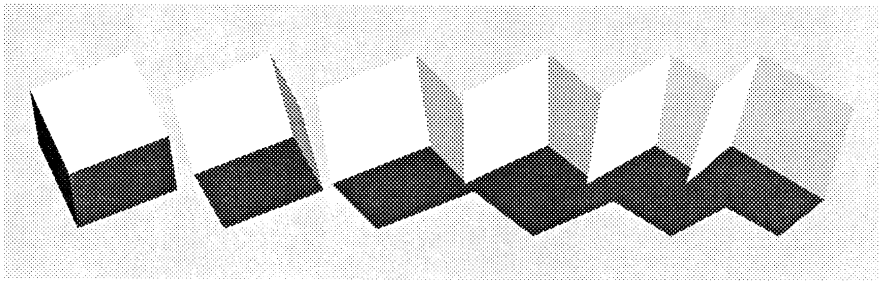
In the present monograph we do not generalize the models to the case where the n configurations have unequal numbers of columns. In other words, we assume throughout that the configurations have the same dimensionality (although they may have different (and thus also deficient) column ranks). When confronted with configurations having unequal numbers of columns, and letting $m = \max(m_j)$, where m_j is the number of columns in configuration j, we advise to append $(m - m_j)$ zero columns to each configuration $\mathbf{X}_j$. But we are well aware of the fact that this procedure may lead to suboptimal solutions (see, e.g., Ten Berge & Knol, 1984).

Moreover, the present book heavily emphazises the algorithmic aspects of the models. Consequently, hardly any real life examples are presented, while the statistical aspects of the models are only very briefly touched upon (in the final, fifth, chapter). Before applying the algorithms to empirical configurations which will usually be contaminated with error, we thought it important first to investigate whether the algorithms would succeed in recovering an a priori known structure in the data. This is the reason why we primarily use geometrically constructed data sets to test the algorithms presented in this book.

1.6 Overview

The book is organized as follows. In Chapter 2 the efficient algorithm of Ten Berge (1977) for the estimation of the unknown parameters in the GPA model is generalized to the case of missing data. Chapter 3 is devoted to the development of the dimension weighting models (1.9) and (1.10). In Chapter 4 the stimulus weighting models (1.11) and (1.12) are discussed. Chapter 5 summarizes the most important results found in the previous chapters, and provides recommendations for further research.

The book abounds with geometrical figures like squares, pentagons, octagons, cubes, tetrahedrons, dodecahedrons, etc. These figures play such a predominant part because they lend themselves perfectly to illustrate the properties of the MATCHALS models. As for the notation, a bold upper case letter will always denote a matrix (e.g., $\mathbf{X}$), a bold lower case letter a column vector ($\mathbf{x}$, for example), and a plain lower case letter a scalar (e.g., x). For a recent textbook on the matrix algebra used in the present study we refer to Magnus and Neudecker (1988).



Chapter 2

Generalized Procrustes analysis

2.1 Introduction

In this chapter the first and most simple model of the MATCHALS program will be discussed. Following Van de Geer's analogy (1984), the problem in this model may be visualised as follows. Suppose we have two configurations $\mathbf{X}_1$ and $\mathbf{X}_2$, both configurations containing the coordinates of the same p stimuli in two dimensions. Further suppose that the stimulus points of $\mathbf{X}_1$ are drawn on a transparent sheet, and that the stimulus points of $\mathbf{X}_2$ are drawn on another transparent sheet. The two sheets are arbitrarily placed on top of each other. The problem to be solved in GPA, the abbreviation of Generalized Procrustes Analysis that we use throughout this book, is: How should the two sheets be moved such that the sum of the squared distances between corresponding pairs of points on the two sheets becomes as small as possible? This criterion in GPA for the evaluation of the match between configurations $\mathbf{X}_1$ and $\mathbf{X}_2$ is illustrated in Figure 2.1, where the stimulus points are the vertices of two pentagons. To minimize the sum of squared distances between corresponding pairs of points, the two transparent sheets may be manipulated in a number of ways. However, there is one important restriction: We only consider those transformations that leave the relative distances between stimulus points within each of the two configurations unchanged. The implications of this restriction are illustrated for configuration $\mathbf{X}_1$ in Figure 2.2, where the stimulus points of $\mathbf{X}_1$ are numbered from

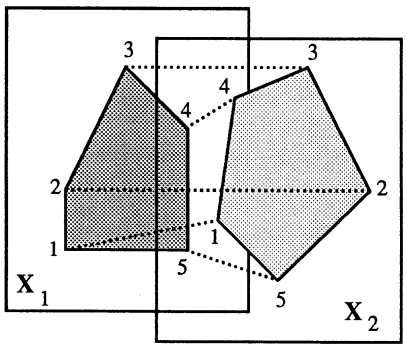


Figure 2.1 Illustration of match criterion in GPA for two configurations. Sum of squared lengths of dotted lines must be as small as possible.

one to five before transformation and from one prime to five prime after transformation.

The first transformation illustrated in Figure 2.2 shows that the stimulus points in a configuration may all be moved into the same direction and along an equal distance. This transformation is called a translation. The second admissible transformation is to turn over the sheet with $\mathbf{X}_1$ on it, known as a reflection. The third admissible transformation is to stretch or shrink the sheet containing $\mathbf{X}_1$ in a uniform fashion, also called a central dilation or uniform rescaling. The last kind of manipulation that

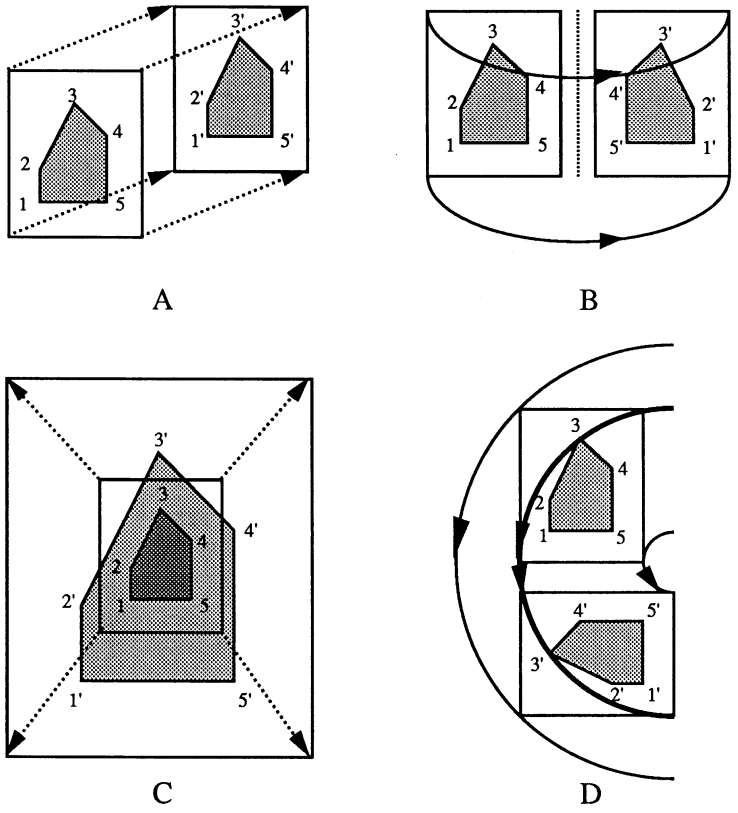


Figure 2.2 Admissible transformations in GPA. A: translation, B: reflection, C: uniform rescaling and D: rotation.

the two sheets are allowed to undergo in GPA in order to enhance their match is a rotation. In Figure 2.2 configuration $\mathbf{X}_1$ is rotated counterclockwise through an angle of 90°. Since, in practice, the problem of rotating and reflecting a configuration can be solved simultaneously, in the sequel these two transformations will jointly be referred to as an *orthonormal transformation.*

None of the four manipulations of the sheet representing configuration $\mathbf{X}_1$ in Figure 2.2 affects the ratios of the distances between the five stimulus point of $\mathbf{X}_1$. These same four transformations may be applied to configuration $\mathbf{X}_2$ in order to reduce the sum of squared distances between the corresponding pairs of points in $\mathbf{X}_1$ and $\mathbf{X}_2$.

In this very simple example consisting of the two configurations $\mathbf{X}_1$ and $\mathbf{X}_2$ shown in Figure 2.1 the reduction in sum of squared distances could roughly be brought about by just moving the two sheets around until the sum of squared distances seems to be as small as possible. But even in this simple case it is difficult to visualise how the two sheets should be stretched or shrunk uniformly in order to enhance their match. Moreover, in practice there will be more than five stimulus points to consider, there will be more than two configurations to match, and last but not least the configurations may be more than two-dimensional.

The groundwork for GPA was laid by Green (1952), Cliff (1966) and Schönemann (1966) who solved the problem of orthonormally transforming (i.e., rotating and reflecting) a matrix such that the sum of squared distances between pairs of points of the latter matrix and another matrix is minimized given that the two configurations have an equal number of columns. The solutions of Green and Cliff only work for matrices of full column rank, while Schönemann generalized the solution to matrices of deficient column rank. Schönemann and Carroll (1970) investigated and solved the problem of fitting one matrix onto another matrix using not only an orthonormal transformation, but also a translation and uniform rescaling.

Kristof and Wingersky (1971) extended orthonormal transformations to the case where more than two configurations with equal numbers of columns are involved. Gower (1975) again solved the problem of fitting more than two configurations using orthonormal transformations as well as translations and central dilations when the configurations have an equal number of columns. He was also the first to coin the term Generalized Procrustes Analysis. In 1977 Ten Berge corrected Gower's method for the determination of the central dilations, and improved Gower's procedure for the

calculation of the orthonormal transformations. Ten Berge and Knol (1984) extended the problem to the case where two or more configurations have a different number of columns (although only for orthonormal transformations). Very recently, Peay (1988) has solved the problem of fitting more than two configurations with differing dimensionalities under the general rule illustrated in Figure 2.2, that is including translations and central dilations.

All these latter extensions of GPA typically are concerned with data sets consisting of matrices having unequal numbers of columns. In this chapter we assume that the matrices have equal column orders, but generalize GPA to the case where the configurations have a different number of *rows*, that is where information about some stimuli in some of the configurations is missing. Previous work in this area has been done by Everitt and Gower (1981) in the context of what they called a 'weighted generalized Procrustes method', of which the problem discussed in this chapter is a special case. However, they did not incorporate central dilations in this method (while these are essential as will be discussed in section 2.3.5), neither did they provide a rigorous treatment of translations. Moreover, a more efficient procedure than Everitt and Gower's method will be proposed in this chapter for the estimation of the orthonormal transformations. Other work on GPA and incomplete configurations has been done by De Leeuw and Meulman (1986) in the context of a Jackknife procedure. They investigated how to match a number of configurations when each configuration has missing information about only one stimulus. De Leeuw and Meulman's solution is a special case of the matching procedure to be developed in this chapter, since we consider the case where the configurations may have any pattern of missing data.

The solution proposed in this chapter is useful in the following general situations. A number of researchers have collected data on the same topic and submitted their data to one or another form of multidimensional scaling or multivariate analysis (like factor or principal components analysis). Each researcher thereby produces a configuration of the stimuli that he or she is studying. However, different researchers not necessarily use the same stimuli when investigating the same topic: some stimulus sets may overlap, others may not. The same thing may happen, of course, if someone repeats an experiment in time. Then too, he or she may choose to use varying numbers of stimuli. In both cases it is of interest to investigate whether the resulting configurations have an identical structure, notwithstanding the fact that the number of stimuli may vary from configuration to configuration.

2.2 The direct and the centroid approach: geometry and algebra

We embark on the solution of the problem of developing the GPA model for configurations containing missing data by giving, without proof, the following identity (see, e.g., Kristof & Wingersky, 1971; Gower, 1975) which is at the core of much that is to follow in this chapter.

Theorem 1. Given n points in m-dimensional space, the sum of the squared distances between these n points always equals n times the sum of the squared distances between the n points and their centroid.

If we let $\mathbf{x}_j$ be the column vector containing the coordinates of point j (j = 1, ..., n) in m-dimensional space, and $\mathbf{z}$ the column vector consisting of the coordinates of the centroid of the n points, that is,

$$\mathbf{z} = (1/n) \sum_{j=1}^{n} \mathbf{x}_j,$$

then Theorem 1 can be expressed algebraically as follows:

$$\sum_{j<k} (\mathbf{x}_j - \mathbf{x}_k)'(\mathbf{x}_j - \mathbf{x}_k) = n \sum_{j=1}^{n} (\mathbf{x}_j - \mathbf{z})'(\mathbf{x}_j - \mathbf{z}) = n \sum_{j=1}^{n} \mathbf{x}_j'\mathbf{x}_j - n^2\, \mathbf{z}'\mathbf{z}.$$

A straightforward application of Theorem 1 in the context of GPA is illustrated in Figure 2.3, which contains the plot of four two-dimensional configurations and their corresponding centroids. The coordinates of all points in Figure 2.3 are given in Table 2.1. The numerical value 99 in Table 2.1 indicates that information about a stimulus in

Table 2.1 Coordinates of stimulus points plotted in Figure 2.3.

stimulus	$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	$\mathbf{X}_4$	$\mathbf{Z}$
1	1 3	-2 3	0 6	-3 8	-1 5
2	99 99	1 5	5 2	3 5	3 4
3	4 -1	99 99	99 99	6 -3	5 -2
4	99 99	1 -4	3 -7	2 -7	2 -6
5	-3 -3	-5 -4	-4.5 -6	-3.5 -7	-4 -5

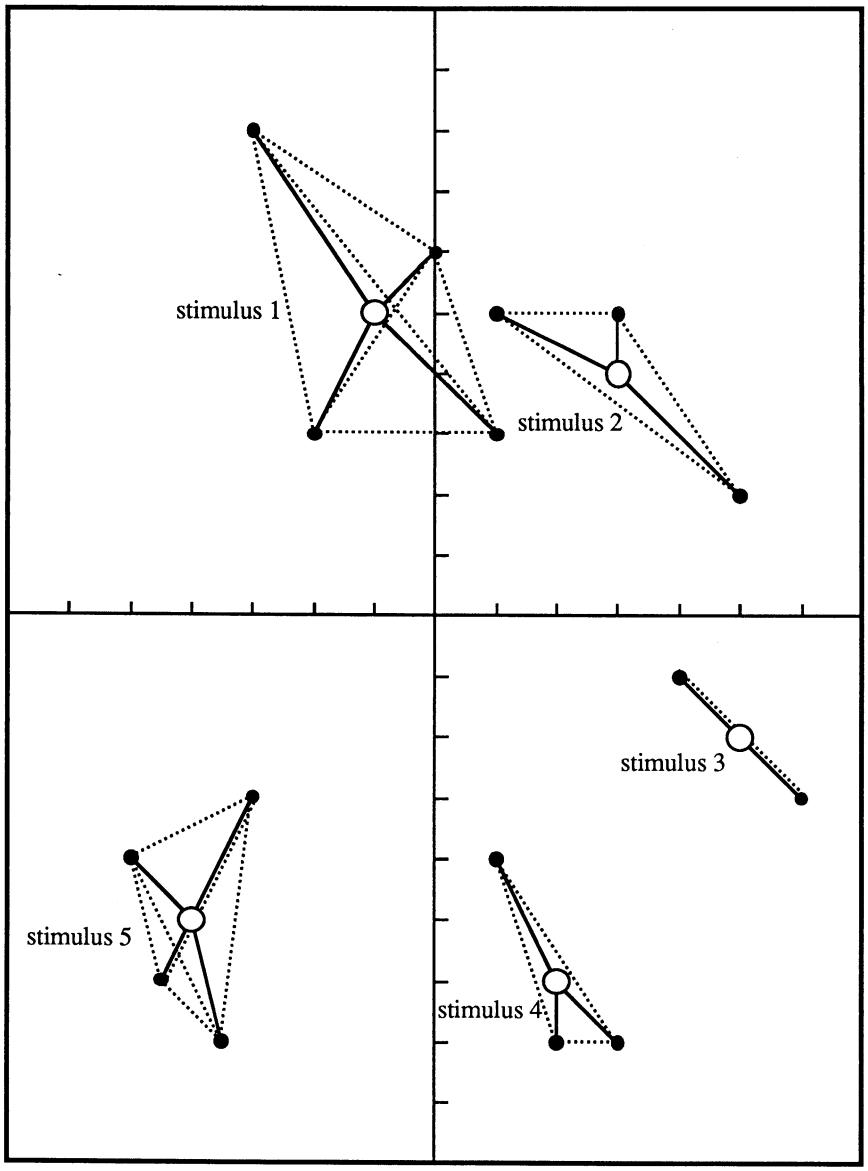


Figure 2.3 Geometry of match for data in Table 2.1.

a configuration is missing. In this example information about stimuli 2 and 4 of configuration $\mathbf{X}_1$ is missing, and stimulus 3 has missing values both in configurations $\mathbf{X}_2$ and $\mathbf{X}_3$.

There are five stimuli in this example and therefore Theorem 1 can be applied five times. The match for stimulus 1 can either be calculated as the sum of squared distances between the four points corresponding to stimulus 1 (dotted lines in Figure 2.3), or as the sum of squared distances between these same four points and their centroid (solid lines in Figure 2.3). For stimulus 1 we have four points to consider because no configuration in this example has missing values in the first row. The match for stimulus 2 may also be evaluated in two ways: either by summing the squared lengths of the dotted lines or by summing the squared lengths of the solid lines for stimulus 2 in Figure 2.3. For stimulus 2 there are only three points to consider since information about this stimulus is missing in configuration $\mathbf{X}_1$. In the same manner the remaining three stimulus points in Figure 2.3 can be evaluated using either match criterion. If we compare the two criteria for each stimulus of the data plotted in Figure 2.3 separately, their ratio is, of course, completely in agreement with Theorem 1. For stimulus 2, for instance, the ratio of the sum of squared lengths of the solid lines and the sum of squared lengths of the dotted lines is 14/42 = 1/3. In general, the ratio between the two criteria will always be $(1/n_i)$, where n_i is the number of non-missing rows in the n configurations $\mathbf{X}_j$ for stimulus i (i = 1, ..., p).

We now address the question how to express these two equivalent match criteria in matrix notation. Given n configurations $\mathbf{X}_j$ (j = 1, ..., n) of order (p x m), let $\mathbf{M}_j$ be the diagonal matrix of order (p x p) with ones on the diagonal if the corresponding rows in $\mathbf{X}_j$ are not missing, and zeroes everywhere else. For instance, $\mathbf{M}_1$ corresponding to $\mathbf{X}_1$ in Table 2.1 is

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Matrix $\mathbf{M}_j$ is always symmetric and idempotent, and if a configuration j has no missing data $\mathbf{M}_j$ is equal to the identity matrix. In matrix notation the sum of squared lengths of the solid lines in Figure 2.3 is equal to

$$\sum_{j=1}^{n} \text{tr } (\mathbf{X}_j - \mathbf{Z})'\mathbf{M}_j(\mathbf{X}_j - \mathbf{Z}), \tag{2.1}$$

where

$$\mathbf{Z} = (\sum_{j=1}^{n} \mathbf{M}_j)^{-1}(\sum_{j=1}^{n} \mathbf{M}_j\mathbf{X}_j) \tag{2.2}$$

is the (p x m) matrix of centroids. The matrix expression for the sum of squared lengths of the dotted lines in Figure 2.3 is

$$\sum_{j<k} \text{tr } (\mathbf{X}_j - \mathbf{X}_k)'\mathbf{M}_j\mathbf{M}_k(\mathbf{X}_j - \mathbf{X}_k). \tag{2.3}$$

The two criteria (2.1) and (2.3) become identical by correcting them for the ratios $(1/n_i)$ that we discussed above. These ratios are equal to the diagonal of $(\sum_{j=1}^{n} \mathbf{M}_j)^{-1}$.

Theorem 2. The generalization of Theorem 1 for n configurations $\mathbf{X}_j$ containing missing data is

$$\sum_{j=1}^{n} \text{tr } (\mathbf{X}_j - \mathbf{Z})'\mathbf{M}_j(\mathbf{X}_j - \mathbf{Z}) = \sum_{j<k} \text{tr } (\mathbf{X}_j - \mathbf{X}_k)'\mathbf{M}_j\mathbf{M}_k(\sum_{j=1}^{n} \mathbf{M}_j)^{-1}(\mathbf{X}_j - \mathbf{X}_k), \tag{2.4}$$

with $\mathbf{Z}$ as defined in (2.2).

Proof. Expanding the right side of equation (2.4) gives

$$\sum_{j<k} \text{tr } (\mathbf{X}_j - \mathbf{X}_k)'\mathbf{M}_j\mathbf{M}_k(\sum_{j=1}^{n} \mathbf{M}_j)^{-1}(\mathbf{X}_j - \mathbf{X}_k)$$

$$= \sum_{j=1}^{n} \text{tr } \mathbf{X}_j'\mathbf{M}_j(\sum_{i\neq j} \mathbf{M}_i)(\sum_{j=1}^{n} \mathbf{M}_j)^{-1}\mathbf{X}_j - 2\sum_{j<k} \text{tr } \mathbf{X}_j'\mathbf{M}_j\mathbf{M}_k(\sum_{j=1}^{n} \mathbf{M}_j)^{-1}\mathbf{X}_k. \tag{2.5}$$

But since

$$\text{tr } \mathbf{X}_j'\mathbf{M}_j(\sum_{i\neq j} \mathbf{M}_i)(\sum_{j=1}^{n} \mathbf{M}_j)^{-1}\mathbf{X}_j = \text{tr } \mathbf{X}_j'\mathbf{M}_j\mathbf{X}_j - \text{tr } \mathbf{X}_j'\mathbf{M}_j(\sum_{j=1}^{n} \mathbf{M}_j)^{-1}\mathbf{X}_j,$$

expression (2.5) may be written as

$$\sum_{j=1}^{n} \text{tr } \mathbf{X}_j'\mathbf{M}_j\mathbf{X}_j - \sum_{j=1}^{n} \text{tr } \mathbf{X}_j'\mathbf{M}_j(\sum_{j=1}^{n} \mathbf{M}_j)^{-1}\mathbf{X}_j - 2\sum_{j<k} \text{tr } \mathbf{X}_j'\mathbf{M}_j\mathbf{M}_k(\sum_{j=1}^{n} \mathbf{M}_j)^{-1}\mathbf{X}_k$$

$$= \sum_{j=1}^{n} \text{tr } \mathbf{X}_j'\mathbf{M}_j\mathbf{X}_j - \text{tr } (\sum_{j=1}^{n} \mathbf{M}_j\mathbf{X}_j)'(\sum_{j=1}^{n} \mathbf{M}_j)^{-1}(\sum_{j=1}^{n} \mathbf{M}_j\mathbf{X}_j). \tag{2.6}$$

It further follows from (2.2) that

$$(\sum_{j=1}^{n} \mathbf{M}_j\mathbf{X}_j) = (\sum_{j=1}^{n} \mathbf{M}_j)\mathbf{Z}, \tag{2.7}$$

and substitution of (2.7) in (2.6) yields

$$\sum_{j=1}^{n} \text{tr } \mathbf{X}_j'\mathbf{M}_j\mathbf{X}_j - \text{tr } \mathbf{Z}'(\sum_{j=1}^{n} \mathbf{M}_j)\mathbf{Z}$$

$$= \sum_{j=1}^{n} \text{tr } \mathbf{X}_j'\mathbf{M}_j\mathbf{X}_j + \text{tr } \mathbf{Z}'(\sum_{j=1}^{n} \mathbf{M}_j)\mathbf{Z} - 2 \text{ tr } \mathbf{Z}'(\sum_{j=1}^{n} \mathbf{M}_j)\mathbf{Z}$$

$$= \sum_{j=1}^{n} \text{tr } (\mathbf{X}_j - \mathbf{Z})'\mathbf{M}_j(\mathbf{X}_j - \mathbf{Z}),$$

which completes the proof.

It may be noted that thus far nothing has been said about the optimization of the match between the n configurations. Equation (2.4) only shows that, wherever in space the configurations are located with respect to one another and in whatever position, the match in GPA can be measured in two different ways yielding exactly the same result. In order to optimize the match between the n configurations the parameters corresponding to the four admissible transformations illustrated in Figure 2.2 must be inserted into the formula of either the first criterion as in (2.1) or the second criterion as on the right side of (2.4).

Let $\mathbf{1}$ be a (p x 1) vector consisting of ones, $\mathbf{u}_j$ an unknown translation vector of order (m x 1), s_j an unknown uniform scaling factor and $\mathbf{R}_j$ an unknown orthonormal rotation and reflection matrix of order (m x m). Also, let the n unknown translation vectors $\mathbf{u}_j$ be collected in the (m x n) matrix $\mathbf{U}$, the uniform scaling factors in the (n x 1) vector $\mathbf{s}$, and the unknown orthonormal matrices in the (nm x m) supermatrix $\mathbf{R}$, then for the first criterion the following least squares loss function must be minimized for unknown translations, uniform scaling factors and orthonormal transformations:

$$f(\mathbf{U},\mathbf{s},\mathbf{R}) = \sum_{j=1}^{n} \text{tr } [s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j - \mathbf{Z}]'\mathbf{M}_j[s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j - \mathbf{Z}], \tag{2.8}$$

where $\mathbf{Z}$ is defined as the (p x m) centroid matrix, that is,

$$\mathbf{Z} = (\sum_{j=1}^{n} \mathbf{M}_j)^{-1}(\sum_{j=1}^{n} s_j\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j). \tag{2.9}$$

Analogously, in GPA the following least squares loss function must be minimized if the second criterion is used

$$g(\mathbf{U},\mathbf{s},\mathbf{R}) = \sum_{j<k} \text{tr}\, [\tilde{\mathbf{X}}_j - \tilde{\mathbf{X}}_k]'\mathbf{M}_j\mathbf{M}_k(\sum_{j=1}^{n} \mathbf{M}_j)^{-1}[\tilde{\mathbf{X}}_j - \tilde{\mathbf{X}}_k], \tag{2.10}$$

where $\tilde{\mathbf{X}}_j = s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j$ and $\tilde{\mathbf{X}}_k = s_k(\mathbf{X}_k - \mathbf{1}\mathbf{u}_k')\mathbf{R}_k$, and again $\mathbf{u}_j$ and $\mathbf{u}_k$ are unknown translation vectors, s_j and s_k are unknown central dilations, and $\mathbf{R}_j$ and $\mathbf{R}_k$ are unknown orthonormal rotation and reflection matrices.

To be able to distinguish between loss functions (2.8) and (2.10), henceforth we will call (2.8) the *centroid approach* in GPA, since in this approach the amount of loss is measured as the sum of squared distances between corresponding stimulus points and their centroid. Loss function (2.10), on the other hand, from here on will be called the *direct approach*, which name expresses the fact that the loss is directly measured as the sum of squared distances between corresponding points and not through the intermediary of their centroid.

2.2.1 Overview of the solutions

In the literature on GPA without missing data one finds that the minimization of the direct approach (2.10) is more efficient than the minimization of the centroid approach (2.8) (cf. Ten Berge, 1977), because in that case the translation vectors are easily eliminated from (2.10) yielding a function of only two sets of parameters: $\mathbf{s}$ and $\mathbf{R}$. In the case of missing data, however, it seems more fruitful to minimize the centroid approach (2.8), since both the translation vectors *and the centroid configuration* can be eliminated from the latter function, while we have found it impossible to eliminate the translation vectors from (2.10).

At this point, the reader probably wonders: how can one possibly eliminate the centroid configuration $\mathbf{Z}$ from (2.8), when $\mathbf{Z}$ is already defined by (2.9)? First, it is important to note that definition (2.9) has a very special property: *it already yields the*

global minimum of (2.8), when considered as a function of **Z**, for fixed **U**, **s**, and **R**. This immediately follows from the fact that the smallest sum of squared distances between n given points and another variable point is obtained by setting the variable point equal to the centroid of the n given points. Because of this property of (2.9), nothing changes by including the matrix of centroids as a variable in (2.8). Moreover, this creates the possibility to eliminate **U** and **Z** from (2.8), and thus to reduce the unknown parameters in (2.8) to **R** and **s**. Recall that no harm is done by treating **Z** as a variable, since, as will be proved below, at the optimum of the centroid loss function it still satisfies (2.9), even after the elimination of **U** from (2.8).

Therefore, in section 2.3 we completely develop the centroid approach instead of the direct approach. In section 2.3.1 we first of all determine the optimal translation vectors $\mathbf{u}_j$ in (2.8), because this allows us to simplify the loss function considerably. In section 2.3.2 we investigate how the optimal centroid configuration **Z** can be determined in the loss function from which the translation vectors have been eliminated. In section 2.3.3 it is shown how the matrix of centroids **Z** can also be eliminated from the simplified loss function (2.8), and in sections 2.3.4 and 2.3.5 the resulting function is minimized with respect to the only remaining unknowns: the orthonormal transformation matrices and the scaling factors. To this end a number of restrictions will be added to the simplified loss function, so as to guarantee that the optimal rotation and reflection matrices $\mathbf{R}_j$ are orthonormal and that the ensuing algorithm does not converge to the trivial solution $s_j = 0$ for each j.

In section 2.4 an algorithm is presented for GPA of configurations with and without missing data. In section 2.5 a number of measures of fit in GPA are discussed and it is shown how the total sum of squares in GPA may be partitioned in such a way that the relative contributions of individual configurations, stimuli and dimensions to the GPA solution can be identified.

In section 2.6 we discuss the relations between the GPA match criterion and Van de Geer's (1984) match criteria MAXBET, MAXNEAR and MAXDIFF when the configurations contain missing data, and in section 2.7, finally, two examples are presented of the generalized Procrustes analysis of configurations containing missing data, the first example consisting of a constructed data set and the second example of empirical data.

2.3 The centroid approach

2.3.1 Translations

In this section the least squares loss function according to the centroid approach

$$f(\mathbf{U},\mathbf{s},\mathbf{R},\mathbf{Z}) = \sum_{j=1}^{n} \text{tr}\ [s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j - \mathbf{Z}]'\mathbf{M}_j[s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j - \mathbf{Z}] \qquad (2.8)$$

is minimized with respect to the unrestricted translation vectors $\mathbf{u}_j$ for fixed $\mathbf{s}$, $\mathbf{R}$, and $\mathbf{Z}$. It may be noted that $\mathbf{Z}$ has now been included in the loss function as a variable. Defining $\mathbf{A}_j = s_j\mathbf{X}_j\mathbf{R}_j - \mathbf{Z}$, and only considering one particular translation vector $\mathbf{u}_j$, function (2.8) can be rewritten as

$$f(\mathbf{u}_j) = s_j^2\mathbf{u}_j'\mathbf{u}_j\mathbf{1}'\mathbf{M}_j\mathbf{1} - 2\ s_j\mathbf{u}_j'\mathbf{R}_j\mathbf{A}_j'\mathbf{M}_j\mathbf{1} + d_j = c_j^2\ \mathbf{u}_j'\mathbf{u}_j - 2\ c_j\mathbf{u}_j'\mathbf{y}_j + d_j, \quad (2.11)$$

where d_j is a term independent of $\mathbf{u}_j$, $\mathbf{y}_j \equiv \mathbf{R}_j\mathbf{A}_j'\mathbf{M}_j\mathbf{1}/\sqrt{\mathbf{1}'\mathbf{M}_j\mathbf{1}}$, and $c_j \equiv s_j\sqrt{\mathbf{1}'\mathbf{M}_j\mathbf{1}}$. Since

$$f(\mathbf{u}_j) = \|c_j\mathbf{u}_j - \mathbf{y}_j\|^2 + d_j - \mathbf{y}_j'\mathbf{y}_j \ \geq\ d_j - \mathbf{y}_j'\mathbf{y}_j,$$

this lower bound, and hence the unique global minimum, is attained for $c_j\mathbf{u}_j - \mathbf{y}_j = \mathbf{0}$, and thus for

$$\mathbf{u}_j = c_j^{-1}\mathbf{y}_j. \qquad (2.12)$$

Re-expressing (2.12) in the original terms yields

$$\mathbf{u}_j = \frac{(s_j\mathbf{X}_j - \mathbf{Z}\mathbf{R}'_j)'\mathbf{M}_j\mathbf{1}}{s_j\mathbf{1}'\mathbf{M}_j\mathbf{1}}. \qquad (2.13)$$

Substitution of (2.13) in $\{s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j - \mathbf{Z}\}$, which is a part of (2.8), gives

$$s_j(\mathbf{X}_j - \frac{\mathbf{1}\mathbf{1}'\mathbf{M}_j(s_j\mathbf{X}_j - \mathbf{Z}\mathbf{R}'_j)}{s_j\ \mathbf{1}'\mathbf{M}_j\mathbf{1}})\mathbf{R}_j - \mathbf{Z} = \mathbf{J}_j(s_j\mathbf{X}_j\mathbf{R}_j - \mathbf{Z}),$$

with $\mathbf{J}_j = \mathbf{I} - \{(\mathbf{1}\mathbf{1}'\mathbf{M}_j)/(\mathbf{1}'\mathbf{M}_j\mathbf{1})\}$.

Hence, the complete loss function can be written as

$$f(\mathbf{s},\mathbf{R},\mathbf{Z}) = \sum_{j=1}^{n} \text{tr } (s_j\mathbf{X}_j\mathbf{R}_j - \mathbf{Z})'\mathbf{J}_j'\mathbf{M}_j\mathbf{J}_j(s_j\mathbf{X}_j\mathbf{R}_j - \mathbf{Z}). \tag{2.14}$$

Result (2.14) implies that, if one wants to match n configurations according to the centroid approach in GPA, the translation problem is simply taken care of by centering the non-missing elements in each column of $\mathbf{X}_j$ *and* the corresponding elements in each column of $\mathbf{Z}$ on the origin of m-dimensional space. In the case of missing data, the geometrical implication of (2.14) is that the matrix of centroids $\mathbf{Z}$ is translated optimally, and thus *differently*, for each configuration j. For each stimulus i, the error is no longer measured as the sum of squared distances between the n_i points in the $\mathbf{X}_j$ and one fixed centroid point (as in Figure 2.3), but as the sum of squared distances between the n_i points in the $\mathbf{X}_j$ centered on the origin and a centroid point that is idiosyncratically translated for each j, and therefore optimally 'moved around' in m-dimensional space. The situation in Figure 2.3 is immediately restored, however, by calculating $\mathbf{u}_j$ for each j according to (2.13) and then plotting $s_j\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j$ for each j. In that case $\mathbf{Z}$ is again at the fixed centroid of the optimally transformed configurations $\mathbf{X}_j$, while the amount of loss in (2.8) is identical to the amount of loss in (2.14).

The nice effect of the substitution of (2.13) in (2.8) is that the translation vectors are eliminated from loss function (2.8) and are implicitly optimized in the simplified loss function (2.14).

We end this section by noting that $\mathbf{M}_j\mathbf{J}_j = \mathbf{J}_j'\mathbf{M}_j\mathbf{J}_j$ is symmetric and idempotent, as is readily verified. It is also important to see that $\mathbf{J}_j\mathbf{1} = \mathbf{0}$, showing that $\mathbf{J}_j$ is always a singular matrix. Since the matrix product $\mathbf{M}_j\mathbf{J}_j = \mathbf{J}_j'\mathbf{M}_j\mathbf{J}_j$ occurs very often in this book, in the sequel we will replace it by the (p x p) matrix $\mathbf{C}_j$. This matrix has the effect of centering the non-missing elements in each column of configuration $\mathbf{X}_j$ *and* the corresponding elements in each column of $\mathbf{Z}$ on the origin. The loss function from which the translation vectors have been eliminated can thus be written as

$$f(\mathbf{s},\mathbf{R},\mathbf{Z}) = \sum_{j=1}^{n} \text{tr } (s_j\mathbf{X}_j\mathbf{R}_j - \mathbf{Z})'\mathbf{C}_j(s_j\mathbf{X}_j\mathbf{R}_j - \mathbf{Z}), \tag{2.15}$$

and henceforth this simplified loss function is used to determine the remaining unknown parameters.

2.3.2 The configuration of centroids Z

In section 2.2 we showed how to calculate the coordinates of the centroids of corresponding points in loss function (2.8) by using (2.9), and discussed that (2.9) also yields the global minimum of (2.8) if **Z** is treated as a variable. After elimination of the translation vectors from (2.8), however, (2.9) only optimizes (2.15) if the optimal translation vectors are calculated using (2.13), and then substituted in (2.9). But this is not what we are after, of course, since the latter procedure would re-introduce the translation vectors that we just eliminated from (2.8). Hence, in this section we investigate the problem of determining the matrix of centroids **Z** optimizing (2.15) without taking recourse to (2.13).

Since in this problem we consider **s** and **R** to be fixed, let $\mathbf{A}_j = s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j$. Also let the matrices $\mathbf{A}_j$ be collected in the supermatrix **A** of order (np x m), and let the centering matrices $\mathbf{C}_j$ be collected in the supermatrix **B** of order (np x p). Then loss function (2.15) may be rewritten as

$$f(\mathbf{Z}) = \text{tr}\,(\mathbf{A} - \mathbf{BZ})'(\mathbf{A} - \mathbf{BZ}). \tag{2.16}$$

To minimize (2.16) with respect to **Z** is the classical multivariate multiple regression problem. The general solution of this problem is well-known, and follows from the normal equations corresponding to (2.16):

$$\mathbf{B}'\mathbf{BZ} = \mathbf{B}'\mathbf{A}. \tag{2.17}$$

Since the matrices $\mathbf{C}_j$ are symmetric and idempotent, re-expressing (2.17) in the original matrices and scalars yields

$$\mathbf{CZ} = \sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j, \tag{2.18}$$

where $\mathbf{C} \equiv (\sum \mathbf{C}_j)$. Hence, the global minimum of (2.15) with respect to **Z** is attained for

$$\mathbf{Z} = \mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j), \tag{2.19}$$

where $\mathbf{C}^{-}$ is the Moore-Penrose generalized inverse of the sum of the $\mathbf{C}_j$'s.

The reason that a generalized inverse must be determined in order to solve (2.18) for unknown $\mathbf{Z}$ is that

$$\mathbf{C1} = \sum_{j=1}^{n} \mathbf{M}_j\mathbf{J}_j\mathbf{1} = \sum_{j=1}^{n} \mathbf{M}_j\mathbf{0} = \mathbf{0}$$

(see also section 2.3.1), indicating that the sum of the $\mathbf{C}_j$'s has no proper inverse. The Moore-Penrose inverse is calculated as follows. Because $\mathbf{C}_j$ is always a symmetric matrix, the sum of the $\mathbf{C}_j$'s is also symmetric. Therefore, let

$$\mathbf{C} = \mathbf{Q}\mathbf{\Phi}\mathbf{Q}'$$

be an eigenvalue-eigenvector decomposition of the sum of the centring matrices. Matrix $\mathbf{\Phi}$ does not contain any negative eigenvalues, since the sum of the $\mathbf{C}_j$'s is positive semi-definite, each $\mathbf{C}_j$ being positive semi-definite. Let $\mathbf{\Phi}^*$ be the submatrix of $\mathbf{\Phi}$ only containing the positive eigenvalues, and let $\mathbf{Q}^*$ be the matrix of eigenvectors corresponding to $\mathbf{\Phi}^*$. The Moore-Penrose generalized inverse is then determined by

$$\mathbf{C}^- = \mathbf{Q}^*\mathbf{\Phi}^{*-1}\mathbf{Q}^{*\prime}.$$

(see, e.g., Van de Geer, 1986).

Geometrically, the matrix $\mathbf{Z}$ calculated with (2.19) does *not* contain the simple averages of corresponding points in the optimally transformed $s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j$'s. This is related to the fact that in (2.15) matrix $\mathbf{Z}$ is optimally translated but this translation is different for each j. Calculating the optimal translation vectors (2.13) for each j after (2.19) has been computed, however, and substituting these optimal $\mathbf{u}_j$'s in (2.9) guarantees that $\mathbf{Z}$ computed according to (2.19) is exactly equal to $\mathbf{Z}$ calculated using (2.9). We will now prove that this is indeed the case.

It follows from (2.9) that

$$(\sum_{j=1}^{n} \mathbf{M}_j)\mathbf{Z} = \sum_{j=1}^{n} s_j\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j,$$

and substitution of (2.13), the optimal translation vectors, in the latter expression gives

2.3.3 The elimination of Z

The loss function corresponding to the centroid approach from which the translation vectors have been eliminated is (2.15), that is,

$$f(\mathbf{s},\mathbf{R},\mathbf{Z}) = \sum_{j=1}^{n} \text{tr } (s_j\mathbf{X}_j\mathbf{R}_j - \mathbf{Z})'\mathbf{C}_j(s_j\mathbf{X}_j\mathbf{R}_j - \mathbf{Z}),$$

and the optimum of this function for unknown $\mathbf{Z}$ is found where

$$\mathbf{Z} = \mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j). \tag{2.19}$$

The key procedure to change (2.15) into a loss function that does not explicitly contain the unknown centroid configuration is to substitute (2.19) into (2.15):

$$f(\mathbf{s},\mathbf{R}) =$$

$$\sum_{j=1}^{n} \text{tr } [s_j\mathbf{X}_j\mathbf{R}_j - \mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)]'\mathbf{C}_j[s_j\mathbf{X}_j\mathbf{R}_j - \mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)] =$$

$$\sum_{j=1}^{n} s_j^2 \text{ tr } \mathbf{R}_j'\mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j - \text{tr } (\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)'\mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j). \tag{2.22}$$

If all configurations are complete $\mathbf{C}_j = \mathbf{I} - (\mathbf{11'/1'1}) \equiv \mathbf{J}$ for each j, and (2.22) may be written as

$$f(\mathbf{s},\mathbf{R}) = \sum_{j=1}^{n} s_j^2 \text{ tr } \mathbf{R}_j'\mathbf{X}_j'\mathbf{J}\mathbf{X}_j\mathbf{R}_j - \text{tr } (\sum_{j=1}^{n} s_j\mathbf{J}\mathbf{X}_j\mathbf{R}_j)'(n\mathbf{J})^{-}(\sum_{j=1}^{n} s_j\mathbf{J}\mathbf{X}_j\mathbf{R}_j)$$

$$= \sum_{j=1}^{n} s_j^2 \text{ tr } \mathbf{R}_j'\mathbf{X}_j'\mathbf{J}\mathbf{X}_j\mathbf{R}_j - (1/n) \text{ tr } (\sum_{j=1}^{n} s_j\mathbf{J}\mathbf{X}_j\mathbf{R}_j)'(\sum_{j=1}^{n} s_j\mathbf{J}\mathbf{X}_j\mathbf{R}_j)$$

$$= (1/n) \sum_{j<k} \text{tr } (s_j\mathbf{J}\mathbf{X}_j\mathbf{R}_j - s_k\mathbf{J}\mathbf{X}_k\mathbf{R}_k)'(s_j\mathbf{J}\mathbf{X}_j\mathbf{R}_j - s_k\mathbf{J}\mathbf{X}_k\mathbf{R}_k).$$

This is the well-known GPA format for the direct approach without translations if all configurations are complete (see, e.g., Gower, 1975; Ten Berge, 1977), and, again, confirms the equivalence between the centroid and the direct approach in GPA.

With (2.22) we are in a position to determine the optimal orthonormal transformations and uniform scaling factors without the slackening effect of $\mathbf{Z}$, and therefore to generalize Ten Berge's (1977) efficient method to the case of missing

$$(\sum_{j=1}^{n} \mathbf{M}_j)\mathbf{Z} = \sum_{j=1}^{n} s_j\mathbf{M}_j(\mathbf{X}_j - \frac{\mathbf{11'M}_j(s_j\mathbf{X}_j - \mathbf{Z}\mathbf{R}_j')}{s_j\mathbf{1'M}_j\mathbf{1}})\mathbf{R}_j$$

$$= \sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j + (\sum_{j=1}^{n} \mathbf{M}_j\mathbf{11'M}_j/\mathbf{1'M}_j\mathbf{1})\mathbf{Z}. \qquad (2.20)$$

Since it is always true that

$$\mathbf{CZ} = [\sum_{j=1}^{n} \mathbf{M}_j(\mathbf{I} - \mathbf{11'M}_j/\mathbf{1'M}_j\mathbf{1})]\mathbf{Z} = (\sum_{j=1}^{n} \mathbf{M}_j)\mathbf{Z} - (\sum_{j=1}^{n} \mathbf{M}_j\mathbf{11'M}_j/\mathbf{1'M}_j\mathbf{1})\mathbf{Z},$$

the following also holds:

$$(\sum_{j=1}^{n} \mathbf{M}_j\mathbf{11'M}_j/\mathbf{1'M}_j\mathbf{1})\mathbf{Z} = (\sum_{j=1}^{n} \mathbf{M}_j)\mathbf{Z} - \mathbf{CZ}. \qquad (2.21)$$

Substitution of (2.21) in (2.20) gives

$$(\sum_{j=1}^{n} \mathbf{M}_j)\mathbf{Z} = \sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j + (\sum_{j=1}^{n} \mathbf{M}_j)\mathbf{Z} - \mathbf{CZ},$$

and therefore

$$(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j) = \mathbf{CZ}.$$

This completes the proof of the equality of **Z** in (2.19) and (2.9) for optimal translation vectors (2.13).

Due to this equality it also follows that, although **Z** in (2.19) does not contain the averages of corresponding points in the $s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j$, it does contain the averages in the $s_j\mathbf{M}_j(\mathbf{X}_j - \mathbf{1u}_j')\mathbf{R}_j$ after updating **U**, indicating that Theorem 2 still holds after elimination of the translation vectors from (2.8), and with matrix **Z** treated as an unknown variable.

data. In the next two sections (2.22) is minimized with respect to the remaining unknowns **R** and **s**.

2.3.4 Orthonormal transformations

The minimization of loss function (2.22), that is, of

$$f(\mathbf{s},\mathbf{R}) = \sum_{j=1}^{n} s_j^2 \operatorname{tr} \mathbf{R}_j'\mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j - \operatorname{tr} (\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)'\mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)$$

with respect to the unknown matrix **R** for fixed **s** is equivalent to the maximization of

$$g(\mathbf{R}) = \operatorname{tr} (\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)'\mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j) \tag{2.23}$$

given the constraint

$$\mathbf{R}_j'\mathbf{R}_j = \mathbf{R}_j\mathbf{R}_j' = \mathbf{I}_m, \qquad \text{for } j = 1, \ldots, n. \tag{2.24}$$

If we let $\mathbf{A}_j = s_j\mathbf{C}_j\mathbf{X}_j$ and consider one particular $\mathbf{R}_j$, function (2.23) can be written as

$$g(\mathbf{R}_j) = \operatorname{tr} \mathbf{R}_j'\mathbf{A}_j'\mathbf{C}^{-}\mathbf{A}_j\mathbf{R}_j + 2 \operatorname{tr} \mathbf{R}_j'\mathbf{A}_j'\mathbf{C}^{-}(\sum_{i\neq j} \mathbf{A}_i\mathbf{R}_i) + d_j, \tag{2.25}$$

where d_j is a term independent of $\mathbf{R}_j$. Therefore, the maximization of (2.25) is equivalent to the maximization of

$$h(\mathbf{R}_j) = \operatorname{tr} \mathbf{R}_j'\mathbf{A}_j'\mathbf{C}^{-}(\sum_{i\neq j} \mathbf{A}_i\mathbf{R}_i) \tag{2.26}$$

under constraint (2.24). Let

$$\mathbf{A}_j'\mathbf{C}^{-}(\sum_{i\neq j} \mathbf{A}_i\mathbf{R}_i) = \mathbf{P}_j\mathbf{\Phi}_j\mathbf{Q}_j' \tag{2.27}$$

be a singular value decomposition of $\mathbf{A}_j'\mathbf{C}^{-}(\sum_{i\neq j} \mathbf{A}_i\mathbf{R}_i)$. Substitution of (2.27) in (2.26) gives

$$h(\mathbf{R}_j) = \operatorname{tr} \mathbf{R}_j'\mathbf{P}_j\mathbf{\Phi}_j\mathbf{Q}_j' = \operatorname{tr} \mathbf{Q}_j'\mathbf{R}_j'\mathbf{P}_j\mathbf{\Phi}_j = \operatorname{tr} \mathbf{B}_j\mathbf{\Phi}_j,$$

where $\mathbf{B}_j = \mathbf{Q}_j'\mathbf{R}_j'\mathbf{P}_j$ is an orthonormal matrix of order (m x m), being the product of orthonormal matrices $\mathbf{Q}_j'$, $\mathbf{R}_j'$ and $\mathbf{P}_j$. Therefore, it is always true that

$$h(\mathbf{R}_j) = \text{tr } \mathbf{B}_j\mathbf{\Phi}_j \leq \text{tr } \mathbf{\Phi}_j, \tag{2.28}$$

and this upper limit is attained for

$$\mathbf{B}_j = \mathbf{Q}_j'\mathbf{R}_j'\mathbf{P}_j = \mathbf{I}_m,$$

and thus for

$$\mathbf{R}_j = \mathbf{P}_j\mathbf{Q}_j'. \tag{2.29}$$

Solution (2.29) obviously satisfies (2.24), and holds regardless of singularity of (2.27). Inequality (2.28) is due to Kristof (1970) and, in a more general form, Ten Berge (1983).

Since (2.23) can not be solved for the n $\mathbf{R}_j$'s simultaneously, an alternative least squares algorithm must be used where (2.23) is consecutively being maximized for j = 1, 2, ..., n and $\mathbf{R}$ is updated after each step. This process is repeated until n steps jointly fail to raise (2.23) above some threshold value.

2.3.5 Scaling constants

In this section we will show how to minimize (2.22) with respect to the unknown uniform scaling constants. When we are dealing with incomplete configurations it is essential to determine optimal scaling constants in GPA. We will illustrate this point with a simple example. The example consists of only two configurations in two dimensions with coordinates given on the left of Table 2.2 and drawn on the left of Figure 2.4. Configuration $\mathbf{X}_1$ has four stimuli that form the vertices of a square. Configuration $\mathbf{X}_2$ is identical to configuration $\mathbf{X}_1$, except that information about the second stimulus is assumed to be missing in $\mathbf{X}_2$. Both configurations have been centered on the origin. Notice that the distances between stimuli 1, 3, and 4 in $\mathbf{C}_1\mathbf{X}_1$ are identical to the distances between stimuli 1, 3, and 4 in $\mathbf{C}_2\mathbf{X}_2$. Loosely speaking the two configurations $\mathbf{C}_1\mathbf{X}_1$ and $\mathbf{C}_2\mathbf{X}_2$ are 'of the same size'. Translations and orthonormal transformations of these two configurations therefore suffice to obtain a perfect solution. In practice this situation will rarely occur, however. Usually the

Table 2.2 Coordinates of the configurations in Figure 2.4 before (left) and after (right) unit normalization.

stimulus	$\mathbf{C}_1\mathbf{X}_1$	$\mathbf{C}_2\mathbf{X}_2$	$\mathbf{C}_1\tilde{\mathbf{X}}_1$	$\mathbf{C}_2\tilde{\mathbf{X}}_2$
1	1 0	1 -1/3	1/2 0	.612 -.204
2	0 -1	99 99	0 -1/2	99 99
3	-1 0	-1 -1/3	-1/2 0	-.612 -.204
4	0 1	0 2/3	0 1/2	0 .408

situation will be as is illustrated on the right of Figure 2.4. There, the same two configurations are drawn after having been unit normalized (i.e., $\text{tr}\,\tilde{\mathbf{X}}_1\mathbf{C}_1\tilde{\mathbf{X}}_1 = \text{tr}\,\tilde{\mathbf{X}}_2\mathbf{C}_2\tilde{\mathbf{X}}_2 = 1$). The coordinates of the unit normalized configurations are given at the right side of Table 2.2. Due to the unit normalization, the distances between stimuli 1, 3, and 4 in $\mathbf{C}_1\tilde{\mathbf{X}}_1$ are no longer identical to the distances between stimuli 1, 3, and 4 in $\mathbf{C}_2\tilde{\mathbf{X}}_2$. Obviously, in this case it is impossible to find a perfect solution if the only admissible transformations are translations, rotations and reflections. This shows that, if the configurations contain missing values, and if no care is taken to correct for differences in 'size' between the configurations, neither the loss function according to the centroid approach nor the loss function using the direct approach is in general bounded from below by zero (i.e., a perfect solution is never found, even if it

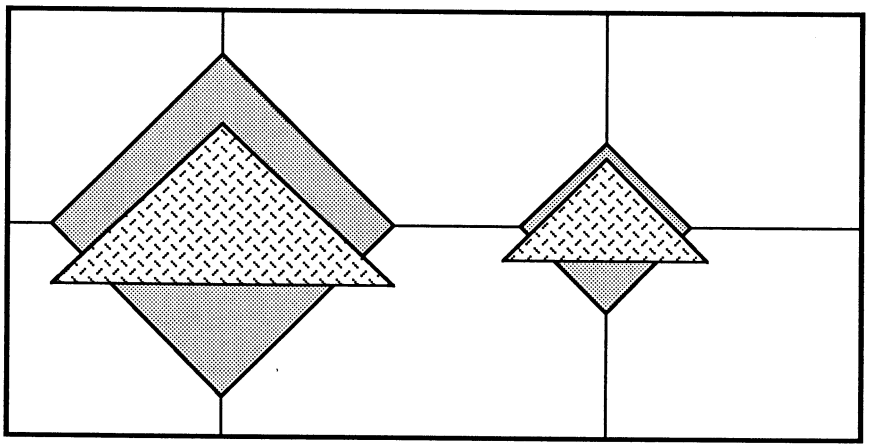


Figure 2.4 Configurations $\mathbf{X}_1$ and $\mathbf{X}_2$ from Table 2.2, both before (left) and after (right) unit normalization.

exists). In the case of missing data, therefore, it is mandatory to implement scaling factors in the corresponding loss functions.

Some kind of restriction must be imposed upon the scaling factors that minimize (2.22) because the ensuing algorithm otherwise will converge to the trivial solution $\mathbf{s} = \mathbf{0}$. The scaling factors will be restricted to satisfy

$$\sum_{j=1}^{n} s_j^2 \operatorname{tr} \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j = \sum_{j=1}^{n} \operatorname{tr} \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j. \tag{2.30}$$

This side condition is a generalization of the constraint that Gower (1975) used in the case without missing data. Restriction (2.30) expresses that the sum of squares about the origin of the scaled configurations must remain equal to the sum of squares of the original configurations centered on the origin.

Considering $\mathbf{R}$ as fixed, and letting $\mathbf{A}_j = \mathbf{C}_j\mathbf{X}_j\mathbf{R}_j$, the loss function to be minimized is

$$f(\mathbf{s}) = \sum_{j=1}^{n} s_j^2 \operatorname{tr} \mathbf{A}_j'\mathbf{A}_j - \operatorname{tr} (\sum_{j=1}^{n} s_j\mathbf{A}_j)'\mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{A}_j). \tag{2.22}$$

Because the scaling constants are constrained to satisfy

$$\sum_{j=1}^{n} s_j^2 \operatorname{tr} \mathbf{A}_j'\mathbf{A}_j = \sum_{j=1}^{n} \operatorname{tr} \mathbf{A}_j'\mathbf{A}_j, \tag{2.30}$$

the minimization of (2.22) for unknown scaling constants is equivalent to the maximization of

$$h(\mathbf{s}) = \operatorname{tr} (\sum_{j=1}^{n} s_j\mathbf{A}_j)'\mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{A}_j). \tag{2.31}$$

Let matrix $\mathbf{Y}$ of order (n x n) be defined as

$$\mathbf{Y} = \begin{bmatrix} \operatorname{tr} \mathbf{A}_1'\mathbf{C}^{-}\mathbf{A}_1 & \operatorname{tr} \mathbf{A}_1'\mathbf{C}^{-}\mathbf{A}_2 & \dots & \operatorname{tr} \mathbf{A}_1'\mathbf{C}^{-}\mathbf{A}_n \\ \operatorname{tr} \mathbf{A}_2'\mathbf{C}^{-}\mathbf{A}_1 & \operatorname{tr} \mathbf{A}_2'\mathbf{C}^{-}\mathbf{A}_2 & \dots & \operatorname{tr} \mathbf{A}_2'\mathbf{C}^{-}\mathbf{A}_n \\ \vdots & \vdots & & \vdots \\ \operatorname{tr} \mathbf{A}_n'\mathbf{C}^{-}\mathbf{A}_1 & \operatorname{tr} \mathbf{A}_n'\mathbf{C}^{-}\mathbf{A}_2 & \dots & \operatorname{tr} \mathbf{A}_n'\mathbf{C}^{-}\mathbf{A}_n \end{bmatrix}. \tag{2.32}$$

Also, define matrix $\mathbf{W}$ of order (n x n) as

$$\mathbf{W} = \begin{bmatrix} \text{tr } \mathbf{A}_1'\mathbf{A}_1 & 0 & \dots & 0 \\ 0 & \text{tr } \mathbf{A}_2'\mathbf{A}_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \text{tr } \mathbf{A}_n'\mathbf{A}_n \end{bmatrix}. \tag{2.33}$$

Finally, let

$$\mathbf{W}^{-1/2}\mathbf{Y}\mathbf{W}^{-1/2} = \mathbf{P}\boldsymbol{\Phi}\mathbf{P}' \tag{2.34}$$

be an eigenvector-eigenvalue decomposition of $\mathbf{W}^{-1/2}\mathbf{Y}\mathbf{W}^{-1/2}$, and let $\mathbf{p}_1$ be the first column of $\mathbf{P}$.

Theorem 3. The scaling factors $\mathbf{s}$ that maximize (2.31) under constraint (2.30) are

$$\mathbf{s} = (\text{tr } \mathbf{W})^{1/2}\mathbf{W}^{-1/2}\mathbf{p}_1, \tag{2.35}$$

where $\mathbf{W}$ and $\mathbf{p}_1$ are defined in (2.33) and (2.34), respectively.

Proof. Function (2.31) can be written as

$$h(\mathbf{s}) = \mathbf{s}'\mathbf{Y}\mathbf{s}, \tag{2.36}$$

which must be maximized under constraint (2.30) that can be written as

$$\mathbf{s}'\mathbf{W}\mathbf{s} = \text{tr } \mathbf{W}. \tag{2.37}$$

Since $\mathbf{W}^{1/2}\mathbf{W}^{-1/2} = \mathbf{I}_n$, expression (2.36) may be rewritten as

$$h(\mathbf{s}) = \mathbf{s}'\mathbf{W}^{1/2}\mathbf{W}^{-1/2}\mathbf{Y}\mathbf{W}^{-1/2}\mathbf{W}^{1/2}\mathbf{s}. \tag{2.38}$$

Substitution of (2.34) in (2.38) gives

$$h(\mathbf{s}) = \mathbf{s}'\mathbf{W}^{1/2}\mathbf{P}\boldsymbol{\Phi}\mathbf{P}'\mathbf{W}^{1/2}\mathbf{s}, \tag{2.39}$$

and if we let $\mathbf{u} = \mathbf{P}'\mathbf{W}^{1/2}\mathbf{s}$, expression (2.39) simplifies into

$$h(\mathbf{s}) = \mathbf{u}'\boldsymbol{\Phi}\mathbf{u}.$$

But, since

$$h(\mathbf{s}) = \mathbf{u}'\mathbf{\Phi u} \leq \phi_1 \mathbf{u}'\mathbf{u},$$

and because

$$\mathbf{u}'\mathbf{u} = \mathbf{s}'\mathbf{W}^{1/2}\mathbf{P}\mathbf{P}'\mathbf{W}^{1/2}\mathbf{s} = \mathbf{s}'\mathbf{W}\mathbf{s} = \text{tr } \mathbf{W},$$

the following always holds:

$$h(\mathbf{s}) = \mathbf{u}'\mathbf{\Phi u} \leq \phi_1(\text{tr } \mathbf{W}), \tag{2.40}$$

whatever the values of the scaling constants are, as long as they satisfy $\mathbf{s}'\mathbf{W}\mathbf{s} = \text{tr } \mathbf{W}$. The upper bound in (2.40) is attained for (2.35) since

$$\begin{aligned} h((\text{tr } \mathbf{W})^{1/2}\mathbf{W}^{-1/2}\mathbf{p}_1) &= [(\text{tr } \mathbf{W})^{1/2}\mathbf{W}^{-1/2}\mathbf{p}_1]'\mathbf{Y}[(\text{tr } \mathbf{W})^{1/2}\mathbf{W}^{-1/2}\mathbf{p}_1] \\ &= (\text{tr } \mathbf{W})\, \mathbf{p}_1'\mathbf{P}\mathbf{\Phi}\mathbf{P}'\mathbf{p}_1 = \phi_1(\text{tr } \mathbf{W}). \end{aligned} \tag{2.41}$$

The optimal scaling constants in (2.35) satisfy constraint (2.37) because

$$\mathbf{s}'\mathbf{W}\mathbf{s} = [(\text{tr } \mathbf{W})^{1/2}\mathbf{W}^{-1/2}\mathbf{p}_1]'\mathbf{W}[(\text{tr } \mathbf{W})^{1/2}\mathbf{W}^{-1/2}\mathbf{p}_1] = \text{tr } \mathbf{W}.$$

This completes the proof.

Concluding, for fixed $\mathbf{R}$ an analytical solution is available for the unknown scaling constants $\mathbf{s}$ which is guaranteed to yield the global optimum of the GPA loss function. To minimize (2.22) with respect to $\mathbf{R}$ and $\mathbf{s}$, however, an iterative procedure must be used. In the next section we present a two-step algorithm for the generalized Procrustes analysis of configurations containing missing data.

2.4 The algorithm

The algorithm for generalized Procrustes analysis of data with and without missing rows can be set up as follows.

1. Initialization step

For each j (j = 1, 2, ..., n) calculate

$$\mathbf{C}_j = \mathbf{M}_j(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}'\mathbf{M}_j}{\mathbf{1}'\mathbf{M}_j\mathbf{1}})$$

of order (p x p) and determine the Moore-Penrose generalized inverse of the sum of the $\mathbf{C}_j$'s as described in section 2.3.2, that is, determine

$$\mathbf{C}^- = (\sum_{j=1}^{n} \mathbf{C}_j)^- .$$

Normalize the n configurations such that

$$\sum_{j=1}^{n} \text{tr } \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j = n.$$

It may be noted that this normalization is not strictly necessary. Our main reason for using it is that it helps to keep the precision of the GPA algorithm (and of the algorithms to be developed in subsequent chapters) within more or less uniform bounds. Construct the matrix

$$\mathbf{W}^{-1/2} = \begin{bmatrix} 1/\sqrt{\text{tr } \mathbf{X}_1'\mathbf{C}_1\mathbf{X}_1} & 0 & \dots & 0 \\ 0 & 1/\sqrt{\text{tr } \mathbf{X}_2'\mathbf{C}_2\mathbf{X}_2} & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1/\sqrt{\text{tr } \mathbf{X}_n'\mathbf{C}_n\mathbf{X}_n} \end{bmatrix} .$$

Initialize all n scaling constants on $s_j = 1$, and set all n initial $\mathbf{R}_j$'s equal to $\mathbf{I}_m$.

2. Orthonormal transformation step

Determine for each j a new orthonormal matrix $\mathbf{R}_j$ using the singular value decomposition (see section 2.3.4)

$$\mathbf{X}_j'\mathbf{C}_j\mathbf{C}^{-}(\sum_{i\neq j} s_i\mathbf{C}_i\mathbf{X}_i\mathbf{R}_i) = \mathbf{P}_j\mathbf{\Phi}_j\mathbf{Q}_j'. \tag{2.27}$$

In practice, the orthonormal transformation procedure in MATCHALS has been programmed as follows. We use the Jacobi method to calculate the eigenvalue-eigenvector decomposition of

$$\mathbf{A}_j'\mathbf{A}_j = \mathbf{Q}_j\mathbf{\Phi}_j^2\mathbf{Q}_j', \tag{2.42}$$

where

$$\mathbf{A}_j \equiv \mathbf{X}_j'\mathbf{C}_j\mathbf{C}^{-}(\sum_{i\neq j} s_i\mathbf{C}_i\mathbf{X}_i\mathbf{R}_i).$$

If $\mathbf{A}_j$ has full column rank (i.e., if $\mathbf{\Phi}_j^2$ has m positive eigenvalues), then the left singular vectors $\mathbf{P}_j$ are computed as

$$\mathbf{P}_j = \mathbf{A}_j\mathbf{Q}_j\mathbf{\Phi}_j^{-1}, \tag{2.43}$$

and the optimal orthonormal $\mathbf{R}_j$ is computed as $\mathbf{R}_j = \mathbf{P}_j\mathbf{Q}_j'$.

If $\mathbf{A}_j$ has rank $r < m$ (i.e., if the diagonal of $\mathbf{\Phi}_j^2$ contains one or more zeroes), $\mathbf{P}_j^*$ of order (m x r) is calculated as

$$\mathbf{P}_j^* = \mathbf{A}_j\mathbf{Q}_j^*\mathbf{\Phi}_j^{*-1}, \tag{2.44}$$

where $\mathbf{\Phi}_j^*$ is the (r x r) submatrix of $\mathbf{\Phi}_j$ only containing the positive singular values, and $\mathbf{Q}_j^*$ is the (m x r) matrix of eigenvectors corresponding to $\mathbf{\Phi}_j^*$. Then (m – r) orthonormal columns collected in the (m x (m – r)) matrix $\bar{\mathbf{P}}_j$ must be appended to $\mathbf{P}_j^*$ satisfying $\mathbf{P}_j^{*\prime}\bar{\mathbf{P}}_j = \mathbf{0}$ and $\bar{\mathbf{P}}_j'\bar{\mathbf{P}}_j = \mathbf{I}_{(m-r)}$.

The nullspace $\bar{\mathbf{P}}_j$ of $\mathbf{P}_j^*$ is calculated as follows. Using a random number generator a vector $\mathbf{y}$ of order (m x 1) containing random numbers is constructed. Then the multiple regression of $\mathbf{y}$ on the r columns of $\mathbf{P}_j^*$ is calculated. If we let $\hat{\mathbf{y}}$ be the vector of predicted values on account of the regression equation, the (m x 1) column vector of residuals $(\mathbf{y} - \hat{\mathbf{y}})$ is calculated and unit normalized. This unit normalized vector of

residuals is guaranteed to be orthogonal to all r columns of $\mathbf{P}_j^*$ and is used as the first column of matrix $\bar{\mathbf{P}}_j$. Then a new (m x 1) vector **y** of random numbers is generated, the multiple regression of this new **y** on the r columns of $\mathbf{P}_j^*$ *and* the first column of $\bar{\mathbf{P}}_j$ is executed, and the unit normalized vector of residuals is used as the second column of $\bar{\mathbf{P}}_j$, etc., etc. This procedure is repeated (m – r) times, yielding an (m x (m – r)) orthonormal nullspace $\bar{\mathbf{P}}_j$ of matrix $\mathbf{P}_j^*$. If we let $\mathbf{P}_j = [\mathbf{P}_j^*, \bar{\mathbf{P}}_j]$, that is, matrix $\mathbf{P}_j^*$ of order (m x r) appended with the (m – r) columns of $\bar{\mathbf{P}}_j$, then the optimal orthonormal transformation is $\mathbf{R}_j = \mathbf{P}_j\mathbf{Q}_j'$.

After the calculation of n new orthonormal matrices $\mathbf{R}_j$ following the above procedure, evaluate loss function (2.22).

3. Rescaling step

Set up matrix **Y** of order (p x p) as defined in section 2.3.5, and determine the eigenvalue-eigenvector decomposition of

$$\mathbf{W}^{-1/2}\mathbf{Y}\mathbf{W}^{-1/2}.$$

To this end use the power method, and only calculate the *first* principal component, that is, only calculate

$$\mathbf{W}^{-1/2}\mathbf{Y}\mathbf{W}^{-1/2} = \mathbf{p}_1\phi_1\mathbf{p}_1' + \mathbf{E},$$

where $\mathbf{p}_1$ is the principal eigenvector, ϕ_1 is the largest eigenvalue and matrix **E** of order (n x n) is a matrix of residuals. The reason that we prefer the power method in this step of the algorithm is that it is more efficient than the Jacobi method as the latter always calculates all eigenvectors and eigenvalues (while only the first one is needed). Then calculate new scaling constants as

$$\mathbf{s} = (\sqrt{n})\mathbf{W}^{-1/2}\mathbf{p}_1. \tag{2.35}$$

Evaluate the loss in this iteration by computing

$$f(\mathbf{s},\mathbf{R}) = n - \mathrm{tr}\,(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)'\mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j), \tag{2.22}$$

and check whether the difference between the loss in this iteration and the loss in the former iteration is smaller than some preset convergence criterion (0.00001, for example). If so, go to step 4, otherwise go to step 2.

4. Results

After the algorithm has converged, determine the centroid configuration using

$$\mathbf{Z} = \mathbf{C}^{-}(\sum_{j=1}^{n} s_j \mathbf{C}_j \mathbf{X}_j \mathbf{R}_j), \tag{2.19}$$

and then calculate the optimal translation vectors $\mathbf{u}_j$ using (2.13). As is easily verified, the whole solution of GPA is unique up to a simultaneous orthonormal transformation of all optimally rotated, reflected and rescaled configurations. For interpretational purposes as well as to make the solution unique the whole solution is rotated to the principal components of $\mathbf{Z}$. Hence, determine the eigenvalue-eigenvector decomposition of

$$\mathbf{Z}'\mathbf{C}\mathbf{Z} = \mathbf{K}\Lambda\mathbf{K}',$$

and print $\mathbf{ZK}$ (i.e., the centroid configuration rotated to its principal components) and $s_j\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j\mathbf{K}$ for each j (i.e., the optimally translated, rotated, reflected and rescaled configurations rotated to the principal components of their centroid configuration). Also print the optimal scaling constants. Finally, print the history of iterations, that is the loss after the orthonormal transformations step and after the rescaling step in each iteration of the algorithm.

This algorithm must converge, although it can not be proved to converge necessarily to the global minimum. Convergence is very fast in practice, and much faster than the first GPA algorithm proposed by Gower (1975). This is due to the fact that in the algorithm proposed in this section the unknowns can be calculated directly, that is, without having to use the centroid configuration as an intermediary.

2.5 Analysis of variation in GPA with missing data

For reasons of simplicity of notation we assume throughout this section that the n configurations have been normalized such that $\sum \text{tr } \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j = n$.

In their 1978 article Lingoes and Borg discussed two measures of fit in the context of the generalized Procrustes analysis model without missing data. The first measure is the squared correlation between the elements of the optimally transformed $\mathbf{X}_j$ and the elements of $\mathbf{Z}$. In Lingoes and Borg's notation the fit for each j is $r^2(\tilde{\mathbf{X}}_j,\mathbf{Z})$, where $\tilde{\mathbf{X}}_j$ is the optimally transformed configuration $\mathbf{X}_j$. They used this measure to assess the relative contribution of each configuration j to the solution of GPA. As a measure of total fit Lingoes and Borg calculated the mean of these configuration fits, that is

$$\text{total fit} = (1/n) \sum_{j=1}^{n} r^2(\tilde{\mathbf{X}}_j,\mathbf{Z}).$$

Lingoes and Schönemann (1974) proposed this measure of fit in the context of fitting one configuration to another if only two configurations are involved. The advantages of this measure are that it is a symmetric measure and that it is not sensitive to differences in scale of the two configurations.

The second measure of fit that Lingoes and Borg briefly discussed is the squared norm of the centroid configuration $\mathbf{Z}$ (i.e., tr $\mathbf{Z}'\mathbf{Z}$), which can be shown to satisfy

$$0 \leq \text{tr } \mathbf{Z}'\mathbf{Z} \leq 1,$$

and therefore can be used as a measure of total fit. Lingoes and Borg pointed to the fact that, if the $\mathbf{X}_j$'s have been unit normalized, a measure of configuration fit could be derived from this total fit by calculating

$$s_j^2 \text{ tr } \mathbf{Z}'\mathbf{Z}$$

for each j. They also noted that in that case there exists the following relation between the measure of fit based on the squared correlation and the one based on the sum of squares of $\mathbf{Z}$:

$$r^2(\tilde{\mathbf{X}}_j,\mathbf{Z}) = s_j^2 \text{ tr } \mathbf{Z}'\mathbf{Z}.$$

In this section we investigate the relations between the two measures mentioned above, and generalize them to the case of configurations containing missing data. We will show that if the configurations are incomplete the two measures are no longer identical, and argue that in this case the second measure of fit (based on the sum of squares of the centroid configuration **Z**) is more appropriate than the use of squared correlations. At the same time we will generalize Gower's (1975) analysis of variance framework to the case of missing data.

We start by generalizing Lingoes and Borg's second measure of fit to missing data, and first note that the linear GPA model underlying loss function (2.15) according to the centroid approach may be written as

$$s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j = \mathbf{C}_j\mathbf{Z} + \mathbf{C}_j\mathbf{E}_j \qquad \text{for } j = 1, ..., n, \tag{2.45}$$

where $\mathbf{E}_j = (s_j\mathbf{X}_j\mathbf{R}_j - \mathbf{Z})$ is a (p x m) matrix of residuals. It follows from (2.45) that

$$\sum_{j=1}^{n} s_j^2 \text{ tr } \mathbf{R}_j'\mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j = \sum_{j=1}^{n} \text{tr } (\mathbf{Z} + \mathbf{E}_j)'\mathbf{C}_j(\mathbf{Z} + \mathbf{E}_j),$$

and therefore that

$$\sum_{j=1}^{n} s_j^2 \text{ tr } \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j = \sum_{j=1}^{n} \text{tr } \mathbf{Z}'\mathbf{C}_j\mathbf{Z} + \sum_{j=1}^{n} \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j + 2\sum_{j=1}^{n} \text{tr } \mathbf{Z}'\mathbf{C}_j\mathbf{E}_j. \tag{2.46}$$

Due to the orthogonality of the predictor and residual space in regression it is true that Σ tr $\mathbf{Z}'\mathbf{C}_j\mathbf{E}_j = 0$. This means that (2.46) may be written as

$$\sum_{j=1}^{n} s_j^2 \text{ tr } \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j = \sum_{j=1}^{n} \text{tr } \mathbf{Z}'\mathbf{C}_j\mathbf{Z} + \sum_{j=1}^{n} \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j,$$

and therefore as

$$n = \text{tr } \mathbf{Z}'\mathbf{C}\mathbf{Z} + f(\mathbf{s},\mathbf{R},\mathbf{Z}), \tag{2.47}$$

with $f(\mathbf{s},\mathbf{R},\mathbf{Z})$ as in (2.15). Equation (2.47) expresses that the total sum of squares of the n configurations about the origin can be partitioned into two parts. The first part, represented by tr $\mathbf{Z}'\mathbf{C}\mathbf{Z}$, is the proportion of the total sum of squares that is accounted for by the GPA model, while the second part, represented by loss function (2.15), is a residual sum of squares. Dividing (2.47) by n gives

$$1 = (1/n) \text{ tr } \mathbf{Z}'\mathbf{C}\mathbf{Z} + (1/n)\, f(\mathbf{s},\mathbf{R},\mathbf{Z}),$$

which proves that the sum of squares of the centroid configuration satisfies the following conditions

$$0 \leq (1/n)\ \mathrm{tr}\ \mathbf{Z}'\mathbf{C}\mathbf{Z} \leq 1,$$

and therefore can be used as a measure of total fit in GPA.

For the direct approach these relations are even more straightforward. The corresponding loss function

$$f(\mathbf{s},\mathbf{R}) = n - \mathrm{tr}\ (\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)'\mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j) \tag{2.22}$$

may be written as

$$f(\mathbf{s},\mathbf{R}) = n - \mathrm{tr}\ \mathbf{Z}'\mathbf{C}\mathbf{Z},$$

due to (2.18). This shows that the minimization of (2.22) with respect to unknown scaling factors and orthonormal transformation matrices implicitly maximizes the sum of the squared averages of corresponding points. For the direct approach this result is implicit as the averages of corresponding points (i.e., $\mathbf{Z}$) are only calculated *after* convergence of the algorithm (see section 2.4).

Another interesting relation is that, since $(1/n)\ \mathrm{tr}\ \mathbf{Z}'\mathbf{C}\mathbf{Z} = (1/n)\ \mathbf{s}'\mathbf{Y}\mathbf{s} = \phi_1$, where $\mathbf{Y}$ and ϕ_1 are defined in section 2.3.5, GPA in fact maximizes the first eigenvalue of $\mathbf{W}^{-1/2}\mathbf{Y}\mathbf{W}^{-1/2}$ (see formula (2.34)).

Hence, this first eigenvalue, or equivalently the sum of squares of $\mathbf{Z}$, plays much the same role as the squared Pearson product moment correlation coefficient in regression analysis. In contrast with the latter technique, however, until now in GPA no statistical sampling theory has been developed. Thus, as yet no statistical tests exist to determine whether the proportion of the total sum of squares of the data accounted for by the GPA model (i.e., the sum of squares of $\mathbf{Z}$) is significant or not.

We now turn our attention to the relation between the sum of squares of $\mathbf{Z}$ and the other measure of fit that Lingoes and Borg consequently used in their models for the matching of n configurations: the squared correlation between the elements of the optimally transformed $\mathbf{X}_j$'s and $\mathbf{Z}$. We start by giving the relation between these two measures of fit for the general case of incomplete configurations, and then derive the relation between the two measures without missing data as a special case.

Let $r^2(s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j,\mathbf{C}_j\mathbf{Z})$ denote the squared correlation between the elements of $s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j$ and the elements of $\mathbf{C}_j\mathbf{Z}$, then it is easily verified that $r^2(s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j,\mathbf{C}_j\mathbf{Z}) = r^2(\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j,\mathbf{C}_j\mathbf{Z})$. Given an optimal $\mathbf{Z}$ and optimal scaling constants s_j, the general relation between the two measures of fit is given by

$$r^2(\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j,\mathbf{C}_j\mathbf{Z}) = \frac{(\text{tr } \mathbf{Z}'\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)^2}{(\text{tr } \mathbf{Z}'\mathbf{C}_j\mathbf{Z})(\text{tr } \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j)} = \frac{s_j^2\ (\text{tr } \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j)[(1/n)\ \text{tr } \mathbf{Z}'\mathbf{C}\mathbf{Z}]^2}{\text{tr } \mathbf{Z}'\mathbf{C}_j\mathbf{Z}}. \quad (2.48)$$

Proof. Assuming that (2.48) indeed holds, multiplying both terms with $(\text{tr } \mathbf{Z}'\mathbf{C}_j\mathbf{Z})(\text{tr } \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j)$ results in

$$(\text{tr } \mathbf{Z}'\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)^2 = s_j^2\ (\text{tr } \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j)^2\ [(1/n)\ \text{tr } \mathbf{Z}'\mathbf{C}\mathbf{Z}]^2, \quad \text{for } j = 1, ..., n, \quad (2.49)$$

and taking the square root of (2.49) gives

$$\text{tr } \mathbf{Z}'\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j = s_j\ (\text{tr } \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j)\ [(1/n)\ \text{tr } \mathbf{Z}'\mathbf{C}\mathbf{Z}], \quad \text{for } j = 1, ..., n. \quad (2.50)$$

Hence, to prove that (2.48) holds, it is sufficient to prove that (2.50) holds. Letting

$$\mathbf{W}^{-1/2}\mathbf{Y}\mathbf{W}^{-1/2} = \mathbf{P}\mathbf{\Phi}\mathbf{P}' \quad (2.34)$$

be the eigenvalue-eigenvector decomposition of $\mathbf{W}^{-1/2}\mathbf{Y}\mathbf{W}^{-1/2}$, where $\mathbf{Y}$ and $\mathbf{W}$ are defined as in section 2.3.5, it is always true that

$$\mathbf{P}\mathbf{\Phi}\mathbf{P}'\mathbf{p}_1 = \phi_1\mathbf{p}_1, \quad (2.51)$$

where ϕ_1 is the largest eigenvalue, and $\mathbf{p}_1$ is the corresponding first eigenvector. Substitution of (2.34) in the left side of (2.51), and premultiplying the result with $(\sqrt{n})\mathbf{W}^{1/2}$ yields

$$(\sqrt{n})\mathbf{Y}\mathbf{W}^{-1/2}\mathbf{p}_1 = \phi_1(\sqrt{n})\mathbf{W}^{1/2}\mathbf{p}_1 = \phi_1(\sqrt{n})\mathbf{W}\mathbf{W}^{-1/2}\mathbf{p}_1,$$

and therefore

$$\mathbf{Y}\mathbf{s} = \phi_1\mathbf{W}\mathbf{s}, \quad (2.52)$$

due to (2.35) (see section 2.3.5). Since, however, according to (2.36) and (2.41) it is true that $(n\ \phi_1) = \mathbf{s}'\mathbf{Y}\mathbf{s}$ given optimal scaling constants, (2.52) may be written as

$$\mathbf{Ys} = [(1/n)\ \mathbf{s'Ys}]\mathbf{Ws}. \tag{2.53}$$

This completes the proof because, collecting the n terms on the left side of (2.50) in a vector of order (n x 1), and doing the same with the n elements on the right side of (2.50), and substituting (2.19) in the elements of both vectors, yields identity (2.53).

We briefly discuss the special cases of identity (2.48). When there are no missing data identity (2.48) becomes

$$r^2(\mathbf{X}_j\mathbf{R}_j,\mathbf{Z}) = s_j^2\ (\text{tr}\ \mathbf{X}_j'\mathbf{X}_j)\ (\text{tr}\ \mathbf{Z'Z}),$$

and if the configurations have been unit normalized (i.e., tr $\mathbf{X}_j'\mathbf{X}_j = 1$ for every j) the identity simplifies into

$$r^2(\mathbf{X}_j\mathbf{R}_j,\mathbf{Z}) = s_j^2\ \text{tr}\ \mathbf{Z'Z},$$

as Lingoes and Borg noted. Due to (2.48), we also have the following simple identity in the case without missing data

$$r^2(\mathbf{X}_j\mathbf{R}_j,\mathbf{Z}) = s_j\ \text{tr}\ \mathbf{Z'X}_j\mathbf{R}_j. \tag{2.54}$$

If the total fit is defined as the mean of the configuration fits, then the relations between the two measures for the total fit are as follows. In the case that the configurations contain no missing data the total fit equals

$$(1/n) \sum_{j=1}^{n} r^2(\mathbf{X}_j\mathbf{R}_j,\mathbf{Z}) = (1/n) \sum_{j=1}^{n} s_j^2\ (\text{tr}\ \mathbf{X}_j'\mathbf{X}_j)\ (\text{tr}\ \mathbf{Z'Z})$$

$$= (1/n)\ (\text{tr}\ \mathbf{Z'Z}) \sum_{j=1}^{n} s_j^2\ \text{tr}\ \mathbf{X}_j'\mathbf{X}_j = \text{tr}\ \mathbf{Z'Z}.$$

Without missing data, therefore, the two measures of fit are identical, the mean of the squared correlations then being equal to the sum of squares of $\mathbf{Z}$.

If there are missing data, however, the total fit is

$$(1/n) \sum_{j=1}^{n} r^2(\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j,\mathbf{C}_j\mathbf{Z}) = (1/n) \sum_{j=1}^{n} \frac{s_j^2\ (\text{tr}\ \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j)[(1/n)\ \text{tr}\ \mathbf{Z'CZ}]^2}{\text{tr}\ \mathbf{Z'C}_j\mathbf{Z}}$$

$$= (1/n)[(1/n)\ \mathrm{tr}\ \mathbf{Z}'\mathbf{C}\mathbf{Z}]^2 \sum_{j=1}^{n} \frac{s_j^2\ (\mathrm{tr}\ \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j)}{\mathrm{tr}\ \mathbf{Z}'\mathbf{C}_j\mathbf{Z}},$$

and the latter expression is not necessarily equal to (1/n) tr $\mathbf{Z}'\mathbf{C}\mathbf{Z}$. Hence, if the configurations contain missing data the mean of the squared correlations is, in general, no longer identical to the 'size' of the centroid configuration.

Thus, with missing data a choice must be made. For several reasons, we have chosen to use the sum of squares of $\mathbf{Z}$ as a measure of total fit. The first reason for our choice is that this measure is nicely embedded in the GPA model, as expression (2.47) demonstrates, while the squared correlation is not. In the second place, if the configurations contain missing data (tr $\mathbf{Z}'\mathbf{C}\mathbf{Z}$) is completely equivalent to the MAXBET match criterion of Van de Geer (1984), as the next section will show. This equivalence does not hold for the mean of the squared correlations. Last but not least, this measure of fit enables us to generalize Gower's analysis of variance framework for GPA to the case of missing data. Again, this is not possible if squared correlations are used.

In the last part of this section we propose a further decomposition of the sums of squares in GPA. To this end the sum of squares of $\mathbf{Z}$ and the residual sum of squares are partitioned in three separate ways, allowing one to assess: a) the contribution of each individual stimulus to the total fit of the solution, and b) the relative importance of each individual configuration to the proportion of the sum of squares explained by the GPA model, and c) the relative contribution of each dimension to the total GPA solution. These decompositions are a generalization of Gower's analysis of variance. Because the expression 'analysis of variance' suggests the possibility of performing statistical tests, however, we prefer the more neutral expression 'analysis of variation'.

The first decomposition of the total sum of squares with respect to each stimulus separately is given in Table 2.3. Whether we are dealing with complete or incomplete configurations, it can be proved that a_{ii} and b_{ii} in the table are independent components of d_{ii} for each stimulus i. Since the diagonal of the (p x p) matrix $\sum \mathbf{M}_j\mathbf{Z}(\mathbf{M}_j\mathbf{E}_j)'$, where $\mathbf{E}_j = s_j\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j$, consists of the covariations between the p elements a_{ii} and b_{ii}, this diagonal contains zeroes only. The decomposition in

Table 2.3 is based on the following, easily verified, relation between n points $\mathbf{x}_j$ and their centroid $\mathbf{z} = (1/n) \sum \mathbf{x}_j$:

$$\sum_{j=1}^{n} \mathbf{x}_j'\mathbf{x}_j = n\, \mathbf{z}'\mathbf{z} + \sum_{j=1}^{n} (\mathbf{x}_j - \mathbf{z})'(\mathbf{x}_j - \mathbf{z}). \tag{2.55}$$

In words, (2.55) expresses that the sum of squared distances between n points $\mathbf{x}_j$ and the origin always equals n times the squared distance between their centroid and the origin *plus* the sum of squared distances between the n points and their centroid. Notice that we use the GPA model

$$s_j\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j = \mathbf{M}_j\mathbf{Z} + \mathbf{M}_j\mathbf{E}_j, \quad \text{for } j = 1, \ldots, n, \tag{2.56}$$

instead of (2.45) in the decomposition in Table 2.3. The reason is that (2.55) can not be applied in (2.45) because, when dealing with incomplete configurations, the stimulus points of the centroid configuration $\mathbf{Z}$ in (2.45) are idiosyncratically translated for each j (cf. section 2.3.1). In (2.56), on the other hand, $\mathbf{Z}$ remains at the fixed centroid of the individual configurations. The use of (2.56) allows for a

Table 2.3 Contribution of individual stimuli to GPA solution including missing data.

stimulus i	SS_{fit}	$SS_{residual}$	SS_{total}
1	a_{11}*	b_{11}**	d_{11}***
2	a_{22}*	b_{22}**	d_{22}***
⋮	⋮	⋮	⋮
p	a_{pp}*	b_{pp}**	d_{pp}***
$\sum_{i=1}^{p}$	n tr $\mathbf{Z}'\mathbf{Z}$°	$\sum_{j=1}^{n}$ tr $\mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j$	n°

* a_{ii} is diagonal element ii of $\sum_{j=1}^{n} \mathbf{M}_j\mathbf{Z}(\mathbf{M}_j\mathbf{Z})'$

** b_{ii} is diagonal element ii of $\sum_{j=1}^{n} (\mathbf{C}_j\mathbf{E}_j)(\mathbf{C}_j\mathbf{E}_j)'$

*** d_{ii} is diagonal element ii of $\sum_{j=1}^{n} [s_j\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j][s_j\mathbf{M}_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j]'$

° *if and only if* all configurations are complete

decomposition in independent components both for complete and incomplete configurations, but at a cost: SS_{fit} and SS_{total} in Table 2.3 do not add up to the sum of squares of $\mathbf{Z}$ and to n, respectively, in the case of missing data.

If we attempt to identify the relative contribution of each separate configuration to the total sum of squares in GPA, more serious problems arise. Even without missing data, Gower's decomposition scheme for each configuration sometimes yields a residual sum of squares for a configuration that is larger than its total sum of squares. Therefore, in Table 2.4 a slightly different decomposition is presented which at least can not lead to such a situation. This scheme has the additional advantage that, if we are dealing with complete configurations, the sum of squares accounted for by a configuration is identical to the squared correlation between the elements of the optimally transformed $\mathbf{X}_j$ and $\mathbf{Z}$. But we emphasize that, whether we have complete or incomplete configurations, the components of the decomposition are not independent. In the absence of missing data there is no problem, since, according to (2.54), (s_j tr $\mathbf{Z}'\mathbf{X}_j\mathbf{R}_j$) is then equal to the squared correlation, allowing one to directly identify the relative contribution of a configuration to the total solution. With missing data, however, one must keep in mind that the covariation terms tr $\mathbf{Z}'\mathbf{C}_j\mathbf{E}_j$ are unequal to zero. Therefore, we substract these covariation terms from the total sum of squares associated with each configuration in Table 2.4. The reason that we substract them from the total sum of squares is as follows. If we would add the covariation terms to (s_j tr $\mathbf{Z}'\mathbf{X}_j\mathbf{R}_j$), the property would be destroyed that SS_{fit} is equal to the squared correlation without missing data. If we would add them to tr $\mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j$, on the other

Table 2.4 Contribution of individual configurations to GPA solution including missing data.

configuration j	SS_{fit}	$SS_{residual}$	SS_{total}
1	$s_1$tr $\mathbf{Z}'\mathbf{C}_1\mathbf{X}_1\mathbf{R}_1$	tr $\mathbf{E}_1'\mathbf{C}_1\mathbf{E}_1$	$\{s_1^2$ tr $\mathbf{X}_1'\mathbf{C}_1\mathbf{X}_1$ – tr $\mathbf{Z}'\mathbf{C}_1\mathbf{E}_1\}$
2	$s_2$tr $\mathbf{Z}'\mathbf{C}_2\mathbf{X}_2\mathbf{R}_2$	tr $\mathbf{E}_2'\mathbf{C}_2\mathbf{E}_2$	$\{s_2^2$ tr $\mathbf{X}_2'\mathbf{C}_2\mathbf{X}_2$ – tr $\mathbf{Z}'\mathbf{C}_2\mathbf{E}_2\}$
⋮	⋮	⋮	⋮
n	$s_n$tr $\mathbf{Z}'\mathbf{C}_n\mathbf{X}_n\mathbf{R}_n$	tr $\mathbf{E}_n'\mathbf{C}_n\mathbf{E}_n$	$\{s_n^2$ tr $\mathbf{X}_n'\mathbf{C}_n\mathbf{X}_n$ – tr $\mathbf{Z}'\mathbf{C}_n\mathbf{E}_n\}$
$\sum_{j=1}^{n}$	tr $\mathbf{Z}'\mathbf{C}\mathbf{Z}$	$\sum_{j=1}^{n}$ tr $\mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j$	n

Table 2.5 Contribution of individual dimensions to GPA solution including missing data.

dimension k	SS_{fit}	$SS_{residual}$	SS_{total}
1	e_{11}*	f_{11}**	g_{11}***
2	e_{22}*	f_{22}**	g_{22}***
⋮	⋮	⋮	⋮
m	e_{mm}*	f_{mm}**	g_{mm}***
$\sum_{k=1}^{m}$	tr **Z'CZ**	$\sum_{j=1}^{n}$ tr $\mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j$	n

* e_{kk} is diagonal element kk of $\mathbf{\Lambda}$

** f_{kk} is diagonal element kk of $\sum_{j=1}^{n} (s_j\mathbf{X}_j\mathbf{R}_j\mathbf{K} - \mathbf{ZK})'\mathbf{C}_j(s_j\mathbf{X}_j\mathbf{R}_j\mathbf{K} - \mathbf{ZK})$

*** g_{kk} is diagonal element kk of $\sum_{j=1}^{n} (s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j\mathbf{K})'(s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j\mathbf{K})$

hand, the contribution of each configuration to the total sum of squares would always be equal to (1/n) tr **Z'CZ**, which follows directly from (2.46).

The third and last way in which the total sums of squares can be partitioned makes it possible to identify the relative contribution of each dimension to the total GPA solution, and is given in Table 2.5. The matrices **K** and $\mathbf{\Lambda}$ in the table are the eigenvectors and eigenvalues of $\mathbf{Z'CZ} = \mathbf{K\Lambda K'}$, which are determined in order to rotate the whole GPA solution to the principal components of **Z** (see also section 2.4). Because the whole solution is rotated to the principal components of **Z**, the proportion of sum of squares accounted for by the first dimension will always be the largest, the proportion accounted for by the second dimension will always be second largest, etc. The sums of squares in Table 2.5 can be determined independently both with and without missing data.

The MATCHALS program computes and prints all three decompositions discussed in this section.

2.6 GPA with missing data and MAXNEAR, MAXBET, and MAXDIFF

In this section the relations between the GPA match criterion and Van de Geer's (1984) match criteria MAXNEAR, MAXBET and MAXDIFF are briefly discussed. To be able to express the GPA match criterion in Van de Geer's notation we first define two new matrices. Let supermatrix $\mathbf{X}$ of order (p x nm) contain the n matrices $s_jC_jX_j$, that is let

$$\mathbf{X} = \left[\; s_1C_1X_1 \;\vdots\; s_2C_2X_2 \;\vdots\; \ldots \;\vdots\; s_nC_nX_n \; \right] .$$

Also define matrix $\mathbf{D}$ of order (nm x nm) as the block-diagonal matrix of $\mathbf{X'X}$, that is: diagonal blocks of $\mathbf{D}$ are identical to the diagonal blocks $\{(s_jC_jX_j)'s_jC_jX_j\}$ of $\mathbf{X'X}$, while off-diagonal blocks are zero matrices.

Van de Geer (1984, p. 81) defines the MAXNEAR match criterion for the linear relation among n sets of variables as

> The best MAXNEAR solution requires that projections of corresponding points are as much as possible close together: the sum of the squared distances between projections of corresponding points must be minimized.

In his notation the function that must be minimized according to the MAXNEAR match criterion is

$$f(\mathbf{s},\mathbf{R}) = \text{tr } \mathbf{R}'(\mathbf{D} - \mathbf{X}'\mathbf{C}^{-}\mathbf{X})\mathbf{R} = \text{tr } \mathbf{R}'\mathbf{D}\mathbf{R} - \text{tr } \mathbf{R}'\mathbf{X}'\mathbf{C}^{-}\mathbf{X}\mathbf{R}. \qquad (2.57)$$

The relation between Van de Geer's MAXNEAR criterion (2.57) and the match criterion in GPA in our notation is

$$\begin{aligned} f(\mathbf{s},\mathbf{R}) &= \text{tr } \mathbf{R}'\mathbf{D}\mathbf{R} - \text{tr } \mathbf{R}'\mathbf{X}'\mathbf{C}^{-}\mathbf{X}\mathbf{R} \\ &= \sum_{j=1}^{n} s_j^2 \text{ tr } \mathbf{R}_j'\mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j - \text{tr } (\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)'\mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j) \\ &= n - \text{tr } (\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)'\mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j), \end{aligned}$$

and the match criterion in GPA therefore is completely identical to MAXNEAR.

The second match criterion discussed by Van de Geer is MAXBET, which he defines as (p. 81):

The best solution maximizes the sum of the squared projections of averages of corresponding points.

Because the matrix **Z** contains these 'averages of corresponding points', this is exactly what happens in GPA, and as MAXBET in Van de Geer's notation is

$$h(\mathbf{s},\mathbf{R}) = \text{tr } \mathbf{R}'\mathbf{X}'\mathbf{C}^{-}\mathbf{X}\mathbf{R},$$

the relation with the match criterion in GPA in our notation is:

$$\begin{aligned} h(\mathbf{s},\mathbf{R}) &= \text{tr } \mathbf{R}'\mathbf{X}'\mathbf{C}^{-}\mathbf{X}\mathbf{R} \\ &= \text{tr } (\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)'\mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j) = \text{tr } \mathbf{Z}'\mathbf{C}\mathbf{Z}. \end{aligned} \tag{2.58}$$

Hence, in GPA the minimization of the MAXNEAR match criterion at the same time results in the maximization of the MAXBET match criterion.

The relation between Van de Geer's MAXDIFF match criterion, finally, and the same criterion in our notation is

$$\begin{aligned} v(\mathbf{s},\mathbf{R}) &= \text{tr } \mathbf{R}'(n\,\mathbf{X}'\mathbf{C}^{-}\mathbf{X} - \mathbf{D})\mathbf{R} = n \text{ tr } \mathbf{R}'\mathbf{X}'\mathbf{C}^{-}\mathbf{X}\mathbf{R} - \text{tr } \mathbf{R}'\mathbf{D}\mathbf{R} \\ &= n \text{ tr } (\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j)'\mathbf{C}^{-}(\sum_{j=1}^{n} s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j) - \sum_{j=1}^{n} s_j^2 \text{ tr } \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j \\ &= n \text{ tr } \mathbf{Z}'\mathbf{C}\mathbf{Z} - n, \end{aligned} \tag{2.59}$$

which must be maximized for unknown **s** and **R**. But since the maximization of (2.59) is equivalent to the maximization of (2.58), in generalized Procrustes analysis MAXBET and MAXDIFF are equivalent match criteria.

If the configurations have unequal numbers of columns, in GPA the three match criteria MAXNEAR, MAXBET and MAXDIFF no longer coincide (cf., Ten Berge, 1988). As this section demonstrates, on the other hand, the three criteria of Van de Geer remain equivalent if the configurations have unequal numbers of rows.

2.7 Illustrations

In this section two examples are presented of the generalized Procrustes analysis of configurations containing missing data. The first example consists of a constructed data set, while the data in the second example are empirical. In both examples the convergence criterion in the algorithm presented in section 2.4 was set on 1E-7. The algorithm itself has been programmed in APL.

2.7.1 Regular solids

We constructed a data set consisting of three of the five completely regular three-dimensional solids. These five solids (also called the 'Platonic solids' as Plato, the famous Greek philosopher, used them to explain the structure of the universe) are: the tetrahedron or four-face, the hexahedron or six-face or cube, the octahedron or eight-face, the dodecahedron or twelve-face and the icosahedron or twenty-face. The coordinates of the vertices of the tetrahedron, the hexahedron and the dodecahedron were used as input to the GPA algorithm described in section 2.4.

We chose to fit these particular three of the five solids with GPA because the tetra-, the hexa- and the dodecahedron perfectly fit into one another. That is, by connecting eight vertices of the dodecahedron a cube is created, and the connection of four vertices of either the dodecahedron or the cube yields a tetrahedron. We were interested in finding out whether our algorithm would converge to this perfect solution, even though we placed the three regular solids in different positions in space and scaled them differently. A second reason why these data are interesting to analyse is that they contain quite a number of missing data: the tetrahedron has only four vertices and the cube only eight vertices, while the dodecahedron is made up of twenty vertices.

The coordinates of the vertices of the three solids are given in Table 2.6, and a picture of the relative size and position in space of the solids before iterations is given in Figure 2.5, where the horizontal axis represents the first dimension, the vertical axis the third dimension and the remaining axis is the second dimension.

The optimal translation vectors, scaling constants, orthonormal transformation matrices and centroid configuration **Z** after generalized Procrustes analysis of the three

Table 2.6 Coordinates of vertices of tetrahedron ($\mathbf{X}_1$), hexahedron ($\mathbf{X}_2$) and dodecahedron ($\mathbf{X}_3$) before iterations.

$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$
missing	1.0 0.0 -0.5	2.2705 -2.5000 4.9098
missing	missing	1.8197 -1.3197 4.1803
-1.5000 -2.0000 2.4330	1.0 1.0 -0.5	3.0000 -0.5902 3.7295
missing	missing	4.1803 -1.3197 4.1803
missing	missing	3.7295 -2.5000 4.9098
missing	missing	1.0902 -1.7705 3.0000
missing	1.0 1.0 -1.5	1.8197 -1.3197 1.8197
missing	missing	3.0000 -0.5902 2.2705
missing	missing	1.8197 -3.6803 4.1803
missing	missing	3.0000 -4.4098 3.7295
-2.0774 -3.0000 0.8000	0.0 0.0 -0.5	4.1803 -3.6803 4.1803
missing	missing	4.9098 -3.2295 3.0000
missing	0.0 1.0 -0.5	4.9098 -1.7705 3.0000
missing	missing	4.1803 -1.3197 1.8197
-2.0774 -1.0000 0.8000	0.0 1.0 -1.5	3.7295 -2.5000 1.0902
missing	missing	4.1803 -3.6803 1.8197
-0.3453 -2.0000 0.8000	1.0 0.0 -1.5	1.0902 -3.2295 3.0000
missing	missing	2.2705 -2.5000 1.0902
missing	0.0 0.0 -1.5	3.0000 -4.4098 2.2705
missing	missing	1.8197 -3.6803 1.8197

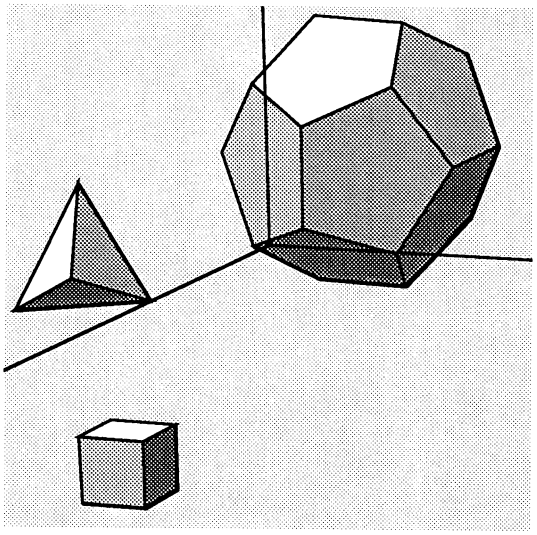


Figure 2.5 Relative size and position in space of the three configurations given in Table 2.6.

Table 2.7 Optimal translation vectors, scaling constants, rotation matrices and centroid configuration **Z** after GPA of data in Table 2.6.

$\mathbf{u}'_1$	$\mathbf{u}'_2$	$\mathbf{u}'_3$
[-.2657 -.3543 .2140]	[.0886 .0886 -.1772]	[.5315 -.4429 .5315]

s_1	s_2	s_3
1.4112	1.9957	0.8454

$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
-.9928 -.0422 .1117	-.8068 .4935 .3248	.9999 -.0125 .0119
-.1192 .4025 -.9076	.3248 .8298 -.4537	.0119 .9986 .0518
.0066 .9144 .4047	.4935 .2606 .8298	-.0125 -.0516 .9986

$$\mathbf{Z} = \begin{bmatrix} -0.1128 & -0.0134 & 0.2843 \\ -0.1769 & 0.1696 & 0.1836 \\ 0.0020 & 0.2800 & 0.1239 \\ 0.1766 & 0.1652 & 0.1878 \\ 0.1057 & -0.0161 & 0.2869 \\ -0.2847 & 0.1127 & 0.0023 \\ -0.1724 & 0.1879 & -0.1695 \\ 0.0048 & 0.2913 & -0.0943 \\ -0.1811 & -0.1834 & 0.1653 \\ -0.0048 & -0.2913 & 0.0943 \\ 0.1724 & -0.1879 & 0.1695 \\ 0.2847 & -0.1127 & -0.0023 \\ 0.2873 & 0.1055 & 0.0091 \\ 0.1811 & 0.1834 & -0.1653 \\ 0.1128 & 0.0134 & -0.2843 \\ 0.1769 & -0.1696 & -0.1836 \\ -0.2873 & -0.1055 & -0.0091 \\ -0.1057 & 0.0161 & -0.2869 \\ -0.0020 & -0.2800 & -0.1239 \\ -0.1766 & -0.1652 & -0.1878 \end{bmatrix}$$

Table 2.8 History of iterations expressed in total loss for data in Table 2.6.

Iteration number	Rotation step	Scaling step
1	0.1795683482	0.0015849889
2	0.0000273636	0.0000261134
3	0.0000002719	0.0000002715
4	0.0000000029	0.0000000029
5	0.0000000001	0.0000000001

Table 2.9 Analysis of variation of GPA solution for data in Table 2.6.

stimulus	SS_{fit}	$SS_{residual}$	SS_{total}
1	0.1875	0.0000	0.1875
2	0.0937	0.0000	0.0937
3	0.2813	0.0000	0.2813
4	0.0937	0.0000	0.0937
5	0.0938	0.0000	0.0938
6	0.0938	0.0000	0.0938
7	0.1875	0.0000	0.1875
8	0.0938	0.0000	0.0938
9	0.0937	0.0000	0.0937
10	0.0938	0.0000	0.0938
11	0.2812	0.0000	0.2812
12	0.0938	0.0000	0.0938
13	0.1875	0.0000	0.1875
14	0.0937	0.0000	0.0937
15	0.2813	0.0000	0.2813
16	0.0937	0.0000	0.0937
17	0.2813	0.0000	0.2813
18	0.0938	0.0000	0.0938
19	0.1875	0.0000	0.1875
20	0.0937	0.0000	0.0937
Σ	3.0000	0.0000	3.0000
configuration			
1	0.3750	0.0000	0.3750
2	0.7500	0.0000	0.7500
3	1.8750	0.0000	1.8750
Σ	3.0000	0.0000	3.0000
dimension			
1	1.0000	0.0000	1.0000
2	1.0000	0.0000	1.0000
3	1.0000	0.0000	1.0000
Σ	3.0000	0.0000	3.0000

configurations are given in Table 2.7. The history of iterations of the algorithm for this example is shown in Table 2.8. As expected, the algorithm converges to a perfect solution. The reason that it does not converge to zero *exactly* is that the coordinates of the vertices given in Table 2.6 have been rounded to four digits. In this example we

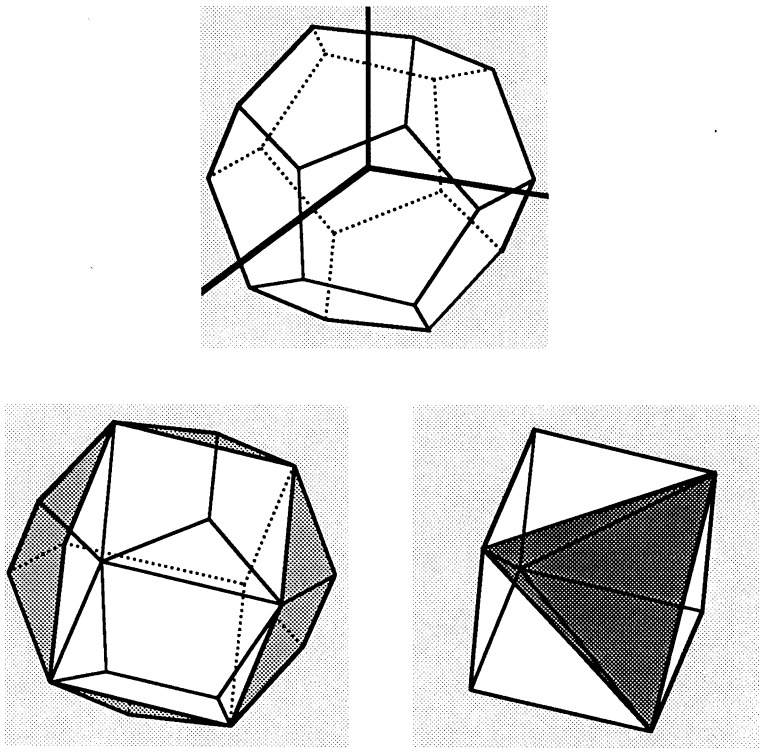


Figure 2.6 Optimally transformed configurations $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$ from Table 2.6 after convergence of the GPA algorithm.

have *not* rotated **Z** to its principal components, for the simple reason that the sum of the three regular solids is completely symmetric. This implies that an eigenvalue-eigenvector decomposition of **Z'CZ** yields three completely identical eigenvalues and thus that all dimensions are equally important.

In Table 2.9 the complete decomposition of the total sum of squares of the three configurations is given with respect to individual stimuli, configurations and dimensions (see section 2.5). Because the solution is perfect, the total residual sum of squares is equal to zero. Therefore, the three-way decomposition of this total residual sum of squares with respect to individual stimuli, configurations and dimensions must also result in zero components, as Table 2.9 points out. The table shows the equal

importance of the three dimensions for the solution. In section 2.5 we discussed that, in the case of missing data, the fitted sum of squares for the stimuli do not add up to the sum of squares of **Z**, and that the total sum of squares of the stimuli do not add up to n. Notice, however, that the fitted sum of squares for the stimuli in this example add up to (n tr **Z**'**Z**), and that the total sum of squares of the stimuli add up to n, notwithstanding the fact that there are two incomplete configurations. This shows that a perfect GPA solution is the one exception to this rule.

The geometry of the solution is illustrated in Figure 2.6. The location of the optimally transformed dodecahedron (i.e., $\mathbf{X}_3$) with respect to the three coordinate axes is shown at the top of the figure. This at the same time demonstrates where the centroid configuration **Z** is located with respect to the coordinate axes since **Z** obviously always is completely identical to the configuration(s) that do(es) not contain any missing data if the solution is perfect. In the bottom left corner of Figure 2.6 is illustrated how the vertices of the optimally transformed hexahedron or cube perfectly fit in the vertices of **Z** (or, which is equivalent in this case, in the vertices of $\mathbf{X}_3$). In the bottom right corner of the figure, finally, is illustrated how the vertices of the optimally transformed tetrahedron again coincide with those of the cube, and therefore with those of the dodecahedron.

In this example the algorithm presented in section 2.4 did exactly what we expected it to do: converge to the global minimum of zero loss. The example discussed here, however, is highly artificial. In the next and last section of this chapter on GPA of configurations containing missing data, a more realistic empirical data set is analyzed.

2.7.2 Personality traits

In this section we analyse a data set consisting of the results of five consecutive studies performed by Van der Kloot (1978) in the years 1969 through 1975. The main objective of these studies was to investigate the structure of the so-called implicit theory of personality, that is, to find out whether there is an underlying common structure in the way people perceive a number of personality traits.

Three of the five studies yielded similarity matrices that could be analyzed with multidimensional scaling techniques. The other two studies resulted in data sets that were analyzed by means of canonical discriminant analysis. All together these analyses again yielded five configurations. What is interesting about these five configurations in the context of this chapter is that different numbers of personality traits were used in the five studies. Our algorithm therefore seems especially appropriate to answer the question whether the five configurations basically have the same structure. In Table 2.10 the coordinates of the five (all of them two-dimensional) configurations are presented.

Table 2.10 Coordinates in two dimensions of five configurations of differing numbers of personality traits.

stimulus	$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	$\mathbf{X}_4$	$\mathbf{X}_5$
aggressive	-.17 .59	-.74 .32	-.39 .63	-.38 .60	-.22 .62
cooperative	.32 -.07	.75 -.07	.79 -.13	.75 -.06	missing
dominant	.22 .43	-.38 .71	.12 .78	.05 .82	missing
intelligent	missing	.67 .22	missing	missing	missing
passive	-.37 -.30	missing	-.20 -.72	-.02 -.91	-.07 -.90
pessimistic	-.32 .00	-.38 -.65	-.32 -.24	-.26 .12	-.04 -.17
submissive	-.03 -.56	-.19 -.84	.10 -.74	.04 -.70	-.19 -.73
unreliable	missing	missing	missing	missing	-.95 -.10

The results of GPA of this data set are given in the following tables. Table 2.11 contains the history of iterations of the GPA algorithm. In Table 2.12 the optimal translation vectors, scaling constants, and orthonormal transformation matrices are given, as well as the coordinates of the centroid configuration rotated to its principal components. Matrix **K** in Table 2.12 is the matrix of eigenvectors of $\mathbf{Z}'\mathbf{C}\mathbf{Z}$, and is used to rotate the five optimally transformed $\mathbf{X}_j$'s and **Z** to the principal components of **Z**. Table 2.13 contains the three-way decomposition of the total sum of squares of

Table 2.11 History of iterations of GPA of data in Table 2.10.

Iteration number	TOTAL LOSS	
	Rotation step	Scaling step
1	0.2749227788	0.1633848176
2	0.1612889220	0.1612885030
3	0.1612805281	0.1612805278
4	0.1612802468	0.1612802467
5	0.1612802426	0.1612802426

Table 2.12 Optimal scaling constants, orthonormal transformation matrices and centroid configuration **Z** after GPA of data in Table 2.10.

s_1	s_2	s_3	s_4	s_5
1.4336	0.8612	0.9772	0.9499	0.9930

$\mathbf{u}_1'$	$\mathbf{u}_2'$	$\mathbf{u}_3'$
[.0404 .0099]	[-.0070 -.1007]	[.1167 -.0085]
$\mathbf{u}_4'$	$\mathbf{u}_5'$	
[.1262 .0273]	[-.3425 -.0960]	

$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
$\begin{bmatrix} .9526 & .3043 \\ -.3043 & .9526 \end{bmatrix}$	$\begin{bmatrix} .9683 & -.2497 \\ .2497 & .9683 \end{bmatrix}$	$\begin{bmatrix} 1.0000 & .0039 \\ -.0039 & 1.0000 \end{bmatrix}$
$\mathbf{R}_4$	$\mathbf{R}_5$	$\mathbf{K}$
$\begin{bmatrix} .9991 & -.0423 \\ .0423 & .9991 \end{bmatrix}$	$\begin{bmatrix} -.9623 & -.2719 \\ -.2719 & .9623 \end{bmatrix}$	$\begin{bmatrix} -.0810 & .9967 \\ .9967 & .0810 \end{bmatrix}$

$$\mathbf{ZK} = \begin{bmatrix} .4039 & -.3018 \\ -.0765 & .3245 \\ .4575 & -.0055 \\ .0739 & .4081 \\ -.4516 & -.2453 \\ -.0749 & -.2973 \\ -.4124 & -.1125 \\ .0801 & .2298 \end{bmatrix}$$

Table 2.13 Analysis of variation of GPA solution for data of Table 2.10.

stimulus	SS_{fit}	$SS_{residual}$	SS_{total}
aggressive	1.2712	0.0062	1.2773
cooperative	0.4445	0.0183	0.4627
dominant	0.8372	0.0121	0.8493
intelligent	0.1720	0.0000	0.1720
passive	1.0563	0.0296	1.0859
pessimistic	0.4699	0.0392	0.5091
submissive	0.9138	0.0560	0.9698
unreliable	0.0592	0.0000	0.0592
Σ	5.2242	0.1613	5.3855
configuration			
1	1.0108	0.0666	1.0774
2	0.9920	0.0415	1.0334
3	1.0719	0.0075	1.0794
4	1.0400	0.0384	1.0784
5	0.7240	0.0073	0.7313
Σ	4.8387	0.1613	5.0000
dimension			
1	3.3061	0.0957	3.4017
2	1.5327	0.0656	1.5983
Σ	4.8387	0.1613	5.0000

the five configurations with respect to individual stimuli, configurations and dimensions.

As can be deduced from Table 2.13, the total fit of the GPA solution for this data set is equal to 4.8387/5.0000 = 0.9677, meaning that the solution accounts for 97% of the sum of squares of the individual configurations. Thus the GPA model fits quite well, suggesting, at least as far as this data set is concerned, that there indeed is an underlying common structure in the way people perceive the eight personality traits investigated in the five studies of Van der Kloot.

The decomposition of the total sum of squares with respect to the individual stimuli shows that the stimuli 'aggressive', 'cooperative', 'dominant', 'intelligent', 'passive',

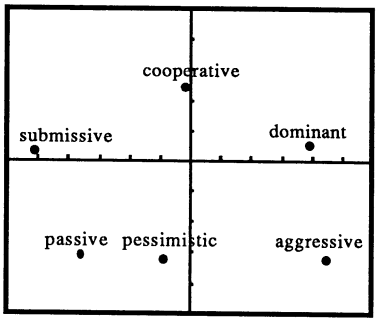


Figure 2.7a $s_1M_1(X_1 - 1u_1')R_1K$.

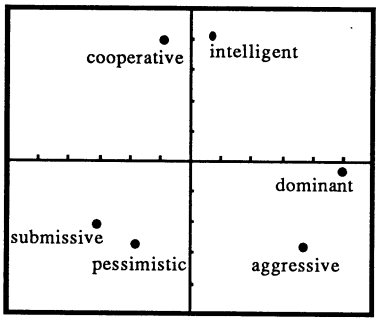


Figure 2.7b $s_2M_2(X_2 - 1u_2')R_2K$.

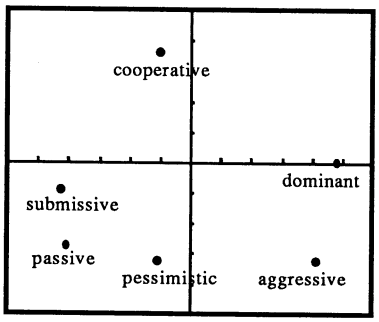


Figure 2.7c $s_3M_3(X_3 - 1u_3')R_3K$.

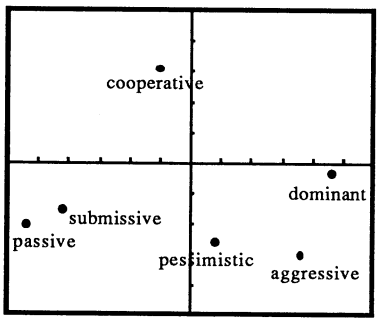


Figure 2.7d $s_4M_4(X_4 - 1u_4')R_4K$.

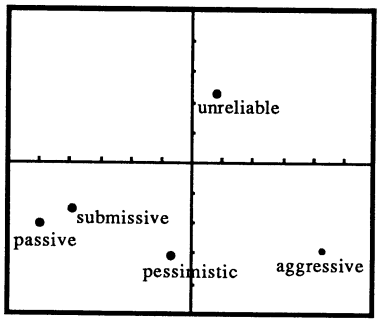


Figure 2.7e $s_5M_5(X_5 - 1u_5')R_5K$.

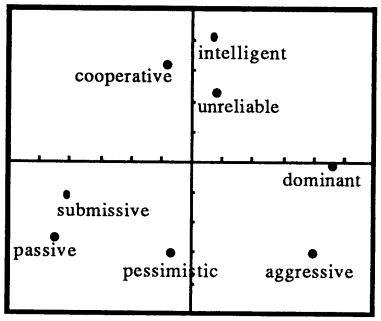


Figure 2.7f ZK.

'pessimistic', 'submissive' and 'unreliable' are accounted for 99.5%, 96.1%, 98.6%, 100%, 97.3%, 92.3%, 94.2% and 100%, respectively. Thus, stimulus 'pessimistic' has the lowest fit, while the stimuli 'intelligent' and 'unreliable' are perfectly represented in the solution. That the fit for the latter two stimuli is perfect is hardly surprising: they both occur only once in the five configurations (see Table 2.10). Therefore no compromises have to be made, and the coordinates of the centroids for these two stimulus points are equal to the coordinates of the stimulus points themselves. Notice that the fitted sums of squares for the eight stimuli do not add up to (tr $\mathbf{Z}'\mathbf{C}\mathbf{Z}$), and that the sum of the total sums of squares is unequal to five, due to the presence of missing data.

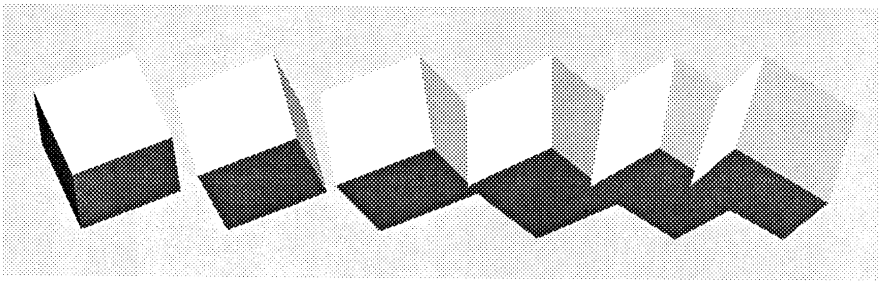
The second decomposition in Table 2.13 demonstrates that the five configurations are accounted for 94%, 96%, 99%, 96% and 99%, respectively. The analysis of variation with respect to the two dimensions shows that the first dimension is accounted for 97%, while the second dimension is accounted for 96%. The two dimensions are therefore almost equally important in the GPA solution.

In Figure 2.7 the geometry of the optimally transformed configurations (rotated to the principal components of $\mathbf{Z}$) is shown together with the optimal centroid configuration $\mathbf{Z}$ for this data set (rotated to its principal components). The points in the last plot of Figure 2.7 are the averages of the corresponding points in the first five plots. The two dimensions of all six plots in Figure 2.7 can be interpreted as representing a submission versus dominance dimension (the first dimension of the plots) and an evaluation dimension (the second dimension of the plots). This result is not new: the dimensions found in studies on the implicit personality theory can usually be interpreted in a similar fashion.

The location of the stimulus point for 'unreliable' in the centroid $\mathbf{Z}$ is strange (see Figure 2.7f). One would not expect this stimulus to be situated near 'cooperative' and 'intelligent', but rather in the neighboorhood of any one of the remaining stimuli. We already noted that the stimuli 'unreliable' and 'intelligent' both are perfectly represented in the solution because information about these two stimuli is only available in one configuration (about 'intelligent' in configuration two, and about 'unreliable' in configuration five). The reason that stimulus 'intelligent', though being only represented once, falls on a very understandable spot in the centroid configuration $\mathbf{Z}$, while stimulus 'unreliable' does not, may well be the following. Stimulus 'unreliable' is *in its own configuration* (Figure 2.7e) already an outlier and

being perceived by the subjects in the fifth study as completely different from the other four stimuli. About the location of this stimulus point, therefore, nothing more is known than that it is 'somewhere else' compared to the rest of the stimuli in this study. Stimulus 'intelligent', on the other hand, is situated far from 'submissive', 'pessimistic', 'aggressive' and 'dominant', *but close to* 'cooperative' (Figure 2.7b). Thus, the location of stimulus 'intelligent' in configuration two is more reliable than the location of 'unreliable' in configuration five, since not only information is available about what 'intelligent' is not, but also about what it is similar to (i.e., 'cooperative').

Concluding, this suggests that in GPA of configurations containing missing data the location of a stimulus that occurs only once in all n configurations, and whose position is not 'pinned down' locally by closely related stimuli in its own configuration, should not be given too much weight in the interpretation of the centroid configuration **Z** (notwithstanding the fact that such a stimulus is always perfectly represented in the solution).



Chapter 3

The dimension weighting models

3.1 Introduction

In the previous chapter we discussed how to match n configurations when the only admissible transformations are those that leave relative distances intact. We also discussed that the transformations satisfying this restriction are translations, orthonormal transformations, and uniform rescalings of the n configurations. In this chapter we investigate the matching of n configurations using a less restrictive set of rules. Specifically, the condition that the configurations may only be rescaled uniformly is relaxed to the effect that *each of the m dimensions* of each configuration is allowed to be stretched or shrunk *differently*. The latter transformation is called dimension weighting, and is illustrated for one two-dimensional configuration in Figure 3.1, where the stimulus points are numbered from one to five before transformation and from one prime to five prime after transformation. As the figure shows, relative distances are no longer preserved in dimension weighting.

This chapter deals with the matching of n configurations when the admissible transformations are translations, orthonormal transformations, and dimension weighting. Lingoes and Borg (1978) discussed how these three transformations quite naturally result in two different so-called dimension weighting models. As we already noted in chapter 1, the way in which Lingoes and Borg developed these two models has an important drawback: the estimation of some of the unknown transformation parameters is based on results derived for two-dimensional configurations only. In the following sections we will not only generalize the dimension weighting models of Lingoes and Borg to m dimensions, but also to the case of missing data. Moreover, a

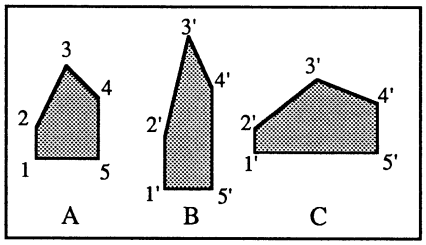


Figure 3.1 Illustration of dimension weighting. A: before weighting, B and C: after differential weighting of the two dimensions of A.

new approach to the fitting of these models will be proposed. While Lingoes and Borg chose to hold on to the centroid configuration **Z** obtained in the GPA model in the development of their dimension weighting models, we propose to determine *new* optimal centroid configurations. This results in a better fit, and yields models that can, at least in principle, be fitted independently from any results obtained in GPA. In the following section we will first explain how Lingoes and Borg set up their models, and then discuss our alternative approach.

3.2 Geometry and algebra

Given that the admissible transformations in the matching of n configurations are translations, orthonormal transformations, and dimension weighting, Lingoes and Borg (1978) proposed to apply these transformations in the following way. Given n configurations $\mathbf{X}_j$ (j = 1, ..., n) of order (p x m), let **Z** of order (p x m) be the matrix of centroids obtained after a generalized Procrustes analysis of these n configurations. Also, let $\mathbf{g}_j$ and $\mathbf{h}_j$ be unknown translation vectors of order (m x 1), $\mathbf{Q}_j$ be an unknown orthonormal matrix of order (m x m), $\mathbf{W}_j$ be an unknown (m x m) diagonal matrix containing dimension weights, and $\mathbf{E}_j$ be a (p x m) matrix of residuals. Using these definitions, Lingoes and Borg proposed the following model:

$$(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j = (\mathbf{Z} - \mathbf{1}\mathbf{h}_j')\mathbf{W}_j + \mathbf{E}_j, \qquad \text{for } j = 1, 2, ..., n, \tag{3.1}$$

where the unknown $\mathbf{g}_j$'s, $\mathbf{h}_j$'s, $\mathbf{Q}_j$'s, and $\mathbf{W}_j$'s are to be estimated by minimizing the sum of squared residuals, that is, by minimizing $\sum \text{tr } \mathbf{E}_j'\mathbf{E}_j$.

Lingoes and Borg discussed that, to obtain optimal results, it is necessary to include still another (admissible) transformation in (3.1). They showed that a rotation of the dimensions of **Z** is essential to get the best out of differential weighting of these same dimensions. A simple geometrical example may illustrate the necessity of an optimal orientation of **Z**. Suppose that the coordinates of the stimuli in **Z** are the vertices of the square shown in Figure 3.2. Also suppose that the four stimuli in one of the individual configurations $\mathbf{X}_j$ form the vertices of the rhombus shown in the same figure. By differential weighting of the two dimensions of the centroid configuration **Z** all kinds of rectangles can be obtained, but never the rhombus representing $\mathbf{X}_j$. If only weights are allowed, therefore, the match between $\mathbf{X}_j$ and **Z**

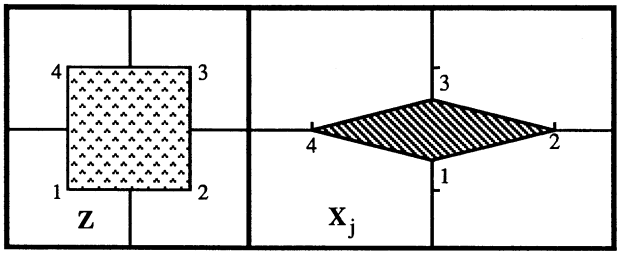


Figure 3.2 Illustration of necessity of rotation of **Z** with respect to dimension weighting.

will never be perfect. But it is immediately apparent how a rotation of **Z** before weighting its dimensions can effectuate a perfect match between $\mathbf{X}_j$ and **Z**. Rotating **Z** counterclockwise through an angle of 45°, and then weighting the first dimension of **Z** with a factor $\sqrt{2}$ and the second dimension with a factor $(1/4)\sqrt{2}$ exactly gives the rhombus representing $\mathbf{X}_j$ in Figure 3.2. In matrix notation this can be expressed as follows:

$$\mathbf{ZSW} = \begin{bmatrix} -1 & -1 \\ 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \cos 45° & \sin 45° \\ -\sin 45° & \cos 45° \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & (1/4)\sqrt{2} \end{bmatrix}$$

$$= \begin{bmatrix} -1 & -1 \\ 1 & -1 \\ 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & (1/4)\sqrt{2} \end{bmatrix} = \begin{bmatrix} 0.0 & -0.5 \\ 2.0 & 0.0 \\ 0.0 & 0.5 \\ -2.0 & 0.0 \end{bmatrix} = \mathbf{X}_j,$$

where **S** is the optimal rotation matrix, and **W** contains the optimal dimension weights to achieve a perfect match between **Z** and $\mathbf{X}_j$. These transformations of **Z** are also illustrated geometrically in Figure 3.3.

Lingoes and Borg next showed that such an orthonormal transformation of **Z**, needed to obtain an optimal match in model (3.1), can be performed in two ways, and that from these two options two different dimension weighting models emerge.

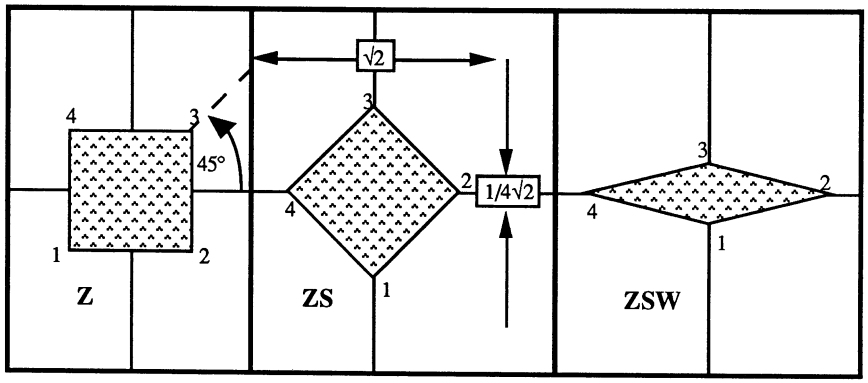


Figure 3.3 Geometry of optimal rotation of **Z** in Figure 3.2 with respect to dimension weighting.

The first way to rotate **Z** is to estimate only one orthonormal transformation of **Z** such that the match is optimized over all n individual configurations simultaneously. The geometry of this model has a rather straightforward psychological interpretation: people (represented by the individual configurations) agree on the underlying dimensions that structure the surrounding world (represented by the optimally rotated centroid or group configuration), but differ in the importance they attach to the dimensions on which the stimuli under investigation are ordered. The algebra corresponding to this model is as follows:

$$(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j = (\mathbf{Z} - \mathbf{1}\mathbf{h}_j')\mathbf{S}\mathbf{W}_j + \mathbf{E}_j \quad \text{for } j = 1, ..., n, \tag{3.2}$$

where **S** is an unknown orthonormal matrix of order (m x m). The matrices $\mathbf{X}_j$, **Z**, $\mathbf{W}_j$, and $\mathbf{E}_j$, and the vectors $\mathbf{g}_j$ and $\mathbf{h}_j$ have been defined earlier in this section.

This model is related to the INDSCAL model proposed by Carroll and Chang (1970). Both model (3.2) and the INDSCAL model are characterized by a group configuration whose dimensions are idiosyncratically being weighted. Therefore, these models have a common psychological interpretation. Here the resemblance stops, however, since Lingoes and Borg's model was designed to match n *configurations* of order (p x m), while Carroll and Chang use the INDSCAL model to analyse n *(dis)similarity matrices* of order (p x p).

The second way to rotate **Z** is to estimate a separate optimal orthonormal transformation for each of the n individual configurations. The psychological

implication of this second dimension weighting model is that individuals not only differ in the importance they attach to the dimensions in the group space, but also differ in which dimensions they choose to weight in this same group configuration. The matrix notation for this model is very similar to (3.2):

$$(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j = (\mathbf{Z} - \mathbf{1}\mathbf{h}_j')\mathbf{S}_j\mathbf{W}_j + \mathbf{E}_j \quad \text{for } j = 1, ..., n, \tag{3.3}$$

the only difference being that the fixed orientation matrix $\mathbf{S}$ has now been replaced by the idiosyncratic orientation matrix $\mathbf{S}_j$. With respect to interpretation, this model is related to the IDIOSCAL model presented by Carroll and Chang in 1972. Again, however, model (3.3) was set up for the analysis of configurations, while Carroll and Chang's IDIOSCAL model was designed for the analysis of (dis)similarity matrices.

What is striking about the way Lingoes and Borg set up these two models is that they chose to 'borrow' the centroid configuration $\mathbf{Z}$ from GPA. The fact that $\mathbf{Z}$ is optimal in GPA, however, does not guarantee that it is optimal in models (3.2) and (3.3). This means that by sticking to $\mathbf{Z}$ a source of error is introduced obscuring the real fit of these models. This is the reason why we propose to replace models (3.2) and (3.3) of Lingoes and Borg by two models that always find the perfect solution if it exists, and eliminate error due to compromising factors resulting from GPA. Instead of using the matrix of centroids $\mathbf{Z}$ obtained in the GPA model, a *new* optimal centroid configuration $\mathbf{Y}$ is estimated.

Taking into account that information may be missing about some stimuli in some configurations, and letting $\tilde{\mathbf{X}}_j$ denote the optimally translated, rotated and rescaled configuration $\mathbf{X}_j$ after GPA rotated to the principal components of $\mathbf{Z}$ (i.e., $\tilde{\mathbf{X}}_j = s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j\mathbf{K}$), we propose the following alternative for model (3.2):

$$\mathbf{M}_j(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j = \mathbf{M}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{W}_j + \mathbf{M}_j\mathbf{E}_j \quad \text{for } j = 1, ..., n. \tag{3.4}$$

In (3.4), $\mathbf{Y}$ is assumed to be an unknown centroid configuration $\mathbf{Y}$ of order (p x m), and $\mathbf{M}_j$ is a given diagonal matrix of order (p x p) with ones on the diagonal if the corresponding rows in $\mathbf{X}_j$ are not missing, and zeroes elsewhere. The reason that we use the optimally transformed configurations $\tilde{\mathbf{X}}_j$ instead of the raw configurations $\mathbf{X}_j$ in model (3.4) is a practical one. We simply expect that the algorithm to be developed in section 3.3 will be more efficient when applied to the $\tilde{\mathbf{X}}_j$'s than to the raw $\mathbf{X}_j$'s,

because the former configurations are already optimal with respect to distance preserving transformations. It may be noted that the orthonormal matrix **S** is lacking in model (3.4). This latter matrix no longer needs to be estimated since the free matrix of centroids **Y** will automatically be in an optimal orientation with respect to dimension weighting. In full we call this alternative model the *dimension weighting model with free optimal centroid configuration*, and will refer to it as the DIMFREE model. The DIMFREE model is completely developed in section 3.3 of this chapter.

Our alternative for model (3.3) can be written as follows:

$$\mathbf{M}_j(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j = \mathbf{M}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{S}_j\mathbf{W}_j + \mathbf{M}_j\mathbf{E}_j \quad \text{for } j = 1, \ldots, n, \tag{3.5}$$

where **Y** is, again, assumed to be an unknown (p x m) centroid configuration. In full we call this model the *dimension weighting model with idiosyncratic optimal orientations of a free optimal centroid configuration.* Henceforth we will refer to this model as the DIMIDIO model. The DIMIDIO model is completely developed in section 3.4 of this chapter.

A characteristic feature of all dimension weighting models discussed so far is that the weighting of dimensions is performed on the centroid or group configuration (whether it be **Y** or **Z**), and not on the individual configurations. In the GPA model, on the other hand, it is typically the individual configurations that are optimally being rescaled. The choice of Lingoes and Borg to weight the centroid configuration instead of the individual configurations was probably based on traditional considerations: in the dimension weighting models INDSCAL and IDIOSCAL of Carroll and Chang (1970, 1972), it is also the dimensions of the group configuration that are being weighted. In the latter models this manner of weighting is born out of necessity, since it does not make sense to weight the dimensions of a (dis)similarity matrix. When analyzing configurations, however, there is, in principle, nothing against using a model like, for instance,

$$\mathbf{M}_j(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j\mathbf{W}_j = \mathbf{M}_j\mathbf{Y} + \mathbf{M}_j\mathbf{E}_j, \quad \text{for } j = 1, 2, \ldots, n, \tag{3.6}$$

where the dimensions of the individual configurations are weighted instead of those of **Y**. In that case we could again apply Theorem 2 (see section 2.2), and show that the minimization of the sum of squared distances between corresponding stimulus points is equivalent to the minimization of the sum of squared distances between

corresponding points and their centroid. But we will not follow this line of reasoning any further, and use the 'traditional' weighting of the dimensions of the centroid configuration.

The remainder of this chapter is organized as follows. In sections 3.3 and 3.4 the dimension weighting models (3.4) and (3.5) are developed and algorithms for the estimation of the corresponding unknown transformation parameters are presented. In section 3.5 measures of fit and an analysis of variation are presented for both models, and in section 3.6 we discuss the results of the analysis of constructed data sets according to the DIMFREE and DIMIDIO model.

3.3 The DIMFREE model

3.3.1 Introduction

To determine the unknown parameters in the DIMFREE model (3.4) the following least squares loss function

$$f(\mathbf{G},\mathbf{H},\mathbf{Q},\mathbf{W},\mathbf{Y}) = \sum_{j=1}^{n} \text{tr } \mathbf{E}_j'\mathbf{M}_j\mathbf{E}_j$$

$$= \sum_{j=1}^{n} \text{tr } [(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j - (\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{W}_j]'\mathbf{M}_j[(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j - (\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{W}_j] \quad (3.7)$$

is defined. The unknown translation vectors $\mathbf{g}_j$ and $\mathbf{h}_j$ are collected in the (m x n) matrices $\mathbf{G}$ and $\mathbf{H}$, respectively, the unknown orthonormal matrices $\mathbf{Q}_j$ in the (nm x m) supermatrix $\mathbf{Q}$, and the unknown dimension weight matrices $\mathbf{W}_j$ in the (nm x m) supermatrix $\mathbf{W}$. In section 3.3.2 we first of all determine the translation vectors $\mathbf{g}_j$ optimizing (3.7), since this makes it possible to simultaneously eliminate $\mathbf{G}$ and $\mathbf{H}$ from (3.7). In sections 3.3.3, 3.3.4 and 3.3.5 the simplified loss function for the DIMFREE model is minimized with respect to the orthonormal matrices $\mathbf{Q}_j$, dimension weights $\mathbf{W}_j$ and new centroid configuration $\mathbf{Y}$, respectively. In section 3.3.6 we briefly go into the possibility of setting up a 'direct approach' for the DIMFREE model by eliminating the unknown matrix $\mathbf{Y}$ from the DIMFREE loss function, and discuss the consequences this has for the optimization of the resulting loss function with respect to the only unknowns left: the orthonormal matrices $\mathbf{Q}_j$, and the dimension weights $\mathbf{W}_j$. In section 3.3.7 we show that the solution for the DIMFREE model is only unique up to a simultaneous weighting of the optimal $\mathbf{Y}$ and $\mathbf{W}_j$'s in (3.7), and discuss a procedure which guarantees a uniquely weighted solution. Finally, in section 3.3.7 an algorithm is presented for the fitting of the DIMFREE model to n configurations.

3.3.2 Translations

In this section we minimize (3.7) with respect to the unrestricted translation vectors **G**. Defining $\mathbf{A}_j = \tilde{\mathbf{X}}_j\mathbf{Q}_j - (\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{W}_j$, and considering only one particular $\mathbf{g}_j$, we may rewrite the loss function as

$$\begin{aligned} f(\mathbf{g}_j) &= \mathbf{1}'\mathbf{M}_j\mathbf{1}\mathbf{g}_j'\mathbf{g}_j - 2\,\mathbf{g}_j'\mathbf{Q}_j\mathbf{A}_j'\mathbf{M}_j\mathbf{1} + d_j \\ &= c_j^2\,\mathbf{g}_j'\mathbf{g}_j - 2\,c_j\mathbf{g}_j'\mathbf{b}_j + d_j, \end{aligned} \tag{3.8}$$

where d_j is a term independent of $\mathbf{g}_j$, $\mathbf{b}_j \equiv \mathbf{Q}_j\mathbf{A}_j'\mathbf{M}_j\mathbf{1}/\sqrt{\mathbf{1}'\mathbf{M}_j\mathbf{1}}$, and $c_j \equiv \sqrt{\mathbf{1}'\mathbf{M}_j\mathbf{1}}$. Applying the same procedure to (3.8) as discussed in section 2.3.1, one finds that the global minimum of (3.8) is attained for

$$\mathbf{g}_j = \frac{[\tilde{\mathbf{X}}_j - (\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{W}_j\mathbf{Q}_j']'\mathbf{M}_j\mathbf{1}}{\mathbf{1}'\mathbf{M}_j\mathbf{1}}. \tag{3.9}$$

Substitution of (3.9) in

$$(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j - (\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{W}_j,$$

which is a part of (3.7), gives

$$(\tilde{\mathbf{X}}_j - \frac{\mathbf{1}\mathbf{1}'\mathbf{M}_j[\tilde{\mathbf{X}}_j - (\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{W}_j\mathbf{Q}_j']}{\mathbf{1}'\mathbf{M}_j\mathbf{1}})\mathbf{Q}_j - (\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{W}_j = \mathbf{J}_j(\tilde{\mathbf{X}}_j\mathbf{Q}_j - \mathbf{Y}\mathbf{W}_j),$$

with $\mathbf{J}_j = (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}'\mathbf{M}_j}{\mathbf{1}'\mathbf{M}_j\mathbf{1}})$.

Hence, defining $\mathbf{C}_j = \mathbf{M}_j\mathbf{J}_j$, and since $\mathbf{C}_j$ is an idempotent matrix, the complete DIMFREE loss function (3.7) can be written as

$$f(\mathbf{Q},\mathbf{W},\mathbf{Y}) = \sum_{j=1}^{n} \mathrm{tr}\,(\tilde{\mathbf{X}}_j\mathbf{Q}_j - \mathbf{Y}\mathbf{W}_j)'\mathbf{C}_j(\tilde{\mathbf{X}}_j\mathbf{Q}_j - \mathbf{Y}\mathbf{W}_j). \tag{3.10}$$

Substitution of (3.9) in (3.7) has the nice effect of simultaneously eliminating the $\mathbf{g}_j$ *and* the $\mathbf{h}_j$ from (3.7), which means that model (3.4) can be simplified to

$$\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j = \mathbf{C}_j\mathbf{Y}\mathbf{W}_j + \mathbf{C}_j\mathbf{E}_j \qquad \text{for } j = 1, 2, ..., n. \tag{3.11}$$

This shows that, just like in GPA, the problem of translations in the DIMFREE model is implicitly solved by centering the non-missing elements in each column of $\tilde{\mathbf{X}}_j$ and the corresponding elements in each column of $\mathbf{Y}$ on the origin of m-dimensional space.

It is always true that $\mathbf{C}_j\mathbf{1} = \mathbf{0}$, and, therefore, that

$$\mathbf{C}_j\tilde{\mathbf{X}}_j = s_j\mathbf{C}_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j\mathbf{K} = s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j\mathbf{K} - s_j\mathbf{C}_j\mathbf{1}\mathbf{u}_j'\mathbf{R}_j\mathbf{K} = s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j\mathbf{K}, \tag{3.12}$$

where s_j, $\mathbf{u}_j$, and $\mathbf{R}_j$ are the optimal central dilation, translation and orthonormal transformation obtained in GPA, and $\mathbf{K}$ is the orthonormal matrix rotating the GPA solution to its principal components. It follows from (3.12) that nothing changes by using $s_j\mathbf{X}_j\mathbf{R}_j\mathbf{K}$ instead of $s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j\mathbf{K}$ when fitting the DIMFREE model on the optimally transformed configurations in GPA. Hence, we define $\tilde{\mathbf{X}}_j = s_j\mathbf{X}_j\mathbf{R}_j\mathbf{K}$ throughout the rest of section 3.3.

3.3.3 Orthonormal transformations

To determine the orthonormal matrices $\mathbf{Q}_j$ minimizing (3.10), define $\mathbf{A}_j = \mathbf{Y}\mathbf{W}_j$ and rewrite (3.10) as

$$f(\mathbf{Q}) = \sum_{j=1}^{n} \text{tr } (\tilde{\mathbf{X}}_j\mathbf{Q}_j - \mathbf{A}_j)'\mathbf{C}_j(\tilde{\mathbf{X}}_j\mathbf{Q}_j - \mathbf{A}_j). \tag{3.13}$$

Considering only one particular $\mathbf{Q}_j$, (3.13) can be written as

$$f(\mathbf{Q}_j) = \text{tr } \mathbf{Q}_j'\tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j - 2 \text{ tr } \mathbf{Q}_j'\tilde{\mathbf{X}}_j'\mathbf{C}_j\mathbf{A}_j + b_j, \tag{3.14}$$

where b_j is a term independent of $\mathbf{Q}_j$. Therefore, the minimization of (3.14) is equivalent to the maximization of

$$g(\mathbf{Q}_j) = \text{tr } \mathbf{Q}_j'\tilde{\mathbf{X}}_j'\mathbf{C}_j\mathbf{A}_j \tag{3.15}$$

under the constraint that $\mathbf{Q}_j'\mathbf{Q}_j = \mathbf{Q}_j\mathbf{Q}_j' = \mathbf{I}_m$. Letting

$$\tilde{\mathbf{X}}_j'\mathbf{C}_j\mathbf{A}_j = \mathbf{K}_j\mathbf{\Phi}_j\mathbf{L}_j' \tag{3.16}$$

be a singular value decomposition of $\tilde{\mathbf{X}}_j'\mathbf{C}_j\mathbf{A}_j$, the global maximum of (3.15) is found for

$$\mathbf{Q}_j = \mathbf{K}_j\mathbf{L}_j'. \tag{3.17}$$

This analytical solution obviously satisfies the constraint that $\mathbf{Q}_j$ must be orthonormal, and holds regardless of singularity of (3.16). Because (3.17) gives the global minimum of (3.14) with respect to $\mathbf{Q}_j$, determining (3.17) for j = 1, ..., n is guaranteed to yield the global minimum of (3.13) with respect to $\mathbf{Q}$.

3.3.4 Dimension weights

For fixed $\mathbf{Q}$ and $\mathbf{Y}$, the problem discussed in this section is how to minimize

$$f(\mathbf{W}) = \sum_{j=1}^{n} \text{tr } (\mathbf{A}_j - \mathbf{B}_j\mathbf{W}_j)'(\mathbf{A}_j - \mathbf{B}_j\mathbf{W}_j), \tag{3.18}$$

where $\mathbf{A}_j \equiv \mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j$ and $\mathbf{B}_j \equiv \mathbf{C}_j\mathbf{Y}$. Considering only one $\mathbf{W}_j$, (3.18) may be written as

$$\begin{aligned} f(\mathbf{W}_j) &= \text{tr } \mathbf{A}_j'\mathbf{A}_j + \text{tr } \mathbf{W}_j^2(\text{diag } \mathbf{B}_j'\mathbf{B}_j) - 2 \text{ tr } \mathbf{W}_j(\text{diag } \mathbf{A}_j'\mathbf{B}_j) \\ &= d_j + \| \mathbf{W}_j(\text{diag } \mathbf{B}_j'\mathbf{B}_j)^{1/2} - (\text{diag } \mathbf{B}_j'\mathbf{B}_j)^{-1/2}(\text{diag } \mathbf{A}_j'\mathbf{B}_j) \|^2, \end{aligned} \tag{3.19}$$

where d_j is a constant with respect to $\mathbf{W}_j$. Therefore, the global minimum of (3.19) is attained for

$$\mathbf{W}_j = (\text{diag } \mathbf{B}_j'\mathbf{B}_j)^{-1}(\text{diag } \mathbf{A}_j'\mathbf{B}_j). \tag{3.20}$$

Because (3.20) gives the global minimum of (3.19), calculating (3.20) for j = 1, ..., n also yields the global minimum of (3.18) with respect to $\mathbf{W}$. In words, the optimal

dimension weights in (3.20) are the raw regression weights in the regression equations

$$\mathbf{a}_{kj} = w_{kj}\mathbf{b}_{kj} + \mathbf{e}_{kj}, \quad \text{for } k = 1, 2, ..., m,$$

where $\mathbf{a}_{kj}$ and $\mathbf{b}_{kj}$ are column k of $\mathbf{A}_j$ and $\mathbf{B}_j$, respectively, and w_{kj} is the dimension weight in $\mathbf{W}_j$ corresponding to dimension k.

The last unknown to be determined in (3.10) is the unrestricted group configuration $\mathbf{Y}$. This problem is discussed in the next section.

3.3.5 The centroid configuration Y

For fixed $\mathbf{Q}$ and $\mathbf{W}$, we want to find the minimum of

$$f(\mathbf{Y}) = \sum_{j=1}^{n} \text{tr}\,(\mathbf{A}_j - \mathbf{C}_j\mathbf{Y}\mathbf{W}_j)'(\mathbf{A}_j - \mathbf{C}_j\mathbf{Y}\mathbf{W}_j), \tag{3.21}$$

where $\mathbf{A}_j \equiv \mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j$. Define $\mathbf{a}_j = (\text{vec } \mathbf{A}_j)$, that is, let $\mathbf{a}_j$ be the (mp x 1) vector consisting of the columns of $\mathbf{A}_j$ stacked on top of one another, and let (vec $\mathbf{C}_j\mathbf{Y}\mathbf{W}_j$) denote the (mp x 1) vector which is obtained by stacking the columns of $\mathbf{C}_j\mathbf{Y}\mathbf{W}_j$ on top of one another. Since (see, e.g., Magnus and Neudecker, 1988)

$$(\text{vec } \mathbf{C}_j\mathbf{Y}\mathbf{W}_j) = (\mathbf{W}_j \otimes \mathbf{C}_j)(\text{vec } \mathbf{Y}),$$

where (vec $\mathbf{Y}$) is the (pm x 1) vector containing the columns of $\mathbf{Y}$ stacked on top of one another and $\otimes$ denotes the right Kronecker product, we may rewrite (3.21) as

$$f(\mathbf{y}) = \sum_{j=1}^{n} (\mathbf{a}_j - \mathbf{D}_j\mathbf{y})'(\mathbf{a}_j - \mathbf{D}_j\mathbf{y}), \tag{3.22}$$

where $\mathbf{y} \equiv (\text{vec } \mathbf{Y})$ of order (mp x 1), and $\mathbf{D}_j \equiv (\mathbf{W}_j \otimes \mathbf{C}_j)$ of order (mp x mp). If we finally let the supervector $\mathbf{a}$ of order (nmp x 1) contain the $\mathbf{a}_j$'s stacked on top of one another, and if we collect the matrices $\mathbf{D}_j$ in the supermatrix $\mathbf{D}$ of order (nmp x mp), (3.22) may be written as

$$f(\mathbf{y}) = (\mathbf{a} - \mathbf{D}\mathbf{y})'(\mathbf{a} - \mathbf{D}\mathbf{y}). \tag{3.23}$$

This is the classical univariate multiple regression problem. The solution follows from the normal equations

$$\mathbf{D'Dy} = \mathbf{D'a}, \tag{3.24}$$

which may be re-expressed in terms of the original matrices as

$$[\sum_{j=1}^{n} (\mathbf{W}_j^2 \otimes \mathbf{C}_j)](\text{vec } \mathbf{Y}) = \sum_{j=1}^{n} (\text{vec } \mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j\mathbf{W}_j). \tag{3.25}$$

Hence, the global minimum of (3.21) is attained for

$$\text{vec } \mathbf{Y} = [\sum_{j=1}^{n} (\mathbf{W}_j^2 \otimes \mathbf{C}_j)]^{-} [\sum_{j=1}^{n} (\text{vec } \mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j\mathbf{W}_j)], \tag{3.26}$$

where $[\Sigma (\mathbf{W}_j^2 \otimes \mathbf{C}_j)]^{-}$ is the Moore-Penrose generalized inverse of $\Sigma (\mathbf{W}_j^2 \otimes \mathbf{C}_j)$. The latter matrix of order (pm x pm) is block-diagonal, and the Moore-Penrose generalized inverse of this matrix can therefore be computed for each of the m symmetric blocks of order (p x p) separately. For the determination of the Moore-Penrose inverse of a singular Gramian matrix we refer to section 2.3.2 of chapter 2.

In the special case that all configurations are complete, and assuming that they have been centered on the origin, (3.26) simplifies to

$$\mathbf{Y} = (\sum_{j=1}^{n} \tilde{\mathbf{X}}_j\mathbf{Q}_j\mathbf{W}_j)(\sum_{j=1}^{n} \mathbf{W}_j^2)^{-1}, \tag{3.27}$$

and we only need to determine the proper inverse of an (m x m) diagonal matrix.

3.3.6 The direct approach in DIMFREE

In this section we briefly discuss the consequences of eliminating the centroid configuration **Y** from the loss function for the DIMFREE model by substitution of (3.26) in (3.10). We have investigated this option because we were interested in finding out whether an analogous situation could be created as in the GPA model (see section 2.3.3 of chapter 2), where the elimination of **Z** resulted in a very efficient method for the determination of the unknown orthonormal transformation matrices and scaling factors.

We state, without proof, that substitution of (3.26) in (3.10) yields the following loss function:

$$f(\mathbf{Q},\mathbf{W}) = n - [\mathrm{vec}\,(\sum_{j=1}^{n} \mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j\mathbf{W}_j)]'[\sum_{j=1}^{n} (\mathbf{W}_j^2 \otimes \mathbf{C}_j)]^{-}[\mathrm{vec}\,(\sum_{j=1}^{n} \mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j\mathbf{W}_j)]. \quad (3.28)$$

Although we will not discuss the solution in detail, we note that it is possible to determine the orthonormal matrices $\mathbf{Q}_j$ minimizing (3.28) for fixed $\mathbf{W}$. This solution for the estimation of each $\mathbf{Q}_j$ in (3.28) is closely related to the solution proposed by Mooijaart and Commandeur (1990) for the weighted orthonormal Procrustes problem, and requires an iterative procedure if the configurations are more than two-dimensional. Because the procedure is iterative, it is much more expensive in terms of computation time than the analytical solution for $\mathbf{Q}$ discussed in section 3.3.3. Moreover, in more than two dimensions it is unclear whether this iterative procedure always yields the global minimum of (3.28) with respect to $\mathbf{Q}$, while the procedure described in section 3.3.3 is guaranteed to give the global minimum of (3.10).

As for the dimension weights, we have not been able to solve the problem of minimizing (3.28) with respect to $\mathbf{W}$ for fixed $\mathbf{Q}$ in the case of incomplete configurations. Without missing data, however, this minimization problem can be shown to be an eigenvalue-eigenvector problem for which an analytical solution is available.

Hence, in the general case of three- and higher dimensional incomplete configurations the 'centroid' approach (3.10) is to be preferred to the direct approach (3.28), because the former results in a more efficient procedure for the estimation of the matrices $\mathbf{Q}_j$, and because at present no solution is available for the estimation of the dimension weights $\mathbf{W}_j$ in (3.28) when the configurations are incomplete.

3.3.7 Uniqueness of the DIMFREE solution

The solution for the DIMFREE model is unique up to a weighting of the optimal dimensions weights $\mathbf{W}_j$ together with an inverse weighting of the columns of the optimal centroid $\mathbf{Y}$. This is easily verified by noting that the solution found for this model will always be only one member of the following family of equivalent solutions:

$$f(\mathbf{Q},\mathbf{W},\mathbf{Y}) = \sum_{j=1}^{n} \text{tr}\,(\tilde{\mathbf{X}}_j\mathbf{Q}_j - \mathbf{Y}\mathbf{L}^{-1}\mathbf{L}\mathbf{W}_j)'\mathbf{C}_j(\tilde{\mathbf{X}}_j\mathbf{Q}_j - \mathbf{Y}\mathbf{L}^{-1}\mathbf{L}\mathbf{W}_j), \tag{3.29}$$

where $\mathbf{L}$ is an arbitrary diagonal matrix of order (m x m).

The freedom we have in choosing a particular diagonal matrix $\mathbf{L}$ can be used to select the matrix $\mathbf{L}$ which guarantees the uniqueness of the solution for the DIMFREE model with respect to dimension weighting. We propose to use the following diagonal matrix, which we will denote by $\mathbf{D}$:

$$\mathbf{D} = (\text{diag}\ \mathbf{Y}'\mathbf{Y})^{1/2}, \tag{3.30}$$

yielding a uniquely weighted DIMFREE solution where

$$\mathbf{Y}^* = \mathbf{Y}\mathbf{D}^{-1} = \mathbf{Y}(\text{diag}\ \mathbf{Y}'\mathbf{Y})^{-1/2} \tag{3.31}$$

and

$$\mathbf{W}_j^* = \mathbf{D}\mathbf{W}_j = (\text{diag}\ \mathbf{Y}'\mathbf{Y})^{1/2}\mathbf{W}_j \tag{3.32}$$

for j = 1, 2, ..., n.

That (3.31) and (3.32) guarantee a uniquely weighted solution can be seen as follows. To postmultiply $\mathbf{Y}$ with a diagonal matrix has the effect of rescaling its columns, and $\mathbf{Y}$ is, therefore, unique up to a differential rescaling of its columns. Since postmultiplying $\mathbf{Y}$ with $\mathbf{D}^{-1}$ has the effect of unit normalizing the columns of $\mathbf{Y}$ (i.e. $(\text{diag}\ \mathbf{D}^{-1}\mathbf{Y}'\mathbf{Y}\mathbf{D}^{-1}) = \mathbf{I}_m$), it immediately follows that (3.31) yields a uniquely weighted $\mathbf{Y}^*$, and it also follows that (3.32) yields uniquely weighted dimension weights $\mathbf{W}_j^*$.

As we will prove in section 3.5.2, if there are no missing data the choice for (3.30) also guarantees that the sum of squared dimension weights $\mathbf{W}_j^*$ for each j becomes equal to the squared correlation between the elements of $\tilde{\mathbf{X}}_j\mathbf{Q}_j$ and the elements of $\mathbf{Y}\mathbf{W}_j$.

An important property of the DIMFREE solution is that the centroid configuration $\mathbf{Y}$ has optimally rotated dimensions since they are the differentially weighted ones. Analogous to the INDSCAL model, in DIMFREE also it is the model itself that, as it were, dictates which axes are to be used for interpretation. Hence, having uniquely weighted the centroid configuration according to (3.31) and the dimension weights according to (3.32), the DIMFREE solution is unique up to permutations and

reflections of the m dimensions. Under these restrictions we are still free to permute the m dimensions such that they are ordered according to the amount of variation accounted for by each dimension. In section 3.5.2 we will give a procedure to determine this order.

Because there is no analytical solution for the unknowns in the DIMFREE model, an alternating least squares algorithm is used for the estimation of the unknown transformation parameters in (3.10).

3.3.8 The algorithm

In the MATCHALS program, the algorithm corresponding to the DIMFREE model consists of the following steps. As input to the algorithm we use the n configurations $\tilde{\mathbf{X}}_j = s_j\mathbf{X}_j\mathbf{R}_j\mathbf{K}$ obtained in GPA.

a) Before starting to iterate, initialize the matrices $\mathbf{Q}_j$ on $\mathbf{Q}_j = \mathbf{I}_m$, and $\mathbf{Y}$ on $\mathbf{Y} = \mathbf{ZK}$ (i.e., the optimal centroid configuration found in GPA rotated to the principal components of the GPA solution).

b) For every j, compute new dimension weights with (3.20).

c) Determine a new orthonormal matrix $\mathbf{Q}_j$ for every j by applying the procedure given in section 3.3.3.

d) Again compute new dimension weights with (3.20) for every j.

e) Calculate a new centroid configuration $\mathbf{Y}$. If there are missing data, update $\mathbf{Y}$ according to (3.26). To determine the Moore-Penrose inverse of the block-diagonal matrix $[\Sigma\ (\mathbf{W}_j^2 \otimes \mathbf{C}_j)]$ of order (pm x pm), compute the Moore-Penrose inverse of each of the m blocks separately. This is more efficient than calculating an inverse of the complete (pm x pm) matrix. If all configurations are complete, however, use (3.27) instead of (3.26) to update $\mathbf{Y}$.

f) Evaluate loss function (3.10). If the difference between the value of the function in this iteration and in the previous iteration is smaller than some predetermined convergence criterion, go to step g). Otherwise, go to step b).

g) Print the history of iterations. Compute $\mathbf{Y}^*$ and $\mathbf{W}_j^*$ using (3.31) and (3.32), respectively, and print the uniquely weighted optimal group configuration $\mathbf{Y}^*$ and the unique dimension weights $\mathbf{W}_j^*$ for each j.

This algorithm is guaranteed to converge, although not necessarily to the global minimum of (3.10). The reason why dimension weights are calculated twice in one iteration (i.e., in steps b) and d) of the algorithm) is that, in comparison with the $\mathbf{Q}_j$ and $\mathbf{Y}$, the estimation of the $\mathbf{W}_j$ is relatively cheap in terms of CPU-time.

We end this section by emphasizing that, since the dimension weights $\mathbf{W}_j^*$ and the group configuration $\mathbf{Y}^*$ are uniquely weighted, the solution for the DIMFREE model is unique up to reflections and permutations of the m dimensions. Because of the unique orientation of the axes of $\mathbf{Y}^*$, just as in the literature on the INDSCAL analysis of (dis)similarities data, it is recommended to use only the uniquely oriented dimensions of $\mathbf{Y}^*$ for interpretational purposes.

3.4 The DIMIDIO model

3.4.1 Introduction

To determine the optimal translations, orthonormal transformations, dimension weights, and centroid configuration in the DIMIDIO model, the following least squares loss function

$$f(\mathbf{G},\mathbf{H},\mathbf{Q},\mathbf{S},\mathbf{W},\mathbf{Y}) = \sum_{j=1}^{n} \| \mathbf{M}_j[(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j - (\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{S}_j\mathbf{W}_j] \|^2 \qquad (3.33)$$

is defined, where $\mathbf{G}$ and $\mathbf{H}$ denote the (m x n) matrices in which the translation vectors $\mathbf{g}_j$ and $\mathbf{h}_j$ are collected, respectively, and $\mathbf{Q}$, $\mathbf{S}$, and $\mathbf{W}$ denote the supermatrices of order (nm x m) in which the n matrices $\mathbf{Q}_j$, $\mathbf{S}_j$, and $\mathbf{W}_j$ are collected, respectively. Matrix $\tilde{\mathbf{X}}_j$ again denotes an optimally translated, rotated and rescaled configuration $\mathbf{X}_j$ after GPA rotated to the principal components of the GPA solution (i.e., $\tilde{\mathbf{X}}_j = s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j\mathbf{K}$). In section 3.4.2 we first of all show that the translation vectors $\mathbf{g}_j$ and $\mathbf{h}_j$ can be eliminated from (3.33). In section 3.4.3 we discuss how the minimization of the simplified loss function for the DIMIDIO model with respect to the orthonormal matrices $\mathbf{Q}_j$, dimension weights $\mathbf{W}_j$, and orthonormal matrices $\mathbf{S}_j$ can be reduced to the much more simple problem of minimizing the loss function with respect to one set of parameters only, from which the optimal $\mathbf{Q}$, $\mathbf{W}$, and $\mathbf{S}$ can be recovered afterwards. In sections 3.4.4 and 3.4.5 the resulting DIMIDIO loss function is minimized with respect to the only two remaining sets of parameters. In section 3.4.6 the uniqueness properties of the DIMIDIO model are discussed, and in section 3.4.7 an alternating least squares algorithm is presented for the minimization of the loss function corresponding to the DIMIDIO model.

3.4.2 Translations

The problem of minimizing (3.33) with respect to the unrestricted translation vectors $\mathbf{G}$ for fixed $\mathbf{H}$, $\mathbf{Q}$, $\mathbf{S}$, $\mathbf{W}$, and $\mathbf{Y}$ is so similar to the problem encountered in section 3.3.2 for the DIMFREE model that we state, without proof, that the global minimum of (3.33) with respect to one particular translation vector $\mathbf{g}_j$, that is, of

$$f(\mathbf{g}_j) = \sum_{j=1}^{n} \| \mathbf{M}_j[(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{Q}_j - (\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{S}_j\mathbf{W}_j] \|^2 \tag{3.34}$$

is attained for

$$\mathbf{g}_j = \frac{[\tilde{\mathbf{X}}_j - (\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{S}_j\mathbf{W}_j\mathbf{Q}_j']'\mathbf{M}_j\mathbf{1}}{\mathbf{1}'\mathbf{M}_j\mathbf{1}}. \tag{3.35}$$

Substitution of (3.35) in (3.33) gives

$$f(\mathbf{Q},\mathbf{S},\mathbf{W},\mathbf{Y}) = \sum_{j=1}^{n} \text{tr } (\tilde{\mathbf{X}}_j\mathbf{Q}_j - \mathbf{Y}\mathbf{S}_j\mathbf{W}_j)'\mathbf{C}_j(\tilde{\mathbf{X}}_j\mathbf{Q}_j - \mathbf{Y}\mathbf{S}_j\mathbf{W}_j), \tag{3.36}$$

where $\mathbf{C}_j \equiv \mathbf{M}_j(\mathbf{I} - \mathbf{1}\mathbf{1}'\mathbf{M}_j/\mathbf{1}'\mathbf{M}_j\mathbf{1})$. Thus, substitution of (3.35) in (3.33) results in the simultaneous elimination of **G** *and* **H** from (3.33). After the elimination of the translation vectors from (3.33), model (3.5) can be simplified to

$$\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j = \mathbf{C}_j\mathbf{Y}\mathbf{S}_j\mathbf{W}_j + \mathbf{C}_j\mathbf{E}_j \qquad \text{for } j = 1, 2, ..., n. \tag{3.37}$$

Since it follows from $\mathbf{C}_j\mathbf{1} = \mathbf{0}$ that

$$\mathbf{C}_j\tilde{\mathbf{X}}_j = \mathbf{C}_j(s_j\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j\mathbf{K} = s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j\mathbf{K},$$

we may as well fit the DIMIDIO model on the optimally rotated and rescaled configurations after GPA. It is for this reason that we define $\tilde{\mathbf{X}}_j = s_j\mathbf{X}_j\mathbf{R}_j\mathbf{K}$ throughout the remainder of this chapter.

3.4.3 Reducing the estimation of Q, S, and W to one set of parameters

We start this section by noting that, since for orthonormal matrices $\mathbf{Q}_j$

$$f(\mathbf{Q},\mathbf{S},\mathbf{W},\mathbf{Y}) = \sum_{j=1}^{n} \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j = \sum_{j=1}^{n} \text{tr } \mathbf{Q}_j\mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j\mathbf{Q}_j',$$

where $\mathbf{E}_j$ is defined in (3.37), the minimization of loss function (3.36) with respect to **Q**, **S**, **W**, and **Y** is equivalent to the minimization of

$$f(\mathbf{Q},\mathbf{S},\mathbf{W},\mathbf{Y}) = \sum_{j=1}^{n} \text{tr}\,(\tilde{\mathbf{X}}_j - \mathbf{Y}\mathbf{S}_j\mathbf{W}_j\mathbf{Q}_j')'\mathbf{C}_j(\tilde{\mathbf{X}}_j - \mathbf{Y}\mathbf{S}_j\mathbf{W}_j\mathbf{Q}_j'). \tag{3.38}$$

Letting $\mathbf{B}_j = \mathbf{S}_j\mathbf{W}_j\mathbf{Q}_j'$ of order (m x m), (3.38) may be written as

$$f(\mathbf{B},\mathbf{Y}) = \sum_{j=1}^{n} \text{tr}\,(\tilde{\mathbf{X}}_j - \mathbf{Y}\mathbf{B}_j)'\mathbf{C}_j(\tilde{\mathbf{X}}_j - \mathbf{Y}\mathbf{B}_j), \tag{3.39}$$

where it is assumed that the n matrices $\mathbf{B}_j$ are collected in the supermatrix $\mathbf{B}$ of order (nm x m). It is important to note that $\mathbf{B}_j$ stands for the matrix product of an orthonormal matrix $\mathbf{S}_j$, a diagonal matrix $\mathbf{W}_j$, and another orthonormal matrix $\mathbf{Q}_j'$. Since, moreover, singular value decomposition can be applied to any matrix, no restrictions have to be imposed on the matrices $\mathbf{B}_j$. If, therefore, we have determined the (unrestricted) matrix $\mathbf{B}$ which gives the global minimum of (3.39) for fixed $\mathbf{Y}$, and given the singular value decompositions

$$\mathbf{B}_j = \mathbf{K}_j\mathbf{\Lambda}_j\mathbf{L}_j' \qquad \text{for } j = 1, \ldots, n, \tag{3.40}$$

then the global minimum of (3.38) with respect to $\mathbf{Q}$, $\mathbf{S}$, and $\mathbf{W}$ is obtained by setting $\mathbf{S}_j = \mathbf{K}_j$, $\mathbf{W}_j = \mathbf{\Lambda}_j$, and $\mathbf{Q}_j = \mathbf{L}_j$ for $j = 1, \ldots, n$.

The reformulation of (3.36) in (3.39) considerably simplifies the task of minimizing the DIMIDIO loss function: instead of having to determine four sets of parameters in (3.36) we need to determine only two sets of parameters in (3.39). Once the optimal matrix $\mathbf{B}$ is obtained, the optimal matrices $\mathbf{S}$, $\mathbf{W}$, and $\mathbf{Q}$ can be recovered using (3.40). The following two sections deal with the minimization of (3.39) with respect to the unrestricted matrices $\mathbf{B}$ and $\mathbf{Y}$.

3.4.4 The regression weights B

To minimize (3.39) with respect to unrestricted $\mathbf{B}$ for fixed $\mathbf{Y}$, consider only one $\mathbf{B}_j$ and write (3.39) as

$$f(\mathbf{B}_j) = d_j + \text{tr}\,(\mathbf{A}_j - \mathbf{D}_j\mathbf{B}_j)'(\mathbf{A}_j - \mathbf{D}_j\mathbf{B}_j), \tag{3.41}$$

where $\mathbf{A}_j \equiv \mathbf{C}_j\tilde{\mathbf{X}}_j$, $\mathbf{D}_j \equiv \mathbf{C}_j\mathbf{Y}$, and d_j is a constant with respect to $\mathbf{B}_j$. This is the classical multivariate multiple regression problem with the well-known solution

$$\mathbf{B}_j = (\mathbf{D}_j'\mathbf{D}_j)^{-1}(\mathbf{D}_j'\mathbf{A}_j), \tag{3.42}$$

which, when calculated for j = 1, ..., n, is guaranteed to yield the global minimum of (3.39) with respect to **B**.

It may be interesting to note that, applying the results of sections 3.4.2, 3.4.3 and 3.4.4 to the least squares loss function corresponding to the original model (3.3) of Lingoes and Borg where the centroid configuration was assumed to be fixed (see section 3.2), one finds that the minimization of this loss function has an *analytical* solution.

3.4.5 The centroid configuration Y

For fixed **B**, we want to find the minimum of

$$f(\mathbf{Y}) = \sum_{j=1}^{n} \text{tr } (\mathbf{A}_j - \mathbf{C}_j\mathbf{Y}\mathbf{B}_j)'(\mathbf{A}_j - \mathbf{C}_j\mathbf{Y}\mathbf{B}_j), \tag{3.43}$$

where $\mathbf{A}_j \equiv \mathbf{C}_j\tilde{\mathbf{X}}_j$. Applying the same procedure as in section 3.3.5, that is, letting **a** be the (nmp x 1) supervector containing the n vectors (vec $\mathbf{A}_j$) on top of one another, letting **D** be the (nmp x mp) supermatrix containing the n matrices ($\mathbf{B}_j' \otimes \mathbf{C}_j$), and defining **y** = (vec **Y**), the problem of minimizing (3.40) can be expressed as the classical univariate multiple regression problem:

$$f(\mathbf{y}) = (\mathbf{a} - \mathbf{D}\mathbf{y})'(\mathbf{a} - \mathbf{D}\mathbf{y}). \tag{3.44}$$

Hence, the global minimum of (3.43) is attained for

$$\text{vec } \mathbf{Y} = (\mathbf{D}'\mathbf{D})^{-}\mathbf{D}'\mathbf{a} = [\sum_{j=1}^{n} (\mathbf{B}_j\mathbf{B}_j' \otimes \mathbf{C}_j)]^{-}[\text{vec} \sum_{j=1}^{n} \mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{B}_j'], \tag{3.45}$$

where $[\sum (\mathbf{B}_j\mathbf{B}_j' \otimes \mathbf{C}_j)]^{-}$ is the Moore-Penrose inverse of $\sum (\mathbf{B}_j\mathbf{B}_j' \otimes \mathbf{C}_j)$. This latter matrix of order (pm x pm) is no longer block-diagonal, and the Moore-Penrose inverse must therefore be determined for the complete matrix.

In the special case that all n configurations are complete, and assuming that they have been centered on the origin, (3.45) simplifies to

$$\mathbf{Y} = (\sum_{j=1}^{n} \tilde{\mathbf{X}}_j \mathbf{B}_j')(\sum_{j=1}^{n} \mathbf{B}_j \mathbf{B}_j')^{-1}, \tag{3.46}$$

and we only need to determine the proper inverse of an (m x m) matrix.

3.4.6 Uniqueness of the DIMIDIO solution

Since both **B** and **Y** in DIMIDIO loss function (3.39) are unrestricted matrices, the value of this function is unchanged by the following transformation:

$$f(\mathbf{B},\mathbf{Y}) = \sum_{j=1}^{n} \text{tr}\, (\tilde{\mathbf{X}}_j - \mathbf{Y}\mathbf{T}^{-1}\mathbf{T}\mathbf{B}_j)'\mathbf{C}_j(\tilde{\mathbf{X}}_j - \mathbf{Y}\mathbf{T}^{-1}\mathbf{T}\mathbf{B}_j), \tag{3.47}$$

where **T** is an arbitrary non-singular matrix of order (m x m). Because of its nice properties, we choose for **T** in (3.47) the special solution

$$\mathbf{G} = \mathbf{H}\mathbf{N}', \tag{3.48}$$

where **H** contains the singular values, and **N** the right singular vectors of the singular value decomposition

$$\mathbf{Y} = \mathbf{M}\mathbf{H}\mathbf{N}'. \tag{3.49}$$

This yields a DIMIDIO solution where

$$\mathbf{Y}^* = \mathbf{Y}\mathbf{G}^{-1} = \mathbf{M}\mathbf{H}\mathbf{N}'(\mathbf{H}\mathbf{N}')^{-1} = \mathbf{M}, \tag{3.50}$$

and

$$\mathbf{B}_j^* = \mathbf{G}\mathbf{B}_j = \mathbf{H}\mathbf{N}'\mathbf{B}_j \tag{3.51}$$

for j = 1, ..., n. Since $\mathbf{Y}^* = \mathbf{M}$, the transformed centroid configuration is columnwise orthonormal. Evidently, having defined (3.51) we use the singular value decompositions

$$\mathbf{B}_j^* = \mathbf{K}_j\mathbf{\Phi}_j\mathbf{L}_j' \qquad \text{for } j = 1, \ldots, n \tag{3.52}$$

instead of (3.40) to determine the optimal $\mathbf{S}_j$, $\mathbf{W}_j$, and $\mathbf{Q}_j$ in DIMIDIO loss function (3.36). That is, we let $\mathbf{S}_j = \mathbf{K}_j$ in (3.52), $\mathbf{W}_j = \mathbf{\Phi}_j$ in (3.52), and $\mathbf{Q}_j = \mathbf{L}_j$ in (3.52).

It follows from (3.50), (3.51), and (3.52) that the $\mathbf{YB}_j$ are decomposed into

$$\mathbf{YB}_j = \mathbf{MK}_j\mathbf{\Phi}_j\mathbf{L}_j' = \mathbf{Y}^*\mathbf{S}_j\mathbf{W}_j\mathbf{Q}_j', \qquad \text{for } j = 1, \ldots, n, \tag{3.53}$$

which, by letting $\mathbf{V}_j = \mathbf{Y}^*\mathbf{S}_j = \mathbf{MK}_j$, may be written as

$$\mathbf{YB}_j = \mathbf{V}_j\mathbf{\Phi}_j\mathbf{L}_j', \qquad \text{for } j = 1, \ldots, n. \tag{3.54}$$

Since $\mathbf{V}_j$ is an orthonormal matrix, being the product of orthonormal matrices $\mathbf{M}$ and $\mathbf{K}_j$, (3.54) is a singular value decomposition of $\mathbf{YB}_j$. Therefore, the matrices $\mathbf{W}_j$ and $\mathbf{Q}_j$ defined by (3.52) have the nice property that they are unique up to reflections. The same holds for the matrix products $\mathbf{Y}^*\mathbf{S}_j = \mathbf{MK}_j$. Only one indeterminacy remains: the matrices $\mathbf{Y}^* = \mathbf{M}$ and $\mathbf{S}_j = \mathbf{K}_j$ themselves are only determined up to a rotation, because we are still free to write (3.53) as

$$\mathbf{YB}_j = \mathbf{MP'PK}_j\mathbf{\Phi}_j\mathbf{L}_j' = \mathbf{Y}^*\mathbf{P'PS}_j\mathbf{W}_j\mathbf{Q}_j', \qquad \text{for } j = 1, \ldots, n, \tag{3.55}$$

where $\mathbf{P}$ is an arbitrary orthonormal matrix of order (m x m). In other words, the matrix product $\mathbf{Y}^*\mathbf{P}'$ just defines another orthonormal basis of the raw $\mathbf{Y}$.

The interpretation of the DIMIDIO model does not require a unique orientation of the centroid configuration, however, because it is the dimensions of the $\mathbf{Y}^*\mathbf{S}_j$ that are differentially being weighted, and the procedure proposed in this section guarantees the uniqueness of the latter matrix products.

As we will prove in section 3.5.3, another advantage of the choice for (3.48) is that, if all configurations are complete, the sum of squared dimension weights $\mathbf{W}_j$ for each j is equal to the squared correlation between the elements of $\tilde{\mathbf{X}}_j\mathbf{Q}_j$ and the elements of $\mathbf{Y}^*\mathbf{S}_j\mathbf{W}_j$.

In the following section we present an alternating least squares algorithm for the estimation of the unknown parameters in the DIMIDIO model.

3.4.7 The algorithm

In the MATCHALS program, the algorithm corresponding to the DIMIDIO model consists of the following steps.

a) Before starting to iterate, initialize $\mathbf{Y}$ on $\mathbf{ZK}$ (i.e., the optimal centroid configuration in GPA rotated to the principal components of the GPA solution).
b) Compute new regression weights with (3.42) for every j.
c) Calculate a new centroid configuration $\mathbf{Y}$. If the configurations are incomplete, formula (3.45) must be used to update $\mathbf{Y}$. If, on the other hand, all configurations are complete, use (3.46) instead of (3.45) since this is much cheaper in terms of CPU-time.
d) Evaluate loss function (3.39). If the difference between the value of the function in this outer iteration and in the previous iteration is smaller than some convergence criterion, go to step e). Otherwise, go to step b).
e) Print the history of iterations. Determine the singular value decomposition of the optimal $\mathbf{Y}$, and compute $\mathbf{Y}^*$ and $\mathbf{B}_j^*$ for each j as defined in (3.50) and (3.51), respectively. Then determine the singular value decompositions $\mathbf{B}_j^* = \mathbf{K}_j\mathbf{\Lambda}_j\mathbf{L}_j'$ for each j, and set $\mathbf{S}_j = \mathbf{K}_j$, $\mathbf{W}_j = \mathbf{\Lambda}_j$, and $\mathbf{Q}_j = \mathbf{L}_j$ for each j. Print $\mathbf{Y}^*$, and the $\mathbf{S}_j$, $\mathbf{W}_j$, and $\mathbf{Q}_j$.

This algorithm must converge although it can not be proved that it will necessarily find the global minimum of (3.36). Since in steps b) and c) both $\mathbf{B}$ and $\mathbf{Y}$ are updated by the computation of regression weights, the iterative part of the DIMIDIO algorithm is entirely based on multiple regression.

The computation of the Moore-Penrose inverse of the (pm x pm) matrix needed in the case of missing data in step c) of the algorithm is quite expensive. If the number of dimensions *and* the number of stimuli is large, in practice the problem of calculating the Moore-Penrose inverse in (3.45) may well prove to be impossible to cope with. This problem does not arise with complete configurations, of course, because then (3.46) can be used to update $\mathbf{Y}$, which only involves the computation of a proper inverse of order (m x m). Still, we emphasize that fitting the DIMIDIO model on large incomplete configurations can become impractical due to computational limits that are easily reached in step c) of the algorithm.

As we already noted in section 3.4.6, the procedure described in step e) of the algorithm guarantees the uniqueness of the DIMIDIO solution up to a rotation of $\mathbf{Y}^*$ together with an inverse rotation of the matrices $\mathbf{S}_j$.

3.5 Analysis of variation in the dimension weighting models

3.5.1 Introduction

In the following sections we will show that the total sum of squares of the n configurations can be partitioned in two components for the two dimension weighting models discussed in this chapter. These two components are again decomposed in three ways in order to allow one to assess the relative contribution of each individual configuration, of each individual stimulus, and of each dimension to the total solution. We first discuss the analysis of variation for the DIMFREE model (3.4) in section 3.5.2, and the analysis of variation for the DIMIDIO model (3.5) is the subject of section 3.5.3.

3.5.2 Partitioning of the sums of squares in DIMFREE

The linear model (3.11) underlying DIMFREE loss function (3.10) is (see section 3.3.2)

$$\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j = \mathbf{C}_j\mathbf{Y}\mathbf{W}_j + \mathbf{C}_j\mathbf{E}_j \qquad \text{for } j = 1, 2, ..., n,$$

from which it follows that

$$\text{tr } \mathbf{Q}_j'\tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j = \text{tr } (\mathbf{Y}\mathbf{W}_j + \mathbf{E}_j)'\mathbf{C}_j(\mathbf{Y}\mathbf{W}_j + \mathbf{E}_j), \tag{3.56}$$

and therefore that

$$\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j = \text{tr } \mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j + \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j + 2 \text{ tr } \mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{E}_j. \tag{3.57}$$

Because $(\text{tr } \mathbf{Y}'\mathbf{C}_j\mathbf{E}_j) = 0$, due to the orthogonality of residual and predictor space in regression analysis, (3.57) may be written as

$$\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j = \text{tr } \mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j + \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j, \qquad \text{for } j = 1, ..., n, \tag{3.58}$$

from which it immediately follows that

$$0 \leq (\text{tr } \mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j)/(\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j) \leq 1, \qquad \text{for } j = 1, \ldots, n, \tag{3.59}$$

showing that (3.59) directly yields a measure of fit for each configuration j in the DIMFREE solution. It also follows from (3.58) that

$$\sum_{j=1}^{n} \text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j = \sum_{j=1}^{n} \text{tr } \mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j + \sum_{j=1}^{n} \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j,$$

and therefore that

$$n = \sum_{j=1}^{n} \text{tr } \mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j + f(\mathbf{Q},\mathbf{W},\mathbf{Y}), \tag{3.60}$$

with $f(\mathbf{Q},\mathbf{W},\mathbf{Y})$ as in (3.10), and thus that

$$0 \leq (1/n) \sum_{j=1}^{n} \text{tr } \mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j \leq 1. \tag{3.61}$$

Hence, (3.61) is perfectly suited as a measure of total fit.

The measure of fit in (3.59) is equal to the squared correlation between the elements of $\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j$ and the elements of $\mathbf{C}_j\mathbf{Y}\mathbf{W}_j$. This simple relation holds, because

$$r^2(\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j,\mathbf{C}_j\mathbf{Y}\mathbf{W}_j) = \frac{(\text{tr } \mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j)^2}{(\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j)(\text{tr } \mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j)},$$

$$= (\text{tr } \mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j)/(\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j). \tag{3.62}$$

In section 3.3.7 we proposed to normalize the raw dimension weights in the DIMFREE model according to (3.32), that is:

$$\mathbf{W}_j^* = \mathbf{D}\mathbf{W}_j = (\text{diag } \mathbf{Y}'\mathbf{Y})^{1/2}\mathbf{W}_j.$$

In the case that all configurations are complete, and assuming that they have been centered on the origin, it is true that

$$\text{tr } [\mathbf{W}_j^*/(\sqrt{\text{tr } \tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_j})]^2 = r^2(\tilde{\mathbf{X}}_j\mathbf{Q}_j,\mathbf{Y}\mathbf{W}_j). \tag{3.63}$$

Table 3.1 Contribution of individual configurations in the DIMFREE model.

configuration j	SS_{fit}	$SS_{residual}$	SS_{total}
1	$\text{tr } \mathbf{W}_1\mathbf{Y}'\mathbf{C}_1\mathbf{Y}\mathbf{W}_1$	$\text{tr } \mathbf{E}_1'\mathbf{C}_1\mathbf{E}_1$	$\text{tr } \tilde{\mathbf{X}}_1'\mathbf{C}_1\tilde{\mathbf{X}}_1$
2	$\text{tr } \mathbf{W}_2\mathbf{Y}'\mathbf{C}_2\mathbf{Y}\mathbf{W}_2$	$\text{tr } \mathbf{E}_2'\mathbf{C}_2\mathbf{E}_2$	$\text{tr } \tilde{\mathbf{X}}_2'\mathbf{C}_2\tilde{\mathbf{X}}_2$
⋮	⋮	⋮	⋮
n	$\text{tr } \mathbf{W}_n\mathbf{Y}'\mathbf{C}_n\mathbf{Y}\mathbf{W}_n$	$\text{tr } \mathbf{E}_n'\mathbf{C}_n\mathbf{E}_n$	$\text{tr } \tilde{\mathbf{X}}_n'\mathbf{C}_n\tilde{\mathbf{X}}_n$
$\sum_{j=1}^{n}$	$\sum_{j=1}^{n} \text{tr } \mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j$	$\sum_{j=1}^{n} \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j$	n

Identity (3.63) follows from substitution of (3.32) in the left side term of (3.63):

$$\text{tr } [\mathbf{W}_j^*/(\sqrt{\text{tr } \tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_j})]^2 = (\text{tr } \mathbf{W}_j^*)^2/(\text{tr } \tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_j) = (\text{tr } \mathbf{W}_j\mathbf{D}^2\mathbf{W}_j)/(\text{tr } \tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_j) =$$

$$(\text{tr } \mathbf{W}_j(\text{diag } \mathbf{Y}'\mathbf{Y})\mathbf{W}_j)/(\text{tr } \tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_j) = (\text{tr } \mathbf{W}_j\mathbf{Y}'\mathbf{Y}\mathbf{W}_j)/(\text{tr } \tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_j) = r^2(\tilde{\mathbf{X}}_j\mathbf{Q}_j, \mathbf{Y}\mathbf{W}_j).$$

In words (3.63) expresses that, without missing data, the squared norm of the normalized dimension weights contained in $\mathbf{W}_j^*/(\sqrt{\text{tr } \tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_j})$ directly reflects the amount of variation in configuration j that is accounted for by the DIMFREE model.

We use the above results to give a further decomposition of the sums of squares in (3.60) allowing one to assess the relative contribution of individual configurations, stimuli and dimensions to the DIMFREE solution. In Table 3.1 the decomposition of (3.60) with respect to the individual configurations is given. This table is, of course, nothing more than (3.58) in tabulated form.

Table 3.2 shows how to assess the relative contribution of each individual stimulus to the DIMFREE solution. It is important to note that in this model no independent sums of squares can be obtained for each stimulus separately, both with and without missing data. Therefore, we propose to calculate the amount of variation accounted for by each stimulus according to the expression given in the column titled 'SS_{fit}' in Table 3.2, since this guarantees that SS_{fit} and $SS_{residual}$ add up to SS_{total} for each stimulus i, and that $\sum SS_{fit}$ is equal to the total sum of squares in (3.61). But it must be kept in mind that these decompositions do not result in independent sums of squares.

Table 3.2 Contribution of individual stimuli in the DIMFREE model.

stimulus i	SS_{fit}	$SS_{residual}$	SS_{total}
1	a_{11}*	b_{11}**	d_{11}***
2	a_{22}*	b_{22}**	d_{22}***
⋮	⋮	⋮	⋮
p	a_{pp}*	b_{pp}**	d_{pp}***
$\sum_{i=1}^{p}$	$\sum_{j=1}^{n}$ tr $\mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j$	$\sum_{j=1}^{n}$ tr $\mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j$	n

* a_{ii} is diagonal element ii of $\sum_{j=1}^{n}\{2\,\mathbf{C}_j\mathbf{Y}\mathbf{W}_j(\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j)' - \mathbf{C}_j\mathbf{Y}\mathbf{W}_j(\mathbf{C}_j\mathbf{Y}\mathbf{W}_j)'\}$

** b_{ii} is diagonal element ii of $\sum_{j=1}^{n}\mathbf{C}_j\mathbf{E}_j(\mathbf{C}_j\mathbf{E}_j)'$

*** d_{ii} is diagonal element ii of $\sum_{j=1}^{n}\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j(\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j)'$

Table 3.3 Contribution of individual dimensions in the DIMFREE model.

dimension k	SS_{fit}	$SS_{residual}$	SS_{total}
1	e_{11}*	f_{11}**	g_{11}***
2	e_{22}*	f_{22}**	g_{22}***
⋮	⋮	⋮	⋮
m	e_{mm}*	f_{mm}**	g_{mm}***
$\sum_{k=1}^{m}$	$\sum_{j=1}^{n}$ tr $\mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j$	$\sum_{j=1}^{n}$ tr $\mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j$	n

* e_{kk} is diagonal element kk of $\sum_{j=1}^{n}\mathbf{W}_j\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{W}_j$

** f_{kk} is diagonal element kk of $\sum_{j=1}^{n}\mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j$

*** g_{kk} is diagonal element kk of $\sum_{j=1}^{n}\mathbf{Q}_j'\tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j$

The partitioning of the sums of squares in (3.60) for each dimension separately is given in Table 3.3. This decomposition yields independent components both with and without missing data, and the information in this table can be used to permute the m dimensions such that they become ordered according to the amount of variation accounted for by each dimension.

3.5.3 Partitioning of the sums of squares in DIMIDIO

The linear model (3.37) underlying DIMIDIO loss function (3.36) is (see section 3.4.3)

$$\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j = \mathbf{C}_j\mathbf{Y}\mathbf{S}_j\mathbf{W}_j + \mathbf{C}_j\mathbf{E}_j, \qquad \text{for } j = 1, 2, ..., n,$$

from which it follows that

$$\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j = \text{tr } (\mathbf{Y}\mathbf{S}_j\mathbf{W}_j + \mathbf{E}_j)'\mathbf{C}_j(\mathbf{Y}\mathbf{S}_j\mathbf{W}_j + \mathbf{E}_j), \text{ for } j = 1, \ldots, n. \tag{3.64}$$

Expanding the trace on the right side of (3.64) gives

$$\text{tr } \mathbf{W}_j\mathbf{S}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{S}_j\mathbf{W}_j + \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j + 2 \text{ tr } \mathbf{W}_j\mathbf{S}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{E}_j =$$

$$\text{tr } \mathbf{Q}_j\mathbf{W}_j\mathbf{S}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{S}_j\mathbf{W}_j\mathbf{Q}_j' + \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j + 2 \text{ tr } \mathbf{Q}_j\mathbf{W}_j\mathbf{S}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{E}_j\mathbf{Q}_j' =$$

$$\text{tr } \mathbf{B}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{B}_j + \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j + 2 \text{ tr } \mathbf{B}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{E}_j\mathbf{Q}_j',$$

where $\mathbf{B}_j \equiv \mathbf{S}_j\mathbf{W}_j\mathbf{Q}_j'$. Also, the optimal matrix $\mathbf{B}_j$ contains the regression weights of the regression of $\mathbf{C}_j\tilde{\mathbf{X}}_j$ on $\mathbf{C}_j\mathbf{Y}$ (see section 3.4.4), so that the sum of cross products $(\text{tr } \mathbf{B}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{E}_j\mathbf{Q}_j')$ is zero. Therefore (3.64) may be written as

$$\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j = \text{tr } \mathbf{B}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{B}_j + \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j, \qquad \text{for } j = 1, ..., n. \tag{3.65}$$

It immediately follows that

$$0 \leq (\text{tr } \mathbf{B}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{B}_j)/(\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j) \leq 1, \tag{3.66}$$

and

$$(\text{tr } \mathbf{B}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{B}_j)/(\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j) = r^2(\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{Q}_j,\mathbf{C}_j\mathbf{Y}^*\mathbf{S}_j\mathbf{W}_j) \tag{3.67}$$

is a measure of fit for each configuration j in DIMIDIO. If all configurations are complete, and assuming that they have been centered on the origin, it is also true that

$$(\text{tr } \mathbf{B}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{B}_j)/(\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j) = (\text{tr } \mathbf{B}_j'\mathbf{Y}'\mathbf{Y}\mathbf{B}_j)/(\text{tr } \tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_j) =$$

$$(\text{tr } \mathbf{Q}_j\mathbf{W}_j\mathbf{S}_j'\mathbf{Y}^{*\prime}\mathbf{Y}^*\mathbf{S}_j\mathbf{W}_j\mathbf{Q}_j')/(\text{tr } \tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_j) = \text{tr } [\mathbf{W}_j/(\sqrt{\text{tr } \tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_j})]^2, \tag{3.68}$$

due to the fact that $\mathbf{Y}^*$ is columnwise orthonormal. In words, (3.68) expresses that, without missing data, the squared norm of the dimension weights $\mathbf{W}_j$ corrected for the size of the corresponding configuration $\tilde{\mathbf{X}}_j$ is equal to the fit of configuration j in the DIMIDIO solution. It also follows from (3.65) that

$$n = \sum_{j=1}^{n} \text{tr } \mathbf{B}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{B}_j + f(\mathbf{B},\mathbf{Y}), \tag{3.69}$$

with $f(\mathbf{B},\mathbf{Y})$ as in (3.39), and therefore that

$$0 \leq (1/n) \sum_{j=1}^{n} \text{tr } \mathbf{B}_j'\mathbf{Y}'\mathbf{C}_j\mathbf{Y}\mathbf{B}_j \leq 1,$$

is a measure of total fit in the DIMIDIO model.

For a further decomposition of the sums of squares in (3.69) with respect to configurations, stimuli, and dimensions we refer to Tables 3.1, 3.2, and 3.3, respectively, where in all tables one should now read $\mathbf{Y}^*\mathbf{S}_j$ instead of $\mathbf{Y}$.

The MATCHALS program computes and prints all decompositions discussed in section 3.5.

3.6 Illustrations

In this section several examples of the matching of configurations with the dimension weighting models are presented. The algorithms used for the analyses were all programmed in APL, and we used a convergence criterion of 1E-7 throughout. In section 3.6.1 the results of two analyses of a constructed data set with the DIMFREE model are discussed, and in section 3.6.2 the results are given of the DIMIDIO analysis of a constructed data set.

3.6.1 DIMFREE analysis of a constructed data set

The data discussed in this section were constructed in such a way that a perfect solution exists when the admissible transformations are those corresponding to the DIMFREE model (3.4). The 'mould' of this data set is a configuration whose coordinates are given in matrix **A** in Table 3.4. This matrix contains the coordinates of

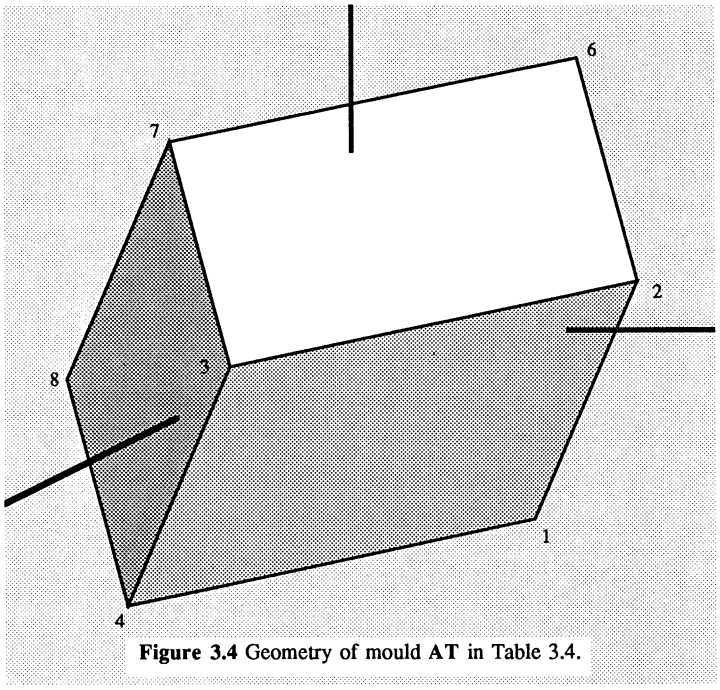


Figure 3.4 Geometry of mould **AT** in Table 3.4.

Table 3.4 Matrices, vectors and central dilations used in the construction of data in Table 3.5.

$$\mathbf{A} = \begin{bmatrix} -0.5 & -0.5 & -0.5 \\ 0.5 & -0.5 & -0.5 \\ 0.5 & 0.5 & -0.5 \\ -0.5 & 0.5 & -0.5 \\ -0.5 & -0.5 & 0.5 \\ 0.5 & -0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ -0.5 & 0.5 & 0.5 \end{bmatrix} \qquad \mathbf{T} = \begin{bmatrix} 0.5000 & 0.1464 & 0.8536 \\ -0.5000 & 0.8536 & 0.1464 \\ -0.7071 & -0.5000 & 0.5000 \end{bmatrix}$$

$$\mathbf{W}_1 = \begin{bmatrix} 4.0 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 2.0 \end{bmatrix} \quad \mathbf{u}_1 = \begin{bmatrix} 2.0 \\ 0.5 \\ 0.8 \end{bmatrix} \quad \mathbf{R}_1 = \begin{bmatrix} 0.3299 & 0.8949 & -0.3004 \\ -0.0501 & 0.3343 & 0.9411 \\ 0.9427 & -0.2955 & 0.1552 \end{bmatrix}$$

$s_1 = 3.0$

$$\mathbf{W}_2 = \begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 2.5 & 0 \\ 0 & 0 & 1.0 \end{bmatrix} \quad \mathbf{u}_2 = \begin{bmatrix} -2.0 \\ -0.4 \\ -1.5 \end{bmatrix} \quad \mathbf{R}_2 = \begin{bmatrix} -0.1065 & 0.1104 & 0.9882 \\ 0.5038 & 0.8628 & -0.0421 \\ 0.8572 & -0.4934 & 0.1475 \end{bmatrix}$$

$s_2 = 0.5$

$$\mathbf{W}_3 = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 3.0 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} \quad \mathbf{u}_3 = \begin{bmatrix} 0.2 \\ -1.0 \\ 3.0 \end{bmatrix} \quad \mathbf{R}_3 = \begin{bmatrix} 0.3231 & 0.9355 & 0.1429 \\ 0.7710 & -0.3477 & 0.5336 \\ -0.5489 & 0.0623 & 0.8336 \end{bmatrix}$$

$s_3 = 1.0$

$$\mathbf{W}_4 = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 4.5 \end{bmatrix} \quad \mathbf{u}_4 = \begin{bmatrix} 4.0 \\ -2.0 \\ 0.0 \end{bmatrix} \quad \mathbf{R}_4 = \begin{bmatrix} 0.9810 & 0.0457 & -0.1876 \\ 0.1906 & -0.3912 & 0.9004 \\ 0.0323 & 0.9193 & 0.3923 \end{bmatrix}$$

$s_4 = 0.8$

the vertices of a cube with edges parallel to the three coordinate axes. As a first step in the construction of artificial data, matrix **A** was rotated by the orthonormal matrix **T** given in Table 3.4. The position in space of the rotated cube **AT** is illustrated in Figure 3.4. The horizontal axis in this figure represents the first dimension, while the

Table 3.5 Constructed data for analysis with DIMFREE model.

$$\mathbf{X}_1 = \begin{bmatrix} -6.9903 & -0.1608 & -2.3074 \\ -0.1939 & 3.7690 & -3.1081 \\ -1.4094 & -1.4322 & 0.0353 \\ -8.2058 & -5.3621 & 0.8360 \\ -6.9244 & -8.8918 & 0.0008 \\ -0.1280 & -4.9619 & -0.7999 \\ -1.3436 & -10.1632 & 2.3435 \\ -8.1400 & -14.0931 & 3.1442 \end{bmatrix} \quad \mathbf{X}_2 = \begin{bmatrix} 0.1432 & -0.1561 & 1.1880 \\ 0.5800 & -0.1867 & 1.4408 \\ 1.2016 & 0.6756 & 1.2091 \\ 0.7649 & 0.7062 & 0.9562 \\ 0.0728 & -0.8499 & 0.9716 \\ 0.5095 & -0.8805 & 1.2245 \\ 1.1312 & -0.0181 & 0.9928 \\ 0.6944 & 0.0124 & 0.7399 \end{bmatrix}$$

$$\mathbf{X}_3 = \begin{bmatrix} 2.0284 & 0.0259 & -2.4452 \\ -2.5157 & 0.5854 & -1.9613 \\ 4.2315 & -1.0048 & -0.6777 \\ 3.7441 & -1.5643 & -1.1616 \\ 0.4744 & -0.4385 & -3.3138 \\ 0.9617 & 0.1210 & -2.8299 \\ 2.6775 & -1.4693 & -1.5463 \\ 2.1901 & -2.0288 & -2.0301 \end{bmatrix} \quad \mathbf{X}_4 = \begin{bmatrix} -2.8487 & -3.1709 & 0.7805 \\ -2.5702 & -0.3845 & 2.0615 \\ -2.5801 & -0.1743 & 2.9133 \\ -2.8585 & -2.9606 & 1.6322 \\ -3.0888 & -1.3700 & 1.1690 \\ -2.8104 & 1.4164 & 2.4500 \\ -2.8202 & 1.6267 & 3.3017 \\ -3.0986 & -1.1597 & 2.0207 \end{bmatrix}$$

vertical axis is the third dimension. Four configurations were constructed from this mould by differentially weighting the dimensions of **AT**. Then, as a final step (to make the confusion complete, as it were), the four identically rotated but differentially weighted configurations were again idiosyncratically translated, rotated and centrally dilated. Table 3.4 contains all transformation parameters that we used to construct this data set. By calculating $\mathbf{X}_j = s_j(\mathbf{ATW}_j - \mathbf{1u}_j')\mathbf{R}_j$ for $j = 1, ..., 4$ we obtained the configurations given in Table 3.5.

The four complete configurations in Table 3.5 were first submitted to a generalized Procrustes analysis. The corresponding algorithm converged in five iterations, yielding a loss of 1.2928. Since the configurations are always normalized on $\sum \mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j = n$, this means that the GPA model accounts for (100 x (4 – 1.2928)/4) = 68% of the sum of squares of the four configurations.

Thereafter, the optimally rotated and rescaled configurations $s_j\mathbf{X}_j\mathbf{R}_j\mathbf{K}$ rotated to the principal components of the GPA solution, and the optimal centroid configuration **ZK**, were used as input for the DIMFREE algorithm described in section 3.3.8. This algorithm converged in 3 iterations yielding a loss of 0.0000. As expected, the DIMFREE model accounts for all of the total variation in the four configurations. The results of the DIMFREE analysis are given in Table 3.6.

Table 3.6 Results of DIMFREE analysis of data in Table 3.5.

History of iterations — TOTAL LOSS

Iteration number	new $\mathbf{W}_j$	new $\mathbf{Q}_j$	new $\mathbf{W}_j$	new $\mathbf{Y}$
1	0.00000139	0.00000092	0.00000092	0.00000038
2	0.00000038	0.00000016	0.00000016	0.00000006
3	0.00000006	0.00000003	0.00000003	0.00000001

$$\mathbf{Y} = \begin{bmatrix} -0.0993 & 0.2260 & 0.1060 \\ -0.0411 & -0.0312 & 0.2559 \\ 0.2978 & -0.0753 & 0.1060 \\ 0.2397 & 0.1819 & -0.0439 \\ -0.2978 & 0.0753 & -0.1060 \\ -0.2397 & -0.1819 & 0.0439 \\ 0.0993 & -0.2260 & -0.1060 \\ 0.0411 & 0.0312 & -0.2559 \end{bmatrix} \qquad \mathbf{Y}^* = \begin{bmatrix} -0.1768 & 0.5303 & 0.2500 \\ -0.0732 & -0.0733 & 0.6035 \\ 0.5303 & -0.1768 & 0.2500 \\ 0.4268 & 0.4268 & -0.1035 \\ -0.5303 & 0.1768 & -0.2500 \\ -0.4268 & -0.4268 & 0.1035 \\ 0.1768 & -0.5303 & -0.2500 \\ 0.0732 & 0.0733 & -0.6035 \end{bmatrix}$$

$$\mathbf{Q}_1 = \begin{bmatrix} 1.0000 & 0.0001 & 0.0000 \\ -0.0001 & 1.0000 & -0.0008 \\ 0.0000 & 0.0008 & 1.0000 \end{bmatrix} \qquad \mathbf{Q}_2 = \begin{bmatrix} 1.0000 & 0.0001 & 0.0000 \\ -0.0001 & 1.0000 & -0.0007 \\ 0.0000 & 0.0007 & 1.0000 \end{bmatrix}$$

$$\mathbf{Q}_3 = \begin{bmatrix} 1.0000 & 0.0001 & 0.0000 \\ -0.0001 & 1.0000 & -0.0008 \\ 0.0000 & 0.0008 & 1.0000 \end{bmatrix} \qquad \mathbf{Q}_4 = \begin{bmatrix} 1.0000 & 0.0001 & 0.0000 \\ -0.0001 & 1.0000 & -0.0008 \\ 0.0000 & 0.0008 & 1.0000 \end{bmatrix}$$

	raw dimension weights $\mathbf{W}_j$				unique dimension weights $\mathbf{W}_j^*$			
	configuration				configuration			
dimension	1	2	3	4	1	2	3	4
1	0.1838	1.8085	1.6821	0.3255	0.1032	1.0156	0.9446	0.1828
2	0.9688	0.9532	0.1478	1.9303	0.4129	0.4062	0.0630	0.8226
3	1.9473	0.7664	1.1138	0.1724	0.8258	0.3250	0.4723	0.0731

In Figure 3.5 a plot is shown of the optimal centroid configuration both in its raw and in its uniquely weighted form. The horizontal axis represents the first dimension, while the vertical axis is the third dimension. Comparing these plots with the mould in

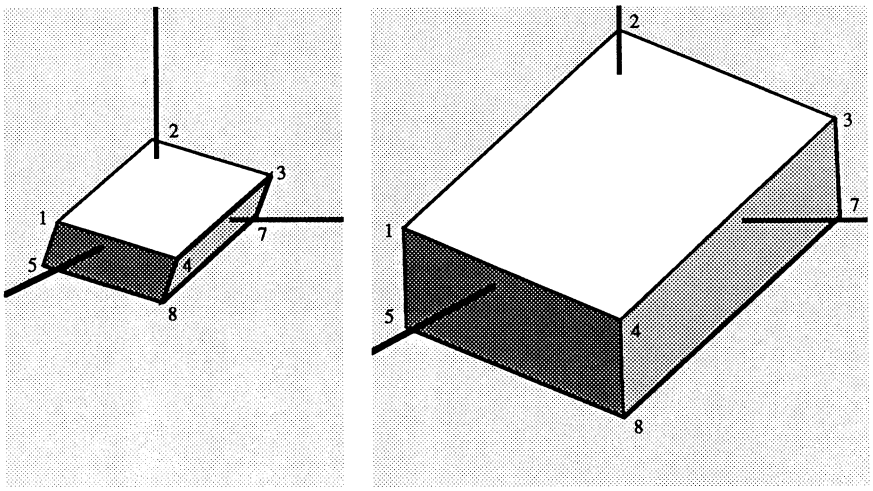


Figure 3.5 Plot of raw centroid configuration **Y** (left), and of uniquely weighted centroid configuration $\mathbf{Y}^*$ (right) from Table 3.6.

Figure 3.4, it becomes clear that we are now dealing with a *permuted* version of the original cube. Specifically, the first dimension in the original in Figure 3.4 has become the third dimension in Figure 3.5, the second dimension has become the first dimension, and the original third dimension now emerges as the second dimension.

Comparing the two plots in Figure 3.5, one notices that the uniquely weighted $\mathbf{Y}^*$ is a blown-up version of the raw **Y**. As can be seen at the bottom of Table 3.6 the raw dimension weights $\mathbf{W}_j$ are accordingly being scaled down to $\mathbf{W}_j^*$ for each j.

Moreover, the coordinates of the stimulus points in $\mathbf{Y}^*$ form the vertices of a cube, as is easily verified by calculating the distances between the stimulus points in Figure 3.5. On the other hand, in the raw centroid configuration **Y**, the stimulus points are the vertices of a parallelepiped, of which all faces are parallelograms, and of which the edges are therefore not orthogonal.

As a logical consequence of the fact that $\mathbf{Y}^*$ is a cube, for each configuration j the ratio of the normalized dimension weights in Table 3.6 is equal to the ratio of the original weights in Table 3.4. This does not hold for the raw dimension weights in Table 3.6. Moreover, if the unique weights $\mathbf{W}_j^*$ in the table are divided by $\sqrt{\text{tr}\,\tilde{\mathbf{X}}'_j\tilde{\mathbf{X}}_j}$ for each j, the sum of the resulting squared weights for each j is equal to one. This follows from the fact that, given n complete configurations, the sum of squared

dimension weights $\mathbf{W}_j^*$ corrected for the size of configuration $\tilde{\mathbf{X}}_j$ is equal to the fit of that configuration in the DIMFREE solution (cf. section 3.5.2). Since we have a perfect solution in this example, the fit is equal to one for all four configurations.

In the second part of this section we show the results of an analysis of configurations containing missing data with the DIMFREE model. The four configurations given in Table 3.5 were analyzed treating the following rows as

Table 3.8 History of iterations of DIMFREE algorithm for incomplete configurations in Table 3.5.

	TOTAL LOSS			
Iteration number	new $\mathbf{W}_j$	new $\mathbf{Q}_j$	new $\mathbf{W}_j$	new $\mathbf{Y}$
1	0.33204141	0.29020618	0.28530994	0.17039250
2	0.15547808	0.11105605	0.10885688	0.07072065
3	0.06736023	0.04530308	0.04461105	0.02856942
4	0.02698703	0.01854676	0.01831891	0.01211514
5	0.01135827	0.00818357	0.00808814	0.00562069
6	0.00527978	0.00397960	0.00393580	0.00287635
7	0.00272453	0.00213260	0.00211047	0.00161267
8	0.00154467	0.00124385	0.00123077	0.00097487
9	0.00094396	0.00077505	0.00076603	0.00062339
10	0.00060895	0.00050636	0.00049955	0.00041453
11	0.00040747	0.00034160	0.00033630	0.00028293
12	0.00027926	0.00023537	0.00023125	0.00019645
13	0.00019441	0.00016446	0.00016128	0.00013798
14	0.00013675	0.00011600	0.00011357	0.00009766
15	0.00009689	0.00008235	0.00008052	0.00006951
16	0.00006899	0.00005874	0.00005737	0.00004967
17	0.00004932	0.00004205	0.00004102	0.00003561
18	0.00003536	0.00003018	0.00002942	0.00002559
19	0.00002541	0.00002171	0.00002115	0.00001842
20	0.00001829	0.00001564	0.00001523	0.00001328
21	0.00001319	0.00001129	0.00001098	0.00000959
22	0.00000952	0.00000815	0.00000793	0.00000693
23	0.00000688	0.00000590	0.00000574	0.00000502
24	0.00000498	0.00000427	0.00000415	0.00000363
25	0.00000361	0.00000309	0.00000301	0.00000263
26	0.00000261	0.00000224	0.00000218	0.00000191
27	0.00000189	0.00000162	0.00000158	0.00000138
28	0.00000137	0.00000118	0.00000115	0.00000100
29	0.00000100	0.00000086	0.00000083	0.00000073
30	0.00000072	0.00000062	0.00000060	0.00000053
31	0.00000053	0.00000045	0.00000044	0.00000038
32	0.00000038	0.00000033	0.00000032	0.00000028
33	0.00000028	0.00000024	0.00000023	0.00000020

Table 3.9 Results of DIMFREE analysis of data set in Table 3.5 containing missing data.

$$\mathbf{Y} = \begin{bmatrix} 0.5464 & 0.1470 & -0.0717 \\ -0.0755 & 0.3550 & -0.0294 \\ -0.1821 & 0.1470 & 0.2127 \\ 0.4398 & -0.0609 & 0.1711 \\ 0.1820 & -0.1470 & -0.2128 \\ -0.4398 & 0.0609 & -0.1704 \\ -0.5463 & -0.1470 & 0.0709 \\ 0.0755 & -0.3550 & 0.0296 \end{bmatrix} \qquad \mathbf{Y}^* = \begin{bmatrix} 0.5303 & 0.2500 & -0.1787 \\ -0.0732 & 0.6036 & -0.0733 \\ -0.1767 & 0.2500 & 0.5306 \\ 0.4269 & -0.1035 & 0.4267 \\ 0.1766 & -0.2500 & -0.5308 \\ -0.4269 & 0.1035 & -0.4250 \\ -0.5303 & -0.2500 & 0.1768 \\ 0.0733 & -0.6035 & 0.0738 \end{bmatrix}$$

$$\mathbf{Q}_1 = \begin{bmatrix} 0.8779 & -0.3944 & 0.2717 \\ 0.3907 & 0.9178 & 0.0698 \\ -0.2769 & 0.0449 & 0.9599 \end{bmatrix} \qquad \mathbf{Q}_2 = \begin{bmatrix} 0.7129 & -0.4162 & 0.5643 \\ 0.5037 & 0.8639 & 0.0008 \\ -0.4879 & 0.2837 & 0.8255 \end{bmatrix}$$

$$\mathbf{Q}_3 = \begin{bmatrix} 0.9602 & -0.2772 & -0.0342 \\ 0.2728 & 0.9570 & -0.0987 \\ 0.0601 & 0.0854 & 0.9945 \end{bmatrix} \qquad \mathbf{Q}_4 = \begin{bmatrix} 0.9504 & -0.2163 & 0.2237 \\ 0.2583 & 0.9493 & -0.1795 \\ -0.1735 & 0.2283 & 0.9580 \end{bmatrix}$$

	raw dimension weights $\mathbf{W}_j$				unique dimension weights $\mathbf{W}^*_j$			
	configuration				configuration			
dimension	1	2	3	4	1	2	3	4
1	0.4827	0.3361	1.3530	1.3169	0.4974	0.3463	1.3940	1.3568
2	1.6913	0.4710	1.1843	0.2050	0.9947	0.2770	0.6965	0.1206
3	0.3101	2.1582	0.1757	0.7507	0.1243	0.8653	0.0704	0.3010

missing: rows 1, 3, 6 and 8 of configuration 2, rows 1, 3, 5 and 7 of configuration 3, and rows 3 and 7 of configuration 4.

A generalized Procrustes analysis of these four configurations containing missing data yielded a loss of 0.8280, meaning that the GPA model now accounts for (100 x (4 – 0.8280)/4) = 79% of the total sum of squares of the four configurations.

Submitting the optimally rotated and rescaled configurations from GPA to the algorithm for the DIMFREE model, we again obtained the expected results. The algorithm converged in 33 iterations, yielding a perfect solution. The history of iterations for this data set is shown in Table 3.8, and the optimal transformation parameters of the DIMFREE solution are given in Table 3.9.

Within rounding errors, and up to permutations of the three dimensions, the uniquely weighted centroid configuration $\mathbf{Y}^*$ in Table 3.9 is equal to $\mathbf{Y}^*$ in Table 3.6. This shows that the DIMFREE algorithm also recovers the original mould **AT** of this artificial data set if missing data are introduced.

Due to the fact that we are dealing with incomplete configurations in this example, the sum of the squared dimension weights $\mathbf{W}_j^*$ shown in Table 3.9 is no longer related to the fit of each configuration $\tilde{\mathbf{X}}_j$ in the DIMFREE solution.

3.6.2 DIMIDIO analysis of a constructed data set

The data discussed in this section were again constructed such that a perfect solution must exist when the admissible transformations are those corresponding to dimension weighting model (3.5). We used the mould and transformation parameters given in Table 3.4 to construct four configurations by calculating $\mathbf{X}_j = s_j(\mathbf{A}\mathbf{R}_j\mathbf{W}_j - \mathbf{1}\mathbf{u}_j')$ for j = 1, ..., 4. Thus, the coordinates of the vertices of the cube **A** were first idiosyncratically rotated by $\mathbf{R}_j$, then differentially weighted by $\mathbf{W}_j$, and finally idiosyncratically translated and centrally dilated. This resulted in the data given in Table 3.10.

A generalized Procrustes analysis of these four complete configurations yielded a solution in six iterations, with a loss of 0.8335. This implies that the GPA model accounts for 79% of the total variation in the configurations in Table 3.10.

Submitting the results from GPA to the DIMFREE algorithm of section 3.3.8, it was found that this algorithm converged in 58 iterations, giving a loss of 0.1784. Thus, the DIMFREE model yielded a solution which accounts for 96% of the total variation, 17% more than the GPA model.

Table 3.10 Constructed data for analysis with DIMIDIO model.

$$\mathbf{X}_1 = \begin{bmatrix} -12.9338 & -0.8184 & -5.1471 \\ -1.3868 & -0.4520 & -4.4265 \\ -2.2232 & -0.7601 & 1.4307 \\ -13.7702 & -1.1264 & 0.7101 \\ -9.7768 & -2.2399 & -6.2307 \\ 1.7702 & -1.8736 & -5.5101 \\ 0.3436 & -2.1816 & 0.3471 \\ -10.6132 & -2.5480 & -0.3735 \end{bmatrix} \quad \mathbf{X}_2 = \begin{bmatrix} 0.6774 & 0.4417 & 0.8746 \\ 0.9783 & 0.1422 & 1.1815 \\ 1.2218 & 0.9034 & 0.9273 \\ 0.9209 & 1.2029 & 0.6204 \\ 0.7782 & -0.5034 & 0.5727 \\ 1.0791 & -0.8029 & 0.8796 \\ 1.3226 & -0.0417 & 0.6254 \\ 1.0217 & 0.2578 & 0.3185 \end{bmatrix}$$

$$\mathbf{X}_3 = \begin{bmatrix} -0.1564 & -2.9912 & -2.9643 \\ -0.5420 & 0.9484 & -2.9883 \\ 2.0971 & 1.5866 & -2.7842 \\ 2.4826 & -2.3530 & -2.7602 \\ -2.4971 & 0.4134 & -3.2158 \\ -2.8826 & 4.3530 & -3.2398 \\ -0.2436 & 4.9912 & -3.0357 \\ 0.1420 & 1.0516 & -3.0117 \end{bmatrix} \quad \mathbf{X}_4 = \begin{bmatrix} -3.1684 & -0.8547 & -3.3878 \\ -3.1241 & 1.8452 & 3.6610 \\ -3.0944 & 3.7627 & 4.3163 \\ -3.1387 & 1.0628 & -2.7325 \\ -3.3056 & -0.5627 & -4.3163 \\ -3.2613 & 2.1372 & 2.7325 \\ -3.2316 & 4.0547 & 3.3878 \\ -3.2759 & 1.3548 & -3.6610 \end{bmatrix}$$

Table 3.11 Results of DIMIDIO analysis of complete data set in Table 3.10.

$$\mathbf{Y} = \begin{bmatrix} -0.3252 & -0.1183 & -0.0534 \\ 0.1595 & -0.0335 & -0.0883 \\ 0.1088 & 0.2432 & -0.0368 \\ -0.3760 & 0.1584 & -0.0019 \\ -0.1088 & -0.2432 & 0.0368 \\ 0.3760 & -0.1584 & 0.0019 \\ 0.3252 & 0.1183 & 0.0534 \\ -0.1595 & 0.0335 & 0.0883 \end{bmatrix} \qquad \mathbf{Y}^* = \begin{bmatrix} -0.4312 & -0.2653 & -0.3445 \\ 0.2115 & -0.0751 & -0.5697 \\ 0.1442 & 0.5456 & -0.2378 \\ -0.4985 & 0.3554 & -0.0126 \\ -0.1442 & -0.5456 & 0.2378 \\ 0.4985 & -0.3554 & 0.0126 \\ 0.4312 & 0.2653 & 0.3445 \\ -0.2115 & 0.0751 & 0.5697 \end{bmatrix}$$

$$\mathbf{S}_1 = \begin{bmatrix} 0.9881 & -0.0571 & 0.1431 \\ 0.0933 & 0.9608 & -0.2610 \\ -0.1226 & 0.2713 & 0.9547 \end{bmatrix} \qquad \mathbf{S}_2 = \begin{bmatrix} -0.5827 & -0.3612 & 0.7280 \\ 0.7698 & 0.0418 & 0.6369 \\ -0.2605 & 0.9315 & 0.2537 \end{bmatrix}$$

$$\mathbf{S}_3 = \begin{bmatrix} 0.9208 & 0.1667 & -0.3527 \\ -0.1797 & 0.9837 & -0.0043 \\ 0.3462 & 0.0674 & 0.9357 \end{bmatrix} \qquad \mathbf{S}_4 = \begin{bmatrix} 0.8348 & -0.0475 & 0.5485 \\ 0.4548 & 0.6210 & -0.6384 \\ -0.3103 & 0.7824 & 0.5400 \end{bmatrix}$$

$$\mathbf{Q}_1 = \begin{bmatrix} 0.9969 & -0.0615 & 0.0493 \\ 0.0683 & 0.9864 & -0.1496 \\ -0.0394 & 0.1525 & 0.9875 \end{bmatrix} \qquad \mathbf{Q}_2 = \begin{bmatrix} -0.6873 & -0.3116 & 0.6561 \\ 0.7074 & -0.0819 & 0.7021 \\ -0.1651 & 0.9467 & 0.2767 \end{bmatrix}$$

$$\mathbf{Q}_3 = \begin{bmatrix} 0.9844 & 0.1567 & -0.0803 \\ -0.1574 & 0.9875 & -0.0022 \\ 0.0790 & 0.0148 & 0.9968 \end{bmatrix} \qquad \mathbf{Q}_4 = \begin{bmatrix} 0.9506 & -0.2234 & 0.2157 \\ 0.2986 & 0.8482 & -0.4376 \\ -0.0852 & 0.4803 & 0.8729 \end{bmatrix}$$

	dimension weights $\mathbf{W}_j$				dimension weights $\mathbf{W}^*_j = \mathbf{W}_j/(\sqrt{\text{tr}\ \tilde{\mathbf{X}}_j\tilde{\mathbf{X}}_j})$			
	configuration				configuration			
dimension	1	2	3	4	1	2	3	4
1	0.9815	0.8226	0.9226	0.8844	0.8889	0.8900	0.8766	0.9775
2	0.4908	0.3290	0.5065	0.1903	0.4444	0.3560	0.4812	0.2103
3	0.1227	0.2632	0.0079	0.0149	0.1111	0.2848	0.0075	0.0165
				$\Sigma\, w^{*2}_{jk}$	1.0000	1.0000	1.0000	1.0000

Thereafter, submitting the optimally rotated and rescaled configurations $s_j\mathbf{X}_j\mathbf{R}_j\mathbf{K}$ and the optimal centroid configuration $\mathbf{ZK}$ of the GPA solution to the DIMIDIO algorithm of section 3.4.7, this algorithm immediately reached a perfect solution in the first iteration. The results of this analysis are given in Table 3.11. As expected, the DIMIDIO model accounts for all the variation in this constructed data set. Of course, in practice one would choose for the results of the DIMFREE model, since this model is much more parsimonious and accounts for almost the same percentage of total variation as the DIMIDIO model. But our main purpose here is to investigate whether the DIMIDIO algorithm finds a perfect solution when it exists.

At the bottom of Table 3.11 is shown that, since all configurations are complete in this example, the sum of squared dimension weights corrected for the size of configuration $\tilde{\mathbf{X}}_j$ is equal to the fit of each configuration. Because we have a perfect solution, the fit is equal to one for all four configurations.

Introducing the same missing data in the configurations of Table 3.10 as in section 3.6.1, that is, treating stimuli 1, 3, 6 and 8 of configuration 2, stimuli 1, 3, 5 and 7 of configuration 3, and stimuli 3 and 7 of configuration 4 as missing, the GPA model yielded a loss of 0.6770. Thus, this model accounts for 83% of the variation in these incomplete configurations.

Submitting the optimally rotated and rescaled configurations $s_j\mathbf{X}_j\mathbf{R}_j\mathbf{K}$ and $\mathbf{ZK}$ from GPA to the DIMFREE algorithm, it was found that this algorithm took 108 iterations to converge. At convergence the loss was 0.0773, meaning that the DIMFREE model accounts for 98% of the total variation.

The DIMIDIO algorithm of section 3.4.7 again converged to a perfect solution, in 48 iterations. The history of iterations for this data set is shown in Table 3.12, and the results of the analysis are given in Table 3.13. In Figure 3.6 we have plotted the coordinates of the optimal centroid configuration $\mathbf{Y}^*$ in Table 3.11 (i.e., after DIMIDIO analysis of the four complete configurations) together with the coordinates of $\mathbf{Y}^*$ in Table 3.13 (i.e., after DIMIDIO analysis of the four incomplete configurations). For the example discussed in the previous section about DIMFREE we saw that the optimal uniquely weighted centroid configurations $\mathbf{Y}^*$ for complete and incomplete data were identical up to permutations and reflections of the columns of the two $\mathbf{Y}^*$'s. In DIMIDIO the centroid configuration $\mathbf{Y}^*$ given in Table 3.13 is a

Table 3.12 History of iterations of DIMIDIO analysis of incomplete data set in Table 3.10.

Iteration number	TOTAL LOSS new $\mathbf{B}_j$	TOTAL LOSS new $\mathbf{Y}$	Iteration number	TOTAL LOSS new $\mathbf{B}_j$	TOTAL LOSS new $\mathbf{Y}$
1	0.04255481	0.01220214	25	0.00004515	0.00004023
2	0.00746972	0.00580469	26	0.00003586	0.00003204
3	0.00494796	0.00439940	27	0.00002864	0.00002571
4	0.00400550	0.00368420	28	0.00002310	0.00002090
5	0.00340545	0.00314840	29	0.00001897	0.00001670
6	0.00290998	0.00268247	30	0.00001495	0.00001319
7	0.00246822	0.00226339	31	0.00001167	0.00001049
8	0.00207090	0.00188833	32	0.00000952	0.00000866
9	0.00171806	0.00155823	33	0.00000790	0.00000721
10	0.00141051	0.00127326	34	0.00000659	0.00000602
11	0.00114752	0.00103174	35	0.00000551	0.00000504
12	0.00092649	0.00083032	36	0.00000461	0.00000421
13	0.00074344	0.00066458	37	0.00000385	0.00000352
14	0.00059369	0.00052967	38	0.00000322	0.00000295
15	0.00047236	0.00042080	39	0.00000270	0.00000247
16	0.00037479	0.00033352	40	0.00000226	0.00000206
17	0.00029678	0.00026390	41	0.00000189	0.00000173
18	0.00023468	0.00020858	42	0.00000158	0.00000145
19	0.00018540	0.00016474	43	0.00000132	0.00000121
20	0.00014640	0.00013006	44	0.00000111	0.00000101
21	0.00011557	0.00010269	45	0.00000093	0.00000085
22	0.00009125	0.00008110	46	0.00000078	0.00000071
23	0.00007209	0.00006409	47	0.00000065	0.00000059
24	0.00005700	0.00005072	48	0.00000054	0.00000050

Table 3.13 Results of DIMIDIO analysis of incomplete data set in Table 3.10.

$$\mathbf{Y} = \begin{bmatrix} -0.4103 & -0.1861 & -0.0702 \\ 0.2804 & -0.0124 & -0.0940 \\ 0.2190 & 0.3092 & -0.0545 \\ -0.4716 & 0.1361 & -0.0313 \\ -0.2190 & -0.3094 & 0.0547 \\ 0.4716 & -0.1357 & 0.0308 \\ 0.4102 & 0.1854 & 0.0710 \\ -0.2804 & 0.0129 & 0.0935 \end{bmatrix} \qquad \mathbf{Y}^* = \begin{bmatrix} -0.4256 & -0.1890 & -0.3963 \\ 0.2635 & -0.1166 & -0.5412 \\ 0.2662 & 0.5092 & -0.2117 \\ -0.4227 & 0.4379 & -0.0702 \\ -0.2662 & -0.5096 & 0.2126 \\ 0.4228 & -0.4370 & 0.0676 \\ 0.4254 & 0.1876 & 0.4006 \\ -0.2634 & 0.1175 & 0.5386 \end{bmatrix}$$

$$\mathbf{S}_1 = \begin{bmatrix} 0.9965 & 0.0802 & -0.0236 \\ -0.0836 & 0.9584 & -0.2728 \\ 0.0008 & 0.2738 & 0.9618 \end{bmatrix} \qquad \mathbf{S}_2 = \begin{bmatrix} -0.4009 & -0.4644 & 0.7897 \\ 0.8590 & 0.1091 & 0.5003 \\ -0.3185 & 0.8789 & 0.3552 \end{bmatrix}$$

$$\mathbf{S}_3 = \begin{bmatrix} 0.4255 & -0.8985 & 0.1074 \\ -0.8347 & -0.3439 & 0.4301 \\ 0.3495 & 0.2727 & 0.8964 \end{bmatrix} \qquad \mathbf{S}_4 = \begin{bmatrix} 0.9336 & -0.0321 & 0.3569 \\ 0.2968 & 0.6274 & -0.7199 \\ -0.2008 & 0.7780 & 0.5953 \end{bmatrix}$$

$$\mathbf{Q}_1 = \begin{bmatrix} 0.9864 & -0.1639 & -0.0133 \\ 0.1600 & 0.9754 & -0.1518 \\ 0.0379 & 0.1476 & 0.9883 \end{bmatrix} \qquad \mathbf{Q}_2 = \begin{bmatrix} -0.6341 & -0.5391 & 0.5543 \\ 0.6989 & -0.0930 & 0.7091 \\ -0.3307 & 0.8371 & 0.4357 \end{bmatrix}$$

$$\mathbf{Q}_3 = \begin{bmatrix} 0.9643 & -0.1556 & 0.2144 \\ -0.2455 & -0.2210 & 0.9439 \\ 0.0994 & 0.9628 & 0.2513 \end{bmatrix} \qquad \mathbf{Q}_4 = \begin{bmatrix} 0.9699 & -0.1969 & 0.1433 \\ 0.2240 & 0.9523 & -0.2074 \\ -0.0956 & 0.2333 & 0.9677 \end{bmatrix}$$

dimension weights
$\mathbf{W}_j$

	configuration			
dimension	1	2	3	4
1	1.1701	0.8825	1.0959	1.2946
2	0.5851	0.3528	0.1268	0.2795
3	0.1463	0.2827	0.0000	0.0219

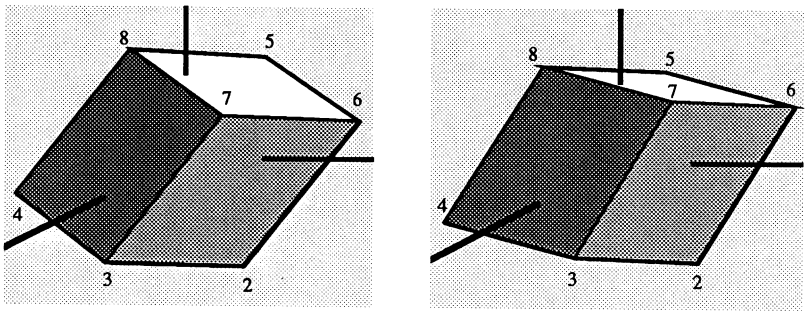
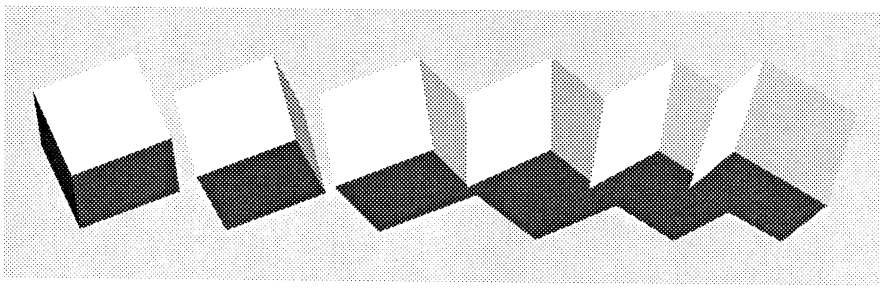


Figure 3.6 Optimal $\mathbf{Y}^*$ after DIMIDIO analysis of complete data set (left), and of incomplete data set (right).

rotated version of the $\mathbf{Y}^*$ given in Table 3.11. This is an immediate consequence of the fact that $\mathbf{Y}^*$ in DIMIDIO is only determined up to a rotation (see section 3.4.6).

We finally note that the sum of squared dimension weights in Table 3.13 is no longer related to the fit of the configurations in the DIMIDIO solution, due to the fact that we are now dealing with incomplete configurations.



Chapter 4

The stimulus weighting models

4.1 Introduction

In this chapter we return to the situation at the end of Chapter 2, and from there we start off into a completely different direction. Instead of matching n configurations by means of differential weighting of dimensions, we now investigate the effect of differentially weighting the p *stimulus points* of each configuration. An illustration is given in Figure 4.1 where stimulus weighting is applied to a two-dimensional configuration consisting of the vertices of a pentagon. The stimulus points in the figure are numbered from one to five before transformation and from one prime to five prime after transformation. For each stimulus point of the configuration a line is generated by connecting the stimulus point with the origin of m-dimensional space, and this line is then allowed to be shortened or lengthened. In other words, the stimulus points are conceived of as vectors emanating from the origin O, and weighting these vectors has the effect of moving the stimulus points either closer to or further away from the origin.

If we would translate the pentagon in Figure 4.1 to a different location and then connect the stimulus points with the origin, this would yield a different set of vectors

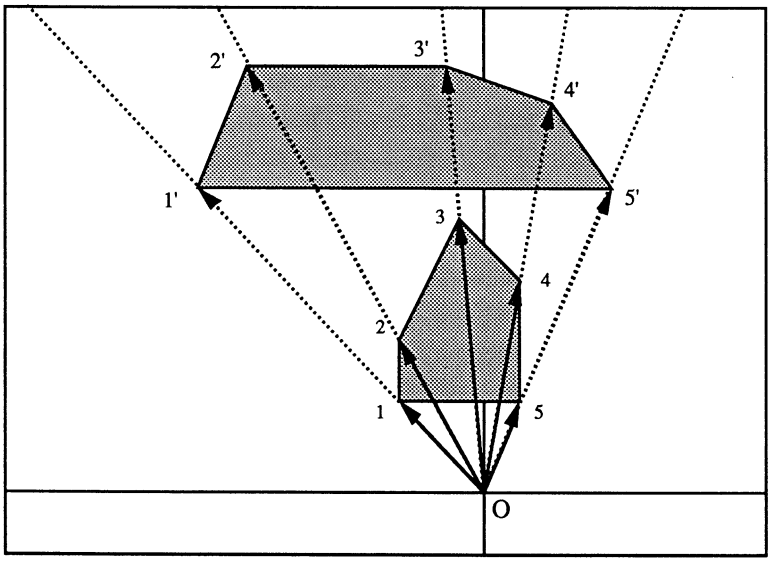


Figure 4.1 Illustration of stimulus weighting.

than the one shown in the figure. Since the weighting of the latter vectors would also result in a different set of configurations, the location of a configuration is clearly of crucial importance in stimulus weighting.

This chapter deals with the matching of n configurations by means of translations, orthonormal transformations, and stimulus weighting. Lingoes and Borg (1978) showed that these three transformations result in two different stimulus weighting models. In the first model only one optimal translation is determined for all configurations, while the second model allows for a different optimal translation for each separate configuration. In the following sections we generalize the stimulus weighting models to m dimensions, as well as to the case of missing data. We also propose a new approach to the fitting of these models, which, just like in the dimension weighting models, depends on determining a new centroid configuration instead of using the centroid configuration **Z** obtained in GPA. Before presenting our alternative approach, however, we first discuss the manner in which Lingoes and Borg constructed their models.

4.2 Geometry and algebra

Given n configurations $\mathbf{X}_j$ (j = 1, ..., n) of order (p x m), let **Z** of order (p x m) be the corresponding matrix of centroids obtained in GPA. Also, let $\mathbf{g}_j$ and $\mathbf{h}_j$ be unknown translation vectors of order (m x 1), $\mathbf{T}_j$ be an unknown orthonormal matrix of order (m x m), $\mathbf{V}_j$ be an unknown (p x p) diagonal matrix containing stimulus weights, and $\mathbf{E}_j$ be a (p x m) matrix of residuals. With these definitions, Lingoes and Borg proposed the following model:

$$(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j = \mathbf{V}_j(\mathbf{Z} - \mathbf{1}\mathbf{h}_j') + \mathbf{E}_j, \qquad \text{for } j = 1, ..., n, \qquad (4.1)$$

where the unknown $\mathbf{g}_j$, $\mathbf{h}_j$, $\mathbf{T}_j$, and $\mathbf{V}_j$ are to be estimated by minimizing the sum of squared residuals, that is, by minimizing $\sum \text{tr } \mathbf{E}_j'\mathbf{E}_j$. The diagonal matrix $\mathbf{V}_j$ in (4.1) has the effect of changing the lengths of the vectors generated by connecting the stimulus points in $(\mathbf{Z} - \mathbf{1}\mathbf{h}_j')$ with the origin. Characteristic of (4.1) is that the weighting of stimulus points is applied to $(\mathbf{Z} - \mathbf{1}\mathbf{h}_j')$, and not to the stimulus points of the individual configurations. In principle, one could also use a model like

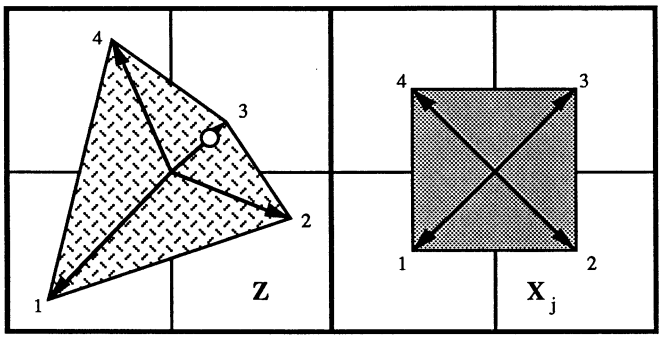


Figure 4.2 Illustration of necessity of translation of **Z** in the case of stimulus weighting.

$$\mathbf{V}_j(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j = (\mathbf{Z} - \mathbf{1}\mathbf{h}_j') + \mathbf{E}_j, \qquad \text{for } j = 1, \ldots, n, \tag{4.2}$$

where the stimulus points of the individual configurations are weighted. But we will not pursue this course, and stick to the stimulus weighting of the centroid configuration, as Lingoes and Borg did.

The translations $\mathbf{h}_j$ play an essential role in model (4.1). The necessity of an optimal translation of **Z** in the case that its stimulus points are weighted is illustrated with a simple example in Figures 4.2 and 4.3. Supposing that the coordinates of the stimulus points in **Z** and in $\mathbf{X}_j$ are represented by the end points of the vectors shown in Figure 4.2, it is impossible to obtain $\mathbf{X}_j$ by just lengthening or shortening the vectors representing the stimulus points contained in **Z**. Therefore, if only stimulus weights are allowed the match between **Z** and $\mathbf{X}_j$ will never be perfect. A perfect match is immediately obtained, however, by translating **Z** to the point (0.5, 0.4) and then weighting the four vectors representing the four stimuli with weights 0.5, 1.0, 5.0 and (1/1.2), respectively. The point (0.5, 0.4) is indicated by a white circle in Figure 4.2. In matrix notation we have

$$\mathbf{V}_j(\mathbf{Z} - \mathbf{1}\mathbf{h}_j') = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 5.0 & 0 \\ 0 & 0 & 0 & 1/1.2 \end{bmatrix} \left(\begin{bmatrix} -1.5 & -1.6 \\ 1.5 & -0.6 \\ 0.7 & 0.6 \\ -0.7 & 1.6 \end{bmatrix} - \begin{bmatrix} 0.5 & 0.4 \\ 0.5 & 0.4 \\ 0.5 & 0.4 \\ 0.5 & 0.4 \end{bmatrix} \right)$$

$$= \begin{bmatrix} -1.0 & -1.0 \\ 1.0 & -1.0 \\ 1.0 & 1.0 \\ -1.0 & 1.0 \end{bmatrix} = \mathbf{X}_j,$$

where $\mathbf{h}_j$ is the optimal translation vector, and $\mathbf{V}_j$ contains the optimal stimulus weights to achieve a perfect match between $\mathbf{Z}$ and $\mathbf{X}_j$. These transformations are also illustrated geometrically in Figure 4.3, where the translation of $\mathbf{Z}$ to the new location (0.5, 0.4) creates the possibility of applying stimulus weights that yield $\mathbf{X}_j$.

In stimulus weighting two translation options are available. The first option is to translate $\mathbf{Z}$ only once such that the match is simultaneously optimized over all n individual configurations. The matrix notation for this model is:

$$(\mathbf{X}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j = \mathbf{V}_j(\mathbf{Z} - \mathbf{1}\mathbf{h}') + \mathbf{E}_j \quad \text{for } j = 1, ..., n, \tag{4.3}$$

where $\mathbf{h}$ is an unknown translation vector of order (m x 1). Lingoes and Borg (1987) discussed that this model has a psychological interpretation only if the stimulus weights are non-negative. People (represented by the individual configurations) then share a common point of view (represented by the origin of the optimally translated centroid configuration ($\mathbf{Z} - \mathbf{1}\mathbf{h}'$)) from where they perceive the surrounding world (represented by the centroid configuration itself), but differ in the *centrality* they

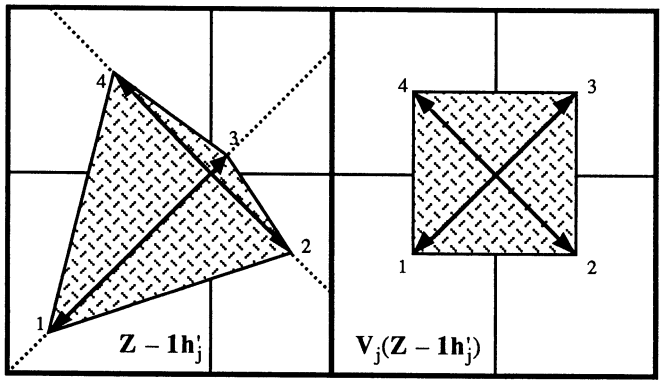


Figure 4.3 Geometry of optimal translation and stimulus weighting of $\mathbf{Z}$ in Figure 4.2.

assign to the stimuli under investigation. That is, some people experience a stimulus as being more peripheral, while it is a more central feature to other people.

If negative weights occur, however, such an interpretation breaks down. In that case a stimulus point is not only moved closer to or further away from the origin, but its associated vector is also made to point in the opposite direction. But even in this case the weights provide valuable information. Since the stimuli in **Z** are weighted from one and the same optimal origin **h** for all n configurations, the stimulus weights in (4.3) are directly comparable over configurations. Inspection of these weights can therefore give information about which specific stimuli are responsible for the possible increase in fit compared to the more parsimonious GPA model. Weights that considerably differ from +1 (that is, weights that are close to zero or even negative, and weights much larger than +1) indicate that the corresponding stimulus points in **Z** do not represent the stimuli in the individual configurations very well.

The alternative for model (4.3) is to translate **Z** optimally for each individual configuration separately. The algebra corresponding to this model has already been given in (4.1). Psychologically, this model implies that people not only differ in the centrality they assign to the stimuli under investigation, but also perceive the group configuration from different points of view, in contrast with the common point of view they share in model (4.3).

Characteristic of the stimulus weighting models of Lingoes and Borg is that they are fitted using the optimal centroid configuration **Z** obtained in GPA. But **Z** is not necessarily optimal in models (4.1) and (4.3). By sticking to **Z** a source of error is introduced which can obscure the real fit of these models. We therefore propose to determine *new* optimal centroid configurations instead of borrowing them from a previous model. This should eliminate the source of error that we just mentioned.

Let $\tilde{\mathbf{X}}_j$ denote the optimally transformed configuration $\mathbf{X}_j$ after GPA (i.e., $\tilde{\mathbf{X}}_j = s_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j\mathbf{K}$, see Chapter 2). Our alternative for model (4.3) is in matrix notation

$$(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j = \mathbf{V}_j\mathbf{Y} + \mathbf{E}_j \quad \text{for } j = 1, \ldots, n, \tag{4.4}$$

where the (p x m) centroid configuration **Y** is assumed to be unknown. It may be noted that the translation vector **h** is lacking in (4.4). This latter vector no longer needs to be estimated since the new matrix of centroids **Y** will automatically be in an optimal location with respect to stimulus weighting. We use the optimally transformed

configurations $\tilde{\mathbf{X}}_j$ instead of the raw $\mathbf{X}_j$'s in (4.4) because we expect that this results in more efficient algorithms, the $\tilde{\mathbf{X}}_j$'s already being optimal with respect to distance preserving transformations. Taking into account that information about some stimuli in some configurations may be missing, model (4.4) becomes

$$\mathbf{M}_j(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j = \mathbf{M}_j\mathbf{V}_j\mathbf{Y} + \mathbf{M}_j\mathbf{E}_j \quad \text{for } j = 1, ..., n, \tag{4.5}$$

where $\mathbf{M}_j$ is a given diagonal matrix of order (p x p) with ones on the diagonal if the corresponding rows in $\mathbf{X}_j$ are not missing, and zeroes elsewhere. In full we call (4.5) the *stimulus weighting model with free centroid configuration*, and will refer to it as the STIMFREE model. The STIMFREE model is developed in section 4.3 of this chapter.

Our alternative for Lingoes and Borg's model (4.1) is in matrix notation

$$(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j = \mathbf{V}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j') + \mathbf{E}_j \quad \text{for } j = 1, ..., n, \tag{4.6}$$

where $\mathbf{Y}$ is, again, an unknown centroid configuration. Generalizing (4.6) to the case of incomplete configurations yields the following model

$$\mathbf{M}_j(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j = \mathbf{M}_j\mathbf{V}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j') + \mathbf{M}_j\mathbf{E}_j \quad \text{for } j = 1, ..., n. \tag{4.7}$$

In full we call this model the *stimulus weighting model with idiosyncratic translations of a free centroid configuration*, and will refer to it as the STIMIDIO model. The STIMIDIO model is the subject of section 4.4.

As we noted earlier in this section non-negative stimulus weights are amenable to a psychological interpretation while negative weights are difficult to interpret. It is for this reason that we will discuss not only how to determine optimal stimulus weights in models (4.5) and (4.7) when the weights may take any value, but also how to determine weights subject to the condition that they must be non-negative.

The remainder of Chapter 4 is organized as follows. In sections 4.3 and 4.4 the stimulus weighting models (4.5) and (4.7) are developed and algorithms for the estimation of the unknown transformation parameters are presented. Section 4.5 deals with measures of fit and an analysis of variation for both models. In section 4.6 results of the analysis of some constructed data sets according to the STIMFREE and STIMIDIO model are shown.

4.3 The STIMFREE model

4.3.1 Introduction

In the following sections the unknown parameters minimizing the least squares loss function

$$f(\mathbf{G},\mathbf{T},\mathbf{V},\mathbf{Y}) = \sum_{j=1}^{n} \text{tr}\, [(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j - \mathbf{V}_j\mathbf{Y}]'\mathbf{M}_j[(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j - \mathbf{V}_j\mathbf{Y}], \qquad (4.8)$$

corresponding to the STIMFREE model (4.5) are determined. In (4.8) it is assumed that the unknown translation vectors $\mathbf{g}_j$ are collected in the (m x n) matrix $\mathbf{G}$, that the unknown orthonormal matrices $\mathbf{T}_j$ are collected in the (nm x m) supermatrix $\mathbf{T}$, and that the unknown diagonal matrices $\mathbf{V}_j$ are collected in the (np x p) supermatrix $\mathbf{V}$. In section 4.3.2 we first determine the translation vectors $\mathbf{g}_j$ minimizing (4.8), and show how to eliminate them from the STIMFREE loss function. In sections 4.3.3, 4.3.4, and 4.3.5 the simplified loss function is optimized with respect to the orthonormal matrices $\mathbf{T}_j$, the stimulus weights $\mathbf{V}_j$, and the centroid configuration $\mathbf{Y}$, respectively. In section 4.3.6 a direct approach is set up for the STIMFREE model, yielding a more efficient procedure for the determination of the orthonormal matrices $\mathbf{T}_j$. The uniqueness properties of the STIMFREE model are discussed in section 4.3.7, and section 4.3.8 contains an algorithm for the fitting of this model on n configurations.

4.3.2 Translations

To minimize (4.8) with respect to the unrestricted translation vectors $\mathbf{G}$ for fixed $\mathbf{T}$, $\mathbf{V}$, and $\mathbf{Y}$, let $\mathbf{A}_j = \tilde{\mathbf{X}}_j\mathbf{T}_j - \mathbf{V}_j\mathbf{Y}$, consider only one particular translation vector $\mathbf{g}_j$, and rewrite the loss function as

$$f(\mathbf{g}_j) = c_j^2\mathbf{g}_j'\mathbf{g}_j - 2\, c_j\mathbf{g}_j'\mathbf{r}_j + d_j, \qquad (4.9)$$

where d_j is independent of $\mathbf{g}_j$, $\mathbf{r}_j \equiv \mathbf{T}_j\mathbf{A}_j'\mathbf{M}_j\mathbf{1}/\sqrt{\mathbf{1}'\mathbf{M}_j\mathbf{1}}$, and $c_j \equiv \sqrt{\mathbf{1}'\mathbf{M}_j\mathbf{1}}$. The solution for (4.9) has been discussed previously (see, e.g., section 2.3.1), and is given by

$$\mathbf{g}_j = c_j^{-1}\mathbf{r}_j. \qquad (4.10)$$

Re-expressing (4.10) in the original matrices and vectors yields

$$\mathbf{g}_j = \frac{(\tilde{\mathbf{X}}_j - \mathbf{V}_j\mathbf{Y}\mathbf{T}_j')'\mathbf{M}_j\mathbf{1}}{\mathbf{1}'\mathbf{M}_j\mathbf{1}}, \tag{4.11}$$

and substitution of (4.11) in (4.8) gives

$$f(\mathbf{T},\mathbf{V},\mathbf{Y}) = \sum_{j=1}^{n} \text{tr } (\tilde{\mathbf{X}}_j\mathbf{T}_j - \mathbf{V}_j\mathbf{Y})'\mathbf{C}_j(\tilde{\mathbf{X}}_j\mathbf{T}_j - \mathbf{V}_j\mathbf{Y}), \tag{4.12}$$

where $\mathbf{C}_j \equiv \mathbf{M}_j(\mathbf{I} - \mathbf{1}\mathbf{1}'\mathbf{M}_j/\mathbf{1}'\mathbf{M}_j\mathbf{1})$. After elimination of the $\mathbf{g}_j$ from the loss function, the STIMFREE model (4.5) can be simplified to

$$\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j = \mathbf{C}_j\mathbf{V}_j\mathbf{Y} + \mathbf{C}_j\mathbf{E}_j, \qquad \text{for } j = 1, 2, ..., n. \tag{4.13}$$

Since $\mathbf{C}_j\mathbf{1} = \mathbf{0}$ it is also true that

$$\mathbf{C}_j\tilde{\mathbf{X}}_j = s_j\mathbf{C}_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j\mathbf{K} = s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j\mathbf{K} - s_j\mathbf{C}_j\mathbf{1}\mathbf{u}_j'\mathbf{R}_j\mathbf{K} = s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j\mathbf{K},$$

and we therefore define $\tilde{\mathbf{X}}_j = s_j\mathbf{X}_j\mathbf{R}_j\mathbf{K}$ throughout the rest of section 4.3.

4.3.3 Orthonormal transformations

To determine the matrix $\mathbf{T}$ minimizing (4.12) for fixed $\mathbf{V}$ and $\mathbf{Y}$, define $\mathbf{A}_j = \mathbf{V}_j\mathbf{Y}$, and rewrite (4.12) as

$$f(\mathbf{T}) = \sum_{j=1}^{n} \text{tr } (\tilde{\mathbf{X}}_j\mathbf{T}_j - \mathbf{A}_j)'\mathbf{C}_j(\tilde{\mathbf{X}}_j\mathbf{T}_j - \mathbf{A}_j). \tag{4.14}$$

We refer to section 3.3.3.of Chapter 3 for a detailed discussion of the solution of (4.14) with respect to the unknown orthonormal matrices $\mathbf{T}_j$. This solution is analytic and guarantees to yield the global minimum of (4.14).

4.3.4 Stimulus weights

For fixed orthonormal matrices $\mathbf{T}$ and centroid configuration $\mathbf{Y}$, the problem discussed in this section is how to minimize

$$f(\mathbf{V}) = \sum_{j=1}^{n} \text{tr } (\mathbf{A}_j - \mathbf{C}_j\mathbf{V}_j\mathbf{Y})'(\mathbf{A}_j - \mathbf{C}_j\mathbf{V}_j\mathbf{Y}) \tag{4.15}$$

where $\mathbf{A}_j \equiv \mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j$. Only considering one $\mathbf{V}_j$, and expanding (4.15) gives

$$f(\mathbf{V}_j) = d_j + \text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{Y} - 2 \text{ tr } \mathbf{A}_j'\mathbf{C}_j\mathbf{V}_j\mathbf{Y}, \tag{4.16}$$

where d_j is a constant with respect to $\mathbf{V}_j$. Let $\mathbf{b}_j = (\text{diag } \mathbf{Y}\mathbf{A}_j')\mathbf{1}$, that is, let $\mathbf{b}_j$ be the (p x 1) vector containing the diagonal elements of $\mathbf{Y}\mathbf{A}_j'$. Let $\mathbf{v}_j = \mathbf{V}_j\mathbf{1}$, that is, let $\mathbf{v}_j$ be the (p x 1) vector consisting of the p unknown stimulus weights. This makes it possible to write (4.16) as

$$f(\mathbf{v}_j) = d_j + \mathbf{v}_j'(\mathbf{C}_j\Theta\mathbf{Y}\mathbf{Y}')\mathbf{v}_j - 2\,\mathbf{b}_j'\mathbf{v}_j, \tag{4.17}$$

where Θ denotes the Hadamard product. But, if configuration $\mathbf{X}_j$ has missing rows, then the corresponding rows *and* columns of $(\mathbf{C}_j\Theta\mathbf{Y}\mathbf{Y}')$ contain zero elements only, and the corresponding elements of $\mathbf{b}_j$ are also zero. If we let p_j be the number of non-missing rows of $\mathbf{X}_j$, this means that we only need to determine p_j stimulus weights, because the values of the remaining $(p - p_j)$ elements of $\mathbf{v}_j$ in (4.17) are irrelevant for the value of the loss function. Therefore, letting $\mathbf{D}_j$ be the $(p_j \text{ x } p_j)$ matrix only containing the non-zero elements of $(\mathbf{C}_j\Theta\mathbf{Y}\mathbf{Y}')$, $\mathbf{b}_j^*$ be the $(p_j \text{ x } 1)$ vector of non-zero elements of $\mathbf{b}_j$, and $\mathbf{v}_j^*$ be the $(p_j \text{ x } 1)$ vector of corresponding elements of $\mathbf{v}_j$, loss function (4.17) may be rewritten as

$$f(\mathbf{v}_j^*) = d_j + \mathbf{v}_j^{*\prime}\mathbf{D}_j\mathbf{v}_j^* - 2\,\mathbf{b}_j^{*\prime}\mathbf{v}_j^* = d_j + e_j + \| \mathbf{D}_j^{1/2}\mathbf{v}_j^* - \mathbf{D}_j^{-1/2}\mathbf{b}_j^* \|^2, \tag{4.18}$$

where e_j is, again, a term independent of $\mathbf{v}_j^*$. It follows that the global minimum of (4.18) is attained for

$$\mathbf{v}_j^* = \mathbf{D}_j^{-1}\mathbf{b}_j^*. \tag{4.19}$$

By replacing the appropriate elements of $\mathbf{v}_j$ of order (p x 1) by the elements of $\mathbf{v}_j^*$ of order $(p_j \text{ x } 1)$ given by (4.19), and by setting the remaining $(p - p_j)$ elements of $\mathbf{v}_j$

equal to some arbitrary number (zero, for instance), the global minimum of (4.17) with respect to $\mathbf{v}_j$ is attained. Finally, given the latter optimal $\mathbf{v}_j$, the global minimum of (4.16) is found where

$$\mathbf{V}_j = \mathbf{I}_p \Theta (\mathbf{v}_j \mathbf{1}'). \tag{4.20}$$

The computation of (4.20) for j = 1, ..., n is guaranteed to yield the global minimum of (4.15) with respect to **V**.

A disadvantage of the calculation of the stimulus weights using (4.19) is that it may result in negative weights. As we discussed in section 4.2, negative stimulus weights do not have a meaningful interpretation. Therefore, we use the nonnegative least squares (NNLS) algorithm of Lawson and Hanson (1974) to determine the global minimum of (4.18) subject to the linear constraint $\mathbf{v}_j^* \geq 0$. For a detailed description of the NNLS algorithm we refer to Lawson and Hanson (1974, p.158 - 165).

4.3.5 The centroid configuration Y

For fixed **T** and **V**, we want to find the minimum of

$$f(\mathbf{Y}) = \sum_{j=1}^{n} \text{tr } (\mathbf{A}_j - \mathbf{B}_j\mathbf{Y})'\mathbf{C}_j(\mathbf{A}_j - \mathbf{B}_j\mathbf{Y}) \tag{4.21}$$

where $\mathbf{A}_j \equiv \mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j$ and $\mathbf{B}_j \equiv \mathbf{C}_j\mathbf{V}_j$. Collecting the n matrices $\mathbf{A}_j$ in the supermatrix **A** of order (np x m), and collecting the n matrices $\mathbf{B}_j$ in the supermatrix **B** of order (np x p), (4.21) may be written as

$$f(\mathbf{Y}) = \text{tr } (\mathbf{A} - \mathbf{B}\mathbf{Y})'(\mathbf{A} - \mathbf{B}\mathbf{Y}). \tag{4.22}$$

This is the classical multivariate multiple regression problem with the well-known solution $\mathbf{Y} = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{A}$, which can be re-expressed in terms of the original matrices as

$$\mathbf{Y} = (\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)^{-1}(\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\mathbf{A}_j). \tag{4.23}$$

The sum of the matrix products $\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j$ yields a matrix of full rank for which a proper inverse exists.

In general the optimal $\mathbf{Y}$ in (4.23) is *not* a column centered matrix, in contrast with the centroid configurations found in the GPA and in the dimension weighting models. This is not surprising since, as we already noted in section 4.2, the location of $\mathbf{Y}$ is essential once stimulus weights are taken into account.

4.3.6 The direct approach in STIMFREE

In this section we show that the centroid configuration $\mathbf{Y}$ can be eliminated from (4.12). This yields a loss function which makes it possible to set up a more efficient procedure for the determination of the $\mathbf{T}_j$'s than the one given in section 4.3.3.

Since the minimization of (4.22) is a multiple regression problem, given optimal regression weights $\mathbf{Y}$, the space spanned by the predicted $\mathbf{BY}$ is orthogonal to the space spanned by the residual matrix $(\mathbf{A} - \mathbf{BY})$. At the global minimum of (4.22) it is therefore true that

$$f(\mathbf{Y}) = \text{tr } (\mathbf{A} - \mathbf{BY})'(\mathbf{A} - \mathbf{BY}) = \text{tr } \mathbf{A}'(\mathbf{A} - \mathbf{BY}), \tag{4.24}$$

where the matrices $\mathbf{A}$ and $\mathbf{B}$ are defined as in section 4.3.5. Substitution of $\mathbf{Y} = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{A}$ in (4.24) yields

$$f((\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{A}) = \text{tr } \mathbf{A}'\mathbf{A} - \text{tr } \mathbf{A}'\mathbf{B}(\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{A},$$

which can be re-expressed in terms of the original matrices as

$$f(\mathbf{T},\mathbf{V}) = \sum_{j=1}^{n} \text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j - \text{tr } (\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j)'(\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)^{-1}(\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j). \tag{4.25}$$

The minimization of (4.25) is equivalent to the maximization of

$$g(\mathbf{T},\mathbf{V}) = \text{tr } (\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j)'(\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)^{-1}(\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j) \tag{4.26}$$

under the constraint that $\mathbf{T}_j'\mathbf{T}_j = \mathbf{T}_j\mathbf{T}_j' = \mathbf{I}_m$ for each j.

It does not seem possible to determine the $\mathbf{V}_j$'s maximizing (4.26) for fixed $\mathbf{T}$. On the other hand, to maximize (4.26) with respect to $\mathbf{T}$ for fixed $\mathbf{V}$, define $\mathbf{D}_j = \mathbf{V}_j\mathbf{C}_j\tilde{\mathbf{X}}_j$, and $\mathbf{F} = (\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)^{-1}$. This allows us to rewrite (4.26) as

$$g(\mathbf{T}) = \text{tr}\,(\sum_{j=1}^{n} \mathbf{D}_j\mathbf{T}_j)'\mathbf{F}(\sum_{j=1}^{n} \mathbf{D}_j\mathbf{T}_j), \tag{4.27}$$

and, if we only consider one $\mathbf{T}_j$, as

$$g(\mathbf{T}_j) = \text{tr}\,\mathbf{T}_j'\mathbf{D}_j'\mathbf{F}\mathbf{D}_j\mathbf{T}_j + 2\,\text{tr}\,\mathbf{T}_j'\mathbf{D}_j'\mathbf{F}(\sum_{i\neq j}\mathbf{D}_i\mathbf{T}_i) + d_j, \tag{4.28}$$

where d_j is a term independent from $\mathbf{T}_j$. Therefore, the maximization of (4.28) is equivalent to the maximization of

$$h(\mathbf{T}_j) = \text{tr}\,\mathbf{T}_j'\mathbf{D}_j'\mathbf{F}(\sum_{i\neq j}\mathbf{D}_i\mathbf{T}_i) \tag{4.29}$$

subject to $\mathbf{T}_j'\mathbf{T}_j = \mathbf{T}_j\mathbf{T}_j' = \mathbf{I}_m$. This is the orthonormal Procrustes problem. We refer to section 2.3.4 of Chapter 2 for a detailed discussion of the solution of (4.29).

Since there is no analytical solution for $\mathbf{T}$ in (4.27), an alternative least squares algorithm must be used where (4.27) is consecutively maximized for j = 1, 2, ..., n and $\mathbf{T}$ is updated after each step. This process must be repeated until n steps jointly fail to raise (4.27) above some threshold value. We have found that this procedure for determining orthonormal matrices $\mathbf{T}_j$ results into a more efficient algorithm for the estimation of the unknown parameters in the STIMFREE model than the procedure discussed in section 4.3.3. That is, while both procedures require about an identical amount of CPU-time within one iteration, convergence of the algorithm is considerably faster when using the updating procedure for the $\mathbf{T}_j$'s described in this section.

4.3.7 Uniqueness of the STIMFREE solution

The solution for the STIMFREE model belongs to the following family of equivalent solutions:

$$f(\mathbf{T},\mathbf{V},\mathbf{Y}) = \sum_{j=1}^{n} \text{tr}\,(\tilde{\mathbf{X}}_j\mathbf{T}_j\mathbf{R} - \mathbf{V}_j\mathbf{K}\mathbf{K}^{-1}\mathbf{Y}\mathbf{R})'\mathbf{C}_j(\tilde{\mathbf{X}}_j\mathbf{T}_j\mathbf{R} - \mathbf{V}_j\mathbf{K}\mathbf{K}^{-1}\mathbf{Y}\mathbf{R}), \tag{4.30}$$

where $\mathbf{R}$ is an arbitrary (m x m) orthonormal matrix, and $\mathbf{K}$ is an arbitrary diagonal matrix of order (m x m). In words, (4.30) expresses that the solution in STIMFREE is unique up to a simultaneous rotation by $\mathbf{R}$ of the whole solution, and, more

importantly, up to a weighting **K** of the optimal stimulus weights $\mathbf{V}_j$ together with an inverse weighting of the optimal centroid **Y**.

The rotational indeterminacy of the STIMFREE model can be used to rotate the solution to its principal components. This point is further discussed in section 4.5. The freedom we have in choosing a particular weighting matrix **K** allows us to choose one yielding a uniquely weighted solution. This is achieved by setting **K** equal to

$$\mathbf{G} = (\text{diag } \mathbf{YY}')^{1/2}, \tag{4.31}$$

which yields a weighted centroid configuration

$$\mathbf{Y}^* = \mathbf{G}^{-1}\mathbf{Y} = (\text{diag } \mathbf{YY}')^{-1/2}\mathbf{Y}, \tag{4.32}$$

with unique stimulus weights

$$\mathbf{V}_j^* = \mathbf{V}_j\mathbf{G} = \mathbf{V}_j(\text{diag } \mathbf{YY}')^{1/2}, \qquad \text{for } j = 1, 2, \ldots, n. \tag{4.33}$$

Since premultiplying **Y** with $(\text{diag } \mathbf{YY}')^{-1/2}$ has the effect of unit normalizing the rows of **Y**, the matrix product $\mathbf{Y}^*\mathbf{Y}^{*\prime}$ contains the cosines of the angles between the stimulus points of **Y** conceived of as vectors connected with the origin.

Except that the unit normalization of the rows of **Y** results in a uniquely weighted solution, it also has another advantage. In the original model (4.3) of Lingoes and Borg (see section 4.2), the effect of a stimulus weight is conditional upon the length of the vector in $(\mathbf{Z} - \mathbf{h})$ that is being weighted. If a stimulus vector in $(\mathbf{Z} - \mathbf{h})$ has a length of 0.08, for example, weighting this vector with a factor 10 has the effect of moving the end point of the vector only $(10)(0.08) - 0.08 = 0.72$ units further away from the origin. But a much smaller weight of 1.144 will do the same job for a vector of length 5.00, since $(1.144)(5.00) - 5.00 = 0.72$ units also. In general, therefore, information about the value of a weight is not, in itself, enough to assess the effect or impact of the weight. This interdependence between weight effect and vector length completely vanishes for the unique stimulus weights $\mathbf{V}_j^*$ given in (4.33). Since the lengths of the stimulus vectors in $\mathbf{Y}^*$ are all equal to one, the impact of the unique weights can be assessed unambiguously in the STIMFREE model. Below, we will propose a special 'impact index' for the STIMFREE stimulus weights. First, however, another problem concerning the stimulus weights must be discussed.

Although the stimulus weights calculated in (4.33) are unique, they are, in general, not comparable over configurations. This is due to the fact that the n configurations $\tilde{\mathbf{X}}_j$ will usually have different sizes (i.e., the values of tr $\tilde{\mathbf{X}}'_j\mathbf{C}_j\tilde{\mathbf{X}}_j$ will be different for $j = 1, \ldots, n$). As a consequence, the values of the stimulus weights corresponding to a 'small' configuration will be relatively larger than those corresponding to a 'large' configuration. This problem is easily remedied, however, by calculating

$$\mathbf{V}^{**}_j = \mathbf{V}^{*}_j/(\text{tr } \tilde{\mathbf{X}}'_j\mathbf{C}_j\tilde{\mathbf{X}}_j)^{1/2} \tag{4.34}$$

for each j, since in that case idiosyncracies in stimulus weighting due to differences in size of the configurations disappear.

As promised, we end this section by presenting an 'impact index' for the stimulus weights in (4.34). This index makes it particularly easy to assess the amount of displacement caused by a stimulus weight. The impact index d_{ij} is defined as follows:

$$d_{ij} = | 1 - v_{ij} |. \tag{4.35}$$

In (4.35), v_{ij} denotes the stimulus weight calculated in (4.33) for stimulus i corresponding to configuration j, and d_{ij} denotes the impact index for v_{ij}. The definition covers positive as well as negative stimulus weights. The impact index is always non-negative. In words, the index d_{ij} is equal to the distance between the two end points of stimulus vector i of $\mathbf{Y}^*$ before and after stimulus weighting with v_{ij}. Therefore, an index of zero indicates that nothing changed after stimulus weighting. The larger the index, the further the end point of the vector has been moved away from its original place by the stimulus weight.

The geometry of the impact index is illustrated in Figure 4.4, where the black arrow represents a unit normalized vector of $\mathbf{Y}^*$. Of the three examples given in the figure, the smallest impact or displacement is achieved by a stimulus weight of 1.5 (middle of the figure), since the displacement of the end point of the vector is only 0.5. A larger displacement of the end point of the stimulus vector of $\mathbf{Y}^*$ is obtained for a stimulus weight of 0.3: the value of the corresponding impact index is 0.7 (left of the figure). The largest impact shown in the figure corresponds to a negative stimulus weight of -0.4.

The impact index can be used to test for differences between configurations or stimuli with standard linear statistical techniques. In this way homogeneous groups of

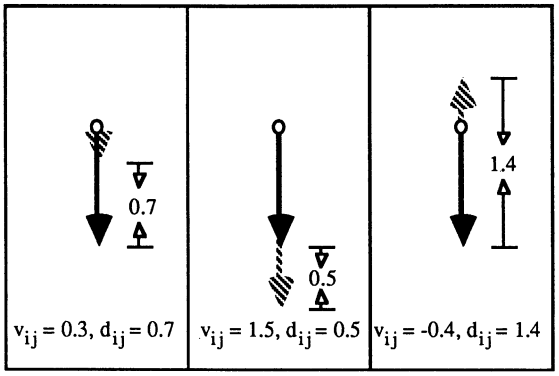


Figure 4.4 Three illustrations of relation between unique stimulus weight and impact index.

stimuli may be formed, for instance, yielding a better fit when analyzed separately with the more parsimonious GPA and dimension weighting models.

4.3.8 The algorithm

In the MATCHALS program the algorithm for the estimation of the unknown parameters in the STIMFREE model consists of the following steps.

a) Before starting to iterate, initialize each $\mathbf{T}_j$ on $\mathbf{T}_j = \mathbf{I}_m$, and $\mathbf{Y}$ on $\mathbf{Y} = \mathbf{ZK}$ (i.e., the optimal centroid configuration found in GPA, rotated to the principal components of the GPA solution).

b) For every j, compute new stimulus weights using the procedure given in section 4.3.4. By default the MATCHALS program computes nonnegative weights.

c) Determine the proper inverse of $(\Sigma\, \mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)$, and calculate a new $\mathbf{T}_j$ for each j using the procedure described in section 4.3.6, where $\mathbf{T}$ must be updated each time a new $\mathbf{T}_j$ has been estimated.

d) Evaluate loss function (4.26). If the difference between the function value in this iteration and in the previous iteration is smaller than some convergence criterion, go to

step e). Otherwise, calculate a new centroid configuration $\mathbf{Y}$ with (4.23) using the proper inverse of $(\Sigma\, \mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)$ computed in step c), and go to step b).

e) Print the history of iterations and $\mathbf{Y}$. Compute the normalized and uniquely weighted group configuration $\mathbf{Y}^*$ and the normalized and uniquely weighted stimulus weights $\mathbf{V}_j^*$ according to (4.32) and (4.33), respectively, and print $\mathbf{Y}^*$ and $\mathbf{V}_j^*/(\text{tr}\, \tilde{\mathbf{X}}'_j\mathbf{C}_j\tilde{\mathbf{X}}_j)^{1/2}$ for each j. Finally, compute the impact index for each stimulus weight using (4.35), and print this also.

The solution calculated in step e) of the algorithm for STIMFREE is unique up to a simultaneous rotation of the complete solution. In section 4.5.2 we show how this last indeterminacy can be used to rotate the solution to its principal components.

Unfortunately, this algorithm often converges to a local minimum. We observed this annoying property when analyzing data constructed in such a way that a perfect solution was known to exist: the algorithm often converged to a solution that, although leaving only a small residual sum of squares, was not the global minimum of zero loss. In section 4.6 examples of this situation are given.

As yet we can only speculate about the possible causes of this problem. Different starting values for $\mathbf{Y}$ in step a) of the algorithm could be tried to investigate the stability of the solution. Instead of using $\mathbf{ZK}$ from GPA as initial estimate for $\mathbf{Y}$, for instance, one could start the STIMFREE algorithm with the centroid configuration obtained in DIMFREE or DIMIDIO. Also, the examples discussed in section 4.6 consist of only four configurations containing the coordinates of eight stimuli. We have not investigated whether better results are obtained when analyzing larger numbers of configurations and stimuli.

4.4 The STIMIDIO model

4.4.1 Introduction

The least squares loss function corresponding to the STIMIDIO model (4.7) is

$$f(\mathbf{G},\mathbf{H},\mathbf{T},\mathbf{V},\mathbf{Y}) = \sum_{j=1}^{n} \| \mathbf{M}_j[(\tilde{\mathbf{X}}_j - \mathbf{1}\mathbf{g}_j')\mathbf{T}_j - \mathbf{V}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j')] \|^2. \tag{4.36}$$

In (4.36) it is assumed that the unknown translation vectors $\mathbf{g}_j$ and $\mathbf{h}_j$ are collected in the (m x n) matrices **G** and **H**, respectively. The unknown orthonormal matrices $\mathbf{T}_j$ are assumed to be collected in the (nm x m) supermatrix **T**, and the unknown diagonal matrices $\mathbf{V}_j$ in the (np x p) supermatrix **V**.

In section 4.4.2 we show how to eliminate the translation vectors **G** from loss function (4.36). In sections 4.4.3, 4.4.4, 4.4.5, and 4.4.6 the simplified loss function is optimized with respect to the orthonormal matrices $\mathbf{T}_j$, the stimulus weights $\mathbf{V}_j$, the translation vectors $\mathbf{h}_j$, and the centroid configuration **Y**, respectively. In section 4.4.7 the uniqueness properties of the STIMIDIO model are investigated, and in section 4.4.8 an algorithm is presented for the estimation of the model parameters.

4.4.2 Translations of the $\mathbf{X}_j$

To minimize (4.36) with respect to **G** for fixed **H**, **V**, **T**, and **Y**, let $\mathbf{A}_j = \tilde{\mathbf{X}}_j\mathbf{T}_j - \mathbf{V}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j')$, and consider only one particular translation vector $\mathbf{g}_j$ to rewrite the loss function as

$$f(\mathbf{g}_j) = d_j + c_j^2\mathbf{g}_j'\mathbf{g}_j - 2\,c_j\mathbf{g}_j'\mathbf{r}_j, \tag{4.37}$$

where d_j is independent of $\mathbf{g}_j$, $\mathbf{r}_j \equiv \mathbf{T}_j\mathbf{A}_j'\mathbf{M}_j\mathbf{1}/\sqrt{\mathbf{1}'\mathbf{M}_j\mathbf{1}}$, and $c_j \equiv \sqrt{\mathbf{1}'\mathbf{M}_j\mathbf{1}}$. The solution for (4.37) has already been discussed several times (see, e.g., section 2.3.1), and is given by

$$\mathbf{g}_j = c_j^{-1}\mathbf{r}_j = \frac{[\tilde{\mathbf{X}}_j - \mathbf{V}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j')\mathbf{T}_j']'\mathbf{M}_j\mathbf{1}}{\mathbf{1}'\mathbf{M}_j\mathbf{1}}. \tag{4.38}$$

Substitution of (4.38) in (4.36) gives

$$f(\mathbf{H},\mathbf{T},\mathbf{V},\mathbf{Y}) = \sum_{j=1}^{n} \text{tr}\,[\tilde{\mathbf{X}}_j\mathbf{T}_j - \mathbf{V}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j')]'\mathbf{C}_j[\tilde{\mathbf{X}}_j\mathbf{T}_j - \mathbf{V}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j')] \tag{4.39}$$

with $\mathbf{C}_j \equiv \mathbf{M}_j(\mathbf{I} - \mathbf{1}\mathbf{1}'\mathbf{M}_j/\mathbf{1}'\mathbf{M}_j\mathbf{1})$. Note that, unlike in the dimension weighting models, substitution of (4.38) in (4.36) does not result in the simultaneous elimination of the translation vectors $\mathbf{H}$. After the elimination of $\mathbf{G}$ from (4.36) the STIMIDIO model (4.7) can be simplified to

$$\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j = \mathbf{C}_j\mathbf{V}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j') + \mathbf{C}_j\mathbf{E}_j, \qquad \text{for } j = 1, 2, \ldots, n. \tag{4.40}$$

Since $\mathbf{C}_j\mathbf{1} = \mathbf{0}$, and therefore

$$\mathbf{C}_j\tilde{\mathbf{X}}_j = s_j\mathbf{C}_j(\mathbf{X}_j - \mathbf{1}\mathbf{u}_j')\mathbf{R}_j\mathbf{K} = s_j\mathbf{C}_j\mathbf{X}_j\mathbf{R}_j\mathbf{K},$$

we define $\tilde{\mathbf{X}}_j = s_j\mathbf{X}_j\mathbf{R}_j\mathbf{K}$ throughout the remaining of Chapter 4.

4.4.3 Orthonormal transformations

To determine the unknown orthonormal matrices $\mathbf{T}_j$ minimizing (4.39) for fixed $\mathbf{H}$, $\mathbf{V}$, and $\mathbf{Y}$, we define $\mathbf{A}_j = \mathbf{V}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j')$ and rewrite (4.39) as

$$f(\mathbf{T}) = \sum_{j=1}^{n} \text{tr}\,(\tilde{\mathbf{X}}_j\mathbf{T}_j - \mathbf{A}_j)'\mathbf{C}_j(\tilde{\mathbf{X}}_j\mathbf{T}_j - \mathbf{A}_j). \tag{4.41}$$

This problem has already been discussed in section 3.3.3 of Chapter 3. Therefore, we refer to the latter section for a detailed description of the solution of this problem. This solution is guaranteed to yield the global minimum of (4.41).

4.4.4 Stimulus weights

For fixed **H**, **T**, and **Y**, the problem is how to minimize

$$f(\mathbf{V}) = \sum_{j=1}^{n} \text{tr}\,(\mathbf{A}_j - \mathbf{C}_j\mathbf{V}_j\mathbf{B}_j)'(\mathbf{A}_j - \mathbf{C}_j\mathbf{V}_j\mathbf{B}_j), \tag{4.42}$$

where $\mathbf{A}_j \equiv \mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j$ and $\mathbf{B}_j \equiv \mathbf{Y} - \mathbf{1}\mathbf{h}_j'$. This problem is again identical to the problem discussed in section 4.3.4. Let p_j be the number of non-missing rows of $\mathbf{X}_j$. Define $\mathbf{D}_j$ as the (p_j x p_j) matrix containing the rows *and* columns of $(\mathbf{C}_j \Theta \mathbf{B}_j\mathbf{B}_j')$ corresponding to the non-missing rows of $\mathbf{X}_j$. Also, define $\mathbf{b}_j^*$ as the (p_j x 1) vector consisting of the elements of $\mathbf{b}_j \equiv (\text{diag}\,\mathbf{B}_j\mathbf{A}_j')\mathbf{1}$ corresponding to the non-missing rows of $\mathbf{X}_j$, and $\mathbf{v}_j^*$ as the (p_j x 1) vector containing the corresponding stimulus weights of $\mathbf{v}_j \equiv \mathbf{V}_j\mathbf{1}$. Then, if we consider only one $\mathbf{V}_j$, function (4.42) may be written as

$$f(\mathbf{v}_j^*) = d_j + \mathbf{v}_j^{*\prime}\mathbf{D}_j\mathbf{v}_j^* - 2\,\mathbf{b}_j^{*\prime}\mathbf{v}_j^* = d_j + e_j + \| \mathbf{D}_j^{1/2}\mathbf{v}_j^* - \mathbf{D}_j^{-1/2}\mathbf{b}_j^* \|^2, \tag{4.43}$$

where d_j and e_j are terms independent of $\mathbf{v}_j^*$. If the stimulus weights are not restricted, the global minimum of (4.43) is attained for

$$\mathbf{v}_j^* = \mathbf{D}_j^{-1}\mathbf{b}_j^*. \tag{4.44}$$

Inserting these p_j optimal weights in the appropriate places in the vector $\mathbf{v}_j$, setting the remaining ($p - p_j$) stimulus weights equal to some arbitrary value, and then calculating $\mathbf{V}_j$ as

$$\mathbf{V}_j = \mathbf{I}_p \Theta (\mathbf{v}_j\mathbf{1}'), \tag{4.45}$$

yields the global minimum of (4.42) with respect to one particular $\mathbf{V}_j$. Calculating (4.44) and (4.45) for $j = 1, \ldots, n$ is guaranteed to give the global minimum of (4.42) with respect to **V**.

The nonnegative least squares algorithm of Lawson and Hanson (1974) can be used to determine the global minimum of (4.43) subject to $\mathbf{v}_j^* \geq 0$.

4.4.5 Idiosyncratic translations of Y

Defining $\mathbf{A}_j = \tilde{\mathbf{X}}_j\mathbf{T}_j - \mathbf{V}_j\mathbf{Y}$, and considering only one $\mathbf{h}_j$, the problem for fixed $\mathbf{T}$, $\mathbf{V}$, and $\mathbf{Y}$ is how to minimize

$$f(\mathbf{h}_j) = \sum_{j=1}^{n} \text{tr } (\mathbf{A}_j + \mathbf{V}_j\mathbf{1}\mathbf{h}_j')'\mathbf{C}_j(\mathbf{A}_j + \mathbf{V}_j\mathbf{1}\mathbf{h}_j')$$

$$= c_j^2\mathbf{h}_j'\mathbf{h}_j + 2\, c_j\mathbf{h}_j'\mathbf{r}_j + d_j, \tag{4.46}$$

where d_j is a term independent of $\mathbf{h}_j$, $\mathbf{r}_j \equiv (\mathbf{A}_j'\mathbf{C}_j\mathbf{V}_j\mathbf{1})/\sqrt{(\mathbf{1}'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{1})}$, and $c_j \equiv \sqrt{(\mathbf{1}'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{1})}$. The global minimum of (4.46) is attained for

$$\mathbf{h}_j = -c_j^{-1}\mathbf{r}_j = \frac{(\mathbf{V}_j\mathbf{Y} - \tilde{\mathbf{X}}_j\mathbf{T}_j)'\mathbf{C}_j\mathbf{V}_j\mathbf{1}}{\mathbf{1}'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{1}}. \tag{4.47}$$

Computing (4.47) for $j = 1, \ldots, n$ is guaranteed to give the global minimum of (4.39) with respect to $\mathbf{H}$ for fixed $\mathbf{T}$, $\mathbf{V}$, and $\mathbf{Y}$.

4.4.6 The centroid configuration Y

To minimize (4.39) with respect to $\mathbf{Y}$ for fixed $\mathbf{H}$, $\mathbf{T}$, and $\mathbf{V}$, define $\mathbf{A}_j = \tilde{\mathbf{X}}_j\mathbf{T}_j + \mathbf{V}_j\mathbf{1}\mathbf{h}_j'$, and rewrite (4.39) as

$$f(\mathbf{Y}) = \sum_{j=1}^{n} \text{tr } (\mathbf{A}_j - \mathbf{V}_j\mathbf{Y})'\mathbf{C}_j(\mathbf{A}_j - \mathbf{V}_j\mathbf{Y}). \tag{4.48}$$

As has already been discussed in section 4.3.5, the global minimum of (4.48) is attained for

$$\mathbf{Y} = (\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)^{-1}(\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\mathbf{A}_j). \tag{4.49}$$

Just as in STIMFREE, this centroid configuration will, in general, not be a column centered matrix.

4.4.7 Non-uniqueness of the STIMIDIO solution

Unfortunately, the STIMIDIO model is so full of indeterminacies that it is not possible to reduce these to a unique solution. We start by noting that the value of the STIMIDIO loss function is unchanged by the following two transformations:

$$f(\mathbf{H},\mathbf{T},\mathbf{V},\mathbf{Y}) = \sum_{j=1}^{n} \| \mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j\mathbf{P} - \mathbf{C}_j\mathbf{V}_j[(\mathbf{Y} - \mathbf{1d}') - (\mathbf{1h}_j' - \mathbf{1d}')]\mathbf{P} \|^2, \quad (4.50)$$

where $\mathbf{P}$ is an arbitrary orthonormal matrix of order (m x m), and $\mathbf{d}$ an arbitrary translation vector of order (m x 1). These transformations of the solution express that the whole solution may be simultaneously rotated, and that the centroid configuration $\mathbf{Y}$ may be translated arbitrarily if a reverse translation is applied to the translation vectors $\mathbf{h}_j$.

We could use the latter indeterminacy to translate $\mathbf{Y}$ to a unique location. An obvious choice would be to translate $\mathbf{Y}$ such that it becomes column centered. This is accomplished by taking

$$\mathbf{d} = \mathbf{Y}'\mathbf{1}/\mathbf{1}'\mathbf{1}, \quad (4.51)$$

since

$$(\mathbf{Y} - \mathbf{1d}')'\mathbf{1} = \mathbf{Y}'\mathbf{1} - \mathbf{d}\mathbf{1}'\mathbf{1} = \mathbf{Y}'\mathbf{1} - \mathbf{Y}'\mathbf{1}\mathbf{1}'\mathbf{1}/\mathbf{1}'\mathbf{1} = \mathbf{0}, \quad (4.52)$$

yielding a relocated centroid configuration

$$\mathbf{Y}^* = \mathbf{Y} - \mathbf{1d}' = \mathbf{Y} - \mathbf{1}\mathbf{1}'\mathbf{Y}/\mathbf{1}'\mathbf{1}, \quad (4.53)$$

and relocated translation vectors

$$\mathbf{h}_j^* = \mathbf{h}_j - \mathbf{d} = \mathbf{h}_j - \mathbf{Y}'\mathbf{1}/\mathbf{1}'\mathbf{1}. \quad (4.54)$$

The rotational indeterminacy of the STIMIDIO solution could be used to rotate it to principal components.

However, this would not make the solution unique. Suppose that we analyze the same data set twice with STIMIDIO using two different initial estimates for the centroid $\mathbf{Y}$. Assume for the sake of simplicity that we obtain perfect solutions in both analyses, and that all configurations are complete. Let $\mathbf{V}_j(\mathbf{Y} - \mathbf{1h}_j')$ be the first STIMIDIO solution, and $\tilde{\mathbf{V}}_j(\tilde{\mathbf{Y}} - \mathbf{1}\tilde{\mathbf{h}}_j')$ the second solution for j = 1, ..., n. Letting $\mathbf{J}$ =

$(\mathbf{I}_p - \mathbf{11'/1'1})$, and since $\mathbf{C}_j = \mathbf{J}$ when a configuration is complete, it follows from (4.50) that the relation between the two solutions will be

$$\mathbf{J}\tilde{\mathbf{X}}_j\mathbf{T}_j = \mathbf{J}\mathbf{V}_j(\mathbf{Y} - \mathbf{1h}_j') = \mathbf{J}\tilde{\mathbf{V}}_j(\tilde{\mathbf{Y}} - \mathbf{1}\tilde{\mathbf{h}}_j')\mathbf{P}, \qquad \text{for } j = 1, \ldots, n, \tag{4.55}$$

where $\mathbf{P}$ is an orthonormal matrix. Thus, the two solutions will be rotated versions of one another. Moreover, since matrix $\mathbf{J}$ has the effect of centering the matrix products in (4.55) on the origin, the matrix products $\mathbf{V}_j(\mathbf{Y} - \mathbf{1h}_j')$ and $\tilde{\mathbf{V}}_j(\tilde{\mathbf{Y}} - \mathbf{1}\tilde{\mathbf{h}}_j')\mathbf{P}$ are free to be located anywhere in m-dimensional space. This means that another indeterminacy crops up in the STIMIDIO solution: dropping the centring matrices $\mathbf{J}$ in (4.55) yields

$$\mathbf{V}_j(\mathbf{Y} - \mathbf{1h}_j') = \tilde{\mathbf{V}}_j(\tilde{\mathbf{Y}} - \mathbf{1}\tilde{\mathbf{h}}_j')\mathbf{P} - \mathbf{1k}', \qquad \text{for } j = 1, \ldots, n, \tag{4.56}$$

where $\mathbf{k}$ is a translation vector of order (m x 1). Due to the indeterminacies in (4.56), it is impossible to obtain a unique centroid configuration $\mathbf{Y}$ and unique stimulus weights $\mathbf{V}_j$ in the STIMIDIO model. Therefore, the STIMIDIO solution is not unique.

In contrast with the STIMFREE model, the stimulus weights in the STIMIDIO model are not comparable over configurations. The reason is that the centroid configuration $\mathbf{Y}$ is translated differently for each configuration j. The optimal translation vectors $\mathbf{h}_j$, however, can be compared over configurations. Since these n translations can be plotted as points in m-dimensional space, the inspection of such a plot could be used to detect clusters of translation points (or 'points of view'), which may be related to some background information if the configurations represent subjects, for instance.

4.4.8 The algorithm

The algorithm corresponding to the STIMIDIO model consists of the following steps.

a) Before starting to iterate, initialize the matrices $\mathbf{T}_j$ on the optimal $\mathbf{T}_j$ found in STIMFREE, and $\mathbf{Y}$ on the optimal and uniquely weighted $\mathbf{Y}^*$ found in STIMFREE. Initialize the vectors $\mathbf{h}_j$ on $\mathbf{h}_j = \mathbf{0}$ for $j = 1, \ldots, n$.

b) For every j, compute new stimulus weights using the procedure given in section 4.4.4. By default the MATCHALS program uses the NNLS algorithm of Lawson and Hanson (1974) to restrict the weights to be nonnegative.
c) Compute new optimal translation vectors $\mathbf{h}_j$ with (4.47) for each j.
d) Determine new orthonormal matrices $\mathbf{T}_j$ in (4.41) using the procedure described in section 3.3.3 of Chapter 3.
e) Again compute new optimal translation vectors $\mathbf{h}_j$ with (4.47) for each j.
f) Compute a new centroid configuration $\mathbf{Y}$ using (4.49).
g) Evaluate loss function (4.39). If the difference between the function value in this iteration and in the previous iteration is smaller than some convergence criterion, go to step h). Otherwise go to step b).
h) Print the history of iterations. Calculate the translated $\mathbf{Y}^*$ and $\mathbf{h}_j^*$ according to (4.53) and (4.54), respectively, and print them. Also print the stimulus weights $\mathbf{V}_j$ for each j.

The reason that the translation vectors are calculated twice within one iteration (i.e., in steps c) and e) of the algorithm) is that, in comparison with the $\mathbf{V}_j$, $\mathbf{T}_j$, and $\mathbf{Y}$, the estimation of the $\mathbf{h}_j$ is relatively cheap in terms of computation time. The STIMIDIO algorithm has the same problem as the STIMFREE algorithm: it easily converges to a local minimum. Moreover, in practice we have found this algorithm sometimes to be very slow in convergence. Examples of both problems are discussed in section 4.6.2. We must warn the reader that the calculation of $\mathbf{Y}^*$ and $\mathbf{h}_j^*$ according to (4.53) and (4.54) in step h) of the algorithm does *not* result in a unique solution.

Although we will provide an analysis of variation for the STIMIDIO model in the following section, and discuss two examples of the analysis of configurations with this model in section 4.6.2, this will mainly be done for the sake of completeness. Considering all the problems involved in STIMIDIO we advise against using this model. Possibly better results would be obtained by imposing extra restrictions on the parameters of the STIMIDIO model. One obvious restriction would be to require that $(\text{diag } \mathbf{YY}') = \mathbf{I}_p$. However, we have not investigated this option.

4.5 Analysis of variation in the stimulus weighting models

4.5.1 Introduction

The following sections deal with the partitioning of the total sum of squares of the n configurations for the stimulus weighting models. In section 4.5.2 we discuss the analysis of variation for the STIMFREE model (4.5); the analysis of variation for the STIMIDIO model (4.7) is presented in section 4.5.3.

4.5.2 Partitioning of the sums of squares in STIMFREE

The linear model (4.13) underlying the STIMFREE model (4.5) is (see section 4.3.2)

$$\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j = \mathbf{C}_j\mathbf{V}_j\mathbf{Y} + \mathbf{C}_j\mathbf{E}_j \quad \text{for } j = 1, 2, ..., n,$$

from which it follows that

$$\text{tr } \mathbf{T}_j'\tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j = \text{tr } (\mathbf{V}_j\mathbf{Y} + \mathbf{E}_j)'\mathbf{C}_j(\mathbf{V}_j\mathbf{Y} + \mathbf{E}_j),$$

and therefore that

$$\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j = \text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{Y} + \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j + 2 \text{ tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{E}_j. \tag{4.57}$$

Whether we use unrestricted or nonnegative weights (see section 4.3.4) it is true for optimal $\mathbf{V}_j$ that $(\text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{E}_j) = 0$. Defining $\mathbf{b}_j$, $\mathbf{b}_j^*$, $\mathbf{v}_j$, $\mathbf{v}_j^*$, and $\mathbf{D}_j$ as we did in section 4.3.4, the latter trace may be written as

$$\begin{aligned}\text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{E}_j &= \text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j - \text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{Y} \\ &= \mathbf{b}_j'\mathbf{v}_j - \mathbf{v}_j'(\mathbf{C}_j\otimes\mathbf{Y}\mathbf{Y}')\mathbf{v}_j = \mathbf{b}_j^{*\prime}\mathbf{v}_j^* - \mathbf{v}_j^{*\prime}\mathbf{D}_j\mathbf{v}_j^*. \end{aligned} \tag{4.58}$$

In the case of unrestricted weights the optimal vector $\mathbf{v}_j^*$ is calculated using (4.19), and substitution of (4.19) in (4.58) yields

$$\text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{E}_j = \mathbf{b}_j^{*\prime}\mathbf{D}_j^{-1}\mathbf{b}_j^* - \mathbf{b}_j^{*\prime}\mathbf{D}_j^{-1}\mathbf{D}_j\mathbf{D}_j^{-1}\mathbf{b}_j^* = 0. \tag{4.59}$$

The same holds for nonnegative stimulus weights. We skip the proof. Thus, (4.57) may be written as

$$\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j = \text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{Y} + \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j. \tag{4.60}$$

It immediately follows from (4.60) that

$$0 \leq (\text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{Y})/(\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j) \leq 1, \qquad \text{for } j = 1, 2, \ldots, n, \tag{4.61}$$

showing that (4.61) is perfectly suited as a measure of fit for each separate configuration j in the STIMFREE solution. It further follows from (4.60) that

$$\sum_{j=1}^{n} \text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j = \sum_{j=1}^{n} \text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{Y} + \sum_{j=1}^{n} \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j,$$

and therefore that

$$n = \text{tr } \mathbf{Y}'(\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)\mathbf{Y} + f(\mathbf{T},\mathbf{V},\mathbf{Y}), \tag{4.62}$$

with $f(\mathbf{T},\mathbf{V},\mathbf{Y})$ as in (4.12), and thus that

$$0 \leq (1/n) \text{ tr } \mathbf{Y}'(\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)\mathbf{Y} \leq 1. \tag{4.63}$$

Hence, (4.63) is a measure of total fit in the STIMFREE model.

The measure of configuration fit (4.61) is equal to the squared correlation between the elements of $\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j$ and the elements of $\mathbf{C}_j\mathbf{V}_j\mathbf{Y}$. This is true because

$$r^2(\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j,\mathbf{C}_j\mathbf{V}_j\mathbf{Y}) = \frac{(\text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j)^2}{(\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j)(\text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{Y})}$$

$$= (\text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{Y})/(\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j), \tag{4.64}$$

due to the fact that $(\text{tr } \mathbf{Y}'\mathbf{V}_j\mathbf{C}_j\mathbf{E}_j) = 0$.

Table 4.1 Contribution of individual configurations in the STIMFREE model.

configuration j	SS_{fit}	$SS_{residual}$	SS_{total}
1	$\mathrm{tr}\,\mathbf{Y}'\mathbf{V}_1\mathbf{C}_1\mathbf{V}_1\mathbf{Y}$	$\mathrm{tr}\,\mathbf{E}_1'\mathbf{C}_1\mathbf{E}_1$	$\mathrm{tr}\,\tilde{\mathbf{X}}_1'\mathbf{C}_1\tilde{\mathbf{X}}_1$
2	$\mathrm{tr}\,\mathbf{Y}'\mathbf{V}_2\mathbf{C}_2\mathbf{V}_2\mathbf{Y}$	$\mathrm{tr}\,\mathbf{E}_2'\mathbf{C}_2\mathbf{E}_2$	$\mathrm{tr}\,\tilde{\mathbf{X}}_2'\mathbf{C}_2\tilde{\mathbf{X}}_2$
⋮	⋮	⋮	⋮
n	$\mathrm{tr}\,\mathbf{Y}'\mathbf{V}_n\mathbf{C}_n\mathbf{V}_n\mathbf{Y}$	$\mathrm{tr}\,\mathbf{E}_n'\mathbf{C}_n\mathbf{E}_n$	$\mathrm{tr}\,\tilde{\mathbf{X}}_n'\mathbf{C}_n\tilde{\mathbf{X}}_n$
$\sum_{j=1}^{n}$	$\mathrm{tr}\,\mathbf{Y}'(\sum_{j=1}^{n}\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)\mathbf{Y}$	$\sum_{j=1}^{n}\mathrm{tr}\,\mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j$	n

The results obtained thus far can be used to further decompose the sums of squares in (4.62) allowing one to assess the relative contribution of individual configurations, stimuli and dimensions to the STIMFREE solution. Table 4.1 shows the decomposition of (4.62) with respect to individual configurations, and is, in fact, nothing more than (4.60) in tabulated form.

Table 4.2 Contribution of individual stimuli in the STIMFREE model.

stimulus i	SS_{fit}	$SS_{residual}$	SS_{total}
1	a_{11}*	b_{11}**	d_{11}***
2	a_{22}*	b_{22}**	d_{22}***
⋮	⋮	⋮	⋮
p	a_{pp}*	b_{pp}**	d_{pp}***
$\sum_{i=1}^{p}$	$\mathrm{tr}\,\mathbf{Y}'(\sum_{j=1}^{n}\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)\mathbf{Y}$	$\sum_{j=1}^{n}\mathrm{tr}\,\mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j$	n

* a_{ii} is diagonal element ii of $\sum_{j=1}^{n}\{2\,\mathbf{C}_j\mathbf{V}_j\mathbf{Y}(\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j)' - \mathbf{C}_j\mathbf{V}_j\mathbf{Y}(\mathbf{C}_j\mathbf{V}_j\mathbf{Y})'\}$

** b_{ii} is diagonal element ii of $\sum_{j=1}^{n}\mathbf{C}_j\mathbf{E}_j(\mathbf{C}_j\mathbf{E}_j)'$

*** d_{ii} is diagonal element ii of $\sum_{j=1}^{n}\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j(\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j)'$

Table 4.3 Contribution of individual dimensions in the STIMFREE model.

dimension k	SS_{fit}	$SS_{residual}$	SS_{total}
1	e_{11}*	f_{11}**	g_{11}***
2	e_{22}*	f_{22}**	g_{22}***
⋮	⋮	⋮	⋮
m	e_{mm}*	f_{mm}**	g_{mm}***
$\sum_{k=1}^{m}$	$\text{tr } \mathbf{Y}'(\sum_{j=1}^{n} \mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)\mathbf{Y}$	$\sum_{j=1}^{n} \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j$	n

* e_{kk} is diagonal element kk of $\mathbf{\Lambda}$

** f_{kk} is diagonal element kk of $\sum_{j=1}^{n} (\tilde{\mathbf{X}}_j\mathbf{T}_j\mathbf{P} - \mathbf{V}_j\mathbf{Y}\mathbf{P})'\mathbf{C}_j(\tilde{\mathbf{X}}_j\mathbf{T}_j\mathbf{P} - \mathbf{V}_j\mathbf{Y}\mathbf{P})$

*** g_{kk} is diagonal element kk of $\sum_{j=1}^{n} (\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j\mathbf{P})'\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j\mathbf{P}$

In Table 4.2 the relative contribution of each individual stimulus to the STIMFREE solution is assessed. Unfortunately, these sums of squares are *not* independent, both with and without missing data. It is for this reason that we propose to calculate the amount of variation for each stimulus according to the expression given in the column titled SS_{fit} in Table 4.2, since this at least guarantees that SS_{fit} and $SS_{residual}$ add up to SS_{total} for each stimulus i, and that $\Sigma\ SS_{fit}$ is equal to the total sum of squares in (4.62). But we repeat that these calculations do not result in independent components.

The last decomposition of the sums of squares in (4.62) is given in Table 4.3. This partitioning allows one to assess the relative contribution of the dimensions to the STIMFREE solution. Since this solution may be simultaneously rotated, just like in GPA, we propose to rotate the whole solution to the principal components of $\mathbf{Y}'(\Sigma\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)\mathbf{Y}$ by determining the eigenvector-eigenvalue decomposition $\mathbf{Y}'(\Sigma\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j)\mathbf{Y} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}'$, and rotating the whole solution with $\mathbf{P}$. This has the effect that the proportion of sum of squares accounted for by the first dimension will always be the largest in STIMFREE, the proportion accounted for by the second dimension will always be second largest, etc. The sums of squares in Table 4.3 can be determined independently both with and without missing data.

4.5.3 Partitioning of the sums of squares in STIMIDIO

The linear model (4.40) corresponding to the STIMIDIO model (4.7) is (see section 4.4.2):

$$\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j = \mathbf{C}_j\mathbf{V}_j(\mathbf{Y} - \mathbf{1}\mathbf{h}_j') + \mathbf{C}_j\mathbf{E}_j \qquad \text{for } j = 1, 2, ..., n.$$

Defining $\mathbf{B}_j = (\mathbf{Y} - \mathbf{1}\mathbf{h}_j')$, and applying the same procedures to (4.40) as those discussed in the previous section, it is easily verified that the following relations hold for the STIMIDIO model as well:

$$\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j = \text{tr } \mathbf{B}_j'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{B}_j + \text{tr } \mathbf{E}_j'\mathbf{C}_j\mathbf{E}_j, \quad \text{for } j = 1, 2, ..., n. \tag{4.65}$$

Therefore

$$0 \leq (\text{tr } \mathbf{B}_j'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{B}_j)/(\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j) \leq 1, \tag{4.66}$$

and thus

$$(\text{tr } \mathbf{B}_j'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{B}_j)/(\text{tr } \tilde{\mathbf{X}}_j'\mathbf{C}_j\tilde{\mathbf{X}}_j) = r^2(\mathbf{C}_j\tilde{\mathbf{X}}_j\mathbf{T}_j, \mathbf{C}_j\mathbf{V}_j\mathbf{B}_j) \tag{4.67}$$

is a measure of fit for each separate configuration j in STIMIDIO. Again, it follows from (4.65) that

$$n = \sum_{j=1}^{n} \text{tr } \mathbf{B}_j'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{B}_j + f(\mathbf{H},\mathbf{T},\mathbf{V},\mathbf{Y}), \tag{4.68}$$

with $f(\mathbf{H},\mathbf{T},\mathbf{V},\mathbf{Y})$ as defined in (4.39); therefore

$$0 \leq (1/n)\sum_{j=1}^{n} \text{tr } \mathbf{B}_j'\mathbf{V}_j\mathbf{C}_j\mathbf{V}_j\mathbf{B}_j \leq 1,$$

is a measure of total fit in the STIMIDIO model.

For a further decomposition of the sums of squares in (4.68) with respect to individual configurations, stimuli and dimensions we refer to Tables 4.1, 4.2, and 4.3, respectively, where in all tables one should now read $\mathbf{B}_j = \mathbf{Y} - \mathbf{1}\mathbf{h}_j'$ instead of $\mathbf{Y}$. The MATCHALS program computes and prints all decompositions discussed in this and the previous section.

4.6 Illustrations

In the next two sections constructed data sets are matched with the stimulus weighting models. The algorithms presented in sections 4.3.8 and 4.4.8 were programmed in APL, and we used a convergence criterion of 1E-7 throughout. In section 4.6.1 the results of two analyses of a constructed data set with the STIMFREE model are given, and in section 4.6.2 we present the results of the STIMIDIO analysis of a constructed data set.

4.6.1 STIMFREE analysis of a constructed data set

To investigate the performance of the STIMFREE algorithm, we constructed a data set such that we knew in advance that a perfect solution exists when the admissible transformations are those corresponding to STIMFREE model (4.5). The 'mould' of this example consists of the coordinates of a regular octagon, a two-dimensional figure, which we located in three-dimensional space. The coordinates of the octagon are given in matrix **A** in Table 4.4. Since the first column of **A** contains zero elements only, the octagon is situated in the plane spanned by the y- and z-axes, and, therefore, orthogonal to the x-axis. This also means that we used a singular matrix as mould (i.e., **A** has rank two instead of three).

By translating this octagon located in three-dimensional space with the vector $\mathbf{t}' = (2.5, 0, 0)$, and by differently weighting the vectors obtained by connecting the vertices of the translated octagon with the origin, a set of four configurations was constructed. The last step in the construction consisted of a different rotation of each configuration. In this way four matrices of full column rank were obtained from a rank-deficient mould. Table 4.4 contains all transformation parameters that we used to construct this data set. By calculating $\mathbf{X}_j = \mathbf{V}_j(\mathbf{A} - \mathbf{j}\mathbf{t}')\mathbf{R}_j$ for $j = 1, \ldots, 4$ we obtained the data given in Table 4.5. The geometry of the octagon in two dimensions, and its location in three-dimensional space after translation is shown in Figure 4.5.

We wanted to find out whether the STIMFREE algorithm would recover the common structure in the data of Table 4.5. A generalized Procrustes analysis of these data yielded a loss of 0.7129, which means that the relative distances preserving model accounts for $(4 - 0.7129)/4 = 82\%$ of the total variation in the data.

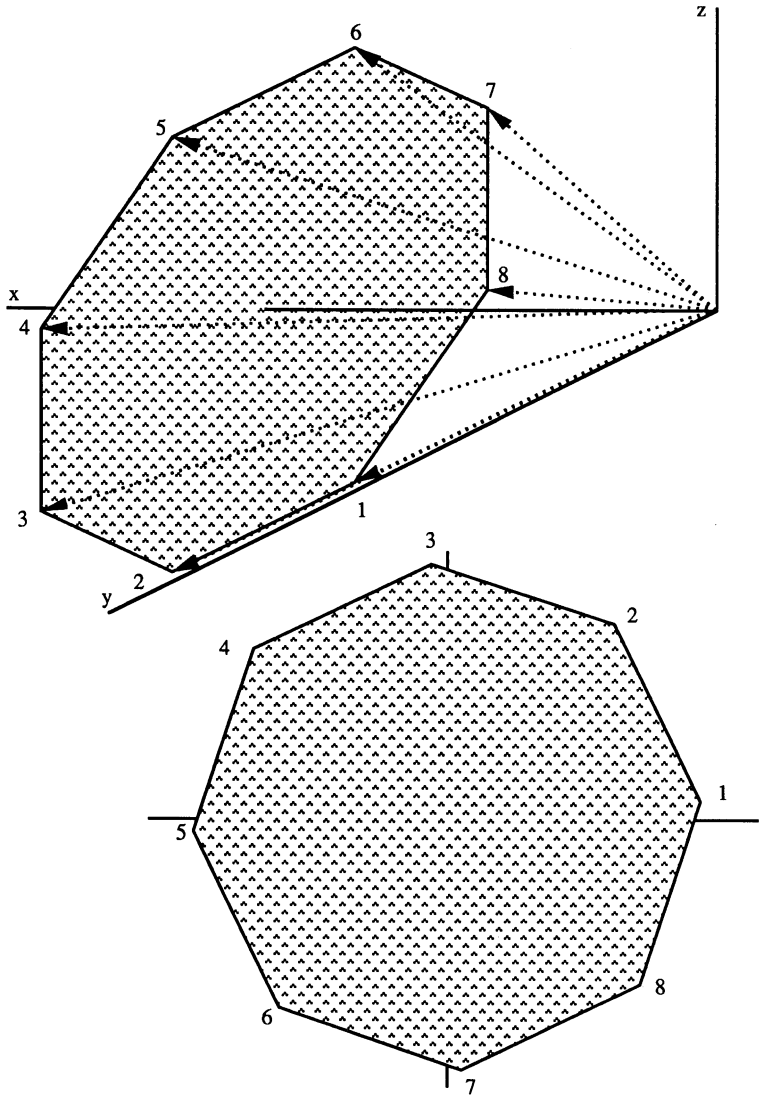


Figure 4.5 Geometry of mould (**A** – **1t'**) in three dimensions (top), and of **A** in two dimensions (bottom).

Table 4.4 Matrices and vector used in the construction of data in Table 4.5.

$$\mathbf{A} = \begin{bmatrix} 0 & -0.5000 & -1.2071 \\ 0 & 0.5000 & -1.2071 \\ 0 & 1.2071 & -0.5000 \\ 0 & 1.2071 & 0.5000 \\ 0 & 0.5000 & 1.2071 \\ 0 & -0.5000 & 1.2071 \\ 0 & -1.2071 & 0.5000 \\ 0 & -1.2071 & -0.5000 \end{bmatrix} \qquad \mathbf{t} = \begin{bmatrix} 2.5 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{R}_1 = \begin{bmatrix} -0.6077 & 0.5803 & -0.5422 \\ 0.7527 & 0.2030 & -0.6263 \\ -0.2534 & -0.7887 & -0.5601 \end{bmatrix} \qquad \mathbf{R}_2 = \begin{bmatrix} 0.9761 & -0.0977 & 0.1940 \\ 0.0487 & 0.9688 & 0.2430 \\ 0.2117 & 0.2277 & -0.9504 \end{bmatrix}$$

$$\mathbf{R}_3 = \begin{bmatrix} 0.2592 & -0.3102 & -0.9146 \\ -0.2659 & 0.8875 & -0.3764 \\ -0.9285 & -0.3408 & -0.1476 \end{bmatrix} \qquad \mathbf{R}_4 = \begin{bmatrix} -0.2315 & -0.1672 & -0.9584 \\ 0.3997 & 0.8818 & -0.2504 \\ 0.8870 & -0.4410 & -0.1373 \end{bmatrix}$$

stimulus weights $\mathbf{V}_j$

	stimulus							
configuration	1	2	3	4	5	6	7	8
1	0.900	0.700	0.600	0.800	1.000	0.300	0.400	0.500
2	1.000	0.750	0.250	0.375	1.375	0.875	1.500	0.500
3	1.000	1.500	0.900	1.300	0.100	0.300	1.200	1.400
4	0.100	0.600	0.500	1.100	0.300	0.700	1.000	0.900

Table 4.5 Constructed data for analysis with STIMFREE model.

$$\mathbf{X}_1 = \begin{bmatrix} 1.3039 & -0.5401 & 2.1103 \\ 1.5410 & -0.2780 & 1.2029 \\ 1.5327 & -0.4867 & 0.5277 \\ 1.8409 & -1.2800 & 0.2556 \\ 1.5898 & -2.3012 & 0.3663 \\ 0.2511 & -0.7513 & 0.2978 \\ 0.1936 & -0.8360 & 0.7326 \\ 0.3687 & -0.6507 & 1.1958 \end{bmatrix} \quad \mathbf{X}_2 = \begin{bmatrix} -2.7202 & -0.5151 & 0.5407 \\ -2.0036 & 0.3403 & 0.5878 \\ -0.6219 & 0.3250 & 0.0709 \\ -0.8534 & 0.5728 & -0.2501 \\ -2.9706 & 1.3799 & -2.0774 \\ -1.9330 & 0.0304 & -1.5346 \\ -3.5898 & -1.2170 & -1.8803 \\ -1.3025 & -0.5195 & -0.1516 \end{bmatrix}$$

$$\mathbf{X}_3 = \begin{bmatrix} 0.6056 & 0.7431 & 2.6529 \\ 0.5096 & 2.4459 & 3.4149 \\ -0.4544 & 1.8155 & 1.7155 \\ -1.8633 & 2.1793 & 2.2860 \\ -0.1902 & 0.0808 & 0.1920 \\ -0.4908 & -0.0239 & 0.6890 \\ -0.9497 & -0.5594 & 3.2005 \\ 0.1920 & -0.1756 & 3.9406 \end{bmatrix} \quad \mathbf{X}_4 = \begin{bmatrix} -0.0692 & 0.0509 & 0.2687 \\ -0.1753 & 0.8347 & 1.4619 \\ 0.3088 & 0.8515 & 1.0812 \\ 1.6550 & 1.3881 & 2.2276 \\ 0.5547 & 0.0980 & 0.6315 \\ 1.0146 & -0.3886 & 1.6488 \\ 0.5397 & -0.8669 & 2.6295 \\ -0.3125 & -0.3833 & 2.4901 \end{bmatrix}$$

Submitting the optimally transformed configurations from GPA to a STIMFREE analysis, it appeared that the algorithm of section 4.3.8 converged in 39 iterations. The complete history of iterations is shown in Table 4.6. The algorithm converged to a local minimum, since the loss at convergence is unequal to zero. At the local minimum, the STIMFREE model accounts for $(4 - 0.0354)/4 = 99\%$ of the total variation in the four configurations. A striking feature in the history of iterations for this data set is the sudden drop in loss at iteration number 10.

The optimal transformation parameters corresponding to the STIMFREE analysis of this data set are given in Table 4.7. In Figure 4.6 the location of the uniquely weighted centroid configuration $\mathbf{Y}^*$ rotated to principal components is shown in three-dimensional space. The coordinates of $\mathbf{Y}^*$ rotated to principal components are given at the top of Table 4.7, where $\mathbf{Q}$ is the matrix of eigenvectors in the eigenvalue-eigenvector decomposition $\mathbf{Y}^{*\prime}\mathbf{Y}^* = \mathbf{Q}\Lambda\mathbf{Q}'$. Figure 4.6 also contains a plot of the 'flattened' centroid configuration $\mathbf{Y}^*$. As coordinates of the flattened configuration, the first two eigenvectors of the eigenvalue-eigenvector decomposition $\mathbf{J}\mathbf{Y}^*(\mathbf{J}\mathbf{Y}^*)' = \mathbf{P}\Phi\mathbf{P}'$ were used, where $\mathbf{J}$ is the centring matrix for the columns of $\mathbf{Y}^*$.

As Figure 4.6 shows, the basic structure of the original mould is recovered by the STIMFREE algorithm. But the recovery is not perfect since the octagon at the bottom

Table 4.6 History of iterations of STIMFREE analysis of data in Table 4.5.

Iteration number	TOTAL LOSS stimulus weighting step	rotation step
1	0.43568246	0.41794632
2	0.38971249	0.38360589
3	0.35985999	0.35345816
4	0.31872274	0.30527786
5	0.25953362	0.23876176
6	0.20702699	0.19223683
7	0.17755962	0.16975370
8	0.16280736	0.15892717
9	0.15542006	0.14343834
10	0.04549479	0.03975107
11	0.03752761	0.03677707
12	0.03620780	0.03598696
13	0.03568135	0.03559487
14	0.03545197	0.03543438
15	0.03540051	0.03539813
16	0.03538608	0.03538524
17	0.03537998	0.03537958
18	0.03537716	0.03537696
19	0.03537583	0.03537573
20	0.03537520	0.03537515
21	0.03537490	0.03537488
22	0.03537476	0.03537475
23	0.03537469	0.03537469

of the figure is not regular. Moreover, the stimulus points of the optimal $\mathbf{Y}^*$ shown at the top of the figure are not exactly located in one plane, as the figure might suggest. This is indicated by the eigenvalues of $\mathbf{JY}^*(\mathbf{JY}^*)'$: the eigenvalues in Φ are 0.7922, 0.7415, and 0.0028. Since the third eigenvalue is unequal to zero, there is some scatter in the stimulus points of $\mathbf{Y}^*$ on the third principal component, while the scatter of the stimulus points in the original mould $\mathbf{A}$ on the third principal component is zero, of course.

Table 4.7 also contains the optimal raw stimulus weights at the local optimum of the STIMFREE loss function, together with the unique weights and the impact indices corresponding to the weights. The latter index shows how far a weight displaces the end point of the (unit normalized) stimulus vector of $\mathbf{Y}^*$. Thus, in this example the largest displacement is caused by stimulus weight 5 for configuration 1, while

Table 4.7 Results of STIMFREE algorithm for data in Table 4.5.

$$\mathbf{Y} = \begin{bmatrix} -0.1066 & 0.4310 & 0.2908 \\ 0.0441 & 0.4053 & 0.4333 \\ 0.0819 & 0.1870 & 0.4005 \\ 0.1087 & 0.0721 & 0.5863 \\ -0.0524 & -0.0062 & 0.4058 \\ -0.2536 & 0.0272 & 0.4687 \\ -0.4772 & 0.2158 & 0.5813 \\ -0.3232 & 0.3415 & 0.4097 \end{bmatrix} \quad \mathbf{Y}^*\mathbf{Q} = \begin{bmatrix} 0.8553 & 0.4946 & 0.1543 \\ 0.9070 & 0.2135 & 0.3631 \\ 0.9238 & -0.1155 & 0.3650 \\ 0.8819 & -0.3987 & 0.2515 \\ 0.9060 & -0.4148 & -0.0847 \\ 0.9035 & -0.1933 & -0.3826 \\ 0.9003 & 0.0987 & -0.4239 \\ 0.9130 & 0.3321 & -0.2368 \end{bmatrix}$$

$$\mathbf{T}_1 = \begin{bmatrix} 0.9925 & 0.1035 & -0.0656 \\ -0.0656 & 0.8339 & 0.5495 \\ 0.1115 & -0.5421 & 0.8329 \end{bmatrix} \quad \mathbf{T}_2 = \begin{bmatrix} 0.9126 & -0.1847 & -0.3647 \\ -0.0820 & 0.7912 & -0.6060 \\ -0.4005 & -0.5829 & -0.7070 \end{bmatrix}$$

$$\mathbf{T}_3 = \begin{bmatrix} 0.9508 & 0.1570 & 0.2670 \\ -0.2561 & 0.8834 & 0.3925 \\ -0.1742 & -0.4416 & 0.8801 \end{bmatrix} \quad \mathbf{T}_4 = \begin{bmatrix} 0.9979 & 0.0588 & 0.0256 \\ -0.0462 & 0.9360 & -0.3490 \\ -0.0444 & 0.3471 & 0.9368 \end{bmatrix}$$

raw stimulus weights $\mathbf{V}_j$

	stimulus							
configuration	1	2	3	4	5	6	7	8
1	1.5321	1.1892	1.2986	0.5265	0.0000	1.2753	0.8749	1.1713
2	1.1686	0.7844	0.3674	0.4129	2.1209	1.0498	1.2016	0.5109
3	1.0967	1.4514	1.1611	1.2423	0.1773	0.3506	0.8892	1.2983
4	0.2349	0.8691	0.9562	1.4980	0.6515	1.1010	1.0650	1.2177

unique stimulus weights $\mathbf{V}_j^*/(\sqrt{\text{tr}\,\mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j})$

	stimulus							
configuration	1	2	3	4	5	6	7	8
1	0.8130	0.7074	0.5836	0.3161	0.0000	0.6803	0.6844	0.7304
2	0.6307	0.4746	0.1680	0.2522	0.8826	0.5696	0.9560	0.3240
3	0.5669	0.8410	0.5083	0.7267	0.0707	0.1822	0.6776	0.7886
4	0.1260	0.5228	0.4345	0.9096	0.2695	0.5939	0.8424	0.7678

impact index of $\mathbf{V}_j^*/(\sqrt{\text{tr}\,\mathbf{X}_j'\mathbf{C}_j\mathbf{X}_j})$

	stimulus							
configuration	1	2	3	4	5	6	7	8
1	0.1870	0.2926	0.4164	0.6839	1.0000	0.3197	0.3156	0.2696
2	0.3693	0.5254	0.8320	0.7478	0.1174	0.4304	0.0440	0.6760
3	0.4331	0.1590	0.4917	0.2733	0.9293	0.8178	0.3224	0.2114
4	0.8740	0.4772	0.5655	0.0904	0.7305	0.4061	0.1576	0.2322

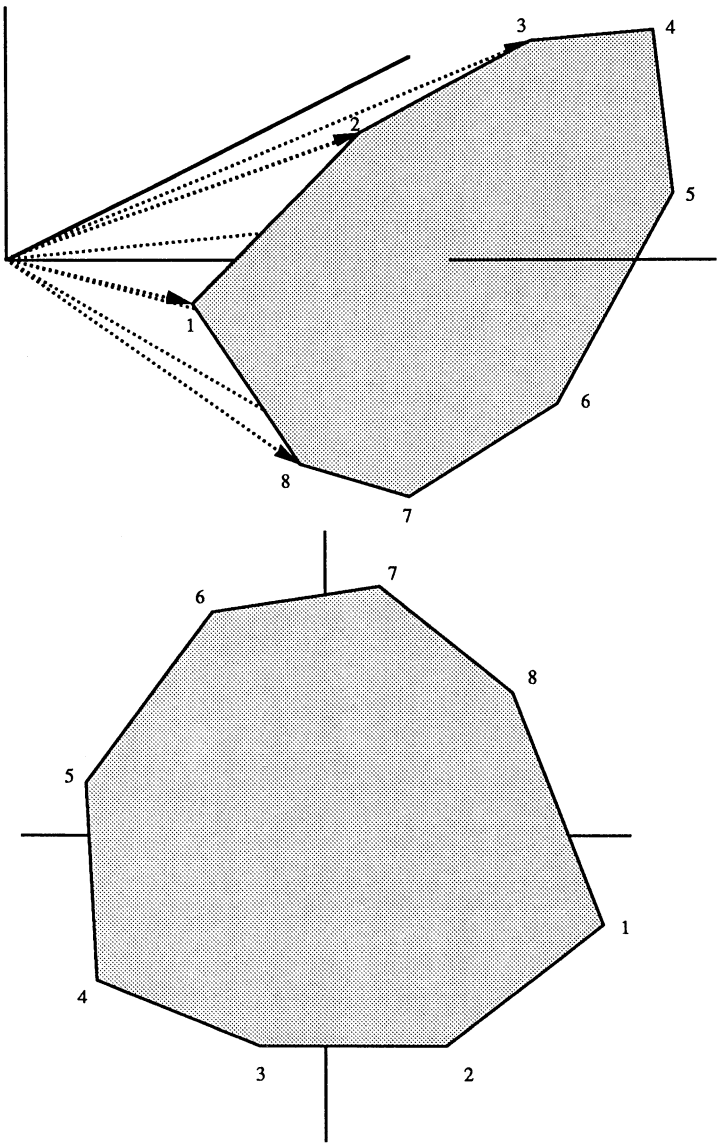


Figure 4.6 Plot of $\mathbf{Y}^*$ rotated to principal components after STIMFREE analysis of data in Table 4.5 (top), and plot of 'flattened' $\mathbf{Y}^*$ (bottom).

stimulus weight 7 for configuration 2 hardly displaces stimulus point 7 of $\mathbf{Y}^*$ at all. Averaged over the four configurations it is stimulus 5 which needs the largest displacement (the mean impact index for this stimulus is 0.6943), while the mean impact for stimulus 7 is the smallest (i.e., 0.2099).

That the weight for stimulus 5 in configuration 1 is equal to zero indicates that this weight would have been negative if we had not used the nonnegative least squares algorithm of Lawson and Hanson (1974) to estimate the weights.

We also analysed the four configurations in Table 4.5 treating the following rows as missing: rows 1, 3, 6, and 8 of configuration 2, rows 1, 3, 5, and 7 of configuration 3, and rows 3 and 7 of configuration 4. A generalized Procrustes analysis of these incomplete configurations yielded a loss of 0.4427, and, thus, a fit of $(4 - 0.4427)/4 = 0.89$.

The algorithm for the STIMFREE model converged in 195 iterations to a local minimum, since the loss at convergence was 0.0469 and not zero. The transformation parameters at the local minimum are given in Table 4.8, and Figure 4.7 contains a plot of the uniquely weighted $\mathbf{Y}^*$ in three dimensions as well as a 'flattened' version of $\mathbf{Y}^*$.

At the local minimum the STIMFREE model accounts for $(4 - 0.0389)/4 = 99\%$ of the total variation in the four incomplete configurations. At the top of Figure 4.7 a plot of $\mathbf{Y}^*$ is shown rotated to its principal components. The corresponding coordinates are given at the top of Table 4.8, where $\mathbf{Q}$ is the matrix of eigenvectors in the eigenvalue-eigenvector decomposition $\mathbf{Y}^{*\prime}\mathbf{Y}^* = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}'$. The coordinates of the 'flattened' version of $\mathbf{Y}^*$ at the bottom of Figure 4.7 are the first two eigenvectors of matrix $\mathbf{P}$ in the eigenvalue-eigenvector decomposition $\mathbf{J}\mathbf{Y}^*(\mathbf{J}\mathbf{Y}^*)' = \mathbf{P}\mathbf{\Phi}\mathbf{P}'$, where $\mathbf{J}$ is the centring matrix for the columns of $\mathbf{Y}^*$. The flattened version of $\mathbf{Y}^*$ illustrates particularly well how much (or, perhaps more appropriately: how little) of the original regular octagon has been preserved in the STIMFREE solution. Stimuli 1 through 5 still more or less form the contour of the original octagon, but the location of stimuli 6, 7, and 8 is quite different from what we would expect. Moreover, information is lost in the flattened version of $\mathbf{Y}^*$. The eigenvalues in $\mathbf{\Phi}$ are 1.9181, 1.4338, and 0.2839, showing that there is some scatter in the stimulus points of $\mathbf{Y}^*$ on the third principal component.

The impact indices corresponding to the unique stimulus weights in Table 4.8 show that stimulus 5 for configuration 2 needs the largest displacement, while

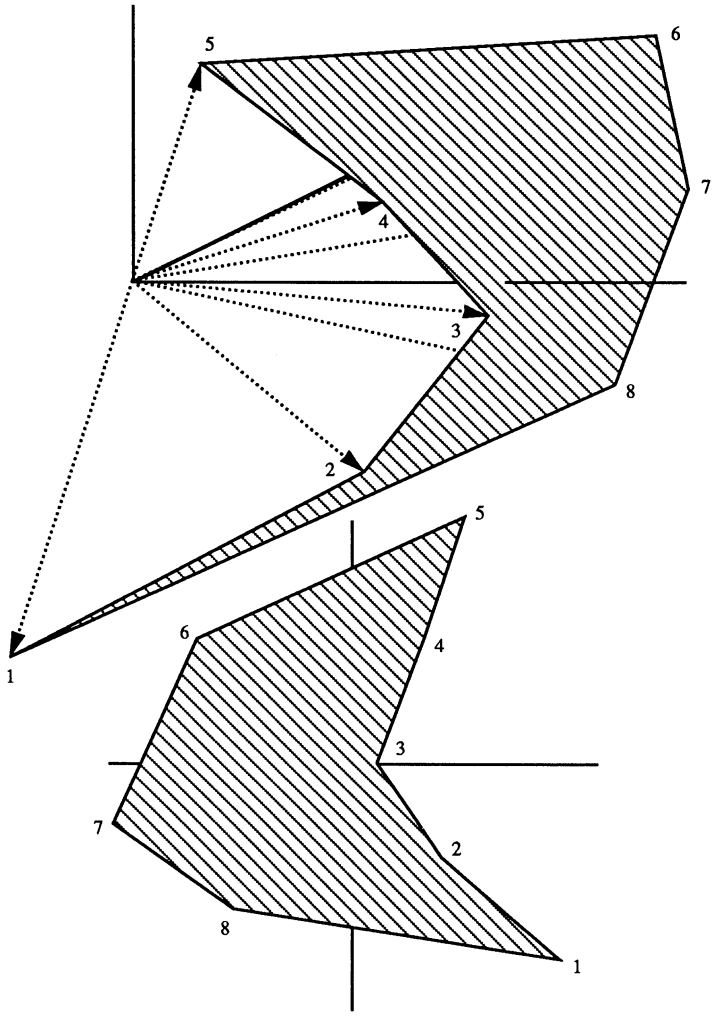


Figure 4.7 Plot of $\mathbf{Y}^*$ rotated to principal components after STIMFREE analysis of incomplete configurations of Table 4.5 (top), and plot of 'flattened' $\mathbf{Y}^*$ (bottom).

Table 4.8 Results of STIMFREE analysis of configurations in Table 4.5 containing missing data.

$$\mathbf{Y} = \begin{bmatrix} 0.1684 & -0.3071 & 0.1562 \\ 0.1947 & -0.2083 & 0.5434 \\ 0.1923 & 0.0042 & 0.6994 \\ 0.4848 & 0.2177 & 0.6786 \\ 0.3677 & 0.2415 & 0.1830 \\ -0.0930 & 0.4020 & 0.4974 \\ -0.3685 & 0.1475 & 0.3632 \\ -0.2334 & -0.0880 & 0.5540 \end{bmatrix} \quad \mathbf{Y}^*\mathbf{Q} = \begin{bmatrix} 0.4099 & 0.7076 & -0.5756 \\ 0.8957 & 0.3265 & -0.3017 \\ 0.9934 & 0.1100 & -0.0338 \\ 0.8968 & 0.2773 & 0.3448 \\ 0.5630 & 0.3919 & 0.7276 \\ 0.7811 & -0.5115 & 0.3581 \\ 0.5581 & -0.8114 & -0.1739 \\ 0.8088 & -0.3818 & -0.4472 \end{bmatrix}$$

$$\mathbf{T}_1 = \begin{bmatrix} 0.6781 & 0.1661 & 0.7160 \\ 0.0037 & 0.9734 & -0.2292 \\ -0.7350 & 0.1581 & 0.6594 \end{bmatrix} \quad \mathbf{T}_2 = \begin{bmatrix} 0.9841 & 0.0473 & -0.1711 \\ -0.1326 & 0.8366 & -0.5316 \\ 0.1180 & 0.5458 & 0.8296 \end{bmatrix}$$

$$\mathbf{T}_3 = \begin{bmatrix} 0.9609 & 0.0465 & -0.2731 \\ 0.0965 & 0.8679 & 0.4872 \\ 0.2597 & -0.4945 & 0.8295 \end{bmatrix} \quad \mathbf{T}_4 = \begin{bmatrix} 0.9969 & 0.0419 & 0.0659 \\ -0.0442 & 0.9985 & 0.0336 \\ -0.0644 & -0.0364 & 0.9973 \end{bmatrix}$$

raw stimulus weights $\mathbf{V}_j$

	stimulus							
configuration	1	2	3	4	5	6	7	8
1	1.5537	1.0312	0.9199	0.9159	1.7497	0.6305	0.4089	0.3055
2	—	1.1950	—	0.8276	0.0000	—	1.2666	—
3	—	0.3647	—	0.8267	—	1.3470	—	1.1738
4	0.1752	0.8420	—	1.1650	0.4158	0.8718	—	1.2540

unique stimulus weights $\mathbf{V}_j^*/(\sqrt{\text{tr}\,\mathbf{X}_j^t\mathbf{C}_j\mathbf{X}_j})$

	stimulus							
configuration	1	2	3	4	5	6	7	8
1	0.5069	0.5384	0.5677	0.6716	0.7092	0.3467	0.1872	0.1579
2	—	0.8125	—	0.7903	0.0000	—	0.7550	—
3	—	0.2736	—	0.8710	—	1.0642	—	0.8718
4	0.0631	0.4851	—	0.9427	0.1860	0.5290	—	0.7153

impact index for $\mathbf{V}_j^*/(\sqrt{\text{tr}\,\mathbf{X}_j^t\mathbf{C}_j\mathbf{X}_j})$

	stimulus							
configuration	1	2	3	4	5	6	7	8
1	0.4981	0.4616	0.4323	0.3284	0.2908	0.6533	0.8128	0.8421
2	—	0.1875	—	0.2097	1.0000	—	0.2450	—
3	—	0.7264	—	0.1290	—	0.0642	—	0.1282
4	0.8984	0.5149	—	0.0573	0.8140	0.4710	—	0.2847

stimulus 6 for configuration 3 and stimulus 4 for configuration 4 are almost not displaced at all. Averaged over configurations, stimulus 5 again needs the largest displacement (the mean index for this stimulus being 0.7016), but the mean impact for stimulus 4 is now the smallest (i.e., 0.1811). The zero weight for stimulus 5 corresponding to configuration 2 indicates that a negative weight would have been obtained if we had not restricted the weights to be nonnegative.

Summarizing, both examples illustrate that the STIMFREE algorithm is not guaranteed to converge to a global minimum, even if a perfect solution exists.

4.6.2 STIMIDIO analysis of a constructed data set

Again, we constructed a data set such that we knew in advance that a perfect solution exists when the admissible transformations are those corresponding to the STIMIDIO model. In the construction of this data set the same transformation parameters were used as in the previous section, except that the regular octagon was translated *differently* for each configuration. The following translation vectors were applied: $\mathbf{t}'_1 = (2.5, 0, 0)$, $\mathbf{t}'_2 = (0.8, -1.5, 3.0)$, $\mathbf{t}'_3 = (5.2, 2.2, -4.0)$, and $\mathbf{t}'_4 = (-0.4, 0, -3.0)$, and four configurations were constructed by calculating $\mathbf{X}_j = \mathbf{V}_j(\mathbf{A} - \mathbf{1}\mathbf{t}'_j)\mathbf{R}_j$ for $j = 1, \ldots, 4$, with $\mathbf{A}$, $\mathbf{V}_j$, and $\mathbf{R}_j$ as given in Table 4.4. This resulted in the data shown in Table 4.9.

A generalized Procrustes analysis of these four complete configurations yielded a loss of 1.3432 in 6 iterations, which means that the GPA model accounts for 66% of the total variation in these configurations. When the results from GPA were submitted to the STIMFREE algorithm of section 4.3.8, this algorithm converged in 126 iterations yielding a loss of 0.1779. Thus, 96% of the total variation is accounted for by the STIMFREE model.

Table 4.9 Constructed data for analysis with STIMIDIO model.

$$\mathbf{X}_1 = \begin{bmatrix} 1.3039 & -0.5401 & 2.1103 \\ 1.5410 & -0.2780 & 1.2029 \\ 1.5327 & -0.4867 & 0.5277 \\ 1.8409 & -1.2800 & 0.2556 \\ 1.5898 & -2.3012 & 0.3663 \\ 0.2511 & -0.7513 & 0.2978 \\ 0.1936 & -0.8360 & 0.7326 \\ 0.3687 & -0.6507 & 1.1958 \end{bmatrix} \quad \mathbf{X}_2 = \begin{bmatrix} -1.6229 & 0.0889 & 4.0863 \\ -1.1807 & 0.7933 & 3.2470 \\ -0.3475 & 0.4760 & 0.9573 \\ -0.4419 & 0.7993 & 1.0795 \\ -1.4618 & 2.2103 & 2.7978 \\ -0.9728 & 0.5588 & 1.5678 \\ -1.9438 & -0.3111 & 3.4381 \\ -0.7538 & -0.2176 & 1.6212 \end{bmatrix}$$

$$\mathbf{X}_3 = \begin{bmatrix} -3.2233 & -1.7349 & 5.3601 \\ -5.2338 & -1.2712 & 7.4756 \\ -3.9004 & -0.4148 & 4.1519 \\ -6.8409 & -1.0421 & 5.8053 \\ -0.5731 & -0.1670 & 0.4627 \\ -1.6395 & -0.7673 & 1.5011 \\ -5.5443 & -3.5331 & 6.4491 \\ -5.1685 & -3.6448 & 7.7306 \end{bmatrix} \quad \mathbf{X}_4 = \begin{bmatrix} 0.1298 & -0.1298 & -0.0504 \\ 1.0185 & -0.2500 & -0.4528 \\ 1.3036 & -0.0524 & -0.5144 \\ 3.8436 & -0.6004 & -1.2827 \\ 1.1516 & -0.4444 & -0.3258 \\ 2.4074 & -1.6541 & -0.5850 \\ 2.5293 & -2.6747 & -0.5616 \\ 1.4781 & -2.0104 & -0.3819 \end{bmatrix}$$

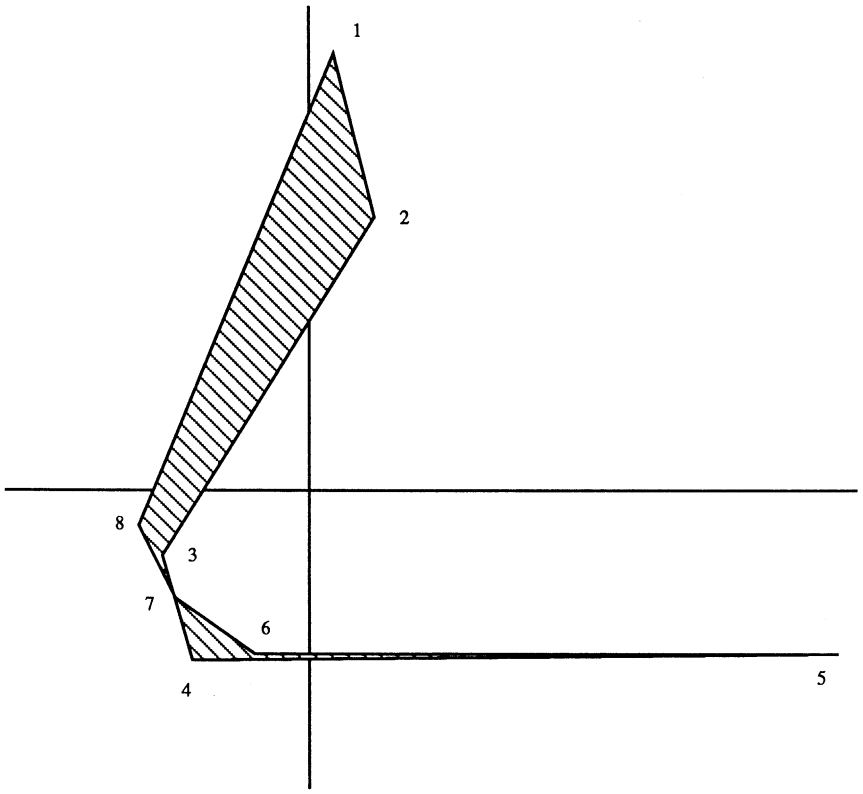


Figure 4.8 Plot of 'flattened' non-unique centroid configuration $\mathbf{Y}^*$ after STIMIDIO analysis of data in Table 4.9.

Then the optimally transformed $\mathbf{X}_j$'s from GPA and the optimal $\mathbf{Y}^*$ obtained in STIMFREE were used as input to the STIMIDIO algorithm of section 4.4.8. Convergence was very slow: it took the algorithm 1660 iterations to converge. The value of the STIMIDIO loss function at convergence was 0.0076, meaning that the solution accounts for almost all of the variation in the configurations. Since the solution was not perfect, however, the STIMIDIO algorithm converged to a local minimum.

The results of the STIMIDIO analysis are given in Table 4.10, where $\mathbf{Y}^*$ denotes the column centered centroid configuration and $\mathbf{h}_j^*$ is a relocated translation vector (cf., section 4.4.7). We emphasize that the parameters in the table are *not* unique. Figure 4.8 contains a plot of the two principal components of $\mathbf{Y}^*$. Clearly, the original mould

Table 4.10 Results of STIMIDIO algorithm for complete configurations in Table 4.9.

$$\mathbf{Y} = \begin{bmatrix} 0.1194 & -1.7662 & -1.0168 \\ -0.0694 & -1.8490 & -1.0795 \\ -0.0200 & -1.3124 & -0.5568 \\ -0.1076 & -1.4060 & -0.5824 \\ -0.5885 & -3.2111 & -1.6892 \\ -0.0233 & -1.6756 & -0.6123 \\ 0.0808 & -1.4554 & -0.4829 \\ 0.1265 & -1.3361 & -0.4587 \end{bmatrix} \qquad \mathbf{Y}^* = \begin{bmatrix} 0.1797 & -0.0148 & -0.2069 \\ -0.0091 & -0.0976 & -0.2697 \\ 0.0403 & 0.4391 & 0.2530 \\ -0.0474 & 0.3455 & 0.2274 \\ -0.5282 & -1.4596 & -0.8794 \\ 0.0370 & 0.0759 & 0.1975 \\ 0.1410 & 0.2960 & 0.3269 \\ 0.1867 & 0.4154 & 0.3512 \end{bmatrix}$$

$$\mathbf{h}_1' = [\ 0.3197 \quad -0.0551 \quad 0.4582\ ] \qquad \mathbf{h}_1^{*\prime} = [\ -0.3800 \quad -1.6964 \quad -1.2681\ ]$$

$$\mathbf{h}_2' = [\ -0.9083 \quad 1.0728 \quad 0.2676\ ] \qquad \mathbf{h}_2^{*\prime} = [\ 0.8480 \quad -2.8242 \quad -1.0774\ ]$$

$$\mathbf{h}_3' = [\ 0.1373 \quad -0.9825 \quad -0.2175\ ] \qquad \mathbf{h}_3^{*\prime} = [\ -0.1976 \quad -0.7689 \quad -0.5923\ ]$$

$$\mathbf{h}_4' = [\ 0.1071 \quad -1.1123 \quad -0.3810\ ] \qquad \mathbf{h}_4^{*\prime} = [\ -0.1674 \quad -0.6391 \quad -0.4288\ ]$$

$$\mathbf{T}_1 = \begin{bmatrix} 0.7598 & -0.4744 & -0.4446 \\ 0.5507 & 0.8331 & 0.0520 \\ 0.3457 & -0.2843 & 0.8942 \end{bmatrix} \qquad \mathbf{T}_2 = \begin{bmatrix} 0.3805 & 0.7914 & 0.4784 \\ -0.6630 & 0.5941 & -0.4555 \\ -0.6447 & -0.1439 & 0.7508 \end{bmatrix}$$

$$\mathbf{T}_3 = \begin{bmatrix} 0.5675 & -0.7199 & -0.3995 \\ 0.6865 & 0.6817 & -0.2532 \\ 0.4546 & -0.1306 & 0.8811 \end{bmatrix} \qquad \mathbf{T}_4 = \begin{bmatrix} 0.9719 & -0.2351 & 0.0087 \\ 0.2057 & 0.8672 & 0.4535 \\ -0.1142 & -0.4390 & 0.8912 \end{bmatrix}$$

	stimulus weights $\mathbf{V}_j$ stimulus							
configuration	1	2	3	4	5	6	7	8
1	2.0101	1.9543	2.8346	2.5131	1.0514	2.4451	2.7935	2.9887
2	0.4358	0.5135	0.8487	0.8213	0.4026	0.6852	0.5339	0.7530
3	0.4782	0.6354	0.9249	1.2568	0.0000	0.1802	1.4389	2.0072
4	0.2462	0.4099	1.5878	2.6265	0.1754	1.3604	2.8486	3.1353

A of Table 4.4 is distorted beyond recognition in the STIMIDIO solution. In contrast with the STIMFREE model, where the stimuli of the unique centroid configuration all lie on the unit (hyper)sphere, and where the angles between the vectors representing these stimuli are uniquely determined, in STIMIDIO the vectors representing the stimuli in $\mathbf{Y}^*$ all have different lengths, and the angles between these vectors are not unique.

Table 4.11 Results of STIMIDIO algorithm for incomplete configurations in Table 4.9.

$$\mathbf{Y} = \begin{bmatrix} -0.6402 & -0.6612 & 0.7795 \\ 0.3585 & -0.0273 & 0.5257 \\ 0.9972 & -0.2586 & 0.4716 \\ 1.0498 & 0.1619 & 0.0010 \\ 1.0198 & 0.2799 & -0.6301 \\ 0.4134 & -0.6230 & -0.0998 \\ 0.3939 & -0.8694 & -0.2797 \\ 0.2602 & -1.1327 & -0.2416 \end{bmatrix} \qquad \mathbf{Y}^* = \begin{bmatrix} -1.1217 & -0.2699 & 0.7136 \\ -0.1231 & 0.3640 & 0.4599 \\ 0.5156 & 0.1327 & 0.4058 \\ 0.5683 & 0.5532 & -0.0648 \\ 0.5382 & 0.6712 & -0.6959 \\ -0.0682 & -0.2317 & -0.1656 \\ -0.0877 & -0.4781 & -0.3455 \\ -0.2214 & -0.7414 & -0.3074 \end{bmatrix}$$

$$\mathbf{h}_1' = [\ -0.2774 \quad 0.3284 \quad 0.0511\ ] \qquad \mathbf{h}_1^{*\prime} = [\ 0.7590 \quad -0.7198 \quad 0.0147\ ]$$

$$\mathbf{h}_2' = [\ 0.6731 \quad 0.0549 \quad -0.3765\ ] \qquad \mathbf{h}_2^{*\prime} = [\ -0.1915 \quad -0.4462 \quad 0.4424\ ]$$

$$\mathbf{h}_3' = [\ -0.1296 \quad -0.1081 \quad -0.7653\ ] \qquad \mathbf{h}_3^{*\prime} = [\ 0.6112 \quad -0.2832 \quad 0.8311\ ]$$

$$\mathbf{h}_4' = [\ -0.3646 \quad -0.6176 \quad 0.6585\ ] \qquad \mathbf{h}_4^{*\prime} = [\ 0.8462 \quad 0.2263 \quad -0.5927\ ]$$

$$\mathbf{T}_1 = \begin{bmatrix} 0.9677 & -0.2518 & 0.0154 \\ 0.2504 & 0.9661 & 0.0628 \\ -0.0306 & -0.0569 & 0.9979 \end{bmatrix} \qquad \mathbf{T}_2 = \begin{bmatrix} 0.7114 & -0.1077 & 0.6945 \\ 0.2952 & 0.9426 & -0.1563 \\ -0.6377 & 0.3162 & 0.7024 \end{bmatrix}$$

$$\mathbf{T}_3 = \begin{bmatrix} 0.9017 & 0.1682 & 0.3983 \\ 0.0468 & 0.8779 & -0.4766 \\ -0.4298 & 0.4484 & 0.7837 \end{bmatrix} \qquad \mathbf{T}_4 = \begin{bmatrix} 0.7713 & 0.5057 & -0.3865 \\ -0.2187 & 0.7809 & 0.5852 \\ 0.5977 & -0.3668 & 0.7129 \end{bmatrix}$$

	stimulus weights $\mathbf{V}_j$							
	stimulus							
configuration	1	2	3	4	5	6	7	8
1	0.0631	0.5037	0.4699	0.6363	0.6891	0.7728	0.5552	0.3727
2	—	0.0000	—	1.3249	0.9820	—	0.4398	—
3	—	0.2010	—	0.2779	—	0.9160	—	0.1146
4	0.6082	0.1348	—	0.5875	0.0763	0.6232	—	0.4711

Introducing missing data in the four configurations in Table 4.9, that is, treating rows 1, 3, and 6 of $\mathbf{X}_2$ as missing, rows 1, 3, 5, and 7 of $\mathbf{X}_3$ as missing, and rows 3 and 7 of $\mathbf{X}_4$ as missing, we obtained the following results. A generalized Procrustes analysis of these incomplete configurations yielded a loss of 0.7392 in 5 iterations. The STIMFREE algorithm converged in 19 iterations to a loss of 0.1070. The STIMIDIO algorithm of section 4.4.7 converged in 440 iterations, yielding a perfect

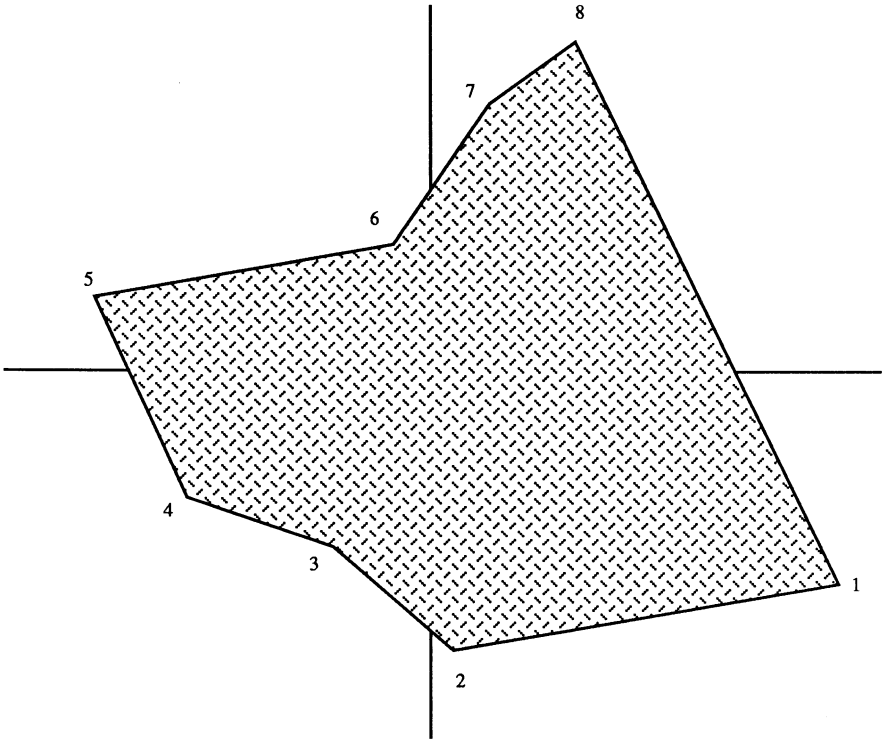
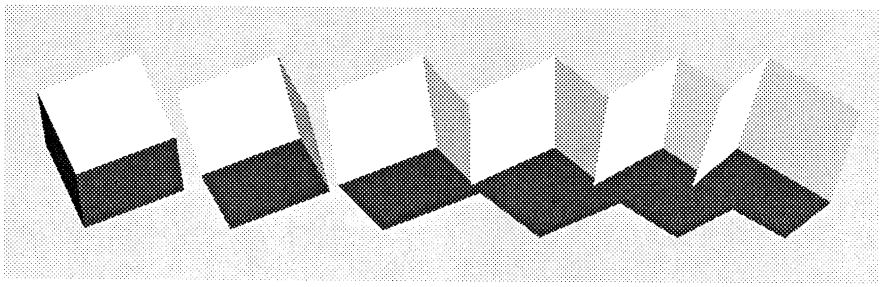


Figure 4.9 Plot of 'flattened' non-unique centroid configuration after STIMIDIO analysis of incomplete configurations.

solution of zero loss. The (non-unique) results of this latter analysis are given in Table 4.11, and Figure 4.9 contains a 'flattened' plot of the centroid configuration $\mathbf{Y}^*$.

Summarizing, these examples show that convergence of the STIMIDIO algorithm can be extremely slow, and that the algorithm easily converges to a local minimum. Because the parameters are also not uniquely determined, the STIMIDIO model is not a very attractive model, and we do not recommend its use.



Chapter 5

Conclusions

5.1 Introduction

In the previous chapters we have developed convergent algorithms for the estimation of the parameters corresponding to the MATCHALS models (1.8) through (1.12) given in Chapter 1. These models are designed to investigate the relationships between n sets of configurations containing the coordinates of the same p stimuli, and are offered as an alternative for the PINDIS models of Lingoes and Borg (1978). Each one of the MATCHALS algorithms is capable of handling m-dimensional complete as well as incomplete configurations and, unlike the PINDIS models and algorithms, nowhere is any transformation parameter being 'borrowed' from another model, except as initial estimate.

The MATCHALS models and algorithms also confront us with new problems and questions, some of which have already been mentioned, and others that we will discuss in this final chapter. First, however, we will summarize the most important conclusions of Chapters 2, 3, and 4. In section 5.2 an overview is given of the main properties of the MATCHALS algorithms. In section 5.3 we recapitulate the uniqueness properties of the MATCHALS models, and in section 5.4 we summarize the results concerning the decompositions of the sums of squares associated with the different models. The last two sections of this chapter are devoted to topics that we have not discussed so far: the statistical aspects of the models (section 5.5), and the possibility of using the models as a form of confirmatory scaling (section 5.6).

5.2 The MATCHALS algorithms

The algorithm for the estimation of the unknown parameters in the GPA model is by far the fastest of the five algorithms presented in this book. In general, it takes only a few iterations to converge, even when a strict convergence criterion is used. This algorithm iterates over two sets of parameters only: the rotation matrices and the scaling factors. Although it can not be proved that the GPA algorithm always converges to a global optimum, it seems to do so in practice.

The algorithms presented for the dimension weighting models consume considerably more CPU-time than the GPA algorithm. The number of iterations required to reach convergence is (much) larger, and within each iteration three sets of

parameters have to be estimated in the DIMFREE algorithm, while the DIMIDIO algorithm iterates over only two sets of parameters. For the estimation of a new centroid configuration in each iteration of the DIMFREE algorithm, it makes quite a difference whether the configurations are complete or not. In the case of incomplete configurations the Moore-Penrose inverse of m matrices of order (p x p+ must be calculated (where m is the number of dimensions and p the number of stimuli), while the nonmissing case only requires the computation of the proper inverse of one (m x m) diagonal matrix. In DIMIDIO the difference in CPU-time between complete and incomplete configurations with respect to the estimation of an optimal centroid configuration is even more dramatic. Without missing data, we only need to determine the proper inverse of one matrix of order (m x m), while the calculation of the Moore-Penrose inverse of a full (pm x pm) matrix is required in the case of incomplete configurations. The latter case may even prove to become impractical for large p and m. Although the DIMFREE and DIMIDIO algorithms must converge, it can not be guaranteed that they will always converge to a global minimum. However, in the case of constructed data sets with known perfect solutions we have found that they did recover the perfect solution.

The algorithms for the estimation of the unknown transformation parameters of the stimulus weighting models are also much more expensive than the GPA algorithm. A most serious disadvantage of the stimulus weighting algorithms is that they seem to converge to local minima very easily.

5.3 Uniqueness properties of the MATCHALS models

The solution of the GPA model is unique up to a simultaneous rotation of the complete solution. We have used this indeterminacy to rotate the solution to the principal components of the optimal centroid or group configuration. But any other set of dimensions in the GPA group configuration can be chosen for interpretational purposes. In GPA, the centroid configuration is always a column-centered matrix.

The latter property of the GPA group configuration is shared by the DIMFREE group configuration. In addition, the DIMFREE group configuration has an important property that is lacking in the GPA model: its axes are unique, because it is this set of axes that is differentially weighted. Therefore, what counts in the interpretation of the

DIMFREE solution is the ordering of the stimuli on these optimally oriented dimensions of the group configuration. Although the axes are unique, the *variances* of the coordinates of the stimuli on these axes are not. We have used this freedom to unit normalize the columns of the centroid configuration, yielding a DIMFREE solution that is unique up to reflections and permutations of the m dimensions (cf., section 3.3.7 of Chapter 3).

The centroid configuration in the DIMIDIO model is, again, columnwise centered, but its columns may be replaced by arbitrarily weighted linear combinations of these same columns. This indeterminacy can be used to replace the 'raw' centroid configuration by an orthonormal basis, yielding a DIMIDIO group configuration that is columnwise orthonormal. This procedure results in unique dimension weights and rotation matrices for the individual configurations, and reduces the previous indeterminacy to a mere rotational indeterminacy (cf., section 3.4.6 of Chapter 3). Since the centroid configuration is rotated idiosyncratically before its dimensions are weighted, the interpretation in DIMIDIO should focus on the idiosyncratic axes of the group configuration.

In contrast with the GPA, DIMFREE, and DIMIDIO models the group configuration obtained in the STIMFREE model will usually not be a column-centered matrix. The *angles* between the vectors generated by connecting the stimulus points in the group configuration with the origin are unique in STIMFREE. Only the lengths of the latter vectors are not determined uniquely. Because this greatly facilitates the interpretation of the stimulus weights, we have used this indeterminacy to unit normalize the rows of the centroid configuration. As a result, the end points of the stimulus vectors all lie on the unit (hyper)sphere (cf., section 4.3.7 in Chapter 4). About the properties of the STIMIDIO model we can be very brief: the STIMIDIO solution is *not* unique (cf., section 4.4.7).

5.4 Analyses of variation

In each of the five MATCHALS models the total sum of squares of the individual configurations (SS_{total}) can be partitioned in two parts: the variation accounted for by the model (SS_{fit}), and the residual variation ($SS_{residual}$). Since the configurations are always normalized such that their total variation equals n, that is, the number of

Table 5.1 Overview of the possible partitionings of SS_{tot}, SS_{fit}, and SS_{res} with respect to individual configurations, stimuli, and dimensions in the five MATCHALS models.

models	by configuration	by stimulus	by dimension
GPA	no	yes	yes
DIMFREE	yes	no	yes
DIMIDIO	yes	no	yes
STIMFREE	yes	no	yes
STIMIDIO	yes	no	yes

configurations, the partitioning takes the following form: $n = SS_{fit} + SS_{res}$. Hence, for all models $(1/n)SS_{fit}$ is a measure of total fit.

In Table 5.1 we have summarized for the five models whether SS_{total}, SS_{fit}, and $SS_{residual}$ can each be further partitioned with respect to individual configurations, stimuli and dimensions. As the table shows, the GPA model is the only model where the sums of squares can not be partitioned with respect to individual configurations (cf., section 2.5 of Chapter 2). In the remaining models the contribution of each configuration to the solution can be assessed unambigously. Moreover, the measure of configuration fit that we derived in the dimension and stimulus weighting models can be shown to be equal to the squared correlation between the elements of the optimally transformed individual configuration and the elements of the optimally weighted centroid configuration. For the stimuli the situation is completely reversed: the GPA model is the only model where SS_{tot}, SS_{fit}, and SS_{res} can be further partitioned with respect to the individual stimuli. Finally, a partition with respect to the separate dimensions can be performed in all five models.

5.5 Statistical aspects

Although measures of total fit are available for all five models, as well as measures of configuration fit for the dimension and stimulus weighting models, an important question remains. For what value of the total fit can we state that a model fits the data well? And, perhaps even more importantly, how small must the value of the fit be

before we decide to reject a particular model? One way to answer these questions, of course, is to design statistical tests. However, as yet no statistical sampling theory has been developed, and no tests are available to determine whether the values of the measures of total and configuration fit are significant or not.

A more feasible approach is to provide baselines against which to evaluate the measures of fit. This typically calls for Monte Carlo studies such as have been performed for the PINDIS models by Langeheine (1982). Baselines can be obtained by generating random configurations using varying numbers of configurations, stimuli, dimensions, and missing rows in the configurations, and then analyzing these data sets with MATCHALS.

Such studies are especially necessary for the dimension and stimulus weighting models proposed in this book. Since new centroid configurations are being estimated in these models, and because we provide better algorithms, it can be expected that, all other things being equal, the values of the measures of fit for these models on random data will be (much) higher than those found by Langeheine.

Some insight in the meaning of the fit values can be gained by comparing the number of parameters estimated in each model with the degrees of freedom of the total sum of squares. An overview of these figures is given in Table 5.2, where n, p, and m are the number of configurations, stimuli, and dimensions, and p_j denotes the number of non-missing rows of configuration $\mathbf{X}_j$. As the table shows, the number of parameters estimated in the GPA, DIMFREE, and DIMIDIO models does not depend on the number of missing rows in the configurations. Therefore, if we analyse a data

Table 5.2 Numbers of free parameters in the MATCHALS models.

model	df of SS_{tot}	estimated parameters	total number of estimated parameters
GPA	$m\sum_{j=1}^{n}(p_j - 1)$	$\mathbf{R}_j$'s, s_j's, $\mathbf{Z}$	$n\binom{m}{2} + (n-1) + m(p-1)$
DIMFREE	$m\sum_{j=1}^{n}(p_j - 1)$	$\mathbf{Q}_j$'s, $\mathbf{W}_j$'s, $\mathbf{Y}$	$n\binom{m}{2} + nm + m(p-1)$
DIMIDIO	$m\sum_{j=1}^{n}(p_j - 1)$	$\mathbf{Q}_j$'s, $\mathbf{W}_j$'s, $\mathbf{Y}$, $\mathbf{S}_j$'s	$n\binom{m}{2} + nm + m(p-1) + n\binom{m}{2}$
STIMFREE	$m\sum_{j=1}^{n}(p_j - 1)$	$\mathbf{T}_j$'s, $\mathbf{V}_j$'s, $\mathbf{Y}$	$n\binom{m}{2} + \sum_{j=1}^{n} p_j + mp$
STIMIDIO	$m\sum_{j=1}^{n}(p_j - 1)$	$\mathbf{T}_j$'s, $\mathbf{V}_j$'s, $\mathbf{Y}$, $\mathbf{h}_j$'s	$n\binom{m}{2} + \sum_{j=1}^{n} p_j + mp + nm$

set only containing complete configurations with GPA, DIMFREE or DIMIDIO, and then re-analyse the same data set treating some rows as missing, the fit of the latter solution will always be better than the fit of the solution for the complete configurations. It is important to keep in mind, however, that such an improvement in fit is only due to the fact that the same number of parameters has been estimated for a smaller number of stimulus points.

It also follows from Table 5.2 that, if we are dealing with four complete configurations of eight stimuli in three dimensions, for example, the total sum of squares has 84 degrees of freedom, while a total number of 36, 45, 57, 64, and 76 parameters is estimated in the GPA, DIMFREE, DIMIDIO, STIMFREE, and STIMIDIO models, respectively. Hence, with four configurations containing eight stimuli the ratio between the total number of estimated parameters and the degrees of freedom of the total sum of squares is quite lop-sided: the DIMFREE model already consumes more than half of the degrees of freedom.

However, the larger the number of stimuli and configurations, the better the ratio becomes. With fifteen complete three-dimensional configurations containing information about 20 stimuli, for instance, the degrees of freedom of SS_{tot} are 855, while a total number of 116, 147, 192, 405, and 450 parameters is consumed by the GPA, DIMFREE, DIMIDIO, STIMFREE, and STIMIDIO models, respectively. Therefore, it can be expected that for fifteen configurations containing 20 stimuli the fit for random data will be much smaller than for four configurations containing information about eight stimuli (although the STIMIDIO model still consumes more than half of the degrees of freedom of the total sum of squares).

But baselines are badly needed, and Monte Carlo studies will have to be performed to be able to decide to what extent a MATCHALS solution is capitalizing on chance. As long as these baselines are not available, however, the following procedure can be used to investigate the significance of the match found for a given set of configurations. Re-analyse the given data set a number of times (say k times) after a separate random permutation of the rows of each configuration, and calculate the mean and standard deviation of the observed fits, including the fit for the original data set. Then compute the *t* ratio for the latter fit, and check whether the *t* value is significant beyond the level of some a priori chosen probability α. If so, this is a strong indication that the match is not merely due to chance.

5.6 Confirmatory versus exploratory analysis

In the PINDIS computer program of Lingoes and Borg an option is available allowing the user to input an externally derived centroid configuration. Such a configuration may be self-designed and based on theoretical grounds (a circumplex, for example), or the result of a previous MDS or other analysis. If this option is chosen in the PINDIS program, all PINDIS models of Lingoes and Borg are fitted on this fixed external centroid configuration. This procedure makes it possible to investigate the match between the (optimally transformed) individual configurations and the 'hypothesis' or 'target' (i.e., the given centroid configuration) that one had in mind, and can in this sense be regarded as a form of confirmatory data analysis.

It is worth mentioning that we have also developed convergent alternating least squares algorithms that fit the GPA model and the dimension and stimulus weighting models given a fixed external centroid configuration. These algorithms have, again, been designed to handle m-dimensional as well as incomplete individual configurations. Since the default option in PINDIS already uses the centroid configuration obtained in GPA as fixed group configuration in the dimension and stimulus weighting models (cf., section 1.2 of Chapter 1), our algorithms for performing confirmatory analyses according to the latter four models are, in fact, improved and generalized versions of the algorithms proposed by Lingoes and Borg.

Appendix

An APL prototype version of the MATCHALS program in which all the algorithms presented in this book have been implemented can be obtained from the author at the following address:

Department of Psychometrics and Research Methodology
Faculty of Social Sciences, University of Leiden
P.O. Box 9555
2300 RB Leiden
The Netherlands

References

Borg, I. & Lingoes, J. (1987). *Multidimensional similarity structure analysis*. New York: Springer Verlag.

Borg, I. (1979). Geometric representation of individual differences. In Lingoes et al. (Eds), *Geometric representations of relational data* (pp. 609 - 656). Ann Arbor: Mathesis Press.

Carroll, J.D. & Chang, J.J. (1970). Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition. *Psychometrika*, **35**, 283 - 319.

Carroll, J.D. & Chang, J.J. (1972). *IDIOSCAL (Individual Differences In Orientation SCALing): A generalization of INDSCAL allowing IDIOsyncratic reference systems as well as an analytic approximation to INDSCAL*. Paper presented at meetings of the Psychometric Society, Princeton, N.J.

Cliff, N. (1966). Orthogonal rotation to congruence. *Psychometrika*, **31**, 33 - 42.

Coxon, A.P.M. (1982). *The user's guide to multidimensional scaling*. London: Heinemann Educational Books.

Davies, P.M. & Coxon, A.P.M. (1983). *The MDS(X) series of multidimensional scaling programs*. Report No 55, University of Edinburgh.

Davison, M.L. (1983).*Multidimensional scaling*. New York: Wiley.

De Leeuw, J. & Meulman, J. (1986). A special jackknife for multidimensional scaling. *Journal of Classification*, **3**, 97-112.

Everitt, B.S. & Gower, J.C. (1981). Plotting the optimum positions of an array of cortical electrical phosphenes. In Barnett (Ed), *Interpreting multivariate data*. (pp. 279 - 287). Chichester: Wiley.

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

Gower, J.C. (1975). Generalized Procrustes analysis. *Psychometrika*, **40**, 33 -51.

Green, B.F. (1952). The orthogonal approximation of an oblique structure in factor analysis. *Psychometrika*, **17**, 429 - 440.

Green, P.E., Carmone, F.J., & Smith, S.M. (1989). *Multidimensional scaling*. Boston: Allyn and Bacon.

Horan, C.B. (1969). Multidimensional scaling: Combining observations when individuals have different perceptual structures. *Psychometrika*, **34**, 139 - 165.

Kristof, W. (1970). A theorem on the trace of certain matrix products and some applications. *Journal of Mathematical Psychology*, **7**, 515 - 530.

Kristof, W. & Wingersky, B. (1971). Generalization of the orthogonal Procrustes rotation procedure to more than two matrices. *Proceedings of the 79th Annual Convention of the American Psychological Association*, **6**, 89 - 90.

Lawson, C.L. & Hanson, R.J. (1974). *Solving least squares problems.* Englewood Cliffs, NJ: Prentice Hall.

Langeheine, R. (1982). Statistical evaluation of measures of fit in the Lingoes-Borg Procrustean individual differences scaling. *Psychometrika*, **47**, 427 - 442.

Lingoes, J.C. & Borg, I. (1978). A direct approach to individual differences scaling using increasingly complex transformations. *Psychometrika*, **43**, 491 - 519.

Lingoes, J.C. & Schönemann, P.H. (1974). Alternative measures of fit for the Schönemann-Carroll matrix fitting algorithm. *Psychometrika*, **39**, 423 - 427.

Magnus, J.R. & Neudecker, H. (1988). *Matrix differential calculus with applications in statistics and econometrics.* Chichester: Wiley.

Mooijaart, A. & Commandeur, J.J.F. (in press). A general solution of the weighted orthonormal Procrustes problem. *Psychometrika.*

Peay, E.R. (1988). Multidimensional rotation and scaling of configurations to optimal agreement. *Psychometrika*, **53**, 199 - 208.

Schönemann, P.H. (1966). A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, **31**, 1 - 10.

Schönemann, P.H. & Carroll, R.M. (1970). Fitting one matrix to another under choice of a central dilation and a rigid motion. *Psychometrika*, **35**, 245 - 255.

Tatsuoka, M.M. (1988). *Multivariate analysis: Techniques for educational and psychological research* (2nd ed.). New York: Macmillan.

Ten Berge, J.M.F. (1977). Orthogonal Procrustes rotation for two or more matrices. *Psychometrika*, **42**, 267 - 276.

Ten Berge, J.M.F. (1983). A generalization of Kristof's theorem on the trace of certain matrix products. *Psychometrika*, **48**, 519-523.

Ten Berge, J.M.F. (1988). Generalized approaches to the Maxbet problem and the Maxdiff problem, with applications to canonical correlations. *Psychometrika*, **53**, 487 - 494.

Ten Berge, J.M.F. & Knol, D.L. (1984). Orthogonal rotations to maximal agreement for two or more matrices of different column orders. *Psychometrika*, **49**, 49 - 55.

Van de Geer, J.P. (1984). Linear relations among k sets of variables. *Psychometrika*, **49**, 79 - 94.

Van de Geer, J. P. (1986). *Introduction to linear multivariate analysis* (2 Vols). University of Leiden: DSWO Press.

Young, F.W. & Hamer, R.M. (1987). *Multidimensional scaling: history, theory, and applications.* NJ: Lawrence Erlbaum Associates.

Van der Kloot, W.A. (1978). Matching five representations of the implicit theory of personality. Unpublished manuscript.

Samenvatting

In de afgelopen decennia is er een groot aantal data analyse technieken ontwikkeld die het letterlijk mogelijk maken om 'de werkelijkheid in kaart te brengen'. Deze veelal exploratieve technieken zijn er op gericht om de relaties tussen objecten te onderzoeken door deze als punten af te beelden in een m-dimensionale ruimte. Meestal wordt hierbij de voorkeur gegeven aan een afbeelding in zo min mogelijk dimensies. Zo'n afbeelding van objecten als punten in m dimensies, ook wel een *configuratie* genoemd, is nuttig omdat deze inzicht kan geven in de onderlinge relaties van de objecten. Bovendien biedt de ordening van de objecten op de afzonderlijke dimensies de mogelijkheid om aan die dimensies een interpretatie te geven.

Mede als gevolg van deze ontwikkelingen is de behoefte ontstaan om configuraties *onderling* met elkaar te vergelijken. In allerlei situaties is het van belang om vast te kunnen stellen in hoeverre configuraties identiek zijn, en zo nee, of er een systematisch patroon in de verschillen zit. Een voorbeeld is de situatie waarbij een aantal onderzoekers gegevens verzameld hebben over eenzelfde onderwerp, en waarbij iedere onderzoeker zijn of haar gegevens heeft omgezet in een configuratie. Het wordt dan interessant om een uitspraak te kunnen doen over de mate van overeenkomst (of: *match*) tussen de configuraties.

Het lijkt misschien alsof we in dit soort gevallen gemakkelijk zouden kunnen volstaan met het 'op het oog' vergelijken van de configuraties om vervolgens op grond daarvan te besluiten of ze al dan niet overeenkomen. Maar deze aanpak is weinig systematisch, en heeft bovendien alleen kans van slagen bij één- of twee-dimensionale configuraties, d.w.z. bij afbeeldingen op de rechte lijn of in het platte vlak. Zodra we te maken hebben met afbeeldingen van objecten in drie of meer dimensies wordt zo'n visuele vergelijking al vrijwel onuitvoerbaar.

Van de eerder in de literatuur voorgestelde methoden voor onderzoek naar de match tussen configuraties biedt het zg. PINDIS programma (Procustean INdividual DIfferences Scaling) van Lingoes en Borg (1978) de meest uitgebreide mogelijkheden. Niet alleen kunnen met dit programma de overeenkomsten tussen twee of meer configuraties worden onderzocht aan de hand van maar liefst vijf verschillende modellen, maar deze modellen zijn bovendien hiërarchisch opgebouwd, zodat nagegaan kan worden of ingewikkelder modellen iets toe te voegen hebben aan meer simpele modellen.

Aan het PINDIS programma kleven echter een aantal belangrijke bezwaren. Met name de algoritmen die in dit programma gebruikt worden zijn duidelijk voor verbetering vatbaar. Het belangrijkste oogmerk van dit boek is dan ook om nieuwe convergerende kleinste kwadraten algoritmen aan te dragen voor het schatten van de parameters in de respectievelijke modellen. Bovendien worden de eerder voorgestelde modellen zodanig uitgebreid dat de overeenkomst tussen twee of meer configuraties ook onderzocht kan worden als de configuraties informatie bevatten over *verschillende aantallen objecten.* De in dit boek voorgestelde algoritmen zijn geïmplementeerd in het zg. MATCHALS programma (MATCHing configurations by Alternating Least Squares).

In hoofdstuk 1 worden de vijf modellen behorend bij het PINDIS programma geïntroduceerd, en wordt een gedetailleerd overzicht gegeven van de bezwaren die tegen deze modellen kunnen worden aangevoerd, alsmede tegen de algoritmen voor het schatten van de betreffende modelparameters. Alternatieve modellen worden gepresenteerd, waarbij rekening wordt gehouden met de mogelijkheid van ontbrekende gegevens.

In hoofdstuk 2 wordt de overeenkomst tussen twee of meer configuraties onderzocht onder transformaties die de relatieve afstanden tussen de object punten onveranderd laten. Deze techniek, die ook wel bekend is onder de naam gegeneraliseerde Procrustes analyse (GPA), wordt uitgebreid naar het geval dat de configuraties ongelijke aantallen rijen hebben. Aangetoond wordt dat het bij onvolledige configuraties essentieel is om te corrigeren voor verschillen in grootte tussen de configuraties. Door eliminatie van twee van de vier sets te schatten parameters ontstaat een efficiënt alternerend algoritme voor het schatten van de overgebleven twee transformaties. Het theoretische gedeelte van het hoofdstuk wordt afgesloten met een decompositie van kwadratensommen, en met een bespreking van de overeenkomsten tussen de criteria MAXBET, MAXDIFF, en MAXNEAR en het match criterium in GPA voor incomplete configuraties. Tenslotte worden de resultaten besproken van de analyse van twee getallenvoorbeelden met het in hoofdstuk 2 ontwikkelde algoritme.

In hoofdstuk 3 worden nieuwe convergerende algoritmen ontwikkeld voor het schatten van de parameters in twee zg. *dimensie gewogen* modellen. Deze modellen hebben gemeen dat de (optimaal gekozen) assen van een (vrij te schatten) groepsconfiguratie verschillend mogen worden opgerekt. In het ene model

(DIMFREE genoemd) zijn deze assen identiek voor alle individuele configuraties, terwijl het andere model (DIMIDIO genoemd) de vrijheid biedt om voor iedere individuele configuratie een eigen optimale assenstand in de groepsconfiguratie te bepalen. Voor beide modellen wordt een ontbinding van kwadratensommen gepresenteerd, en de in hoofdstuk 3 ontwikkelde algoritmen worden uitgetest op een aantal geconstrueerde data sets.

In hoofdstuk 4 worden convergerende algoritmen opgezet voor het bepalen van de parameters in twee zg. *stimulus gewogen* modellen. De overeenkomst tussen deze twee modellen is dat de object punten van een (vrij te schatten) groepsconfiguratie verbonden worden met de oorsprong, en dat de hieruit resulterende vectoren verschillend mogen worden verlengd of ingekort. In het ene model (STIMFREE genaamd) is de oorsprong waarmee de object punten van de groepsruimte worden verbonden identiek voor alle individuele configuraties. In het andere model (STIMIDIO genoemd) wordt voor iedere individuele configuratie een eigen optimale oorsprong geschat. Het hoofdstuk wordt weer afgesloten met een decompositie van kwadratensommen voor de twee stimulus gewogen modellen, waarna de resultaten van het STIMFREE en het STIMIDIO algoritme worden geïllustreerd met een aantal geconstrueerde getallenvoorbeelden.

In hoofdstuk 5 tenslotte worden de belangrijkste resultaten uit de voorafgaande hoofdstukken samengevat, en een aantal suggesties gegeven voor verder onderzoek.

MATCHING CONFIGURATIONS

For many multidimensional data analysis methods, such as multidimensional scaling, principal components analysis and correspondence analysis, the major result consists of a configuration of points, that is, a geometrical representation of the relationships between the observational units, or objects. The growing use of these methods has led to an increased interest in the possibility of comparing (or: matching) configurations. Whenever one analyses the same data with different mutidimensional techniques, or when one is interested in the question whether the configurations obtained in a number of studies in some common field of research are identical, the problem arises how to evaluate the similarity of the outcomes.

This book starts out from the PINDIS framework for the matching of two or more configurations. It discusses five models in detail, providing convergent alternating least squares algorithms for the estimation of the corresponding model parameters. The first and most parsimonious model investigates the match under transformations that leave the relative distances between the object points intact: Generalized Procrustes analysis. In the second and third model a transformation called dimension weighting is introduced, where dimensions are allowed to be stretched or shrunk differently. In terms of interpretation these cases are related to the well-known INDSCAL and IDIOSCAL models. The fourth and fifth model are characterized by a so-called stimulus weighting transformation, where the centrality of the object points is differentially increased or decreased. These five models form two hierarchies, within each of which it is possible to assess whether more complex transformations have anything new to add over the simpler ones.
When the configurations have been obtained from different studies, it often occurs that the number of objects varies from configuration to configuration. In this case some or all configurations contain missing points. An important new feature of the five algorithms presented in this book is that they can also be used to investigate the match between incomplete configurations.

DSWO PRESS

ISBN 90-6695-052-8