SPRINGER BRIEFS IN STATISTICS

Ingwer Borg
Patrick J. F. Groenen
Patrick Mair

# Applied Multidimensional Scaling

Springer

# SpringerBriefs in Statistics

For further volumes:
http://www.springer.com/series/8921

Ingwer Borg · Patrick J. F. Groenen
Patrick Mair

# Applied Multidimensional Scaling

Springer

Ingwer Borg
Zeppelin University
Friedrichshafen
Germany

Patrick J. F. Groenen
Econometric Institute
Erasmus University Rotterdam
Rotterdam
The Netherlands

Patrick Mair
Institute for Statistics and Mathematics
Vienna University of Economics
and Business
Vienna
Austria

ISSN 2191-544X ISSN 2191-5458 (electronic)
ISBN 978-3-642-31847-4 ISBN 978-3-642-31848-1 (eBook)
DOI 10.1007/978-3-642-31848-1
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012942918



Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

This book is a brief introduction to applied Multidimensional Scaling (MDS). It builds on our combined (over 75 years) experience in developing MDS methods and programs, and in using them in substantive research. The book is related to the much more comprehensive book "Modern Multidimensional Scaling" by Borg and Groenen (published by Springer in 2005 in its 2nd edition), and you can use the exercises presented there to test and deepen your understanding of MDS.[1]

The book is, however, not just an abbreviated discussion of MDS. Rather, it chooses a particular perspective, stressing the issues that always come up when MDS is used in substantive research, and presenting answers that are particularly relevant for the substantive researcher:

- What is the purpose of MDS?
- How to generate or select data for MDS?
- How to pick a particular MDS model?
- How to find a best-possible MDS representation in a space of given dimensionality?
- How to assess whether this solution is good enough?
- How to align MDS with substantive theory?
- How to avoid the many (minor or major) mistakes that MDS users tend to make?
- How to use PROXSCAL (in SPSS) or SMACOF (in R) for running MDS?

Why should one be interested in MDS at all? One reason is that MDS is by now an established method for doing data analysis. It belongs to the toolbox of psychologists and social scientists, in particular, but it is also relevant for many other substantive areas that deal with proximity data. There are numerous research papers which use MDS. To understand them fully, one must know the method.

[1] See also Patrick Groenen's website `people.few.eur.nl/groenen`. It offers additional information about this book, and it provides access to 24 data sets that are used to illustrate different MDS models.

Also, there are many publications that do not use MDS, but where MDS may have been the better method to understand the data (or additional features of the data) than main stream methods such as factor analysis. MDS is generally easy to do, and all major statistics packages provide MDS modules, and so re-analyzing published research via MDS to gain additional insights is often an interesting option.

Then, psychologists in particular often study MDS because it originated as a psychological model on how persons form judgments of similarity or preferential choice. The model explains such judgments by a composition rule where the differences of objects (real or virtual) on a number attributes are aggregated to an overall judgment of the objects' dissimilarity or to a value judgment, respectively. This process can be formalized as a distance function in multidimensional space. Much research has been devoted to study this model and its numerous variants, and because of the fundamental nature of similarity and choice in psychology, much of psychology's history is connected to MDS models.

While MDS as a psychological model of judgment is an area that does not seem to offer many new research questions, MDS as a data analysis method is far from complete. Exploratory MDS as a method to visualize proximity data in low-dimensional spaces is well developed, but confirmatory MDS, where theoretical expectations or hypotheses are enforced onto the data representations as side constraints, is a field where more developments are needed. What can be done today by the user, is shown and illustrated in this book. What cannot be done, or what is difficult to do with today's software, also becomes clear, and so this may serve also to stimulate further research on MDS.

# Contents


**1 First Steps** . . . . . 1
Summary . . . . . 6

**2 The Purpose of MDS** . . . . . 7
2.1 MDS for Visualizing Proximity Data . . . . . 7
2.2 MDS for Uncovering Latent Dimensions of Judgment . . . . . 10
2.3 Distance Formulas as Models of Judgment . . . . . 12
2.4 MDS for Testing Structural Hypotheses . . . . . 16
2.5 Summary . . . . . 18
References . . . . . 19

**3 The Goodness of an MDS Solution** . . . . . 21
3.1 Fit Indices and Loss Functions . . . . . 21
3.2 Evaluating Stress . . . . . 23
3.3 Variants of the Stress Measure . . . . . 25
3.4 Summary . . . . . 26
References . . . . . 26

**4 Proximities** . . . . . 27
4.1 Direct Proximities . . . . . 27
4.2 Derived Proximities . . . . . 29
4.3 Proximities from Index Conversions. . . . . 30
4.4 Co-Occurrence Data . . . . . 31
4.5 The Gravity Model for Co-Occurrences . . . . . 32
4.6 Summary . . . . . 34
References . . . . . 34

**5 Variants of Different MDS Models** . . . . . 37
5.1 Ordinal and Metric MDS . . . . . 37
5.2 Euclidean and Other Distances . . . . . 39

5.3 Scaling Replicated Proximities . . . . . 39
5.4 MDS of Asymmetric Proximities . . . . . 40
5.5 Modeling Individual Differences in MDS . . . . . 42
5.6 Unfolding . . . . . 45
5.7 Summary . . . . . 47
References . . . . . 47

**6 Confirmatory MDS** . . . . . 49
6.1 Weak Confirmatory MDS . . . . . 50
6.2 External Side Constraints on the Point Coordinates . . . . . 51
6.3 Regional Axial Restrictions . . . . . 54
6.4 Challenges of Confirmatory MDS . . . . . 56
6.5 Summary . . . . . 57
References . . . . . 58

**7 Typical Mistakes in MDS** . . . . . 59
7.1 Using the Term MDS Too Generally . . . . . 59
7.2 Using the Distance Notion Too Loosely . . . . . 60
7.3 Assigning the Wrong Polarity to Proximities . . . . . 60
7.4 Using Too Few Iterations . . . . . 61
7.5 Using the Wrong Starting Configuration . . . . . 61
7.6 Doing Nothing to Avoid Suboptimal Local Minima . . . . . 61
7.7 Not Recognizing Degeneracy in Ordinal MDS . . . . . 63
7.8 Meaningless Comparisons of Different MDS Solutions . . . . . 65
7.9 Evaluating Stress Mechanically . . . . . 68
7.10 Always Interpreting "the Dimensions" . . . . . 70
7.11 Poorly Dealing with Disturbing Points . . . . . 74
7.12 Scaling Almost Equal Proximities . . . . . 75
7.13 Over-Interpreting Dimension Weights . . . . . 78
7.14 Unevenly Stretching MDS Plots . . . . . 79
7.15 Summary . . . . . 79
References . . . . . 80

**8 MDS Algorithms** . . . . . 81
8.1 Classical MDS . . . . . 81
8.2 Iterative MDS Algorithms . . . . . 84
8.3 Summary . . . . . 86
References . . . . . 86

**9 Computer Programs for MDS** . . . . . 87
9.1 PROXSCAL . . . . . 87
9.1.1 PROXSCAL by Menus . . . . . 88
9.1.2 PROXSCAL by Commands . . . . . 91

9.2 The R Package SMACOF . . . . . 96
9.2.1 General Remarks on SMACOF . . . . . 97
9.2.2 SmacofSym . . . . . 98
9.2.3 SmacofIndDiff . . . . . 101
9.2.4 SmacofRect . . . . . 103
9.2.5 SmacofConstraint . . . . . 104
9.2.6 Final Remarks . . . . . 105
9.3 Other Programs for MDS . . . . . 105
References . . . . . 106

**Author Index** . . . . . 109

**Subject Index** . . . . . 111

# Chapter 1
# First Steps

**Abstract** The basic ideas of MDS are introduced doing MDS by hand. Then, MDS is done using a computer program. The goodness of the MDS configuration is evaluated by correlating its distances with the data.

**Keywords** MDS configuration · Iteration · Proximities · Dimensional interpretation · Goodness of fit

The basic ideas of MDS are easily explained using a small example. Consider Table 1.1. It contains correlations for the frequencies of different crimes in 50 U.S. states. These correlations show, for example, that if there are many cases of assault in a state, then there are also many cases of murder ($r = 0.81$). In contrast, the murder rate is not correlated with the rate of larceny ($r = 0.06$).

We now scale these correlations via MDS. This means that we try to represent the seven crimes by seven points in a geometric space so that any two points lie the *closer* together the *greater* the correlation of the two crimes that they represent. For this we proceed as follows.

We take seven cards, and write the name of one crime on each of them, from Murder to Auto Theft. These cards are placed on a table in an arbitrary arrangement as shown in Fig. 1.1. We then measure the distances among all cards (Fig. 1.2) and compare these values with the correlations in Table 1.1. This comparison makes clear that the configuration of cards in Fig. 1.1 does not represent the data in the desired sense. For example, the cards Murder and Assault should be relatively close together, because these crimes are correlated with 0.81, whereas the cards Murder and Larceny should be farther apart, as these crimes are correlated with only 0.06. We therefore try to move the cards repeatedly in small steps ("*iteratively*") so that the distances correspond more closely to the data. Figure 1.3 demonstrates in which directions the cards should be shifted, by some small amounts, to improve the correspondence of data and distances.

Since iterative modifications of a given configuration by hand can be fairly tedious and since they do not guarantee that an *optimal* configuration is found in the end,

I. Borg et al., *Applied Multidimensional Scaling*, SpringerBriefs in Statistics,
DOI: 10.1007/978-3-642-31848-1_1, 

**Table 1.1** Correlations of crime rates over 50 U.S. states

| Crime | Murder | Rape | Robbery | Assault | Burglary | Larceny | Auto theft |
|---|---|---|---|---|---|---|---|
| Murder | 1.00 | 0.52 | 0.34 | 0.81 | 0.28 | 0.06 | 0.11 |
| Rape | 0.52 | 1.00 | 0.55 | 0.70 | 0.68 | 0.60 | 0.44 |
| Robbery | 0.34 | 0.55 | 1.00 | 0.56 | 0.62 | 0.44 | 0.62 |
| Assault | 0.81 | 0.70 | 0.56 | 1.00 | 0.52 | 0.32 | 0.33 |
| Burglary | 0.28 | 0.68 | 0.62 | 0.52 | 1.00 | 0.80 | 0.70 |
| Larceny | 0.06 | 0.60 | 0.44 | 0.32 | 0.80 | 1.00 | 0.55 |
| Auto theft | 0.11 | 0.44 | 0.62 | 0.33 | 0.70 | 0.55 | 1.00 |

we did not continue these iterations by hand but used an MDS computer program instead. It reports the solution shown in Fig. 1.4.

One such MDS program is PROXSCAL, a module of SPSS. To use PROXSCAL, we first save the correlation matrix of Table 1.1 in a file that we call 'CorrCrimes.sav'. Then, we only need some clicks in PROXSCAL's menus or, alternatively, the following commands:

```
GET FILE='CorrCrimes.sav'.
PROXSCAL VARIABLES=Murder to AutoTheft
     /PROXIMITIES=SIMILARITIES .
```

The `PROXIMITIES` sub-command informs the program that the data—called *proximities* in this context, a generic term for both similarity and dissimilarity data—must be interpreted as similarities. That is, small data values should be mapped into large distances, and large data values into small distances. No further specifications are needed. The program uses its default specifications to generate an MDS solution. We will show later how these specifications can be changed by the user if desired.

Many other programs exist for MDS. One example with nice graphics is the MDS module in SYSTAT. SYSTAT can be run using commands, or by clicking on various options in a graphical user interface. Having loaded the data file with the correlations, and then calling the MDS procedure, we get the menu in Fig. 1.5. In this menu, we select the variables 'Murder', 'Rape', etc. and leave all other specifications as they are, except the one for "Regression" (marked by the arrow on the left-hand side), where we request that the MDS program should optimize the relation of data to distances in the sense of a least-squares *linear* regression. (The default is *ordinal* regression which is discussed later; see p. 37f)

Both computer programs—PROXSCAL in SPSS and the MDS module of SYSTAT—generate essentially the same MDS solution for the correlations in Table 1.1. This solution is not only optimal, but also quite good, as Fig. 1.6 shows: The relation of data and distances is almost perfectly linear ($r = -0.99$). Hence, the distances among the points of Fig. 1.3 contain the same information as the correlations of Table 1.1. Expressed differently, the data are properly *visualized* so that one can interpret the distances as empirical evidence: The closer two points in the MDS plane, the higher the correlation of the variables they represent.

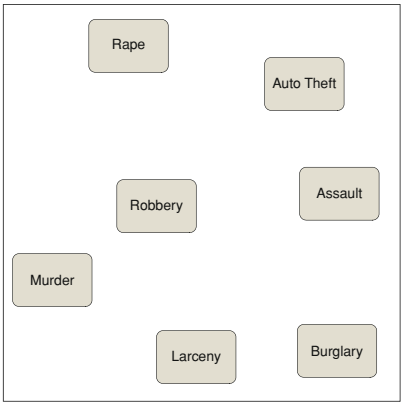


**Fig. 1.1** Starting configuration for an MDS of the data in Table 1.1

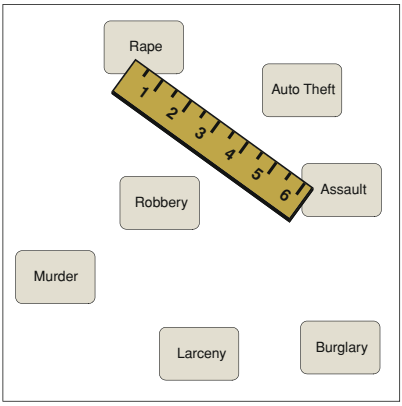


**Fig. 1.2** Measuring distances with a ruler

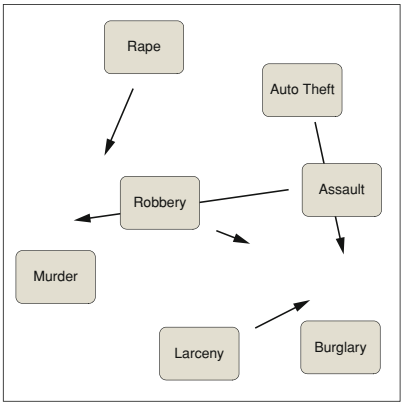


**Fig. 1.3** Directions for point movements to improve the MDS configuration

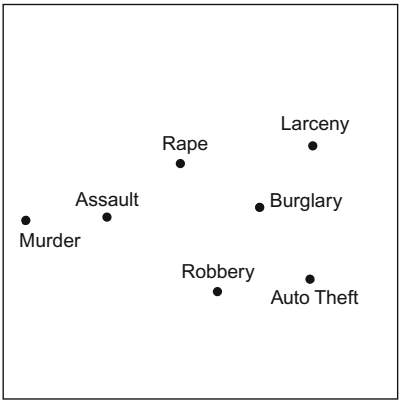


**Fig. 1.4** MDS representation of the correlations in Table 1.1 after several iterations

What has been gained by analyzing the crime data via MDS? First, instead of 21 different *numerical* indexes (i.e., correlations), we get a simple *visual* representation of the empirical interrelations. This allows us to actually *see* and, therefore, more easily explore the structure of these data. As shown in Fig. 1.7, the various crimes form certain *neighborhoods* in the MDS plane: Crimes where persons come to harm are one such neighborhood, and property crimes form another neighborhood. This visualizes, for example, that if the murder rate is high in a state, then assault and rape also tend to be relatively frequent. The same applies to property crimes. Robbery lies between these neighborhoods, possibly because violent crimes not only damage the victims' properties but also their bodies.

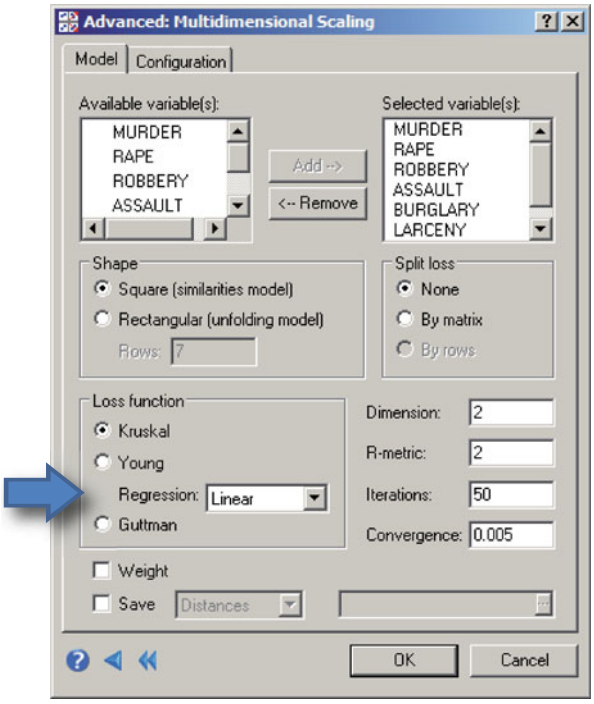


**Fig. 1.5** GUI for the MDS module of SYSTAT

This interpretation builds primarily on the first *principal axis*.[1] This axis corresponds to the horizontal direction of the graph. (Most computer programs for MDS automatically rotate their graphs so that the coordinate axes of MDS plots correspond to principal axes.) The second principal axis is difficult to interpret in this example. On this axis, Larceny and Robbery are farthest apart. Hence, these two crimes might lead us to a meaningful interpretation of the second dimension. Yet, no compelling interpretation seems to offer itself for this dimension: It may simply represent a portion of the "error" component of the data. So, one can ask whether it may suffice to represent the given data in a 1-dimensional MDS space. This is easy to answer: One simply sets "Dimension=1" in the GUI in Fig. 1.5 and then repeats the MDS analysis, leaving all other specifications as before, to get the desired solution.

Figure 1.8 shows the 1-dimensional solution. It closely reproduces the first principal axis of Fig. 1.4. However, its distances correlate with only $r = 0.88$ with the data, i.e. this MDS solution does not represent the data that well. This is also evident from the regression graph in Fig. 1.9, which has a much larger scatter than

[1] The first principal axis is a straight line which runs through the point cloud so that it is closest to the points. That is, the sum the (squared) distances of the points from this line is minimal. Or, expressed differently: The variance of the projections of the points onto this line is maximal. The second major axis is perpendicular to the first and explains the maximum of the remaining variance.

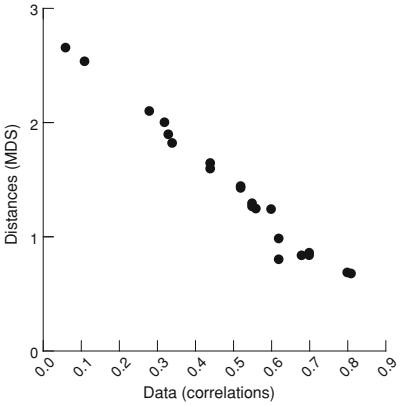


**Fig. 1.6** Relation of data in Table 1.1 and distances in Fig. 1.4

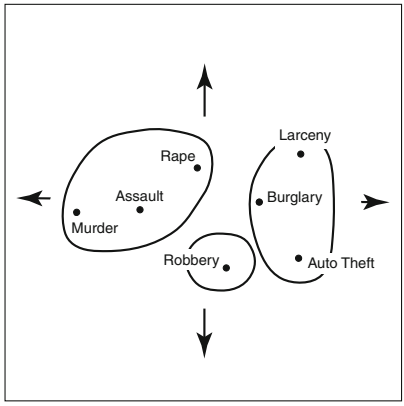


**Fig. 1.7** MDS solution with two interpretations: neighborhoods and dimensions

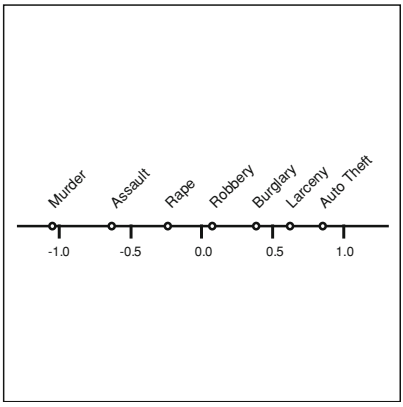


**Fig. 1.8** An 1-dimensional MDS solution for the crime data

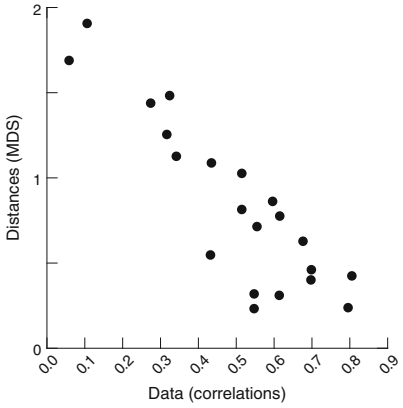


**Fig. 1.9** Relation of data in Table 1.1 and distances in Fig. 1.9

the graph for the 2-dimensional MDS solution in Fig. 1.6. One should therefore be cautious when interpreting this configuration, because it is partly misleading. For example, Larceny and Auto Theft correlate much lower ($r = 0.55$) than Larceny and Burglary ($r = 0.80$), but the configuration in Fig. 1.8 does not represent this difference correctly. Rather, the respective two distances are about equal in size.

## Summary

Multidimensional scaling (MDS) represents proximity data (i.e., measures of similarity, closeness, relatedness etc.) as distances among points in a multidimensional (typically: 2-dimensional) space. The scaling begins with some starting configuration. Its points are then moved iteratively so that the fit between distances and data is improved until no further improvement seems possible. Computer programs (such as SYSTAT or PROXSCAL) exist for that purpose. The more precisely the data correspond to the distances in the MDS space, the better the MDS point configuration represents the structure of the proximities. If the fit of the MDS solution is good, it can be inspected visually in an attempt to interpret it in terms of content. A popular approach for doing this is to look for dimensions, mostly principal axes, that make sense in terms of what is known or assumed about the objects represented by the points.

# Chapter 2
# The Purpose of MDS

**Abstract** The different purposes of MDS are explained: MDS as a psychological model of similarity judgments; MDS for visualizing proximity data; and MDS for testing structural hypotheses.

**Keywords** Latent dimension · Distance axiom · Minkowski distance · Euclidean distance · City-block distance · Dominance metric · Partition · Facet · Radex · Cylindrex

Modern MDS is mainly used for general data analysis, especially for visualizing data. This was not always so. Historically, MDS served a different purpose: It was a psychological model of how persons form judgments about the similarity of objects. In many modern MDS applications, traces of this original model can still be found (e.g., in the way MDS solutions are interpreted or in the terminology used in MDS), even if the scaling method is used as a mere statistical tool. In the following, we begin by discussing a recent application that uses MDS as a visualization tool. Then, we consider typical examples of the early days of MDS.

## 2.1 MDS for Visualizing Proximity Data

Over the recent years, MDS has been predominantly used as a tool for analyzing proximity data of all kinds (e.g., correlations, similarity ratings, co-occurrence data). Most of all, MDS serves to visualize such data, making them accessible to the eye of the researcher. Let us consider a typical visualization application of MDS. Figure 2.1 shows a case from industrial psychology. Its 27 points represent 25 items and two indexes from an employee survey in an international IT company (Liu et al. 2004). Two examples for the items are: "All in all, I am satisfied with my pay", and "I like my work", both employing a Likert-type response scale ranging from "fully agree" to "fully disagree." The two indexes are scale values that summarize the employees' responses to a number of items that focus on their affective commitment to the

I. Borg et al., *Applied Multidimensional Scaling*, SpringerBriefs in Statistics,
DOI: 10.1007/978-3-642-31848-1_2, © The Author(s) 2013

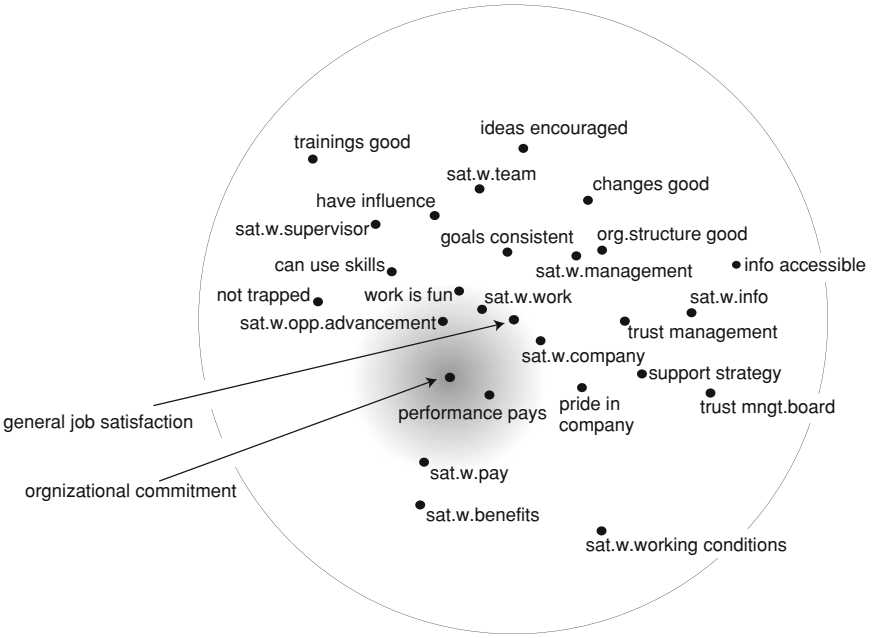


**Fig. 2.1** MDS representation of the intercorrelations of 25 items and 2 indexes of an employee survey in an international IT company. The grayed area around organizational commitment contains likely drivers of commitment

company and on their general job satisfaction, respectively. The distance between two points in Fig. 2.1 represents (quite precisely) the correlation of the respective variables. As all variables are non-negatively intercorrelated, it is particularly easy to interpret this MDS configuration: The closer two points, the higher the correlation of the variables they represent. Hence, one notes, for example, that since "satisfied with pay" and "satisfied with benefits" are close neighbors in the MDS plane (see lower left-hand corner of the plot), employees rated these issues similarly: Those who were satisfied with one job aspect where also satisfied with the other, and vice versa. In contrast, being satisfied with pay is far from "encouraged to voice new ideas" (see top of the plot), and, hence, these two items are essentially uncorrelated.

The value of this MDS configuration is based on the notion that a picture is worth more than a 1,000 words or numbers. Indeed, most researchers and practitioners find it much easier to study such a plot than studying a $27 \times 27$ correlation matrix with its 351 coefficients. It is almost impossible to understand the structure of the data in such large arrays of numbers, while their graphical display in an MDS plane can be explored with considerably less effort.

The fact that 351 correlations can be represented by the distances of 27 points that lie in a merely 2-dimensional space makes clear, moreover, that the data are highly structured. Random data would require much higher-dimensional spaces. Hence, the

persons who answered this employee survey must have generated their answers from a consistent system of attitudes and opinions, and not by generating evasive random ratings, because such ratings would not be so orderly interlocked.

The ratings also make sense psychologically, because items of similar content are grouped in close neighborhoods in the MDS space. For example, the various items related to management (e.g., trust management, trust management board, support strategy) form such a neighborhood of items that received similar ratings in the survey.

One also notes that the one point that represents general job satisfaction lies somewhere in the central region of the point configuration. This central position reflects the fact that general job satisfaction is positively correlated with each of the 25 items of this survey. Items located more at the border of the MDS plot are substantially and positively correlated with the items in their neighborhood, but not with items opposite of them in the configuration. With them, they are essentially uncorrelated.

The plot leads to many more insights. One notes, for example, that the employees tend to be the more satisfied with their job overall the more they like their work and the more they are satisfied with their opportunities for advancement. Satisfaction with working conditions, in contrast, is a relatively poor predictor of general job satisfaction in this company.

Because the company suffered from high turnover of its employees, the variable 'organizational commitment' was of particular interest in this survey. Management wanted to know what could be done to reduce turnover. The MDS configuration can be explored for answers to this question. One begins by studying the neighborhood of the point representing 'organizational commitment' (see dark cloud around the commitment point in Fig. 2.1), looking for items that offer themselves for action. That is, one attempts to find points close to commitment that have low scores and where actions that would improve these scores appear possible. Expressed in terms of the MDS configuration, this can be understood as grabbing such a point and then pulling it upwards so that the whole plane is lifted like a rubber sheet, first of all in the neighborhood of commitment. Managers understand this notion and, if guided properly, they are able to identify and discuss likely "drivers" of the variable of interest efficiently and effectively. In the given configuration, one notes, for example, that the employees' commitment is strongly correlated with how they feel about their opportunities for advancement (42 % are satisfied with them, see Borg 2008, p. 311f.); with how much they like the work they do (69 % like it); with how satisfied they are with the company overall (88 % satisfied); and, most of all, with how positive they feel about "performance pays" (only 36 % positive). Thus, if one interprets this network of correlations causally, with the variables in the neighborhood of commitment as potential drivers of commitment, it appears that the employees' commitment can be enhanced most by improving the employees' opinions about the performance-dependency of their pay and about their advancement opportunities. Improving other variables such as, for example, the employees' attitudes towards management, is not likely to impact organizational commitment that much.

In this example, MDS serves to visualize the intercorrelations of the items. This makes it possible for the user to see, explore, and discuss the whole structure of the data. This can be useful even if the number of items is relatively large, because each additional item adds just one new point to an MDS plot, while it adds as many new coefficients to a correlation matrix as there are variables.

## 2.2 MDS for Uncovering Latent Dimensions of Judgment

One of the most fundamental issues of psychology is how subjective impressions of similarity come about. Why does Julia look like Mike's daughter? How come that a Porsche appears to be more similar to a Ferrari than to a Cadillac? To explain such judgments or perceptions, distance models offer themselves as natural candidates. In such models, the various objects are first conceived as points in a *psychological space* that is *spanned* by the subjective *attributes* of the objects. The distances among the points then serve to generate overall impressions of greater or smaller similarity. Yet, the problem with such models is that one hardly ever knows what attributes a person assigns to the objects under consideration. This is where MDS comes in: With its help, one attempts to infer these attributes from given global similarity judgments.

Let us consider an example that is typical for early MDS applications. Wish (1971) wanted to know the attributes that people use when judging the similarity of different countries. He conducted an experiment where 18 students were asked to rate each pair of 12 different countries on their overall similarity. For these ratings, an answer scale from "extremely dissimilar" (coded as '1') to "extremely similar" (coded as '9') was offered to the respondents. No explanation was given on what was meant by "similar": "There were no instructions concerning the characteristics on which these similarity judgments were to be made; this was information to discover rather than to impose" (Kruskal and Wish 1978, p. 30). The observed similarity ratings, averaged over the 18 respondents, is exhibited in Table 2.1.

An MDS analysis of these data with one of the major MDS programs, using the usual default parameters,[1] delivers the solution shown in Fig. 2.2. Older MDS programs generate only the Cartesian coordinates of the points (as shown in Table 2.2 in columns "Dim. 1" and "Dim. 2", respectively, together called *coordinate matrix*, denoted as **X** in this book). Modern programs also yield graphical output as in Fig. 2.2. The plot shows, for example, that the countries Jugoslavia and USSR are represented by points that are close together. In Table 2.1 we find that the similarity rating for these two countries is relatively high (=6.67, the largest value). So, this relation is properly represented in the MDS plane. We note further that the points representing Brazil and China are far from each other, and that their similarity rating is small (=2.39). Thus, this relation is also properly represented in the MDS solution.

[1] Most early MDS programs were set, by default, to deliver a 2-dimensional solution for data that were assumed to have an ordinal scale level.

**Table 2.1** Mean similarity ratings for 12 countries (Wish 1971)

| Country | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Brazil | 1 | – | | | | | | | | | | | |
| Congo | 2 | 4.83 | – | | | | | | | | | | |
| Cuba | 3 | 5.28 | 4.56 | – | | | | | | | | | |
| Egypt | 4 | 3.44 | 5.00 | 5.17 | – | | | | | | | | |
| France | 5 | 4.72 | 4.00 | 4.11 | 4.78 | – | | | | | | | |
| India | 6 | 4.50 | 4.83 | 4.00 | 5.83 | 3.44 | – | | | | | | |
| Israel | 7 | 3.83 | 3.33 | 3.61 | 4.67 | 4.00 | 4.11 | – | | | | | |
| Japan | 8 | 3.50 | 3.39 | 2.94 | 3.83 | 4.22 | 4.50 | 4.83 | – | | | | |
| China | 9 | 2.39 | 4.00 | 5.50 | 4.39 | 3.67 | 4.11 | 3.00 | 4.17 | – | | | |
| USSR | 10 | 3.06 | 3.39 | 5.44 | 4.39 | 5.06 | 4.50 | 4.17 | 4.61 | 5.72 | – | | |
| USA | 11 | 5.39 | 2.39 | 3.17 | 3.33 | 5.94 | 4.28 | 5.94 | 6.06 | 2.56 | 5.00 | – | |
| Jugoslavia | 12 | 3.17 | 3.50 | 5.11 | 4.28 | 4.72 | 4.00 | 4.44 | 4.28 | 5.06 | 6.67 | 3.56 | – |

Checking more of these correspondences suggests that the MDS solution is a proper representation of the similarity data.

If we want to assume that the similarity ratings were indeed generated by a distance model, and if we are willing to accept that the given MDS plane exhibits the essential structure of the similarity data, we can proceed to interpret this *psychological map*. That is, we now ask what psychologically meaningful "dimensions" span this space. Formally, the map is spanned by what the computer program delivers in terms of "Dimension 1" and "Dimension 2". These dimensions are the *principal axes* of the point configuration. However, one can also *rotate* these dimensions in any way one wants (holding the configuration of points fixed), because any other system of two coordinate axes also spans the plane. Hence, one has to look for a coordinate system that is most plausible in psychological terms. Wish (1971) suggests that rotating the coordinate system in Fig. 2.2 by 45° leads to dimensions that correspond most to psychologically meaningful scales. On the diagonal from the North–West to the South–East corner of Fig. 2.2, countries like Congo, Brazil, and India are on one end, while countries like Japan, USA, and USSR are on the other end. On the basis of what he knows about these countries, and assuming that the respondents use a similar knowledge base, Wish interprets this opposition as "underdeveloped versus developed". The second dimension, the North–East to South–West diagonal, is interpreted as "pro-Western versus pro-Communist".

These interpretations are meant as hypotheses about the attributes that the respondents (not the researcher!) use when they generate their similarity judgments. That is, the respondents are assumed to look at each pair of countries, compute their differences in terms of Underdeveloped/Developed and Pro-Western/Pro-Communist, respectively, and then derive an overall distance from these two *intra-dimensional* distances. Whether this explanation is indeed valid cannot be checked any further with the given data. MDS only suggests that this is a model that is compatible with the observations.

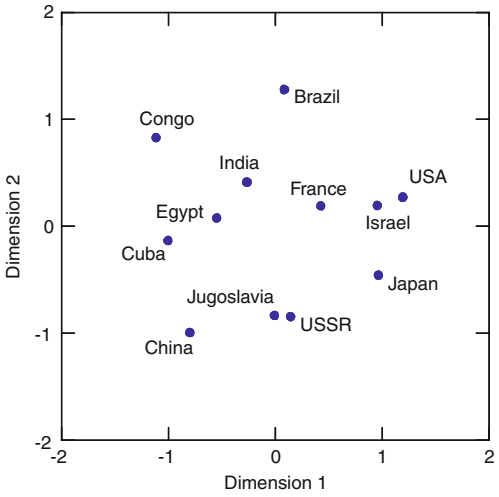


**Fig. 2.2** MDS representation of similarity ratings in Table 2.1

## 2.3 Distance Formulas as Models of Judgment

The above study on the subjective similarity of countries does not explain in detail how an overall similarity judgment is generated based on the information in the psychological space. A natural model that explicates how this can be done is a distance formula based on the coordinates of the points. We will discuss this in the context of an example.

Distances (also called metrics) are *functions* that assign a real value to two arguments of elements from one set. They map all pairs of objects $(i, j)$ of a set of objects (here often "points") onto real values. Distance functions—in the following denoted as $d_{ij}$—have the following properties:

1. $d_{ii} = d_{jj} = 0 \leq d_{ij}$ (Distances have *nonnegative values*; only the self-distance is equal to zero.)
2. $d_{ij} = d_{ji}$ (*Symmetry*: The distance from $i$ to $j$ is the same as the distance from $j$ to $i$.)
3. $d_{ik} \leq d_{ij} + d_{jk}$ (*Triangle inequality*: The distance from $i$ to $k$ via $j$ is at least as large as the direct "path" from $i$ to $k$.)

One can check if given values for pairs of objects (such as the data in Table 2.1) satisfy these properties. If they do, they are distances; if they do not, they are not distances (even though they may be "approximate" distances).

A set $M$ of objects together with a distance function $d$ is called *metric space*. A special case of a metric space is the Euclidean space. Its distance function does not only satisfy the above distance axioms, but it can also be interpreted as the distance of

**Table 2.2** Coordinates **X** of points Fig. 2.2; Economic development and number of inhabitants show further measurements on these countries in 1971

| Country | No. | **X** Dim. 1 | **X** Dim. 2 | Economic development | Number of inhabitants (Mio) |
|---|---|---|---|---|---|
| Brazil | 1 | 0.08 | 1.28 | 3 | 87 |
| Congo | 2 | −1.12 | 0.83 | 1 | 17 |
| Cuba | 3 | −1.01 | −0.13 | 3 | 8 |
| Egypt | 4 | −0.56 | 0.08 | 3 | 30 |
| France | 5 | 0.42 | 0.19 | 8 | 51 |
| India | 6 | −0.27 | 0.41 | 3 | 500 |
| Israel | 7 | 0.95 | −0.20 | 7 | 3 |
| Japan | 8 | 0.96 | −0.46 | 9 | 100 |
| China | 9 | −0.80 | −0.99 | 4 | 750 |
| USSR | 10 | 0.14 | −0.84 | 7 | 235 |
| USA | 11 | 1.19 | 0.27 | 10 | 201 |
| Jugoslavia | 12 | −0.01 | −0.83 | 6 | 20 |

the points $i$ and $j$ of a multi-dimensional Cartesian space. That means that Euclidean distances can be computed from the points' Cartesian coordinates as

$$d_{ij}(\mathbf{X}) = \sqrt{(x_{i1} - x_{j1})^2 + \cdots + (x_{im} - x_{jm})^2}, \tag{2.1}$$

$$= \left( \sum_{a=1}^{m} (x_{ia} - x_{ja})^2 \right)^{1/2}, \tag{2.2}$$

where $\mathbf{X}$ denotes a configuration of $n$ points in $m$-dimensional space, and $x_{ia}$ is the value ("coordinate") of point $i$ on the coordinate axis $a$. This formula can be easily generalized to a family of distance functions, the *Minkowski distances*:

$$d_{ij}(\mathbf{X}) = \left( \sum_{a=1}^{m} |x_{ia} - x_{ja}|^p \right)^{1/p}, \quad p \geq 1. \tag{2.3}$$

Setting $p = 2$, formula 2.3 becomes the Euclidean distance. For $p = 1$, one gets the *city-block distance*. When $p \to \infty$, the formula yields the *dominance metric*.

As a model for judgments of (dis-)similarity, the city-block distance ($p = 1$) seems to be the most plausible "composition rule", at least in case of "analyzable" stimuli with "obvious and compelling" dimensions (Torgerson 1958, p. 254). It claims that a person's judgment is formed by first assessing the distance of the respective two objects on each of the $m$ dimensions of the psychological space, and then adding these intra-dimensional distances to arrive at an overall judgment of dissimilarity.

If one interprets formula (2.3) literally, then it suggests for $p = 2$ that the person first squares each intra-dimensional distance, then sums the resulting values, and

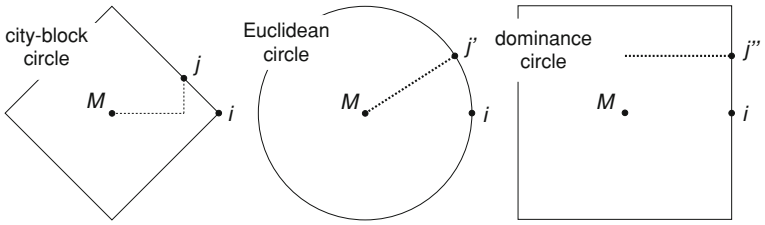


**Fig. 2.3** Three circles with the same radius in the city-block plane, the Euclidean plane, and the dominance plane, respectively

finally takes the square root. This appears hardly plausible. However, one can also interpret the formula somewhat differently. That is, the parameter $p$ of the distance formula can be seen as a *weight* function: For values of $p > 1$, relatively large intra-dimensional distances have an over-proportional influence on the global judgment, and when $p \to \infty$, only the largest intra-dimensional distance matters. Indeed, for $p$-values as small as 10, the global distance is almost equal to the largest intra-dimensional distance.[2] Thus, one could hypothesize that when it becomes more difficult to make a judgment (e.g., because of time pressure), persons tend to pay attention to the largest intra-dimensional distances only, ignoring dimensions where the objects do not differ much. This corresponds, formally, to choosing a large $p$ value. In the limit, only the largest intra-dimensional distance matters.

Another line of argumentation is that city-block composition rules make sense only for analyzable stimuli with their obvious and compelling dimensions (such as geometric figures like rectangles, for example), whereas for "integral" stimuli (such as color patches, for example), the Euclidean distance that expresses the length of the direct path through the psychological space is more adequate (Garner 1974).

Choosing parameters other than $p = 2$ has surprising consequences, though: It generates geometries that differ substantially from those we are familiar with. What we know, and what is called the *natural* geometry, is Euclidean geometry. It is natural because distances and structures in Euclidean geometry are as they "should" be. A circle, for example, is "round". If $p \neq 1$, circles do not seem to be round. In the city-block plane (with simple orthogonal coordinate axes[3]), for example, a circle *looks* like a square that sits on one of its corners (see left panel of Fig. 2.3). Yet, this geometrical figure *is* indeed a circle, because it is the set of all points that have the same distance from their midpoint $M$. The reason for its peculiar-looking shape is that the distances of any two points in the city-block plane correspond to the length of a path between these points that can run only in North–South or West–East directions,

[2] This is easy to see from an example: If point $i$ has the coordinates $(0, 0)$ and $j$ the coordinates $(3, 2)$, we get the intra-dimensional distances $|0-3| = 3$ and $|0-2| = 2$, respectively. The overall distance $d_{ij}$, with $p = 1$, is thus equal to $2 + 3 = 5.00$. For $p = 2$, the overall distance is 3.61. For $p = 10$, it is equal to 3.01.

[3] For the consequences of choosing other coordinate systems and for the many peculiar laws of such "taxicab geometries", see http://taxicabgeometry.net.

**Table 2.3** Dissimilarity ratings for rectangles of Fig. 2.4; ratings are means over 16 subjects and 2 replications

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | | | | | | |
| 2 | 4.33 | | | | | | | | | | | | | | |
| 3 | 6.12 | 4.07 | | | | | | | | | | | | | |
| 4 | 7.21 | 5.62 | 3.24 | | | | | | | | | | | | |
| 5 | 2.38 | 5.76 | 7.12 | 7.57 | | | | | | | | | | | |
| 6 | 4.52 | 2.52 | 5.48 | 6.86 | 4.10 | | | | | | | | | | |
| 7 | 6.00 | 4.52 | 3.38 | 5.21 | 6.10 | 4.31 | | | | | | | | | |
| 8 | 7.76 | 6.21 | 4.40 | 3.12 | 6.83 | 5.45 | 4.00 | | | | | | | | |
| 9 | 3.36 | 6.14 | 7.14 | 8.10 | 2.00 | 4.71 | 6.52 | 7.71 | | | | | | | |
| 10 | 5.93 | 4.24 | 6.07 | 6.93 | 5.00 | 2.81 | 5.43 | 5.67 | 4.38 | | | | | | |
| 11 | 6.71 | 5.60 | 4.29 | 5.90 | 6.86 | 4.50 | 2.64 | 5.21 | 6.26 | 3.60 | | | | | |
| 12 | 7.88 | 6.31 | 5.48 | 5.00 | 7.83 | 5.55 | 4.43 | 2.69 | 7.21 | 5.83 | 3.60 | | | | |
| 13 | 3.69 | 6.98 | 7.98 | 8.45 | 2.60 | 5.95 | 7.69 | 7.86 | 1.60 | 4.31 | 6.95 | 7.43 | | | |
| 14 | 5.86 | 4.55 | 6.64 | 7.17 | 4.86 | 2.88 | 5.40 | 6.50 | 4.14 | 1.19 | 3.79 | 5.88 | 4.17 | | |
| 15 | 7.36 | 5.88 | 4.55 | 6.79 | 6.93 | 4.50 | 3.50 | 5.55 | 5.95 | 3.95 | 1.48 | 4.60 | 6.07 | 4.02 | |
| 16 | 8.36 | 7.02 | 5.86 | 5.40 | 7.57 | 5.86 | 4.52 | 3.50 | 6.86 | 5.17 | 3.71 | 1.62 | 7.07 | 5.26 | 3.45 |

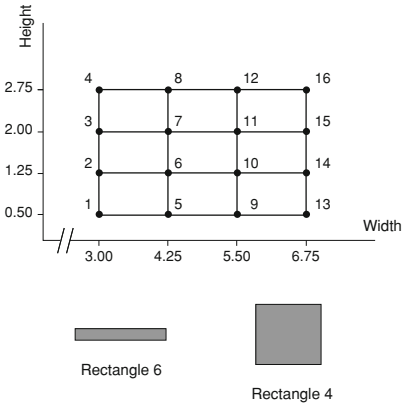


**Fig. 2.4** Design configuration for 16 rectangles with different widths and heights; *lower panel* shows two rectangles in a pair comparison

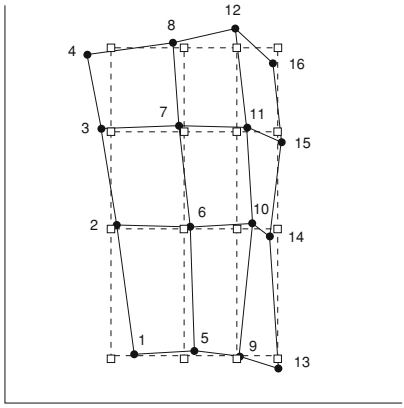


**Fig. 2.5** MDS configuration with city-block distances for data of Table 2.3 (*points*) and design configuration of Fig. 2.4 (*squares*) fitted to MDS configuration

but never along diagonals—just like walking from A to B in Manhattan, where the distance may be "two blocks West and three blocks North". Hence the name city-block distance. For points that lie on a line parallel to one of the coordinate axes, all Minkowski distances are equal (see points $M$ and $i$ in Fig. 2.3); otherwise, they are not equal. If you walk from $M$ to $j$ (or to $j'$ or $j''$, respectively) on a Euclidean path ("as the crow flies"), the distance is shorter than choosing the city-block path which

runs around the corner. The shortest path corresponds to the dominance distance: The largest intra-dimensional difference will get you from $M$ to the other points. This is important for the MDS user because it shows that rotating the coordinate axes generally changes all Minkowski distances, except Euclidean distances.

To see how the distance formula can serve as a model of judgment, consider an experiment by Borg and Leutner (1983). They constructed rectangles on the basis of the grid design in Fig. 2.4. Each point in this grid defines one rectangle. Rectangle 6, for example, had a width of 4.25 cm and a height of 1.25 cm; rectangle 4 was 3.00 cm wide and 2.75 cm tall. A total of 21 persons rated (twice) the similarity of each pair of these 16 rectangles (see example in Fig. 2.4, lower panel) on a 10-point answer scale ranging from "0=equal, identical" to "9=very different". The means of these ratings over persons and replications are shown in Table 2.3.

The MDS representation (using city-block distances) of these ratings is the grid of solid points in Fig. 2.5. From what we discussed above, we know that this configuration must not be rotated relative to the given coordinate axes, because rotations would *change* its (city-block) distances and, since the MDS representation in Fig. 2.5 is the best-possible data representation, it would deteriorate the correspondence of MDS distances and data.

If one allows for some re-scaling of the width and height coordinates of the rectangles, one can fit the design configuration quite well to the MDS configuration (see grid of dashed lines in Fig. 2.5). The optimal re-scaling makes psychological sense: It exhibits a logarithmic shrinkage of the grid lines from left to right and from bottom to top, as expected by psychophysical theory.

The deviations of the re-scaled design configuration and the MDS configuration do not appear to be systematic. Hence, one may conclude that the subjects have indeed generated their similarity ratings by a composition rule that corresponds to the city-block distance formula (including a logarithmic re-scaling of intra-dimensional distances according to the Weber–Fechner law). The MDS solution also shows that differences in the rectangles' heights are psychologically more important for similarity judgments than differences in the rectangles' widths.

## 2.4 MDS for Testing Structural Hypotheses

A frequent application of MDS is using it to test structural hypotheses. In the following, we discuss a typical case from intelligence diagnostics (Guttman and Levy 1991). Here, persons are asked to solve several test items. The items can be classified on the basis of their content into different categories of two design factors, called *facets* in this context. Some test items require the testee to solve computational problems with *numbers* and numerical operations. Other items ask for *geometrical* solutions where figures have to be rotated in 3-dimensional space or pictures have to be completed. Other test items require *applying* learned rules, while still others have to be solved by *finding* such rules. One can always code test items in terms of such facets, but the facets are truly interesting only if they exert some control over the observations, i.e.

**Table 2.4** Intercorrelations of eight intelligence test items, together with codings on two facets

| Format | Requirement | Item | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|
| N | A | 1 | 1.00 | 0.67 | 0.40 | 0.19 | 0.12 | 0.25 | 0.26 | 0.39 |
| N | A | 2 | 0.67 | 1.00 | 0.50 | 0.26 | 0.20 | 0.28 | 0.26 | 0.38 |
| N | I | 3 | 0.40 | 0.50 | 1.00 | 0.52 | 0.39 | 0.31 | 0.18 | 0.24 |
| G | I | 4 | 0.19 | 0.26 | 0.52 | 1.00 | 0.55 | 0.49 | 0.25 | 0.22 |
| G | I | 5 | 0.12 | 0.20 | 0.39 | 0.55 | 1.00 | 0.46 | 0.29 | 0.14 |
| G | A | 6 | 0.25 | 0.28 | 0.31 | 0.49 | 0.46 | 1.00 | 0.42 | 0.38 |
| G | A | 7 | 0.26 | 0.26 | 0.18 | 0.25 | 0.29 | 0.42 | 1.00 | 0.40 |
| G | A | 8 | 0.39 | 0.38 | 0.24 | 0.22 | 0.14 | 0.38 | 0.40 | 1.00 |

if the *distinctions* they make are mirrored somehow in corresponding *effects* on the data side. The data in our small example are the intercorrelations of eight intelligence test items shown in Table 2.4. The items are coded in terms of the facets "Format = {N(umerical), G(eometrical)}" and "Requirement = {A(pply), I(nfer)}".

A 2-dimensional MDS representation of the data in Table 2.4 is shown in Fig. 2.6. We now ask if the facets Format and Requirement surface in some way in this plane. For the facet Format we find that the plane can indeed be *partitioned* by a straight line such that all points labeled as "G" are on one side, and all "N" points on the other (Fig. 2.7). Similarly, using the codings for the facet Requirement, the plane can be partitioned into two subregions, an A- and an I-region. For the Requirement facet, we have drawn the partitioning line in a curved way, anticipating test items of a third kind on this facet: Guttman and Levy (1991) extent the facet Requirement by adding the element "Learning". They also extent the facet Format by adding "Verbal".

For the intercorrelations of items in this $3 \times 3$ design, that is, for items coded in terms of two 3-element facets, MDS leads to structures with a partitioning system as shown in Fig. 2.8. This pattern, termed *radex*, is often found for items that combine a qualitative facet (such as Format) and an ordered facet (such as Requirement). For the universe of typical intelligence test items, Guttman and Levy (1991) suggest yet another facet, called Communication. It distinguishes among Oral, Manual, or Paper-and-Pencil items. If there are test items of all $3 \times 3 \times 3$ types, MDS leads to a 3-dimensional *cylindrex* structure as shown in Fig. 2.9. Such a cylindrex shows, for example, that the items of the type Infer have relatively high intercorrelations (given a certain mode of Communication), irrespective of their Format. It is interesting to see that Apply is "in between" Infer and Learn. We also note that our small sample of test items of Table 2.4 fits perfectly into the larger structure of the universe of intelligence test items.

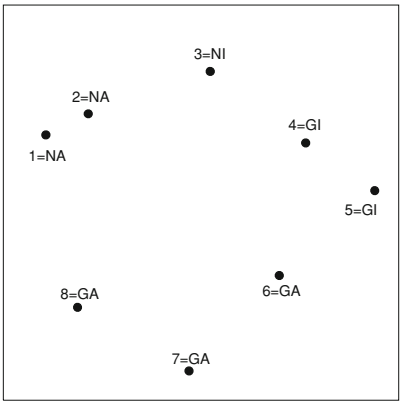


**Fig. 2.6** MDS solution for correlations in Table 2.4

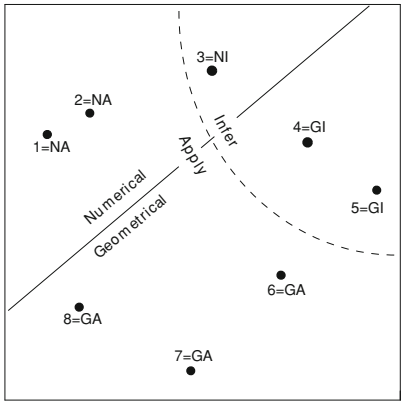


**Fig. 2.7** MDS configuration partitioned by two facets

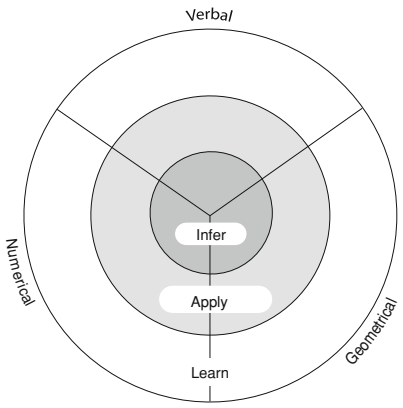


**Fig. 2.8** Schematic radex of intelligence test items

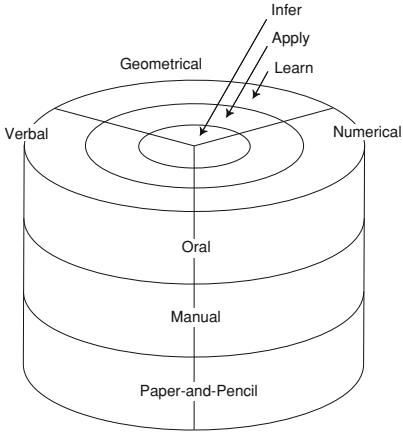


**Fig. 2.9** Cylindrex of intelligence test items

## 2.5 Summary

Originally, MDS was a psychological model for how persons arrive at judgments of similarity. The model claims that the objects of interest can be understood as points in a space spanned by the objects' subjective attributes, and that similarity judgments are generated by computing the distance of two points from their coordinates, i.e. by summing the intra-dimensional differences of any two objects over the dimensions of the space. Different variants of Minkowski distances imply that the intra-dimensional differences are weighted by their magnitude in the summing process. Today, MDS is

used primarily for visualizing proximity data so that their structure becomes accessible to the researcher's eye for exploration or for testing certain hypotheses. Structural hypotheses are often based on content-based classifications of the variables of interest in one or more ways. Such classifications should then surface in the MDS space in corresponding (ordered or unordered) regions. Certain types of regionalities (e.g., radexes) are often found in empirical research.

## References

Borg, I. (2008). *Employee surveys in management: Theories, tools, and practical applications*. Cambridge, MA: Hogrefe.

Borg, I., & Leutner, D. (1983). Dimensional models for the perception of rectangles. *Perception and Psychophysics, 34*, 257–269.

Garner, W. R. (Ed.). (1974). *The processing of information and structure*. Potomac, MD: Erlbaum.

Guttman, L., & Levy, S. (1991). Two structural laws for intelligence tests. *Intelligence, 15*, 79–103.

Kruskal, J. B., & Wish, M. (1978). *Multidimensional scaling*. Beverly Hills, CA: Sage.

Liu, C., Borg, I., & Spector, P. E. (2004). Measurement equivalence of a German job satisfaction survey used in a multinational organization: Implications of Schwartz's culture model. *Journal of Applied Psychology, 89*, 1070–1082.

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

Wish, M. (1971). Individual differences in perceptions and preferences among nations. In C. W. King & D. Tigert (Eds.), *Attitude research reaches new heights* (pp. 312–328). Chicago: American Marketing Association.

# Chapter 3
# The Goodness of an MDS Solution

**Abstract** Ways to assess the goodness of an MDS solution are discussed. The Stress measure and some of its variants are introduced. Criteria for evaluating Stress are presented.

**Keywords** Stress · Disparity · Shepard diagram · Stress-1 · Stress-norm · S-Stress

MDS always searches for coordinate values of $n$ points in $m$ dimensions (i.e., for an $n \times m$ coordinate matrix $\mathbf{X}$) whose distances represent the given proximities as precisely as possible ("optimally"). In the following, we will discuss exactly how the goodness of an MDS solution can be measured.

## 3.1 Fit Indices and Loss Functions

The goodness of an MDS solution is visualized in a *Shepard diagram*, a scatter plot of the data (proximities) versus the corresponding distances in MDS space. Figure 3.1 exhibits the Shepard diagram for the country similarity data discussed in Sect. 2.2. The plot shows that the distances of the MDS solution in Fig. 2.2 become smaller, in general, if the corresponding proximities of Table 2.1 become greater. The closeness of this regression trend can be measured, for example, by computing a correlation coefficient as an index of *fit*. Alternatively, one can formulate a *loss function* that shows how much data information is lost in the MDS representation. Geometrically, this loss corresponds to the (vertical) scatter of the points in a Shepard diagram about a regression line that is optimally fitted to the points.

The regression line in Fig. 3.1 is not the usual *linear* one, but a *monotone* step function, because we used *ordinal* MDS to scale these data.[1] Ordinal MDS attempts

[1] More precisely, it is "weakly monotonically descending", where "weak" means that it admits horizontal steps. A "strictly" monotonically descending function, in contrast, always runs downwards from left to right. Strictness is theoretically more desirable but mathematically more complicated and practically irrelevant because the angle of descent can be arbitrarily small.

I. Borg et al., *Applied Multidimensional Scaling*, SpringerBriefs in Statistics,
DOI: 10.1007/978-3-642-31848-1_3, 

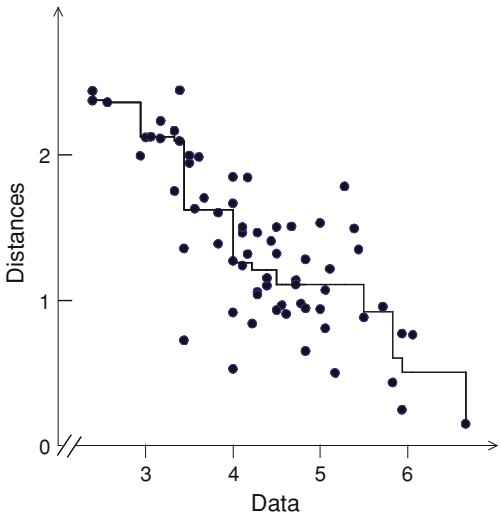


**Fig. 3.1** Shepard diagram for the MDS solution in Fig. 2.2. The vertical distance between a point and the regression line gives the error of the corresponding distance in the MDS representation

to map the proximities into distances so that the greater the proximity of two objects $i$ and $j$, the smaller the distance of the corresponding points $i$ und $j$ in MDS space. Differences, ratios, or other metric properties of the data are ignored in ordinal MDS, because they are not considered important or reliable under this model.

In the Shepard diagram in Fig. 3.1, the loss of information of the MDS representation can be measured as the sum of squared distances of the points from the regression line in the vertical direction (residuals, errors),

$$\sum_{i<j} e_{ij}^2 = \sum_{i<j} \big(f(p_{ij}) - d_{ij}(\mathbf{X})\big)^2 , \tag{3.1}$$

for all non-missing proximities $p_{ij}$. Missing proximities are skipped. The $d_{ij}(\mathbf{X})$'s are distances (computed by formula 2.3 for the configuration $\mathbf{X}$) and the $f(p_{ij})$'s are *disparities*, i.e. proximities optimally re-scaled (within the bounds set by the scale level assigned to the data) so that they approximate the distances as much as possible. Expressed more technically, disparities are computed by regression (of type $f$) of the proximities onto the distances so that $f(p_{ij}) = \widehat{d}_{ij}$ while minimizing (3.1). The distances $d_{ij}(\mathbf{X})$ are Euclidean distances in most MDS applications, computed by formula (2.1).

Since (3.1) is minimized both over both $\mathbf{X}$ and the $\widehat{d}_{ij}$s, an obvious but trivial solution is to choose $\mathbf{X} = \mathbf{0}$ and all $\widehat{d}_{ij} = 0$. To avoid this, (3.1) needs to be normalized. This can be done by dividing (3.1) by the sum of the squared distances. Doing so and taking the square root[2] gives the usual Stress-1 loss function of MDS:

[2] The square root has no deeper meaning here; its purpose is to make the resulting values less condensed by introducing more scatter.

$$\textit{Stress-1} = \sqrt{\sum_{i<j} \left(d_{ij}(\mathbf{X}) - \widehat{d}_{ij}\right)^2 / \sum_{i<j} d_{ij}^2(\mathbf{X})}. \tag{3.2}$$

This normalization has the advantage that the Stress values do not depend on the size of configuration $\mathbf{X}$.

## 3.2 Evaluating Stress

A perfect MDS solution has *Stress-1* $= 0$. If this is true then the distances of the MDS configuration represent the data precisely (in the desired sense). The MDS solution in Fig. 2.2 has a Stress-1 value of 0.19. Hence, it represents the data only "approximately".

This leads to the question whether *approximately correct* is also *good enough*. What is often considered as the "nullest of all null" answers to this question is that the observed Stress-1 must be clearly (e.g., two standard deviations) smaller than the Stress-1 value expected for random data. If this is not true, then it is impossible to interpret the MDS distances in any meaningful sense because then the distances are not reliably related to the data. In this case, the points in MDS space are not fixed; rather, they can be moved around more or less arbitrarily without affecting the Stress.

In practical applications, however, one almost always finds that the observed Stress-1 value is clearly smaller than most Stress-1 values that can expected for random data. For example, for $n = 12$ points in $m = 2$ dimension, Figs. 3.2 and 3.3 show that the expected random Stress-1 is about 0.24, with a standard deviation of 0.012 (according to a simulation study by Spence and Ogilvie (1973)). The Stress-1 value for the MDS solution in Fig. 2.2 is 0.19, and thus it is clearly smaller than random Stress-1.

Another question in this context is whether increasing the dimensionality of an MDS solution leads to "significantly" smaller Stress values. To answer this question, one first computes MDS solutions in, say, 1-, 2-, 3-, and higher-dimensional spaces and then checks how the Stress values decrease when the dimensionality of the MDS solution is increased. One way to evaluate these values is to compare them with Stress values for random data and look for an elbow in the decreasing Stress-versus-dimensionality function, similar to scree tests in factor analysis. As simulation studies show (Spence and Graef 1974), the elbow indicates the dimensionality where additional dimensions represent only random components of the data. In real (not simulated) data, however, elbows are rarely pronounced. Rather, when the MDS dimensionality is increased, the Stress values typically tend to drop smoothly just like the values of an exponential decay function.

Evaluating a given Stress value is a complex matter. It involves a number of different parameters and considerations:

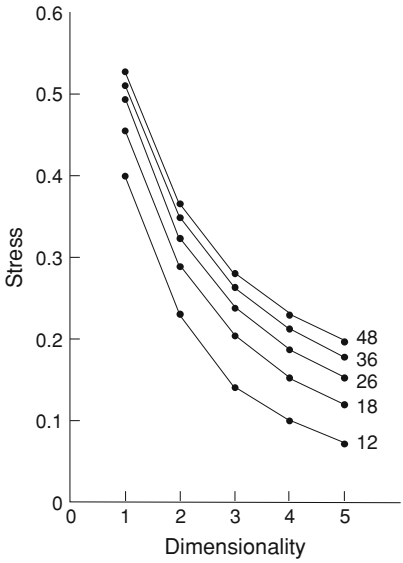


**Fig. 3.2** Average Stress-1 values for random proximities of $n$ objects; ordinal MDS in 1 to 5 dimensions, with $n = 12, 18, \ldots, 48$ points

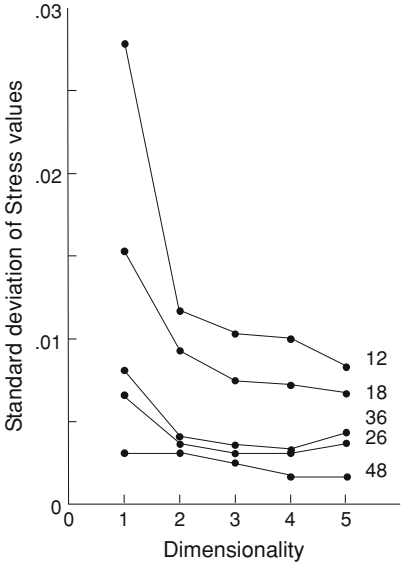


**Fig. 3.3** Standard deviations of curves in Fig. 3.2

- The number of point ($n$). The greater $n$, the larger the expected Stress (because the number of distances in an MDS solution grows almost quadratically as a function of $n$).
- The dimensionality of the MDS solution ($m$). The greater $m$, the smaller the expected Stress (because higher-dimensional spaces offer more freedom for an optimal positioning of points).
- The error component of the data. The greater the noise in the data, the larger the expected Stress (random data require maximal dimensionality).
- The MDS model. Metric (e.g., linear) MDS leads to higher Stress values than ordinal MDS because more restrictive models leave less freedom for choosing optimal $\widehat{d}_{ij}$ values.
- The number of ties when using the primary approach to ties in ordinal MDS (see Sect. 5.1). The more ties (=equal values) in the proximities, the smaller the expected Stress. The reason is that the primary approach to ties does not require that ties be mapped into equal distances, so MDS has more freedom to find an optimal solution.
- The proportion of missing proximities (missing data). The more data are missing, the easier it is to find an MDS solution with small Stress.

- Outliers and other special cases. Different points contribute differently to the total Stress; eliminating particular points or setting certain data as missing (e.g., because they are errors), can reduce the total Stress considerably.

## 3.3 Variants of the Stress Measure

The Stress in formula 3.2 is called Stress-1 or "Kruskal's Stress". If you read about Stress in publications, and if no further specifications are made, then you may assume that this is what is meant by "Stress".

Besides Stress-1, various other versions exist for Stress. One such variant is used in PROXSCAL, i.e. *normalized Stress*,

$$\textit{Stress-norm} = \sum_{i<j} \left(d_{ij}(\mathbf{X}) - \widehat{d}_{ij}\right)^2 / \sum_{i<j} \widehat{d}_{ij}^2.$$

This Stress variant differs from Stress-1 by using (a) in its denominator the sum of squared d-hats, not the sum of squared distances; and (b) by omitting the final square root transformation. It is nice to know, however, that when convergence is reached for an MDS solution, then normalized Stress is equal to the square of Stress-1:

$$\textit{Stress-norm} = \textit{Stress-1}^2.$$

In the literature, one usually reports Stress-1. One reason is tradition, another one is that Stress-1 values for good and for poor MDS solutions, respectively, are numerically more different and, thus, easier to discriminate. However, both Stress measures are zero if the MDS solution is perfect, and larger values indicate poorer data-distance fit.

From normalized Stress, two further variants can be derived. One of them is Dispersion Accounted For (DAF). DAF is equal to $1 - \textit{Stress-norm}$. It measures the proportion of the sum of squared disparities accounted for by the distances. The second measure is Tucker's *congruence coefficient*, $c = 1 - \sqrt{\text{DAF}}$. It can be interpreted similarly as the usual correlation coefficient, except that it does not extract the overall mean out of the distances and the $\widehat{d}_{ij}$'s.

Occasionally, one also encounters *S-Stress* (squared Stress),

$$\textit{S-Stress} = \sum_{i<j} \left(d_{ij}^2(\mathbf{X}) - \widehat{d}_{ij}^2\right)^2 / \sum_{i<j} \widehat{d}_{ij}^4,$$

which is used in the MDS program ALSCAL, an alternative MDS module in SPSS. Compared to Stress-1, S-Stress weights the fit of large distances much more heavily than the fit of small distances. Hence, in a program that minimizes S-Stress, small distances do not matter much in finding an optimal solution. This may be a sensible

choice for particular data, but as a general approach we do not recommend to use S-Stress and ALSCAL with this automatic built-in weighting.

The weighting in S-Stress can also be inverted for MDS, that is, using greater weightings for small rather than large distances. However, to do such weightings automatically is not recommended in general MDS. Hence, minimizing normalized Stress or Stress-1 is the criterion of choice for standard MDS, because it treats small and large distances equally.

## 3.4 Summary

The formal goodness of an MDS solution can be measured by different indices. The common way to do this is by computing the solution's Stress. Stress is a loss function: It is zero when the solution is perfect; otherwise, it is greater than zero. Stress aggregates into one measure the deviations of the points from the regression line in a data-versus-distances plot (Shepard diagram). The type of regression depends on the scale level of the data: For ordinal data, one chooses ordinal regression; for data on an interval scale, linear regression is the proper form. When evaluating the Stress value of a particular MDS solution, the user must assess it in the context of various parameters and contingencies such as the number of points, the dimensionality of the MDS space, the rigidity of the particular MDS model, and the reliability of the data. A minimum criterion for an acceptably low Stress value is that it is clearly smaller than the Stress expected for random data.

## References

Spence, I., & Graef, J. (1974). The determination of the underlying dimensionality of an empirically obtained matrix of proximities. *Multivariate Behavioral Research, 9*, 331–341.

Spence, I., & Ogilvie, J. C. (1973). A table of expected stress values for random rankings in nonmetric multidimensional scaling. *Multivariate Behavioral Research, 8*, 511–517.

# Chapter 4
# Proximities

**Abstract** The data for MDS, proximities, are discussed. Proximities can be collected directly as judgments of similarity; proximities can be derived from data vectors; proximities may result from converting other indices; and co-occurrence data are yet another popular form of proximities.

**Keywords** Similarity ratings · Sorting method · Feature model · LCJ model · Co-occurrence data · S-coefficient · Jaccard coefficient · Simple matching coefficient

A major advantage of MDS over related structural analysis methods (such as, e.g., factor analysis) is that MDS can handle very different data as long as these data can be interpreted as indexes of similarity or dissimilarity. Collectively, such indexes are called proximities. They can be collected either directly (e.g., as numerical ratings of similarity) or they can be derived from other data (e.g., correlations).

## 4.1 Direct Proximities

In Sect. 2.2, we discussed a study where the similarity of different countries was assessed by asking persons to directly rate all pairs of 12 different countries on a 9-point response scale. More concretely, each pair of countries (e.g., "Japan–China") was presented to the respondents, together with a rating scale with nine categories numbered from 1 to 9 and labeled as "very different" (for category 1) to "very similar" (for category 9). This method generated 66 pairwise ratings per person, enough data to scale each single person via MDS.

A similar procedure was used in Sect. 2.3, where a sample of subjects was asked to judge the pairwise similarities of 16 different rectangles on a 10-point rating scale ranging from "0=equal, identical" to "9=very different".

Pairwise similarity ratings can become difficult for the respondents. The rating scale may be too fine-grained (or too coarse) for some respondents so that their

I. Borg et al., *Applied Multidimensional Scaling*, SpringerBriefs in Statistics,
DOI: 10.1007/978-3-642-31848-1_4, © The Author(s) 2013

ratings become unreliable. Market researchers, therefore, typically prefer a *sorting method* over ratings, where the testees work with a deck of cards. Each card exhibits exactly one pair of objects (e.g., the pair "Japan–China"). The testees are asked to sort the cards such that the card with the most similar pair is on top of the stack, and the card showing the most dissimilar pair at the bottom. Since complete sortings of all cards are often too difficult, the sorting can be simplified: the testees are asked to begin by sorting the cards into only two stacks, one for the "similar" pairs of objects, and one for the "dissimilar" pairs. For each stack, this two-stacks sorting is repeated several times until the testees feel that they cannot reliably split the remaining stacks any further. One then numbers the various stacks and assigns these stack numbers to the pairs in the respective stacks. Thus, pairs that belong to the same stack receive the same proximity value.

These examples show that collecting direct proximities can be done on the basis of relatively simple judgments. However, pairwise ratings and card sortings also have their drawbacks. They can both lead to an excessively large number of pairs that must be judged by the subjects. For the $n = 12$ countries of the study in Sect. 2.2, for example, each subject had to rate 66 different pairs of countries. That seems acceptable, but for $n = 20$ countries, say, the number of different pairs goes up to $n \cdot (n-1)/2 = 190$. Assessing that many pairs (without any replications!) is a challenge even for a very motivated test person. To alleviate this problem, various designs for reducing the number of pair comparisons were developed. It was found that a random sample of all possible pairs is not only a simple but also a good method of reduction: One collects only data on the pairs that belong to the sample, and sets the proximities of all other pairs to "missing".

Spence and Domoney (1974) showed in extensive simulation studies that the proportion of missing data can be as high as 80 % and (ordinal) MDS is still able to recover an underlying MDS configuration quite precisely. One should realize, however, that these simulations made a number of simplifying assumptions that cannot automatically be taken for granted in real applications. The simulations first defined some random configuration in $m$-dimensional space. The distances among its points were then superimposed with random noise and taken as proximities. Finally, certain proximities were eliminated either randomly or per systematic design. The $m$-dimensional MDS configurations computed from these data were then compared with the $m$-dimensional configurations that served to generate the data. The precision with which MDS was able to reconstruct the original configurations from the proximities was found to depend on the proportion of missing data; on the proportion of random noise superimposed onto the distances; and on the number of points. In all simulated cases, the dimensionality of the MDS solution was equal to the true dimensionality, and the number of points was relatively large from an MDS user's point of view (i.e., 32 or more). Under these conditions, MDS was able to tolerate large proportions of missing data. If, for example, one third of the proximities is missing and the error component is equal to 15 %, then the MDS distances can be expected to correlate with $r = 0.97$ with the original distances!

One can improve the robustness of MDS by paying particular attention to collecting proximities for pairs of objects that seem very dissimilar rather than similar,

because one thus has proximities for the large distances and they are particularly important for the precision of recovery (Graef and Spence 1979).

The labor involved in data collection can be further reduced by simplifying the individual similarity judgments. Rather than asking for graded ratings on, say, a 10-point scale, one may offer only two response categories, "similar" (1) and "not similar" (0) for each pair of objects. Summing such dichotomous data over replications or over respondents leads to *confusion frequencies* or, after dividing by the number of cases, to *confusion probabilities*. Yet, such aggregations are not necessarily required. Green and Wind (1973) showed in a simulation study that robust MDS is possible using coarse data—given advantageous side constraints such as scaling in the true dimensionality and having many points. The study shows, though, that *some* grading of the data is better than 1–0 data, but very fine grading has essentially no effect on the robustness of MDS. Hence, if one collects direct proximities, it is not necessary to measure up to many decimals (if that is possible at all); rather, nine or ten scale values are sufficient for MDS.

## 4.2 Derived Proximities

Direct proximities are rather rare in practice. Most applications of MDS are based on proximity indices derived from pairs of data vectors. One example are the proximities used in Chap. 1, where the correlations of the frequencies of different crimes in 50 U.S. states were taken as proximities of these crimes.

Indexes for the similarity of data profiles are often used in market research. Assume we want to assess the subjective similarity of different cars. Proximities could be generated by first asking a sample of test persons to rate the cars we are interested in on such attributes as design, fuel consumption, costs, and performance. Then, the correlations of the ratings of the test persons for each pair of cars can be taken as an indicator of perceived similarity.

Instead of using correlation coefficients, one can also consider measuring profile similarity by the Euclidean or by the city-block distance. Such distances can differ substantially from correlations. If two data profiles have the same "shape" of ups and downs so that their values differ by an additive constant $c$ only, their correlation is equal to 1, but their distance is not equal to zero but rather equal to $c$. Conversely, two profiles with the same distance $c$ can correlate with 0 if, for example, their profiles are not parallel but if they cross in the form of an X (Borg and Staufenbiel 2007). Hence, whether one wants to use a correlation or a distance for measuring the proximity of profiles, must be carefully considered. If additive constants are not reliable because the data are at most on an interval scale, distances of profiles are not meaningful.

Besides Minkowski distances, many other distance functions are used in data analysis. An interesting case is discussed by Restle (1959). In his feature models of similarity, he defines the distance between two psychological objects as the relative proportion of the elements in their mental representations that are specific for each object. That is, for example, if a person associates with Japan the features X, Y, and

Z, and with China A, B, X, and Z, then their psychological distance is $3/5 = 0.6$, because there is a total of five different mental elements and three of them (Y, A, and B) are specific ones.

## 4.3 Proximities from Index Conversions

Proximities can sometimes be generated by theory-guided conversions of given measurements on pairs of objects. Here is one example. Glushko (1975) was interested in assessing the psychological "goodness" of dot patterns. He constructed a set of different patterns and printed each possible pair on a separate card. Twenty subjects were then asked to indicate which pattern in each pair is the "better" one. The pattern judged "better" in a pair received a score of 1, the other one a 0. These scores were summed over all subjects. A dissimilarity measure was constructed from these sums by subtracting 10 (i.e., the expected value for each pair of patterns if all 20 subjects decide their preferences randomly) from each sum and then taking the absolute value of this difference.

Borg (1988) used a different conversion to turn dominance probabilities into proximities. In the older psychological literature, many data sets are reported where $N$ persons are asked to judge which object in a pair of objects possesses more of a certain property. For example, considering crimes, the persons decide if murder is "more serious" than arson or not. Or, for paintings, is picture A "prettier" than picture B? The object chosen by the subjects receives a score of 1; the other object is rated as 0. If one then adds these "dominance" scores over all $N$ subjects, and divides by $N$, dominance probabilities, $w_{ij}$, are generated. These probabilities can be scaled by using Thurstone's Law-of-Comparative-Judgment procedure (Thurstone 1927). It assumes that each $w_{ij}$ is related to the distance $d_{ij}$ on a 1-dimensional scale by a cumulative normal function. This rather strong assumption can be replaced by a weaker model that gives the data more room to speak for themselves: this model simply postulates that the dominance probabilities are related to distances by a monotonically increasing function, without specifying the exact form of this function. To find the function that best satisfies this model, ordinal MDS can be used. First, however, one needs to convert the dominance probabilities into dissimilarities via $\delta_{ij} = |w_{ij} - 0.5|$. Then, the distances generated by ordinal MDS are plotted against the $w_{ij}$ probabilities. If Thurstone's model is correct, the regression trend should form an S-shaped function running from the lower left-hand side to the upper right-hand side.

More examples for index conversions are discussed in Borg and Groenen (2005). We do not pursue this topic here any further, because the two examples above should have made clear that it makes no sense to report such conversions one after the other in statistical textbooks. Rather, they always require substantive-theoretical considerations that can be quite specific for the particular setting.

**Table 4.1** Frequencies of four combinations of events $X$ and $Y$

| | $X = 1$ | $X = 0$ | Sum |
|---|---|---|---|
| $Y = 1$ | $a$ | $b$ | $a + b$ |
| $Y = 0$ | $c$ | $d$ | $c + d$ |
| Sum | $a + c$ | $b + d$ | $a + b + c + d$ |

## 4.4 Co-Occurrence Data

An interesting special case of proximities are co-occurrence data. Here is one example. Coxon and Jones (1978) studied the categories that people use to classify occupations. Their subjects were asked to sort a set of 32 occupational titles (such as barman, statistician, and actor) on the basis of their overall similarity into as many or as few groups as they wished. The result of this sorting can be expressed, for each subject, by a $32 \times 32$ *incidence matrix*, with an entry of 1 wherever its row and columns entries are sorted into the same group, and 0 elsewhere.

The incidence matrix in the example above is a data matrix of directly collected same-different proximities. This is not always true for co-occurrence data, as the following study by England and Ruiz-Quintanilla (1994) demonstrates. These authors studied "the meaning of working". For that purpose, they asked large samples of persons to consider a variety of statements such as "if it is physically strenuous" or "if you have to do it", and check those statements that would define work for them. The similarity of two statements (within the context of work) was then defined as the frequency with which these statements were both checked by each of the respondents. Note that here the similarity of two statements was never assessed directly. Rather, it was defined by the researchers. No person in these surveys was ever asked to judge the "similarity" or the "difference" of two statements.

Co-occurrence data are typically aggregated over persons. This is done by first adding the various combinations of occurrence, non-occurrence, and co-occurrence of any two events of the set of events of interest. Assume that $X$ und $Y$ are two such events of interest. Each event either occurs, or it does not occur. We denote this by $X = 1$ and by $X = 0$, respectively. There are four possible cases of the events $X$ and $Y$ to occur or to not occur. We denote the frequencies of these cases as $a$, $b$, $c$, and $d$, respectively, as shown in Table 4.1.

On the basis of Table 4.1, one can define an amazing number of different similarity measures. The two most prominent ones in a system of such coefficients proposed by Gower (1985) are

$$S_2 = a/(a + b + c + d),$$

the frequency of events where both $X$ and $Y$ occur, relative to the frequency of all possible combinations of $X$ and $Y$ ($= a + b + c + d$). Another coefficient is

$$S_3 = a/(a + b + c),$$

the frequency of a joint occurrence of $X$ and $Y$, relative to the frequency of cases where at least one of the events $X$ and $Y$ occurs (*Jaccard similarity index*).

Choosing a particular S-index over another such index can have dramatic consequences. Bilsky et al. (1994) report a study on different behaviors exhibited in family conflicts, ranging from calm discussions to physical assault. The intention was to find out in which psychologically meaningful ways such behaviors can be scaled. They conducted a survey asking which of a list of different conflict behaviors had occurred in the respondent's family in the last five years. If one assesses the similarities of the reported behaviors by using $S_3$, then an MDS generates a 1-dimensional scale on which the behaviors are arrayed from low to high aggression. This order makes psychological sense. If one uses $S_2$, however, then this simple solution falls apart. The reason is that the very aggressive behaviors are also relatively rare, which inflates $d$ so that these behaviors become highly dissimilar in the $S_2$ sense.

Of the many further variants of $S$-coefficients (Gower 1985; Cox and Cox 2000; Borg and Groenen 2005) we here mention the *simple matching coefficient*,

$$S_4 = (a + d)/(a + b + c + d),$$

which interprets both the joint occurrence and the joint non-occurrence of events as a sign of similarity. In the above example of conflict behaviors, $S_4$ would assess the rare forms of behavior as similar because they are rare.

## 4.5 The Gravity Model for Co-Occurrences

Direct analyses of co-occurrence data through MDS can yield uninteresting solutions if some of the objects are much more popular than others. One way to correct for such differential popularity of objects is the gravity model.[1] Consider the following example. We took Canadian newspapers that appeared in the time period between June and September 2009 and searched for articles that contained the word "aboriginal." A total of 92 articles was found. In these articles, we determined the frequencies of other meaningful[2] words (e.g., "tribal", "moose", "arctic", and "health"), and then counted the co-occurrences of the most frequent 112 words on our list of words.

Not surprisingly, a few words co-occur much more often than most others in the context of "aboriginal". Examples are "nations" (813 co-occurrences), "first" (460 co-occurrences), "communities" (266 co-occurrences), and "government" (214 co-occurrences). To avoid that these words dominate an MDS representation of co-occurrence data, we use the gravity model. Let $c_{ij}$ be the elements in the co-occurrence matrix, and let its diagonal contain the sum of the co-occurrences ($c_{ii} = \sum_{j \neq i} c_{ij}$) as a measure of the popularity of each word $i$. Then, the gravity model defines the

[1] The gravity model is based on a Newtonian law from physics that models the gravity forces between large masses such as the earth and the moon.

[2] Words such as "and", "but", "they" etc. were omitted.

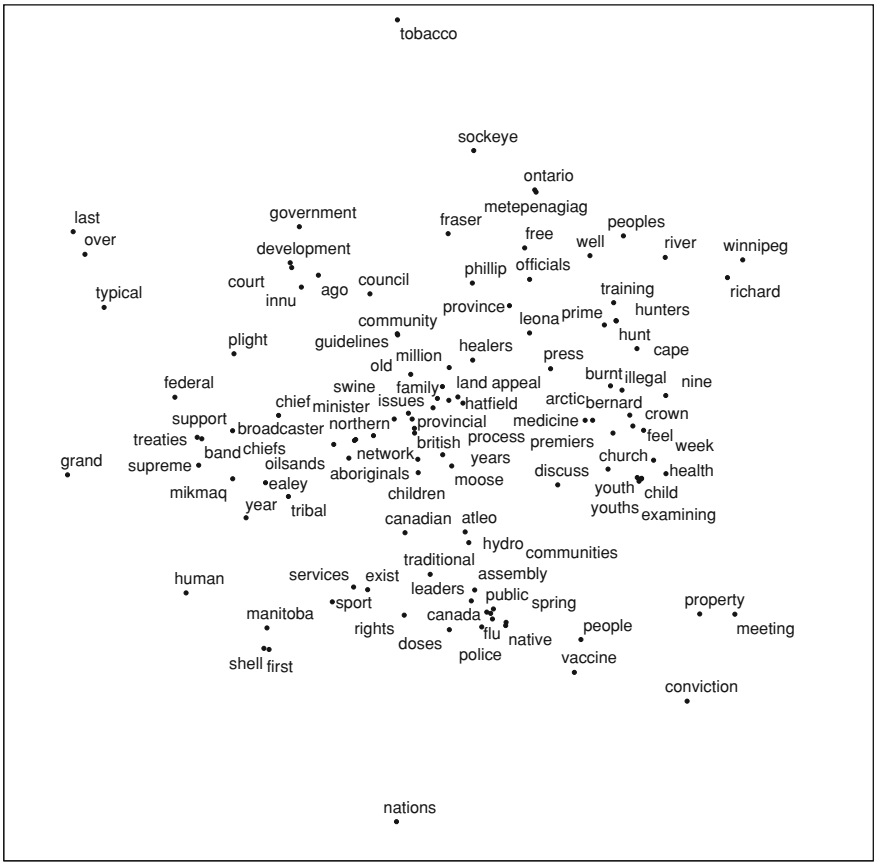


**Fig. 4.1** Ordinal MDS solution (PROXSCAL with Ties = Keep) for the gravity model using co-occurrences of 112 words that appear together with word "aboriginal" in 92 Canadian newspaper articles

dissimilarity of words $i$ and $j$ as

$$\delta_{ij} = \sqrt{\frac{c_{ii}c_{jj}}{c_{ij}}}, \quad \text{if} \quad c_{ij} > 0. \tag{4.1}$$

In (4.1), $1/c_{ij}$ transforms the similarity measure of co-occurrences into a dissimilarity measure, the multiplication by $c_{ii}c_{jj}$ standardizes the measure for the popularities of the respective words, and the square root follows from the physical law, but is immaterial for ordinal MDS.

Formula (4.1) leaves open what to do if two words do not co-occur, that is, if $c_{ij} = 0$. For such $ij$, we define $\delta_{ij}$ to be missing so that these values are skipped by the Stress function. This adaptation is important as most co-occurrences in our example

are zero. Without this adaptation, the 97 % zero co-occurrences would dominate the scaling, yielding an uninformative result (see Sect. 7.12).

The solution using the gravity model is given in Fig. 4.1. It has a low Stress of 0.024. To interpret it, one can focus on groups of words at the periphery of the plot. The words in the center have a similar pattern of relative co-occurrences with the outlying words. One notes that the word "grand" on the left side is associated relatively often with "supreme", "federal", and "treaties". At the right-hand side in the middle, words such as "youth", "child", "examining", "church", and "health" co-occur relatively often. Some words such as "nation" and "tobacco" are far from each other. They are also far from other words which indicates that they do not occur very often. To the right above the middle, many co-occurrences arise around the words "hunters", "hunt", "illegal", and "burnt". These associations may stem from articles on illegal hunting. Other groups of words could be identified and studied similarly.

## 4.6 Summary

MDS builds on proximity data. Many data qualify as proximity data. In psychological research, proximities are often collected directly by asking persons to rate the perceived similarity of objects of interest on a numerical rating scale. A popular alternative is sorting a stack of cards, with one card per object pair, in terms of the objects' similarity. Proximities can also be derived from other measures. The inter-correlations of the variables in a typical person-by-variables data matrix, for example, is a popular example. Sometimes, proximities can be constructed by converting other measures on pairs of objects such as, for example, probabilities with which object $i$ dominates object $j$. Yet another form of proximity data are measures that build on co-occurrence data, where the frequencies with which the events $i$ and $j$, respectively, occur or do not occur at time $t$ are combined into an index of co-occurrence such as Gower's $S$-indexes or the Jaccard index. For co-occurrence data with very skewed distributions, the gravity model offers one possibility to generate dissimilarities that lead to meaningful MDS solutions.

## References

Bilsky, W., Borg, I., & Wetzels, P. (1994). Assessing conflict tactics in close relationships: A reanalysis of a research instrument. In J. J. Hox, P. G. Swanborn, & G. J. Mellenbergh (Eds.), *Facet theory: Analysis and design* (pp. 39–46). Zeist, The Netherlands: Setos.

Borg, I. (1988). Revisiting Thurstone's and Coombs' scales on the seriousness of crimes and offences. *European Journal of Social Psychology*, *18*, 53– 61.

Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling* (2nd ed.). New York: Springer.

Borg, I., & Staufenbiel, T. (2007). *Theorien und Methoden der Skalierung* (4th ed.). Bern: Huber.

Cox, T. F., & Cox, M. A. A. (2000). *Multidimensional scaling* (2nd ed.). London: Chapman & Hall.

Coxon, A. P. M., & Jones, C. L. (1978). *The images of occupational prestige: A study in social cognition*. London: Macmillan.

England, G., & Ruiz-Quintanilla, S. A. (1994). How working is defined: Structure and stability. In I. Borg & S. Dolan (Eds.), *Proceedings of the Fourth International Conference on Work and Organizational Values* (pp. 104–113). Montreal: ISSWOV.

Glushko, R. J. (1975). Pattern goodness and redundancy revisited: Multidimensional scaling and hierarchical clustering analyses. *Perception & Psychophysics*, *17*, 158–162.

Gower, J. C. (1985). Measures of similarity, dissimilarity, and distance. In S. Kotz, N. L. Johnson, & C. B. Read (Eds.), *Encyclopedia of statistical sciences* (Vol. 5, pp. 397–405). New York: Wiley.

Graef, J., & Spence, I. (1979). Using distance information in the design of large multidimensional scaling experiments. *Psychological Bulletin*, *86*, 60–66.

Green, P. E., & Wind, Y. (1973). *Multivariate decisions in marketing: A measurement approach*. Hinsdale, IL: Dryden.

Restle, F. (1959). A metric and an ordering on sets. *Psychometrika*, *24*, 207–220.

Spence, I., & Domoney, D. W. (1974). Single subject incomplete designs for nonmetric multidimensional scaling. *Psychometrika*, *39*, 469–490.

Thurstone, L. L. (1927). A law of comparative judgment. *Psychological Review*, *34*, 273–286.

# Chapter 5
# Variants of Different MDS Models

**Abstract** Various form of MDS are discussed: Ordinal MDS, metric MDS, MDS with different distance functions, MDS for more than one proximity value per distance, MDS for asymmetric proximities, individual differences MDS models, and unfolding.

**Keywords** Ordinal MDS · Interval MDS · Ratio MDS · Drift-vector model · INDSCAL · IDIOSCAL · Unfolding

MDS is really a *family* of related models. They all have in common that they map proximities into distances between points of an $m$-dimensional space. What leads to different models are different assumptions about the scale level of the proximities on the data side, and using different distances on the side of the geometry of the model. These choices allow the user to use MDS for many forms of data and for a variety of purposes.

## 5.1 Ordinal and Metric MDS

A main difference of various MDS models is the scale level that the models assume for the proximities. The most popular MDS model in research publications using MDS has been *ordinal* MDS, sometimes also—less precisely—called *non-metric* MDS. Ordinal MDS builds on the premise that the proximities $p_{ij}$ are on an ordinal scale. That is, only their ranks are taken as reliable and valid information. Therefore, the task of ordinal MDS is to generate an $m$-dimensional configuration $\mathbf{X}$ so that the distances over $\mathbf{X}$ are ordered as closely as possible as the proximities. Expressed differently, the rank order of the distances should optimally correspond to the rank order of the data. Hence, in ordinal MDS, the function

$$f : p_{ij} \rightarrow d_{ij}(\mathbf{X}) \tag{5.1}$$

I. Borg et al., *Applied Multidimensional Scaling*, SpringerBriefs in Statistics,
DOI: 10.1007/978-3-642-31848-1_5, © The Author(s) 2013

is *monotone* so that

$$f : p_{ij} < p_{kl} \rightarrow d_{ij}(\mathbf{X}) \leq d_{kl}(\mathbf{X}) \tag{5.2}$$

for all pairs $i$ and $j$, and $k$ and $l$, respectively, for which data (here assumed to be dissimilarities) are given. Proximities that are not defined ("missing data") are skipped by these formulas. That is, if $p_{ij}$ is missing, it imposes no restriction onto the MDS solution so that the distance $d_{ij}(\mathbf{X})$ can be chosen arbitrarily.

An important distinction between two forms of ordinal MDS results from how it treats *ties* (equal data values). The default in most programs is that ties can be *broken* in the MDS solution, that is, equal proximities need *not* be mapped into equal distances. This is called the *primary approach to ties*. The *secondary* approach to ties ("keep ties tied") leads to an additional requirement for ordinal MDS, namely

$$f : p_{ij} = p_{kl} \rightarrow d_{ij}(\mathbf{X}) = d_{kl}(\mathbf{X}). \tag{5.3}$$

The primary approach to ties is usually more meaningful in terms of the data. Consider, for example, the data discussed in Sect. 2.2, where subjects had to judge the similarity of different countries on 9-point rating scales. Here, ties are unavoidable for formal reasons, because the rating scale had only nine different levels: with 66 different pairs of countries, this will automatically lead to the same proximity values for some pairs, whether or not the subject really feels that the respective countries are "exactly" equally similar. Moreover, no respondent can really make reliable distinctions on a 66-point scale. Also, each single judgment is more or less fuzzy, so that equal ratings should not be interpreted too closely.

A second class of MDS models, called *metric* MDS, goes back to the beginnings of MDS in the 1950s (Torgerson 1952). Such models specify an analytic (usually monotone) function for $f$ rather than requiring that $f$ must be only "some" monotone function. Specifying analytic mapping functions for $f$ has the advantage that it becomes easier to develop the mathematical properties of such models. Moreover, metric MDS also avoids some technical problems of ordinal MDS such as, in particular, degenerate solutions (see Sect. 7.7). Their disadvantage is that they require data on a higher scale level. Moreover, they typically lead to solutions with a poorer fit to the data, because it is generally more difficult to represent data in more restrictive models.

The standard model of metric MDS is *interval MDS*, where

$$p_{ij} \rightarrow a + b \cdot p_{ij} = d_{ij}(\mathbf{X}). \tag{5.4}$$

Interval MDS wants to preserve the data linearly in the distances. This makes sense only if the data are taken as interval-scaled. That is, it is assumed that no meaningful information of the data is lost if they are scaled by multiplying them by an arbitrary constant $b$ (except, of course, by $b = 0$) or by adding any constant $a$ to each data value. All statements about the data that remain invariant under such linear transformations are considered meaningful; all other statements (e.g., statements about the ratio of certain data values) are not meaningful.

More MDS models follow easily by choosing other mapping functions $f$ (e.g., an exponential function). However, if $f$ is not at least weakly monotone, then such functions do not lead to easily interpretable results.

A popular MDS model that is "stronger" than interval MDS is *ratio MDS*, often considered the most restrictive model in MDS. It drops the additive constant $a$ of interval MDS as an admissible transformation and searches for a solution that preserves the proximities up to a scaling factor $b$ ($b \neq 0$).

The researcher chooses a particular MDS model $f$ for a variety of reasons. One important criterion is the scale level he or she assigns to the data. If theoretical or empirical reasons speak for a certain scale level, then it usually makes sense to pick a corresponding MDS model. In practice, however, one often scales given proximities with both ordinal and interval MDS: Ordinal MDS normally leads to smaller Stress values, but it can also over-fit the data (rather than smoothing out noise in the distances) and, occasionally, it can lead to largely meaningless degenerate solutions (see Sect. 7.7).

## 5.2 Euclidean and Other Distances

A second criterion for classifying MDS models is choosing a particular distance function. In psychology, the family of Minkowski metrics (specified in formula 2.3) used to be popular for modeling subjective similarity judgments of different types of stimuli (analyzable vs. integral stimuli) under different conditions (such as different degrees of time pressure). However, applications of MDS in the current literature almost always use Euclidean distances as these are the only ones that correspond to the natural notion of distance between points "as the crow flies".

Euclidean distances, as all other Minkowski distances, imply a *flat* geometry. In special cases, it can be useful to construct MDS representations in *curved* spaces. As an example, one can think of distances on a sphere. Here, the distance between two points is the shortest path ("geodesic path") in the 2-dimensional curved space (i.e., on the sphere), which is the length of a cord spanned between two points over the surface of the sphere. Curved geometries can sometimes be useful (e.g., in psychophysics), but they are never used in general data analysis situations.

## 5.3 Scaling Replicated Proximities

In older applications of MDS, there is always exactly one data value per distance, or a missing value. Hence, the data can always be shown in the lower half of a proximity matrix as, for example, in Tables 2.1 and 2.3. Often, such matrices are generated by averaging the data of $N$ persons and/or by aggregating data over several replications.

Modern MDS programs allow the user to skip such pre-processings of the data. They allow using not just one proximity ($p_{ij}$) for each distance $d_{ij}$ but two or more data values ($p_{ij}^{(k)}$, $k = 1, 2, \ldots$). Instead of one data matrix, one can therefore think

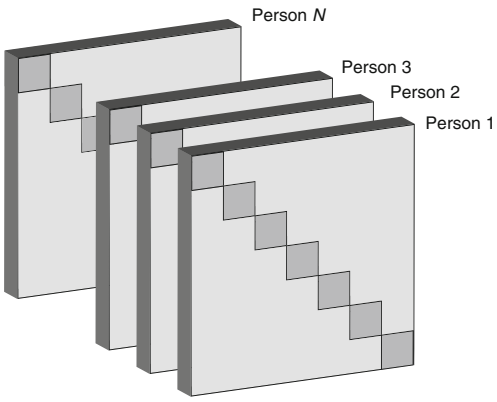


**Fig. 5.1** Data cube that consists of complete proximity matrices for $N$ persons

of the data as a "data cube", as shown in Fig. 5.1. The values in such a data cube could first be averaged over all persons, and then averaged once more over the two halves of the resulting complete matrix. The data in Table 2.3 were generated in this fashion. Alternatively, one could inform the MDS program that what we have here is one complete data matrix for each of $N$ persons, and that we want the program to find a solution where each $d_{ij}$ represents, as precisely as possible, up to $N \cdot 2$ proximities, where the "up to" means that missing data are possible. Only the main diagonals in each proximity matrix can *never* impact the MDS solution because $d_{ii} = 0$, for any $i$, is always true in any distance geometry.

## 5.4 MDS of Asymmetric Proximities

Distances are always symmetric, i.e. it always holds that $d_{ij} = d_{ji}$, for all $i, j$. Therefore, proximities that are not symmetric cannot be represented in MDS models. However, as long as the asymmetries in the data are just error-based, no real problem arises because MDS can smooth out such asymmetries or because the user has eliminated or at least reduced them by first averaging corresponding data values.

Asymmetries can, however, be reliable and valid pieces of information. Examples are the asymmetries in an import-export matrix, where country $X$ imports more from county $Y$ than vice versa. Another example is a social network that can be studied in terms of how much each person $i$ "likes" the other person $j$. Such liking measurements can, of course, also be asymmetric, and this can be very meaningful.

A simple approach of dealing with asymmetric proximities in the MDS context is the *drift vector model*. The model requires the user to first form two matrices from the proximity matrix $\mathbf{P}$. First, the symmetric component of $\mathbf{P}$ is computed by averaging corresponding cells, $\mathbf{S} = (\mathbf{P} + \mathbf{P}')/2$, with elements $s_{ij} = (p_{ij} + p_{ji})/2$. This matrix is then scaled as usual with MDS. The rest of the proximities, $\mathbf{A} = \mathbf{P} - \mathbf{S}$, with elements $a_{ij} = p_{ij} - s_{ij}$, is the *skew-symmetric* component of $\mathbf{P}$. It can be represented within the MDS solution for $\mathbf{S}$ by attaching an arrow on each point $i$ that points

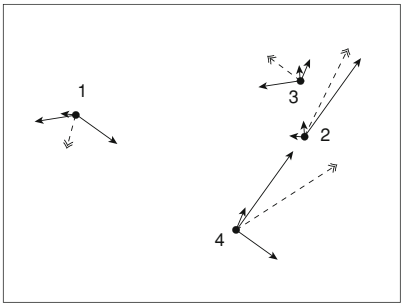


**Fig. 5.2** Vector field over an MDS solution; *dashed arrows* are resultants

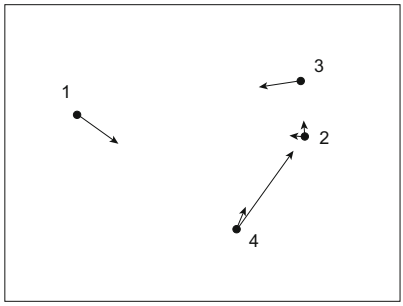


**Fig. 5.3** Vector field reduced to positive asymmetries

towards point $j$ or away from point $j$, depending on the sign of the asymmetry. The length of this arrow is chosen as $k \cdot |a_{ij}|$, with $k$ some scaling factor (e.g., $k = 1$). To simplify this display and to reduce the number of arrows, one can plot the resultant of all arrows emanating from each point $i$.

We demonstrate this model using a small example. Let $\mathbf{P}$ be a matrix of similarity values (e.g., measures on how much person $i$ likes person $j$, $j = 1, \ldots, 4$):

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} 0 & 4 & 6 & 13 \\ 5 & 0 & 37 & 21 \\ 4 & 38 & 0 & 16 \\ 8 & 31 & 18 & 0 \end{bmatrix} = \mathbf{S} + \mathbf{A} \\ &= \begin{bmatrix} 0.0 & 4.5 & 5.0 & 10.5 \\ 4.5 & 0.0 & 37.5 & 26.0 \\ 5.0 & 37.5 & 0.0 & 17.0 \\ 10.5 & 26.0 & 17.0 & 0.0 \end{bmatrix} + \begin{bmatrix} 0.0 & -0.5 & -2.0 & 2.5 \\ 0.5 & 0.0 & 0.5 & -5.0 \\ 2.0 & -0.5 & 0.0 & -1.0 \\ -2.5 & 5.0 & 1.0 & 0.0 \end{bmatrix}. \end{aligned} \tag{5.5}$$

For $\mathbf{S}$, interval MDS yields the point configuration in Fig. 5.2. In this plot, we insert the values of $\mathbf{A}$ as arrows. On point 1, we draw an arrow with a length of 2.5 units pointing towards point 4; additionally, we add an arrow of length 0.5 that points away from point 2; and, finally, we attach another arrow of length 2.0 pointing away from point 3. The resultant of these three arrows is the *drift vector*, represented by the dashed arrow on point 1. For the remaining points, we proceed analogously.

The *vector field* display can be simplified in various ways. Figure 5.3 exhibits only the arrows that are positively pointing from $i$ to $j$. One notices in this plot that persons 2 and 3 like each other a lot, and also symmetrically (because the points $i$ and $j$ are so close, and because the drift vector is so short). For persons 4 and 2, the mutual affection is clearly smaller and, moreover, it is also quite asymmetric. Otherwise, one notices a peculiar circular structure of the asymmetries from 1 to 4, from 4 to 2, and from 3 to 1.

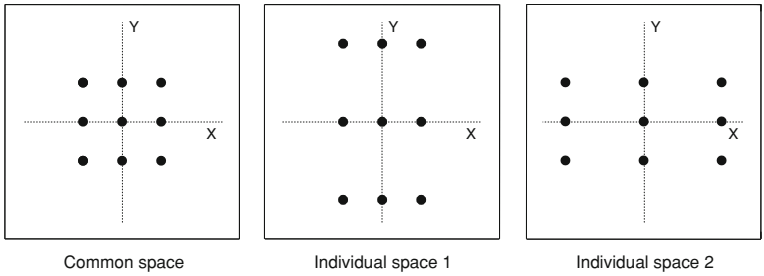


**Fig. 5.4** Illustration of the dimensional weighting model

One can experiment somewhat with how one wants to represent the asymmetries (e.g., show all arrows, only resultants, only positive vectors; use different scale factors for lengths of arrows). However, since there are presently no simple computer programs for such experiments, they entail a lot of cumbersome work. Nevertheless, the result may be worth the effort, because the resulting vector field can reveal asymmetries (over the symmetric base structure) that may be hard to detect in the data matrix.

## 5.5 Modeling Individual Differences in MDS

A popular variant of MDS is the *dimensional weighting model*, often called the INDSCAL model (Carroll and Chang 1970). The basic idea of this "model of subjective metrics" (Schönemann and Borg 1983) is illustrated schematically in Fig. 5.4. The plots show the psychological spaces of two individuals. The spaces are different, but they can both be generated from one *common space* by appropriate weightings of the dimensions $X$ and $Y$. In contrast to the usual MDS solutions that can be arbitrarily rotated, the dimensions in the INDSCAL model are fixed, in general.

If one drops this restriction of common dimensions for all individuals, a more general model arises that allows for person-specific (*idiosyncratic*) rotations of the common space (IDIOSCAL model). The consequence of allowing for a rotation of the common space before stretchings or compressions is that the point grid in Fig. 5.4 will be sheared, in general.

Expressed more formally, the INDSCAL model says that

$$d_{ijk}(\mathbf{X}) = \sqrt{\sum_{a=1}^{m} w_{ak}(x_{ia} - x_{ja})^2}, \quad w_{ak} > 0, \tag{5.6}$$

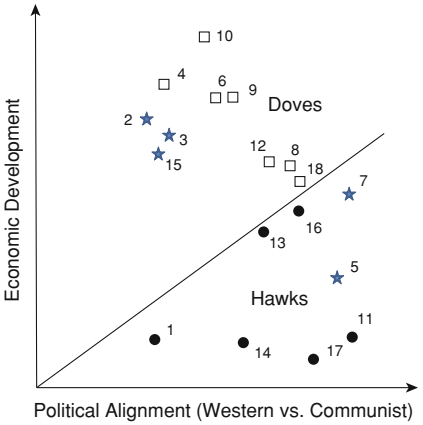


**Fig. 5.5** Dimensional weights for 18 persons in an INDSCAL analysis of the country similarity data from Sect. 2.2; *points* represent hawks, *squares* are doves, and *stars* are moderates

where the parameter $k = 1, \ldots, N$ stands for different individuals or cases. The weight $w_{ak}$ can be interpreted as the *salience* of dimension $a$ for person $k$ (Horan 1969).

Kruskal and Wish (1978) used this model to scale the raw data that led to the proximities in Table 2.1. Instead of first averaging the ratings over the 18 subjects of their experiment, they analyzed these ratings directly with a *three-mode MDS*, namely the INDSCAL model. This type of MDS yields a solution for the common space that is quite similar to the configuration shown in Fig. 2.2. The only real difference is that it is rotated by some 45° so that its $X$ and $Y$ axes closely correspond to the dimensions Wish (1971) had used in Sect. 2.2 for interpreting the MDS solution of the averaged proximities: Pro-Western versus Pro-Communist ($X$ dimension) and Economic Development ($Y$ dimension). The weights computed by INDSCAL for the 18 students are shown in Fig. 5.5. This plot (called the *subject space* in INDSCAL) shows that person 11 strongly compresses the common space along the $Y$ axis or, expressed differently, strongly over-weights the $X$ dimension. That is, this person pays relatively little attention to Economic Development in his or her judgments of similarity or, conversely, pays relatively much attention to the countries' political alignment. For person 4, the opposite is true. This interpretation is buttressed by additional data on these persons. On the basis of a question on the Vietnam war, the students were sorted into three groups: Hawks, doves, and moderates. These groups appear in the subject space in the expected regions of weights.[1]

The INDSCAL procedure is easily over-interpreted if used naively. One delicate issue is the question whether the fit of the model would be clearly worse if all

[1] The distance of a point $i$ from the origin in INDSCAL's subject space represents the goodness of the INDSCAL solution for person $i$. Figure 5.5 therefore exhibits that, for example, the data of person 1 are only relatively poorly explained by the INDSCAL solution, while the opposite is true for persons 10, 7, or 11.

dimension weights would be set to the same value. Borg and Lingoes (1978) report an example where the dimension weights scatter substantially even though they explain very little additional variance over unit weights. Hence, very different weights may represent very little variance, and then the unique orientation of the dimensions is not very strong either.

The dimension weights depend on the particular dimension system of the common space. Yet, one cannot infer from Fig. 5.5 that person 11 weights the countries' political alignment six times as strongly as their economic development. The reason is that the norming of the common space is *arbitrary*. That is, if the common space is stretched or compressed along its dimensions, different weights entail for each person while the overall fit of the MDS solution remains the same. What one can interpret in Fig. 5.5, for example, is that person 11 weights the horizontal dimension more than person 10, since this relation remains invariant under horizontal or vertical compressions or stretchings of the common space.

The idea of the dimension weighting model can also be realized in a more stepwise approach which avoids some of the interpretational problems. To do this, one first scales each of the given $N$ data matrices individually by MDS. One then uses Procrustean transformations to fit the $N$ resulting configurations to each other by admissible transformations (rotations, reflections, size adjustments, and translations) and computes the average configuration as the "common" configuration (*centroid configuration*). In this configuration, one then identifies the dimensions that, if weighted, optimally explain the individual MDS solutions. This *hierarchical* approach is used by the program PINDIS[2] (Lingoes and Borg 1978). It allows to user to check, in particular, how much variance is explained by using individual dimension weights over setting all weights equal for all individuals. For more information on Procrustean analysis, we refer to Borg and Groenen (2005).

The dimension weighting model is interesting but, in practice, it rarely leads to MDS solutions that are convincing in terms of fitting the data or in terms of substantive theory as in the case of Doves and Hawks above.[3] A more general reason for this finding is that such dimensional approaches to individual differences scaling are often inappropriate psychological models for judgments of similarity. Research has shown that different persons may generate very different "dimensionalizations" of even the simplest stimuli where the dimensions seem obvious and compelling apriorily (Schönemann 1994). In addition, they may use different distance functions or functions that do not even satisfy the basic axioms of distances (e.g., in asymmetric judgments; see Tversky 1977). However, formulating and studying models like INDSCAL have greatly contributed to today's more refined understanding of the psychology of similarity judgments.

---

[2] This program is included in the NEWMDSX package.

[3] Note also that if the number of points is small and the dimensionality is high (e.g., if $n = 5$ and $m = 3$), the model has many free parameters which help to generate a good model fit. For theory construction, such special cases are usually of little interest.

## 5.6 Unfolding

Another popular family of MDS models is called *Unfolding*, a model for dominance data,[4] in particular for preference data of $N$ persons for $n$ objects. The small example in Table 5.1 illustrates this case with data of five persons and four political parties. The scores are preference scores, ranging from 4 (= high) to 1 (= low). Person 3 has the strongest preference for party A, in second place comes party D, then C, and finally B. Such a data matrix can be understood as a special case of a proximity matrix where (a) the data express how close a particular person is to a particular party, and (b) where entire blocks of data are *missing*, namely the proximities among the parties, and also the proximities among the persons. Using regular MDS to scale these data, we get $5 + 4 = 9$ points, 5 for the persons, and 4 for the parties, as shown in Fig. 5.6. The points representing the persons are called *ideal points* in unfolding, because they are the points of maximal preference in space: The closer a party to

| | | Parties | | | | Persons | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | A | B | C | D | 1 | 2 | 3 | 4 | 5 |
| Parties | A | - | - | - | - | 1 | 4 | 4 | 4 | 3 |
| | B | - | - | - | - | 3 | 2 | 1 | 1 | 2 |
| | C | - | - | - | - | 2 | 3 | 2 | 3 | 1 |
| | D | - | - | - | - | 4 | 1 | 3 | 2 | 4 |
| Persons | 1 | 1 | 3 | 2 | 4 | - | - | - | - | - |
| | 2 | 4 | 2 | 3 | 1 | - | - | - | - | - |
| | 3 | 4 | 1 | 2 | 3 | - | - | - | - | - |
| | 4 | 4 | 1 | 3 | 2 | - | - | - | - | - |
| | 5 | 3 | 2 | 1 | 4 | - | - | - | - | - |

**Table 5.1** Fictitious preference values of 5 persons for 4 parties

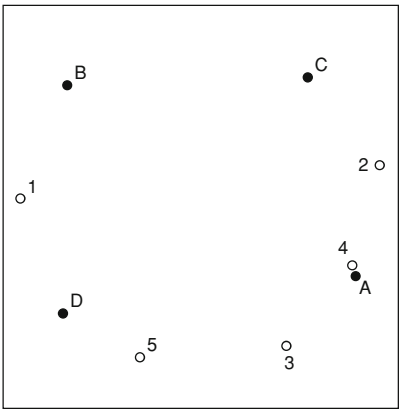


**Fig. 5.6** Ordinal unfolding solution for the preference data in Table 5.1

the ideal point of a person, the stronger this person's preference for this party. The distances between a person's ideal point and the various object points thus represent this person's perceived preference intensities. Hence, folding the unfolding plane of Fig. 5.6 in the ideal point of a person[5] leads to the person's preference scores (ratings, ranks, etc.) for the various parties. Folding the plane in another point generates other preference scores.

[4] Such data express the extent to which $i$ *dominates* $j$ in some sense. For example, dominance could mean "X is better than Y by x units", "I would vote for A rather than for B", or "I agree most with X".

[5] Just like folding an umbrella, or like picking up a handkerchief at this point.

As a psychological model of preference, the unfolding model rests on the strong assumption that all persons share the *same perception* of the objects. They may differ only in what they find ideal. Thus, all persons should agree where to place each party on a left-right continuum, but some persons are conservative, others are left-wingers. This assumption can be quite wrong, of course. Analyzing data from German voters, Borg and Staufenbiel (2007) showed that some voters located the German Liberals to the right of the Conservatives, while other voters swapped the order of these parties. If the two groups of voters are thrown into one single unfolding analysis, a hard-to-interpret 2-dimensional solution is needed to represent these data. If, however, the two groups are analyzed separately ("multiple unfolding"), unfolding leads to a 1-dimensional solution for each group, where the Liberals are to the left of the Conservatives in one data set, and to the right in the other. In other words, the multi-dimensional unfolding representation for the German voting preferences appears to be an aggregation artifact that does not properly represent the preferential space of any single person.

From a geometric point-of-view, unfolding can easily lead to unstable solutions. This is so because the model rests on data that constrain only a sub-set of the distances, namely the distances among ideal points and object points, but not the distances among ideal points and also not the distances among object points (see Table 5.1).

Moreover, in real data, object points and ideal points are often not thoroughly mixed. That is, many preference orders that are theoretically possible do not appear at all or only very infrequently, because most persons prefer or reject the same objects. This can lead to major indeterminacies of the unfolding solution, where single points can be moved around arbitrarily in ample solution regions without affecting the Stress at all (see Borg and Groenen 2005).

Still another problem is that the risk to obtain degenerate solutions can be considerable in both ordinal and interval unfolding. The solutions, then, show peculiar patterns where all distances between ideal points and objects are essentially equal. Such undesirable solutions are sometimes easily recognized, for example, if the person points are all located on a circular arc, while the object points are lumped together in the center of the circle. To avoid this problem, most MDS programs use a modified Stress function in unfolding, *Stress-2*, which slightly modifies the denominator of formula 3.2 to $\sum_{i<j}(d_{ij}(\mathbf{X}) - \overline{d})^2$. Although *Stress-2* seems to reduce the degeneracy problem, it does not make it impossible (see, e.g., Carroll 1980). However, a systematic approach that avoids degenerate solutions was proposed by Busing, Groenen and Heiser (2005) and implemented in the PREFSCAL module of SPSS. It penalizes the loss function whenever the MDS configuration tends to be modified in the optimization process into the direction of equal distances. We recommend this program in case of linear and ordinal unfolding.

In unfolding, one should also consider if one really wants to assume that the data are comparable across rows. In our small demo example, one may doubt that the preferential value "4" of person 1 is truly equal to the "4" of person 2. If one does not want to assume that equal data values have the same meaning, one should not request that the distance from ideal point 1 to point D must have the exact same value as the distance from ideal point 2 to point A. Rather, one may feel that only the preference

values *within* in each row can be compared among themselves. This would require a *row-conditional* unfolding analysis: Splitting the data matrix into horizontal stripes of the lower diagonal block in Table 5.1, allows the unfolding program to choose a different monotone mapping for each row, for example (in case of a linear unfolding model) $d_{1D} \approx a_1 + b_1 \cdot 4$ for person 1, and $d_{2A} \approx a_2 + b_2 \cdot 4$ for person 2, so that it is generally true that $d_{1D} \neq d_{2A}$. Such conditionalities may be theoretically desirable, but mathematically they lead to problems, because they further reduce what is already reduced in unfolding, i.e. the constraints that the data exert onto the distances of the MDS representation. For row-conditional unfolding, one should therefore have "plenty" of data (rule of thumb: at least 15 persons, all with different preference profiles).

## 5.7 Summary

MDS is a family of different models. They differ in the way they map proximities into distances, and in the distance functions they employ. The various mapping functions optimally preserve certain properties of the data such as the ranks of the data in ordinal MDS, the relative differences of any two data values in interval MDS, or the ratios of the data in ratio MDS. Typically, Euclidean distances are chosen as the targets in MDS. City-block distances or dominance metrics are also used in psychological modeling. Some MDS models allow using multiple proximities per distance. Asymmetric proximities can be handled by the drift-vector model: It represents their symmetric part by the distances of an MDS configuration, and their skew-symmetric part by drift vectors attached to the points. A popular MDS model is INDSCAL which represents a set of $N$ proximity matrices, one for each of $N$ individuals, by one common MDS space and by $N$ sets of weights for its dimensions. Another special MDS model is unfolding. It uses preference data, representing $N$ persons by $N$ ideal points in a joint space together with the points for the different choice objects.

## References

Borg, I, & Groenen, P. J. F. (2005). *Modern multidimensional scaling* (2nd ed.). New York: Springer.

Borg, I., & Lingoes, J. C. (1978). What weights should weights have in individual differences scaling? *Quality and Quantity, 12*, 223–237.

Borg, I., & Staufenbiel, T. (2007). *Theorien und Methoden der Skalierung* (4th ed.). Bern: Huber.

Busing, F. M. T. A., Groenen, P. J. F., & Heiser, W. J. (2005). Avoiding degeneracy in multidimensional unfolding by penalizing on the coefficient of variation. *Psychometrika, 70*, 71–98.

Carroll, J. D. (1980). Models and methods for multidimensional analysis of preferential choice or other dominance data. In E. D. Lantermann & H. Feger (Eds.), *Similarity and choice*. Bern: Huber.

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*, 283–320.

Horan, C. B. (1969). Multidimensional scaling: Combining observations when individuals have different perceptual structures. *Psychometrika, 34*, 139–165.

Kruskal, J. B., & Wish, M. (1978). *Multidimensional scaling*. Beverly Hills, CA: Sage.

Lingoes, J. C., & Borg, I. (1978). A direct approach to individual differences scaling using increasingly complex transformations. *Psychometrika, 43*, 491–519.

Schönemann, P. H. (1994). Measurement: The reasonable ineffectiveness of mathematics in the social sciences. In I. Borg & P. P. Mohler (Eds.), *Trends and perspectives in empirical social research* (pp. 149–160). Berlin: De Gruyter.

Schönemann, P. H., & Borg, I. (1983). Grundlagen der metrischen mehrdimensionalen Skaliermethoden. In H. Feger & J. Bredenkamp (Eds.), *Enzyklopädie der Psychologie: Messen und Testen* (pp. 257–345). Göttingen: Hogrefe.

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

Tversky, A. (1977). Features of similarity. *Psychological Review, 84*, 327–352.

Wish, M. (1971). Individual differences in perceptions and preferences among nations. In C. W. King & D. Tigert (Eds.), *Attitude research reaches new heights* (pp. 312–328). Chicago: American Marketing Association.

# Chapter 6
# Confirmatory MDS

**Abstract** Different forms of confirmatory MDS are introduced, from weak forms with external starting configurations, to enforcing theoretical constraints onto the MDS point coordinates or onto certain regions of the MDS space.

**Keywords** Confirmatory MDS · External scales · Dimensional constraints · Shearing · Axial partition · Penalty function

The MDS models discussed so far impose no particular restrictions onto the MDS configurations: The MDS computer programs are free to position the points anywhere in space as long as this reduces the cofigurations' Stress values. This type of *exploratory* MDS lets the data "speak for themselves". If one has specific hypotheses about the structure of the data in MDS space, however, one may be less interested in blindly minimizing Stress, but rather in finding an MDS solution that is not only Stress-optimal but also theory-consistent. In *confirmatory* MDS, additional constraints derived from substantive theory are imposed onto the solution. In case of a dimensional theory, for example, one may request that the points form a particular grid of points (as in Fig. 5.4) and then check how precisely such a solution would represent the given data.

In practice, one often assumes that if the solution of an exploratory MDS closely approximates the configuration that is expected for theoretical reasons, then a perfectly theory-consistent structure has a Stress value that is only "somewhat" higher. If, however, exploratory MDS does not lead to a solution that comes close to what one predicts for theoretical reasons, then it is impossible to tell if a theory-consistent solution with a reasonably low Stress value exists or not. Borg and Groenen (2005, p. 230) report an example for data similar to those used in Sect. 2.3 (this one using ellipses rather than rectangles) where the minimal-Stress MDS solution is radically different from a theory-compatible solution but both solutions have almost the same Stress. Whether this is true or not, must be tested by enforcing the theory onto the MDS solution, that is, by confirmatory MDS.

Confirmatory MDS comprises various approaches that allow the user to formulate *external* constraints that are imposed onto the dimensions of the MDS configuration;

I. Borg et al., *Applied Multidimensional Scaling*, SpringerBriefs in Statistics,
DOI: 10.1007/978-3-642-31848-1_6, © The Author(s) 2013

onto clusters of its points; onto certain partitions of the configuration; or onto the distribution of its points on geometric figures such as circles. Such additional constraints can be *strictly* enforced in some MDS procedures, while other programs only systematically push the iterations into directions that *approximate* such solutions.

## 6.1 Weak Confirmatory MDS

A weak confirmatory MDS approach is running the MDS with a user-defined external starting configuration that is set up on the basis of theoretical considerations rather than leaving it to the program to choose its own starting configuration. This makes it more likely to obtain an MDS solution that is similar to the theory based starting configuration. As an example, consider the case discussed in Sect. 2.3, where we had a clear hypothesis on how judgments on the similarity of different rectangles are generated, that is, by a city-block composition rule based on a log-transform of the design configuration in Fig. 2.4. If one leaves it to MDS to generate its own starting configuration, the program does so on formal grounds. It is blind to the content of the data, as it knows nothing about substantive theory. Hence, this information is not taken into account when setting up a starting configuration. The researcher, moreover, is often not interested to run an MDS analysis that blindly grinds out a minimal Stress solution but rather in finding out how well his/her theory (with an optimal scaling of all free parameters) explains the data. Hence, pushing the MDS solution into this direction from the start makes sense in terms of theory building, wether or not the theory is accepted in the end.

One can also fit given MDS solutions to theory-based *target* configurations. In the above case of rectangles, the design configuration of Fig. 2.4, appropriately stretched or compressed along its dimensions, can serve as a target in *Procrustean* transformations of an MDS configuration for the rectangle data in Table 2.3 (see Sect. 7.8 for Procrustean transformations). However, with its default starting configuration, an exploratory MDS is not pushed from the start into a theory-generated direction so that a combination of using a theory-derived starting configuration with subsequent Procrustean transformations promises to be more effective for theory testing and development.

Providing an external starting configuration can also help to make a set of different MDS solutions more similar. An application example is a study by Dichtl et al. (1980). These authors used a common space (in the sense of the INDSCAL model in Fig. 5.4) as the starting configuration in five different MDS analyses of data on consumer perceptions of various automobiles collected year after year for five years. Using a common starting configuration for each of the MDS scalings makes it more likely that the solutions are more comparable by reducing irrelevant differences such as rotations in space.

**Table 6.1** Coordinates of the stimuli (rectangles) in design of Fig. 2.4

| Rectangle | Width | Height |
|---|---|---|
| 1 | 3.00 | 0.50 |
| 2 | 3.00 | 1.25 |
| 3 | 3.00 | 2.00 |
| 4 | 3.00 | 2.75 |
| 5 | 4.25 | 0.50 |
| ⋮ | ⋮ | ⋮ |
| 15 | 6.75 | 2.00 |
| 16 | 6.75 | 2.75 |

## 6.2 External Side Constraints on the Point Coordinates

A *strict* confirmatory MDS approach *enforces* a solution that *fully* satisfies the external constraints. The Stress value is optimized, but its absolute value does not matter. Stress, therefore, may reach high values, which tells the user that the data are not compatible with the particular theoretical model.

As an application example, we use the rectangle study from Sect. 2.3. Exploratory MDS of the data of Table 2.3 leads to a solution that closely approximates the grid of the design configuration (Fig. 2.5). We now employ confirmatory MDS to enforce such a grid onto the solution and then check if this leads to a Stress value that is still acceptably low. For this scaling, we use PROXSCAL, an MDS program in SPSS. PROXSCAL allows the user to request that an MDS solution $\mathbf{X}$ is generated by optimally scaling the column vectors in $\mathbf{Y}$. In our example, the columns of $\mathbf{Y}$ are the coordinates of the points in the design grid, i.e. the columns "Width" and "Height" in Table 6.1.

To run PROXSCAL, we first store the external scales in the file "RectangleDesign.sav" and the proximities of Table 2.3 in "RectanglesData.sav". We then request for the MDS solution that the dimensions of $\mathbf{X}$ must be generated from the columns of $\mathbf{Y}$, allowing ordinal re-scalings of the coordinate values that preserve ties. The commands for this MDS job in PROXSCAL are:

```
GET FILE='RectangleData.sav'.
PROXSCAL VARIABLES=Rectangle1 to Rechtangle16
   /TRANSFORMATION=ORDINAL
   /RESTRICTIONS=VARIABLES ('RectangleDesign.sav')
    Width(ORDINAL(KEEPTIES))
      Height(ORDINAL(KEEPTIES)).
```

With these commands, PROXSCAL yields the solution in Fig. 6.1. It is almost perfectly theory-compatible except for the slight *shearing* of the point grid which cannot be suppressed in the present version of PROXSCAL.[1] The increment in Stress

[1] The unavoidable shearing can become extreme with other data. It can make the solutions essentially worthless. Presently, you can only cross your fingers and hope that shearings do not become so strong.

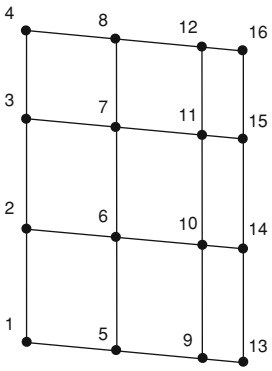


**Fig. 6.1** Ordinal MDS solution based on ordinally re-scaled coordinates of rectangles

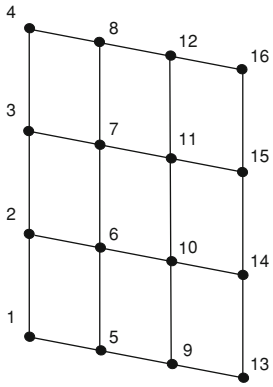


**Fig. 6.2** Ordinal MDS solution based on linearly re-scaled coordinates of rectangles

compared to the Stress of an exploratory MDS[2] is very small (0.102, compared to 0.092). One can conclude, therefore, that the jitter of the grid in Fig. 2.5 does not help to explain the data markedly better. Rather, it seems that it essentially represents some of the error contained in the data. Hence, there is no reason to reject the theory that the dissimilarity judgments for the rectangles are generated by a composition rule that can be modeled by a (Euclidean or city-block) distance formula operating on the rectangles' width and height coordinates.

To see what happens if the external scales are treated as interval scales, we now set `Width (INTERVAL) and Height (INTERVAL)` in the PROXSCAL commands. This entails that the program is restricted to homogeneous stretchings or compressions of the external scales or, expressed differently, to simple dimensional weightings (plus shearings, as before). Under this condition the Stress grows only slightly to 0.120, but the successively smaller compressions of the grid along its dimensions under the ordinal constraints (Fig. 6.1) is more desirable in terms of psychophysical theory (i.e., the Weber–Fechner law) than the even spacings in the interval case (Fig. 6.2).

Enforcing dimensional structures onto an MDS solution can be described by the equation $\mathbf{X} = \mathbf{YC}$, with $\mathbf{X}$ the MDS solution, $\mathbf{Y}$ the external scales, and $\mathbf{C}$ a matrix of parameters to be chosen such that the distances of $\mathbf{X}$ minimize the MDS model's Stress. The matrix $\mathbf{C}$ represents a *linear transformation* of the configuration $\mathbf{Y}$. Any linear transformation can be decomposed into rotations and dimensional strechings/compressions of $\mathbf{Y}$. Algebraically, this means that $\mathbf{C}$ can be "diagonalized" by singular value decomposition into the product $\mathbf{PMQ}$, where $\mathbf{P}$ and $\mathbf{Q}$ represent

[2] The MDS solution in Fig. 2.5 was generated by SYSTAT using city-block distances which PROXSCAL does not offer. If Euclidean distances are used in the exploratory MDS, however, the solution is almost identical to the one shown in Fig. 2.5, with almost the same Stress value of 0.090.

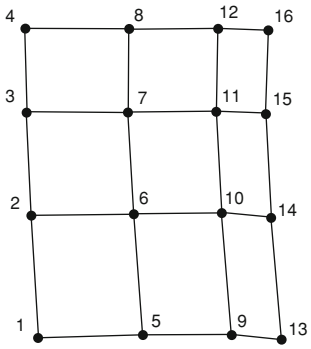


**Fig. 6.3** Ordinal MDS solution with external constraints on the distances that enforce an orthogonal grid

rotations and **M** a diagonal matrix of dimension weights. Thus, **C** first rotates the configuration **Y** in some way, then stretches and/or compresses this *rotated* configuration along its dimensions, and finally rotates the result once more. If **C** is a diagonal matrix, then the column vectors of **Y** are weighted directly. If **C** is not diagonal, then **Y** is first rotated and then dimensionally weighted, and this is what causes the shearing in Figs. 6.2 and 6.1, respectively.[3] Thus, to avoid the shearing, **C** must be diagonal.

The possibility to constrain **C** to be diagonal is offered only by the SMACOF program (described in detail in Sect. 9.2). However, when enforcing this constraint, SMACOF does not allow the user to also set the scale level of the external scales to "ordinal". So, a solution with orthogonal dimensions that are optimally scaled in an ordinal sense cannot be generated by the present version of SMACOF either.

A completely different approach to impose external constraints onto the MDS solution is to focus on the distances of the MDS configuration, not on its coordinates. If, for example, one requests for the rectangle data that $d(1, 6) = d(2, 5)$, $d(6, 11) = d(7, 10)$ and $d(11, 16) = d(12, 15)$ must hold in the MDS solution, shearings of the point grid are avoided. To guarantee that a grid is generated in the first place, one can additionally enforce that some of the horizontal grid distances be equally long, i.e. that $d(1, 5) = d(2, 6) = d(3, 7) = d(4, 8)$, $d(5, 9) = d(6, 10) = d(7, 11) = d(8, 12)$, and $d(9, 13) = d(10, 14) = d(11, 15) = d(12, 16)$. Similarly, for the vertical distances it should hold that $d(1, 2) = d(5, 6) = d(9, 10) = d(13, 14)$, $d(2, 3) = d(6, 7) = d(10, 11) = d(14, 15)$, and $d(3, 4) = d(7, 8) = d(11, 12) = d(15, 16)$. Combined this amounts to nine sets of restrictions that can be imposed on the MDS configuration by the program CMDA (Borg and Lingoes 1980).[4]

The way CMDA proceeds is as follows. First, the various restrictions are formulated in terms of *pseudo data*, $p'(i, j)$, that express the constraints numerically. One defines, for example, for $p'(1, 6)$ and for $p'(2, 5)$ the same pseudo-data value (e.g.,

[3] To see this graphically, first rotate any of the configurations in Fig. 5.4 by 30°, say, and then stretch or compress it along the $X$- and the $Y$-axis.

[4] CMDA is, unfortunately, an old MS-DOS program that is not easy to use. A more user-friendly version that allows one to impose external constraints onto the MDS distances does not exist.

3 or 7); similarly, one sets $p'(6, 11) = p'(7, 10) = 5$, for example; etc., for each of the nine sets of restrictions. Then, MDS is set to use the *secondary* approach to ties ("keep ties") but only *conditionally* within each set so that the equality of pseudo data is to be preserved within each set only.[5] Given some starting configuration, one can then minimize the configuration's Stress not just relative to the proximities, as usual, but also relative to the pseudo data. CMDA begins its iterations by focussing on the proximities only, and then successively shifts more and more weight onto the pseudo data when computing the combined total Stress. In the end, only the pseudo data count. This means that an initially exploratory MDS is penalized more and more for deviations from the external constraints.[6]

The solution that CMDA yields after 100 iterations is shown in Fig. 6.3. It obviously closely approximates the desired orthogonal grid. Its Stress value is 0.108, which is only minimally larger than the Stress (0.102) obtained for the sheared solution in Fig. 6.1. The shear, therefore, explains almost no additional variance and, thus, can be discarded as substantively meaningless.

## 6.3 Regional Axial Restrictions

One can use the methods discussed above to solve confirmatory MDS problems that arise quite frequently in applied research, that is, impose particular *axial partitions* onto the MDS solution. Here is an example. Rothkopf (1957) studied to what extent test persons confused different acoustic Morse signals. He used 36 different signals, the 26 letters of the alphabet, and the natural numbers from 0 to 9. The signal for A, for example, is "di" (a beep with a duration of 0.05 s), followed by a pause (0.05 s) and then by "da" (0.15 s). We code this as 1–2 or 12 for di-da.

The symmetrized confusion probabilities collected for these signals from hundreds of test persons can be represented quite well in a 2-dimensional MDS configuration (Fig. 6.4). The partitioning lines were inserted by hand. They cut the plane in two ways, related to two facets: The nine solid lines discriminate the signals into classes of signals with the same total duration (from 0.05 to 0.95 s); the five dashed lines separate the signals on the basis of their composition (e.g., signals containing only long beeps are all on the right-hand side). The pattern of these partitioning lines is not very simple, though, but partially rather curvy and hard to describe. Particulary the dashed lines are so twisted that the pattern of the emerging regions does not exhibit a simple law of formation. Rather, the partitioning seems over-fitted. The substantive researcher, therefore, would probably not bet that it can be completely replicated with new data.

---

[5] Note that CMDA does not only allow the user to impose equality constraints, as in this example. Order constraints are also possible, for example requesting d(1,5) > d(5,9) > d(9,13). Moreover, CMDA can handle equality constraints both under the primary and also under the secondary approach to ties.

[6] PROXSCAL, in contrast, begins from the start with a configuration that perfectly satisfies the external restrictions, and then successively optimizes its fit to the proximities.

**Table 6.2** Two external scales for regional restrictions on an MDS representation of Morse signals (duration and type), together with ordinally equivalent internal scales

| | $\mathbf{r}_1$ | $\mathbf{r}_2$ | $\mathbf{z}_1$ | $\mathbf{z}_2$ | | $\mathbf{X}$ | |
|---|---|---|---|---|---|---|---|
| Signal | Duration | Type | External scale 1 | External scale 2 | | Internal scale 1 | Internal scale 2 |
| 1 | 0.05 | 1 | 1 | 1 | | 1.0 | 2.1 |
| 11 | 0.15 | 1 | 2 | 1 | | 2.2 | 2.2 |
| 2 | 0.15 | 2 | 2 | 5 | | 1.9 | 4.7 |
| 21 | 0.25 | $1 = 2$ | 3 | 3 | $\rightarrow$ | 2.5 | 2.9 |
| 12 | 0.25 | $1 = 2$ | 3 | 3 | | 3.0 | 3.3 |
| 111 | 0.25 | 1 | 3 | 1 | | 3.3 | 2.0 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | | ⋮ | ⋮ |
| 22222 | 0.95 | 2 | 9 | 5 | | 27 | 4.8 |

We now want to straighten the two sets of partitioning lines. For that purpose we again use the $\mathbf{X} = \mathbf{YC}$ restriction. To generate the scales in $\mathbf{Y}$, we use the signals' codes $\mathbf{r}_1$ and $\mathbf{r}_2$, as shown in Table 6.2. For the external scale 1, $\mathbf{r}_1$, we simply use the lengths of the signals, shown in Fig. 6.4 by the black boxes. If we allow for an ordinal re-scaling of these values in $\mathbf{y}_1$, then we might also choose other values instead of the duration values, for example the values shown in $\mathbf{z}_1$.

For the second scale, we use $\mathbf{r}_2$ (or, equivalently, $\mathbf{z}_2$). Column $\mathbf{r}_2$ assigns values to each signal that correspond to the region shown on the upper boundary of Fig. 6.4, numbered from 1 to 5 for the boxes labeled as "1", "$1 > 2$", "$1 = 2$", "$2 > 1$", and "2" ($\mathbf{r}_2$), respectively.

We also specify that we want to use the primary approach to ties so that ties in the external scale values can be broken in the signals' internal scale values. That is, signals with the same codes on the internal scale must fall into *stripes* or *bands* (not lines) that are *ordered* as the external scale values. This is illustrated in Table 6.2. Here, the "2" of the external scale is mapped into the values 2.2 and 1.9, respectively. These coordinate values of $\mathbf{X}$ are different, but they are both greater than the coordinate values that correspond to "1" on the external scale, and smaller than all coordinate values that correspond to "3" on the external scale ($= 2.5, 3, 3.3$).

With these constraints, MDS[7] delivers the solution in Fig. 6.5. The simple-to-interpret MDS solution has almost the same overall Stress as the exploratory MDS solution in Fig. 6.4 (0.21 vs. 0.18). Upon closer investigation one notes, however, that the confirmatory solution moved only few points by more than a small amount. Particularly point 1 (at the bottom, to the right) was moved a lot so that the substantive researcher may want to study this signal (and its relationship to other stimuli such as signal 2) more closely. Overall, though, the simpler and, probably, also more replicable solution in Fig. 6.5 appears to be the better springboard for further research.

[7] The present version of PROXSCAL (in SPSS 20) does *not* properly handle ordinal re-scalings of external scales if the primary approach to ties is chosen. The solution in Fig. 6.5 was generated by an experimental MDS program written by Patrick Groenen in MATLAB.

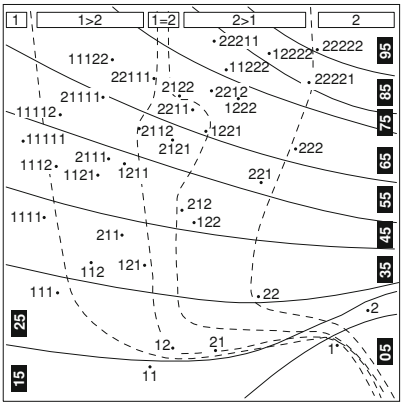


**Fig. 6.4** Exploratory MDS representation for 36 Morse signals; *lines* correspond to two typologies for the signals

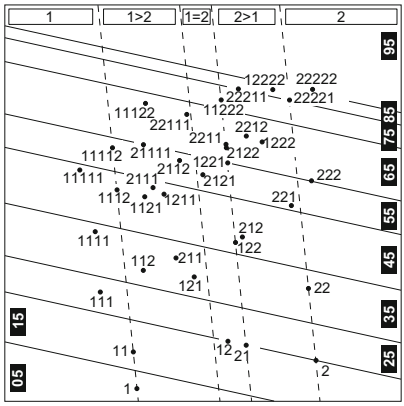


**Fig. 6.5** Confirmatory MDS solution with two perfect axial partitioning lines

## 6.4 Challenges of Confirmatory MDS

The challenges of confirmatory MDS for applied researchers consist of first formulating their theoretical expectations so that they can be expressed mathematically in, say, a penalty function, a pseudo-data matrix, or a system of equations that can be solved by an existing confirmatory MDS program. Confirmatory MDS, therefore, is often much harder than exploratory MDS, because it requires the user to not only develop explicit theories but also translate them into a proper computational language. So far, the MDS programs accessible to the general user can handle only relatively simple confirmatory analyses. As we showed above, dimensional restrictions are relatively easy to test, while confirmatory MDS analyses with regional restrictions are typically difficult to set up and solve (Groenen and Van der Lans 2004). Computer programs that allow all forms of restrictions (combined, in addition, with particular MDS models, certain missing data patterns, or distances other than Euclidean distances) do not exist yet. Rather, in such cases, a suitable MDS algorithm must be programmed ad hoc.

If the users succeed to generate a confirmatory MDS solution, a number of additional challenges await them. They have to evaluate not only the absolute Stress values as described in Chap. 3, but also the Stress increment resulting from adding the particular external constraints to the MDS analysis. Typically, such evaluations amount to deciding whether the Stress increment is "substantial" or not, given the number of points, the dimensionality of the MDS space, the MDS model, the distance function, and the quality of the data (error level). These and further criteria are summarized by Lingoes and Borg (1983) in a quasi-statistical decision procedure.

An important additional criterion is the strength of the external constraints. These constraints may be easy to satisfy for a given number of points in a given

dimensionality, but they may also be quite demanding. An approach for assessing this question is described in Borg et al. (2011). They use data from an experiment where a sample of employees assess 54 organizational culture themes (such as 'being competitive', 'working long hours', or 'being careful') in terms of how important they are for them personally. The correlations of these importance ratings are represented in a theory-compatible MDS solution, where the 54 points are forced into the quadrants of a 2-dimensional coordinate system on the basis of apriori codings of the items derived from the Theory of Universals in Values (TUV). The strength of the external constraints is assessed by studying the Stress values that result from running 1,000 different confirmatory MDS analyses, each one using a random permutation of these codings. It is found that the theory-based assignment of the codes to the 54 items does indeed lead to a Stress value that is smaller than any of the Stress values that result if random permutations of the codings are enforced onto the MDS solution. Hence, the codings are *not trivial* in the sense that random assignments of the codings would lead to equally good MDS solutions when enforced onto the configuration.

At the end of the day, however, assessing confirmatory MDS solutions is more than a mere statistical issue. Rather, it must be embedded into a process of cumulative theory construction where formal models are constructed and modified over time in partnership with substantive theory and empirical observations.

## 6.5 Summary

MDS is mostly used in an exploratory way where the points are positioned in space such that the resulting distances optimally represent the data. Confirmatory MDS enforces additional structure onto the MDS configuration, or it at least tries to push the solution into the direction of a theoretically expected structure. Confirmatory MDS solutions may be quite different from solutions that are Stress-optimal in the exploratory sense, but they can have Stress values that are not much worse than exploratory solutions. However, their Stress values may also be much higher or unacceptably high so that the theory must be rejected. Without confirmatory MDS, such questions cannot be answered. One way to push an MDS solution towards a theoretical structure is using a theory-derived starting configuration. To enforce a theory-compatible outcome, one must use special MDS methods. In case of dimensional expectations, theoretical structures can be strictly enforced in PROXSCAL or in SMACOF by optimally re-scaling a theoretical coordinate system, and in CMDA by penalizing various sub-sets of distances if they deviate from predicted patterns. Axial regionalities can also be enforced onto an MDS space, but more general patterns (such as radexes) are difficult to specify.

## References

Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling* (2nd ed.). New York: Springer.

Borg, I., Groenen, P. J. F., Jehn, K. A., Bilsky, W., & Schwartz, S. H. (2011). Embedding the organizational culture profile into Schwartz's Theory of Universals in Values. *Journal of Personnel Psychology, 10*, 1–12.

Borg, I., & Lingoes, J. C. (1980). A model and algorithm for multidimensional scaling with external constraints on the distances. *Psychometrika, 45*, 25–38.

Dichtl, E., Bauer, H. H., & Schobert, R. (1980). Die Dynamisierung mehrdimensionaler Marktmodelle am Beispiel des deutschen Automobilmarkts. *Marketing, 3*, 163–177.

Groenen, P. J. F., & Van der Lans, I. (2004). Multidimensional scaling with regional restrictions for facet theory: An application to Levy's political protest data. In M. Braun & P. P. Mohler (Eds.), *Beyond the horizon of measurement* (pp. 41–64). Mannheim: ZUMA.

Lingoes, J. C., & Borg, I. (1983). A quasi-statistical model for choosing between alternative configurations derived from ordinally constrained data. *British Journal of Mathematical and Statistical Psychology, 36*, 36–53.

Rothkopf, E. Z. (1957). A measure of stimulus similarity and errors in some paired-associate learning. *Journal of Experimental Psychology, 53*, 94–101.

# Chapter 7
# Typical Mistakes in MDS

**Abstract** Various mistakes that users tend to make when using MDS are discussed, from conceptual fuzziness, over using MDS for the wrong type of data, or using MDS programs with suboptimal specifications, to misinterpreting MDS solutions.

**Keywords** Global optimum · Local optimum · Termination criterion · Starting configuration · Degenerate solution · Dimensional interpretation · Regional interpretation · Procrustean transformation

We now discuss some of the most frequent mistakes that (new but also experienced) users of MDS tend to make in practice. These mistakes can easily lead to suboptimal MDS solutions, and to wrong or at least naive interpretations of MDS solutions.

## 7.1 Using the Term MDS Too Generally

The term multidimensional scaling is normally reserved for the models discussed in this book. However, some people denote any technique that gives a visual representation in low-dimensional space (such as principal components analysis or correspondence analysis) as a "multidimensional scaling" procedure. Yet, each of these techniques differs in what it exactly shows in the visualization and how its plots should be interpreted. Using the term MDS in a very general way can therefore blur these differences, leading to confusion and misinterpretations. We highly recommend to reserve the term multidimensional scaling for models that display proximities among objects of interest by distances between points in a low-dimensional space and not use the term for anything else.

I. Borg et al., *Applied Multidimensional Scaling*, SpringerBriefs in Statistics,
DOI: 10.1007/978-3-642-31848-1_7, 

## 7.2 Using the Distance Notion Too Loosely

The notion of distance is often used quite loosely. For example, numerous publications and manuals are calling dissimilarity data "distances" or "distance-like" measures. This can lead to confusion about the purpose and the processes of MDS. What MDS always does is *representing* proximities, as precisely as possible, as distances. Distances, therefore, always exist *on the model side* of MDS, but rarely ever on its data side. The distances on the model side, moreover, are special distances, i.e. Minkowski distances or, actually, one particular kind of Minkowski distances (mostly city-block distances or Euclidean distances).

Proximities are distances if, and only if, they satisfy the distance axioms presented on p. 14. In most applications, however, not all of the axioms can always be tested. One reason is that typically one does not have all the data that are needed for such tests. For example, one rarely collects data both on the similarity of $i$ with $j$, and also on the similarity of $j$ with $i$. So, symmetry cannot be checked, and simply assuming that $p_{ij}$ would be equal to $p_{ji}$ if both were collected can be quite wrong in many contexts (see, e.g., Tversky 1977). Another reason is that the scale level of the data may be too weak to test the distance axioms. With proximities on an interval scale, for example, the triangle inequality can always be satisfied by an admissible transformation, i.e. by adding an additive constant to all values that is large enough to reverse the inequality that is most violated. Hence, in many applications, the given proximities can be *converted* into values that do *not violate* the testable distance axioms. However, that does not make them distances, let alone Minkowski distances. Indeed, that the data can be admissibly transformed into (a particular variety of) Minkowski distances in an $m$-dimensional space is a hypothesis that is tested by MDS.

## 7.3 Assigning the Wrong Polarity to Proximities

A frequent beginner's mistake is to scale proximities with the wrong polarity. This means that the data are similarities, but that MDS treats them as if they were dissimilarities, or vice versa. MDS then generates an uninterpretable solution with very high Stress. If the proximities are read as data into an MDS program rather than being computed within the program or its surrounding statistics package (e.g., as correlations of variables), then the MDS program cannot know how to interpret these indices and, therefore, works with its default interpretation of the polarity of the data (usually: dissimilarities). Yet, correlations, for example, are similarities, because greater correlation coefficients indicate higher similarity and thus should be represented by small distances. If the user incorrectly specifies the data type, or if the program works with an incorrect default definition, then MDS cannot generate meaningful solutions.

## 7.4 Using Too Few Iterations

Almost all MDS programs have suboptimal default specifications. In particular, they typically terminate the iterations of their optimization algorithms too early, that is, before the process has actually converged at a local minimum. This premature termination is caused by setting the termination criteria too defensively. Many programs set the maximum number of iterations to 100 or less, a specification that dates back to the times when computing was slow and expensive. For example, the GUI box of SYSTAT in Fig. 1.5 shows that, per default, this MDS program allows at most 50 iterations. The iterations are also stopped if the Stress does not go down by more than 0.005 per iteration. However, one can show that very small Stress reductions do not always mean that all points remain essentially fixed in further iterations. We therefore recommend to always *clearly* change these default values to allow the program to work longer. Instead of a maximum of 50 one can easily require 500 or even 1,000 iterations. The convergence criterion, in turn, could be set to 0.000001, that is, to a very small value indeed. The only disadvantage of such extreme specifications is that the program may run a few seconds longer.

## 7.5 Using the Wrong Starting Configuration

All MDS programs automatically generate a ("rational") starting configuration if the user does not import an external starting configuration into the program. It is a common fallacy to assume that internally generated starting configurations will always lead to optimal MDS solutions. For example, we have found in many tests that the default starting configuration used in PROXSCAL (called "SIMPLEX") is often not optimal. We recommend to use the option INITIAL=TORGERSON (see Fig. 9.8) instead. Yet, *no* starting configuration—rational or user-provided—always guarantees the best-possible final solution, and so the user should try out some sensible alternatives before accepting a particular MDS solution all too early as the final solution.

Random starting configurations can also be used in MDS. Indeed, *many* random configurations can be used without much effort. For example, for the solution in Fig. 1.8 we used PROXSCAL with the option RANDOM=1,000, i.e. we asked the program to repeat the scaling with 1,000 different random starting configurations and then report the solution with the lowest Stress value. That only took seconds with this small data set.

## 7.6 Doing Nothing to Avoid Suboptimal Local Minima

An MDS solution is almost always found through series of small movements of the points so that the Stress value goes down. The algorithms used today for computing such iterations are guaranteed to find a *local minimum* solution, that is, a configuration

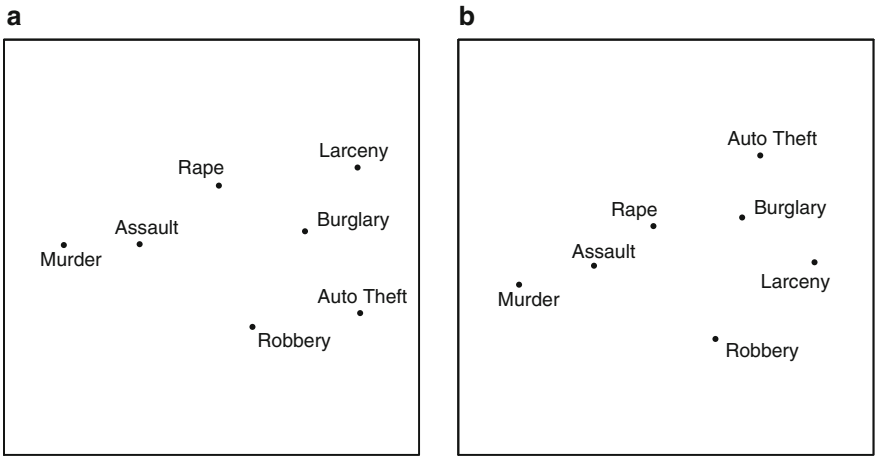


**Fig. 7.1** Two local minima solutions of ordinal MDS of the data in Table 1.1; panel **a** exhibits the global minimum with Stress = 0; panel **b** is a local minimum with Stress = 0.089

where any small movements of the points will lead to higher Stress values. The problem is that the iterations have to begin with some starting configuration, and depending on this configuration they may end up in different local minima (if they exist).

We illustrate this problem using the data of Table 1.1. The ordinal MDS solution in Fig. 7.1 has a Stress value of 0. Now, we take this configuration, swap the positions of Auto Theft and Larceny, and then use this configuration as the starting configuration for another ordinal MDS. This MDS ends up in a different local minimum that has Stress=0.089 (Fig. 7.1b). Thus, depending on the starting configuration, the MDS program may report different MDS solutions.

MDS always attempts to find the local minimum with the smallest possible Stress, i.e. the *global minimum*. MDS users can do their share to help find this global minimum by keeping an eye on the following issues:

- A good starting configuration is the best way to avoid suboptimal local minima. If you have a theory, then a user-defined configuration (as, e.g., described in Sect. 6.1) is what you should always use. If you do not have a theory, you must leave it to the MDS program to define its own starting configuration. In that case, we recommend using the solution of classical MDS (also known as the *Torgerson solution*) as a start.
- Another precaution against suboptimal local minima is using multiple random starts. As modern MDS programs are extremely fast, one can easily require the program to repeat the scaling with a very large number of different random starts (e.g., with 1,000 or more).
- City-block distances increase the risk to end up in suboptimal local minima. General MDS programs are particularly sensitive in this regard. There exist MDS

programs that are optimized for city-block distances, but they are hard to obtain and typically require expert support for using them.

- The greater the dimensionality of the MDS space, the smaller the risk for suboptimal local minima. Even if you want, say, a 2-dimensional MDS solution, using the space spanned by the first two principal components of the 3-dimensional MDS solution may, therefore, serve as a good starting configuration.
- Suboptimal local minima are particularly likely in case of 1-dimensional MDS. Standard programs almost never find the global minimum. If you must do 1-dimensional MDS, you want to consider using an MDS program that is specialized for this case. Again, such programs are not easily accessible and may be difficult to use.

## 7.7 Not Recognizing Degeneracy in Ordinal MDS

Of all MDS models, ordinal MDS is the model that is used most often. It requires data that are only on an ordinal scale level, but it nevertheless produces stable metric solutions. The reasons for this apparent paradox is that the function (5.2) defines an order relation for each *pair* of distances, and this quickly leads to a huge number of restrictions on the configuration. With just $n = 12$ objects (as in the country similarity example in Fig. 2.2), there are 12 points and $n(n-1)/2 = 12 \cdot 11/2 = 66 = k$ distances. Hence, there are $k(k-1)/2 = 66 \cdot 65/2 = 2.145$ order relations that ordinal MDS must bring in agreement with the data. With $n = 20$ objects, we arrive at 17.955 restrictions, with $n = 50$ at 749.700!

Inspite of so many restrictions, ordinal MDS can run into a special problem that the user should keep an eye on, i.e. it can lead to a *degenerate* solution. We illustrate this problem with the following example. Table 7.1 exhibits the intercorrelations of eight test items of the Kennedy Institute Phonics Test (KIPT), a test for reading skills (Guthrie 1973). If we scale these data by ordinal MDS in the plane, we obtain the configuration shown in Fig. 7.2. Its Stress value is almost zero, so this MDS solution seems practically perfect. Yet, the Shepard diagram in Fig. 7.3 reveals a peculiar relation of data and distances. Although the data scatter evenly over the interval from 0.44 to 0.94, they are not represented by distances with a similar distribution, but rather by two clearly distinct classes of distances so that the regression line has essentially just one step.

One notes in the MDS configuration that the program positioned the points in three clusters that have the same distance from each other. This configuration represents all large correlations ($r \geq 0.78$) by similarly small distances, and all large correlations ($r < 0.72$) by essentially the same large distance. This solution correctly displays one of the data relations,[1] but it takes advantage of the freedom to position the points in ordinal MDS to an extent that is most likely not intended by the user, because one

[1] The correlations of test items from two of the subgroups {NP,...,NR}, {SLP} and {CCR, ILR}, respectively, are always smaller than the correlations of test items from the same group.

**Table 7.1** Correlations (lower half) of some test items of the KIPT and their ranks (upper half)

| | NP | LVP | SVP | CCP | NR | SLP | CCR | ILR |
|---|---|---|---|---|---|---|---|---|
| Nonsense word production (NP) | – | 9 | 4 | 1 | 6 | 19 | 10 | 12 |
| Long vowel production (LVP) | 0.78 | – | 1 | 7 | 5 | 21 | 20 | 22 |
| Short vowel production (SVP) | 0.87 | 0.94 | – | 3 | 2 | 17 | 16 | 23 |
| Consonant cluster production (CCP) | 0.94 | 0.83 | 0.90 | – | 7 | 14 | 11 | 16 |
| Nonsense word recognition (NR) | 0.84 | 0.85 | 0.91 | 0.83 | – | 17 | 15 | 18 |
| Single letter production (SLP) | 0.53 | 0.47 | 0.56 | 0.60 | 0.56 | – | 13 | 16 |
| Consonant cluster recognition(CCR) | 0.72 | 0.48 | 0.57 | 0.69 | 0.59 | 0.62 | – | 8 |
| Initial letter recognition (ILR) | 0.66 | 0.45 | 0.44 | 0.57 | 0.55 | 0.57 | 0.82 | – |

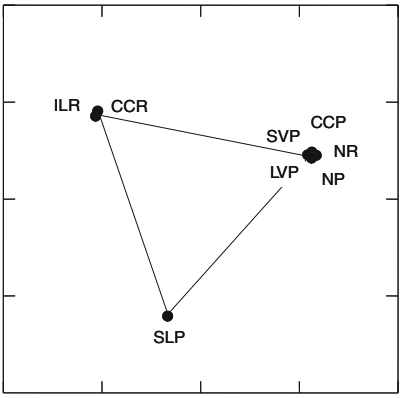


**Fig. 7.2** Ordinal MDS configuration for data of Table 7.1

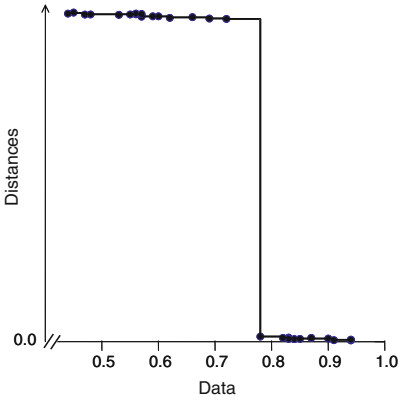


**Fig. 7.3** Shepard diagram for MDS solution in Fig. 7.2

would probably not want to assume that the differences in the size of the correlations have no real meaning at all. In other words, one would prefer a mapping of data into distances that is not so discrete but somewhat more smooth and continuous.

The excellent Stress value of the solution in Fig. 7.2 is, moreover, deceptive. The large and the small distances, respectively, *can be re-ordered arbitrarily* as long as all within-block similarities remain greater than all between-block similarities. Any such reordering will have almost no effect on the Stress value. Indeed, one can drive the Stress *infinitely close to zero* by collapsing the three point clusters more and more into three single points so that one ends up with a perfect equilateral triangle.

The reason for the degenerate solution is that the data have a peculiar structure. They form three subgroups, with high within- and low between-correlations. From the point-of-view of ordinal MDS, such data can always be perfectly represented by

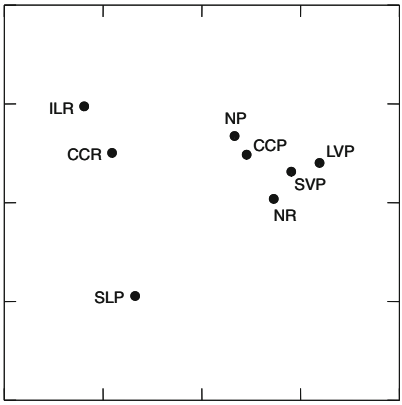


**Fig. 7.4** Interval MDS configuration for data of Table 7.1

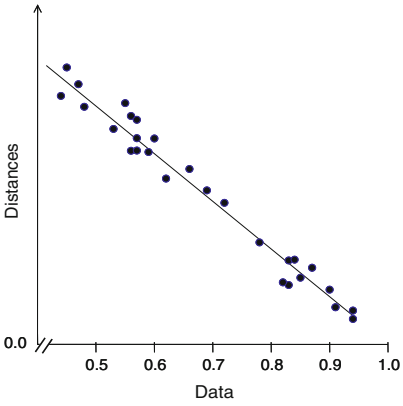


**Fig. 7.5** Shepard diagram for MDS solution in Fig. 7.4

an equilateral triangle. Of course, the data structure here is a contrived case, selected to demonstrate the degeneracy issue. In practice, one should rarely find such cases, but they become more likely if the number of variables is small ($n \leq 8$).

If the Shepard diagram suggests that the MDS solution is degenerate, then the easiest next step for the user is testing a stronger MDS model and compare the solutions. Using interval MDS with the above data yields the solution in Fig. 7.4. It too shows the three clusters of test items, but it does not contract them as much as the ordinal MDS solution. Its Shepard diagram (Fig. 7.5) makes clear that the interval solution preserves a linear relationship of the data in Table 7.1 to the distances in Fig. 7.4.

One may not want to enforce linearity onto this mapping but only *smoothness*. For that case, some MDS programs such as PROXSCAL offer *spline* functions, that is, smooth "elastic" functions (i.e., piecewise polynomial functions of degree $k$) that are set to run through a number of evenly spread out "knots" (Borg and Groenen 2005). The user of MDS does not have to understand the mathematics of this method: it is simple to use and one can always check the results graphically, particularly the resulting regression function in the Shepard diagram.

## 7.8 Meaningless Comparisons of Different MDS Solutions

A frequent issue in MDS applications is comparing two or more MDS solutions. Consider a simple case. Figure 7.6 (left panel) exhibits an MDS representation of correlations among 13 items on work values, collected for a representative West German sample in 1991 (Table 7.2, lower half). The items ask the respondents to rate different themes in their work life (such as 'high income' or 'good chances for

**Table 7.2** Intercorrelations of 13 work values (with ERG codings) assessed in the ALLBUS 1991 (lower/upper half: West-/East-Germany; decimal points omitted)

| Work value | | ERG | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Interesting work | G | | 47 | 43 | 38 | 28 | 37 | 29 | 28 | 27 | 16 | 15 | 21 | 28 |
| 2 | Independent work | G | 51 | | 53 | 31 | 27 | 34 | 23 | 25 | 28 | 25 | 16 | 15 | 26 |
| 3 | Work with responsibility | G | 42 | 57 | | 39 | 32 | 42 | 38 | 38 | 41 | 24 | 16 | 09 | 25 |
| 4 | Meaningful and sensible work | R | 37 | 30 | 33 | | 20 | 33 | 38 | 44 | 29 | 24 | 13 | 08 | 33 |
| 5 | Good chances for advancement | E | 28 | 29 | 33 | 18 | | 43 | 19 | 25 | 15 | 39 | 52 | 27 | 34 |
| 6 | Recognized and respected work | E | 18 | 23 | 34 | 24 | 43 | | 37 | 39 | 29 | 37 | 29 | 21 | 35 |
| 7 | Work helping others | R | 20 | 19 | 31 | 33 | 17 | 32 | | 48 | 49 | 16 | 10 | 14 | 26 |
| 8 | Work useful for society | R | 20 | 17 | 28 | 40 | 18 | 37 | 56 | | 32 | 23 | 16 | 18 | 30 |
| 9 | Contact with other people | R | 31 | 34 | 39 | 31 | 21 | 24 | 43 | 34 | | 16 | 11 | 10 | 19 |
| 10 | Secure position | E | 14 | 17 | 18 | 19 | 39 | 37 | 24 | 25 | 17 | | 40 | 18 | 38 |
| 11 | High income | E | 20 | 26 | 25 | 05 | 54 | 32 | 05 | 08 | 11 | 32 | | 27 | 29 |
| 12 | Work with much spare time | E | 25 | 22 | 13 | 09 | 19 | 30 | 13 | 18 | 19 | 16 | 30 | | 25 |
| 13 | Healthy working conditions | E | 32 | 31 | 23 | 37 | 25 | 20 | 25 | 23 | 24 | 33 | 16 | 23 | |

advancement') on a scale from "not important" to "very important" to them personally. The MDS configuration of the correlations shows three types of neighborhoods that can be predicted from the ERG theory by Alderfer (1972). (For the meaning of E, R, and G, see Sect. 7.10.)

Borg and Braun (1996) wanted to know how West Germans differ from East Germans in their work values, in particular in terms of how they structure work values. They first scaled the lower half of Table 7.2 via MDS and obtained Fig. 7.6. One can repeat this for the East German data, and then compare the two MDS configurations.

When comparing two MDS configurations, one must pay attention to discard *meaningless* differences. Such differences are those that can be *eliminated* by *admissible transformations*, that is, by *rigid transformations* (rotations, reflections, and translations) and by global enlargements or shrinkages of the MDS configurations, together called *similarity transformations*. Similarity transformations preserve the geometric "shape" of the configuration: a figure such as a star or a triangle remains a star or a triangle, for example, but it may be larger, smaller, rotated, reflected, or shifted in space. Similarity transformations do not change the ratio of the distances in a configuration and, therefore, they do not change the configuration's relationship to the data (i.e., Stress). Thus, differences between two configurations that can be eliminated by similarity transformations cannot possibly be meaningful, because they are not anchored in the data.

The similarity transformations that remove meaningless differences as much as possible are called *Procrustean transformations* (Gower and Dijksterhuis 2004). One begins by picking one "pleasing" configuration as the fixed *target*. Then, all other

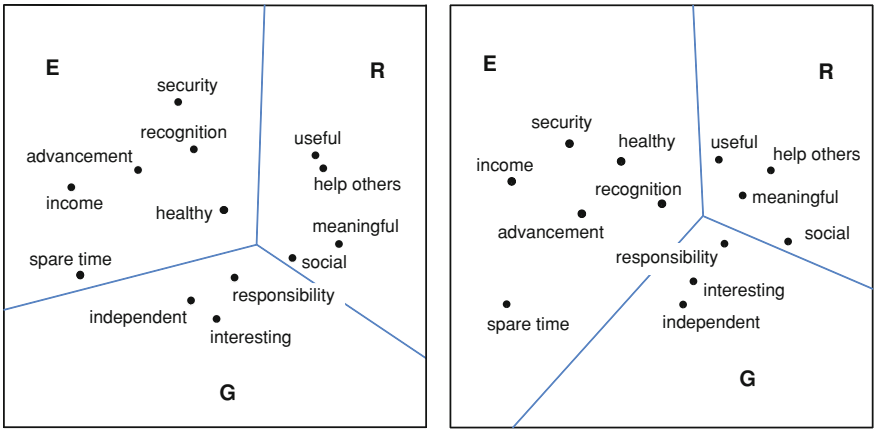


**Fig. 7.6** West- and East-German structures of work values, both with partitions based on the ERG theory

configurations are optimally fitted to this target, one by one.[2] Differences that remain after Procrustean fittings are meaningful and interpretable.

Procrustean transformations rest on the assumption that the given MDS configurations must not be changed. This restriction too should be anchored in the data. In case of the East- and West-German work value structures, it is only partly true. Figure 7.6 shows in the left panel the West German structure, and in the right panel the East German MDS result. The two configurations look quite similar. However, this similarity depends to some extent on how the MDS was done, and not on the data only: when computing the East-German configuration, the West-German MDS solution was used as a starting configuration so that the MDS program would begin from a position that corresponds to the hypothesis "both configurations are equal". If one leaves it to the MDS program to pick its own starting configuration (by one method or another), then this can lead to solutions that differ substantially. What remains the same in all these solutions, however, is that they all exhibit the same three groups of work values (E, R, and G, respectively). This finding is interesting in itself. It can be taken an indication that only this structural aspect is robustly data-driven. When comparing the East- and West-German MDS solutions, the stability of

[2] Unfortunately, few statistics packages offer Procrustean transformations and if they do (such as SYSTAT, for example), then the proper modules can be difficult to find and use. If the users have a program for matrix algebra (e.g., MatLab or R), they can quite easily compute Procrustean fittings themselves. Let $\mathbf{X}$ be the target configuration and $\mathbf{Y}$ the configuration to be fitted. Then, compute $\mathbf{C} = \mathbf{X}'\mathbf{Z}\mathbf{Y}$, and find for it the singular value decomposition $\mathbf{C} = \mathbf{P}\mathbf{\Phi}\mathbf{Q}'$. (In the centering matrix $\mathbf{Z} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}'$, $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is a vector of ones.) The optimal rotation/reflection for $\mathbf{Y}$ is done by $\mathbf{T} = \mathbf{Q}\mathbf{P}'$; the optimal central dilation factor is $s = \text{trace}(\mathbf{X}'\mathbf{Z}\mathbf{Y}\mathbf{T})/\text{trace}(\mathbf{Y}'\mathbf{Z}\mathbf{Y})$; the optimal translation vector is $\mathbf{t} = n^{-1}(\mathbf{X} - s\mathbf{Y}\mathbf{T})'\mathbf{1}$. Hence, the solution is $\hat{\mathbf{Y}} = s\mathbf{Y}\mathbf{T} + \mathbf{1}\mathbf{t}'$ (Borg and Groenen 2005).

this *meta-structural* feature of the MDS solutions is, therefore, more important than simple point-by-point correspondences.

## 7.9 Evaluating Stress Mechanically

A frequent mistake of recipients of MDS results is that they are often all too quick in rejecting an MDS solution because its Stress seems "too high". The Stress value is, however, merely a *technical* index, a target criterion for the optimization algorithm of the MDS program. An MDS solution can be robust and replicable, even if its Stress value is high. Stress, moreover, is *substantively blind* (Guttman 1977), that is, it says nothing about the compatibility of a content theory with the MDS configuration, or about its interpretability.

Stress, moreover, is a *summative* index for *all* proximities. It does not inform the user how well a *particular* proximity value is represented in the given MDS space. Consider the example in Fig. 2.2. This configuration does not represent all proximities equally good. This is made clear by the Shepard diagram in Fig. 3.1 where one notes an outlier point, with coordinates (3.44, 0.80), in the lower left-hand corner. This outlier contributes over-proportionally to the total Stress because it lies—measured in the vertical direction ("error")—very far from the regression line. Substantively, the value 3.44 is the average similarity rating for the pair 'Egypt versus Brazil' in Table 2.1. The distance of the points Egypt and Brazil in Fig. 2.2 is therefore represented relatively poorly in the MDS solution. That can have many reasons. One possibility is that the test persons found it particularly difficult to judge the overall similarity of these two countries, thus introducing measurement error. Another possibility is that the test persons used other or additional or individually different attributes when comparing these countries than what they used for the other countries.

On a higher level of aggregation, one can ask how good each single object is represented in an MDS configuration. This is measured by *Stress per point* (SPP), which is simply the average of the squared error terms for each point. For the country similarity data represented in Fig. 2.2, and using SMACOF as described on p.100ff., one gets all SPP values, sorted and rounded, by the command `round(sort(res.wish $spp, decreasing = TRUE), 3)`:

```
 FRANCE   BRAZIL   ISRAEL     CUBA    INDIA    JAPAN    EGYPT
  0.054    0.052    0.042    0.041    0.040    0.040    0.036
  CONGO    CHINA      USA  YUGOSLAV   RUSSIA
  0.021    0.020    0.019    0.014    0.008
```

One can also plot these results by

```
plot(res.wish, plot.type = "stressplot", main = "Stress Plot",
+ ylab = "Stress Contribution (\%)", ylim = c(0, 15),
xlim=c(0,13))
```

This command generates Fig. 7.7. It exhibits that France and Brazil contribute most to the overall Stress. It also shows that France and Brazil are not true outliers

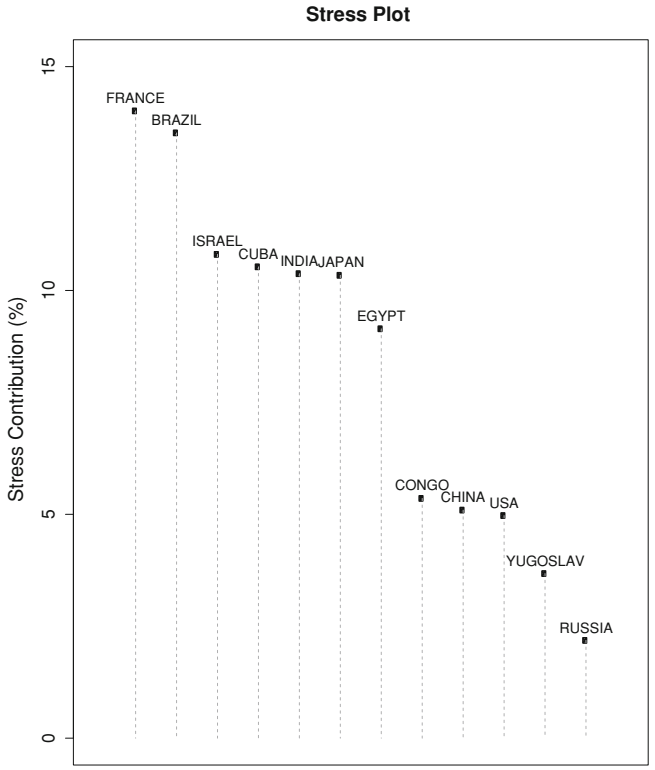


**Fig. 7.7** Stress-per-point values for all variables of country similarity experiment

in the distribution of the countries' SPP values, and so there is no pressing need to ponder about reasons why these two countries seem particularly difficult to compare with the other countries, or why our 2-dimensional composition rule for generating similarity judgments for countries does not seem to hold for these countries.

Not all MDS programs compute SPP values (or similar point-fit measures). However, most programs allow saving the configuration's distances so that one can compute appropriate point-fit measures with standard data analysis programs (e.g., the correlation between the proximities and the corresponding MDS distances).

A simple way to deal with ill-fitting points is to eliminate them from the analysis. This is a popular approach, based on the rationale that such points have a special relation to the other points that needs additional considerations. Another solution is to increase the dimensionality of the space so that these points can move into the extra space and form new distances. The rationale in this case is that the proximities of the objects represented by these points to the other points are based on additional dimensions that are not relevant in other comparisons.

In any case, accepting or rejecting an MDS representation on the basis of overall Stress only can be too simple. This is easy to see from an example. Consider Fig. 7.6. If we increase the dimensionality of this solution to $m = 3$, the Stress goes down from 0.17 to 0.09. If we proceed in the same way in case of Fig. 2.2, we get the same reduction in Stress. However, in the latter case, the reduction in Stress is caused by essentially two points only. That is, "healthy working conditions" and, in particular, "spare time" clearly move out of the plane in Fig. 7.6 into the third dimension. In the former case, all points jitter (some more, some less) about the plane, which looks as if the third dimension is capturing essentially only unsystematic variance contained in the data ("noise").

For data with large noise components, therefore, low-dimensional MDS solutions can have high Stress values, but they may still be better in terms of theory or replicability than higher-dimensional solutions with lower Stress values. In that case, a low-dimensional solution may be an effective *data smoother* that brings out the true structure of the data more clearly than an over-fitted higher-dimensional MDS representation.

## 7.10 Always Interpreting "the Dimensions"

Interpreting an MDS solution can be understood as projecting given or assumed content knowledge onto the MDS configuration. The country similarity example of Sect. 2.1 demonstrates how this is typically done: what one interprets are *dimensions*. MDS users often *automatically* ask for the meaning of "the" dimensions, by which they often mean the axes of the plot that the MDS program delivers. These axes are almost always the principal axes of the solution space, in Fig. 2.2 labeled as "Dimension 1" and "Dimension 2", respectively. Yet, this dimension system can be arbitrarily rotated and reflected, and oblique dimensions would also span the plane. Hence, users do not have to interpret the dimensions offered by the MDS program, but they could look for $m$ dimensions (in $m$-dimensional space) that are more meaningful.

Then, there is no natural law that guarantees that dimensions must always be meaningful. Thus, one should be open for other ways of interpreting MDS solutions. One possibility is to look for meaningful *directions* rather than for dimensions. There can be more than $m$ meaningful directions in $m$-dimensional space. Like dimensions, each of them can be conceived as an *internal scale*. It is generated by perpendicularly projecting the points onto a directed line, adding a "ruler" to this line to generate measurements, and then studying the distribution of the points' dimensional values. To avoid messy plots, one can run all internal scales through a common point such as the centroid of the configuration. Points to the left of this anchor point are given negative scale values; those to the right of it receive positive values. To interpret the internal scale, one studies the point distribution with a focus on content questions such as these: What points lie at the extremes of the scale? How do they differ in terms of content? What is the attribute where they differ most? Why are the points

$i, j, \ldots$ so close together? What do they have in common? Answering such questions gives meaning to the scale.

Additional data can be helpful in such interpretations. We show this for the country similarity example. Table 2.2 exhibits the coordinates of the MDS solution in Fig. 2.2 and the countries' values on two *external scales*, Economic Development and Number of Inhabitants. These scales can be fitted into the MDS space. Geometrically, this fitting can be thought of as rotating a line (running through the origin) in space until the points' projections correlate maximally with their external scale values. Computationally, the best-fitting line is found by multiple regression, where the external scale is the criterion and the coordinate vectors are the predictors. For example, Economic Development = additive constant $c + b_1 \cdot \text{Dim.1} + b_2 \cdot \text{Dim.2}$, where $b_a$ is the regression weight of dimension $a$ and "Dim.1" and "Dim.2" are the coordinate vectors in Table 2.2. For this equation, any statistics program yields as the optimal solution[3] $b_1 = 3.27$ and $b_2 = -1.45$. With these weights, the best-fitting line can be drawn. It runs through the origin of the coordinate system, and through the point with the $X$-coordinate 3.27 and the $Y$-coordinate $-1.45$ (Fig. 7.8, left panel).

The points' projections onto this line correlate with the external scale values with $r = 0.96$. Thus, Economic Development could indeed be an attribute that underlies the respondents' judgments of similarity. The Number of Inhabitants scale, in contrast, cannot be embedded that well into the MDS solution ($r = 0.39$).[4] This property, therefore, cannot really explain the country similarity ratings (if one accepts the 2-dimensional MDS solution as the proper representation of the true similarities).

Directions are but special cases of *regions*. Regions are sub-sets of points of an MDS space that are *connected* (i.e., each pair of points in a region can be joined by a curve whose points lie completely within this region), *non-overlapping*, and *exhaustive* (i.e, each point lies in exactly one region). For interpretational purposes, we ask to what extent certain classifications or orderings of the objects on the basis of *content facets* correspond to regions of the MDS space. Expressed differently, we ask whether the MDS configuration can be *partitioned* into substantively meaningful regions, and, if so, how these regions can be described.

An example for such a partitioning is shown in Fig. 7.6. Here, the different objects ("work values") were first classified into three categories on the basis of a theory by Alderfer (1972): work values related to outcomes that satisfy existential-material needs (E), social-relational needs (R), or cognitive-growth needs (G). This ERG typology surfaces in MDS space in certain neighborhoods that can be separated from each other by cutting the plane in a wedge-like fashion. The same type of partitioning is possible both in the West-German and also in the East-German MDS plane. Hence, the two solutions are equivalent in the ERG sense (Borg and Braun 1996).

---

[3] The optimal weights are the non-standardized or raw weights of the multiple regression solution (betas).

[4] That this scale cannot be easily fitted into the MDS solution can be seen, for example, from the closeness of the points representing Israel and the USA, two countries with vastly different numbers of inhabitants.

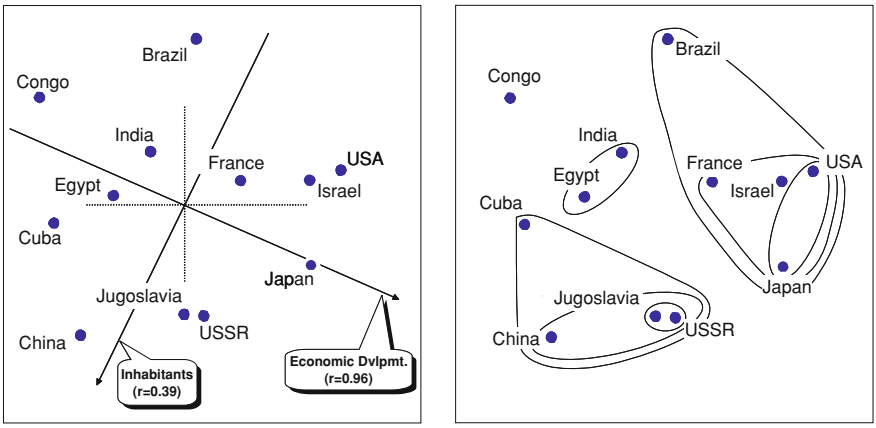


**Fig. 7.8** MDS configuration of Fig. 2.2 with fitted external scales (*left panel*) and with hierarchical clusters (*right panel*)

Partitioning an MDS space is done *facet by facet*. For each facet $F_i$, one generates a *facet diagram*. This is simply a copy of the MDS configuration where each point is replaced by the code that indicates to which category of $F_i$ the respective point belongs. One then checks to what extent and in which way this facet diagram can be partitioned into regions that contain only codes of one particular type. The emerging regions should be as "simple" as possible, e.g. with straight partitioning lines. This is desirable because simple partitions can also be characterized by simple laws of formation that promise to be more robust and replicable than complicated patterns that are fitted too closely to the particular data and its noise.

Although there exist computer programs that yield partitions for facet diagrams that are optimal in some sense (Borg and Shye 1995), it is typically more fruitful for the user to work with pencil and eraser on a print-out of the facet diagram. This way, partitioning lines can be drawn, re-drawn, and simplified in an open-eyed fashion, paying attention to content and substantive theory. One may decide, for example, to admit some incorrect placements of points in wrong regions, because simple overall patterns with some errors are better than perfect partitions with overly complicated partitions.

Three prototypical regionalities that often arise in practice are shown in Fig. 7.9: *axial*, *modular*, and *polar* partitions. Axial and modular partitions are either based on ordered facets, or they suggest ordered facets. Polar partitions, in contrast, are typically related to unordered (nominal) facets. Of course, if the sectors in a polar partition are arranged similarly in many replications, then one should think about reasons for this *circular* order.

Regionalizations—simple ones, in particular—become unlikely to result by chance if the number of points goes up. That is easy to see from a thought experiment. Assume you take a set of $n$ ping-pong balls and label some of them with "a", others

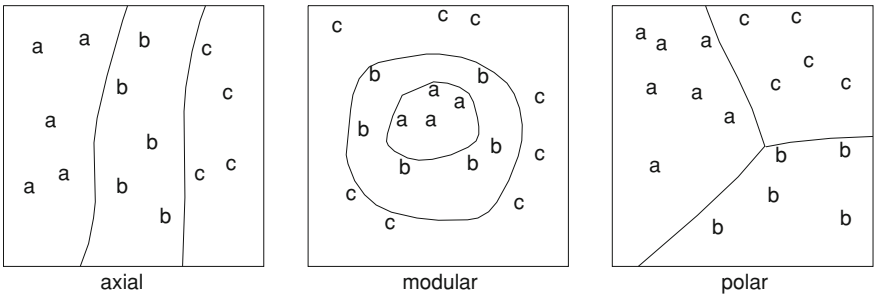


**Fig. 7.9** Prototypical partitioning of MDS configurations by three facets, each one with three categories (a, b, c)

with "b", and still others with "c". Then, throw them all into a bucket, mix them thoroughly, and pour the bucket onto the floor. After the balls come to their parking positions, try to partition the resulting configuration into regions. This will be difficult or even impossible if one wants simple regions as in Fig. 7.9. It is even less likely that the regionality that one finds in one case can be replicated when the experiment is repeated. A simple regional pattern, therefore, suggests a lawful relationship of the facet on which it is based and the MDS representation of the proximities: the facet seems to structure the observations. This notion becomes even more powerful if an MDS configuration can be partitioned by more than one facet at the same time so that the different organizational patterns can be stacked on top of each other as, for example, in the radex in Fig. 2.8.

An MDS solution can be partitioned, in principle, by as many facets as the user can think of. There is no fixed relation between the number of facets and the dimensionality of the space. This is different for dimensions: in an $m$-dimensional space, one always seeks to interpret exactly $m$ dimensions. A dimensional interpretation corresponds to a combination of $m$ axial facets (see Fig. 7.9, left panel), each generating an ordered set of (infinitely) narrow bands with linear boundary lines so that a linear mesh (as, e.g., in Fig. 6.5) is generated.

Regions are sometimes confused with *clusters*. Clusters, however, are but special cases of regions. They are often defined as lumps (or chains) of points surrounded by empty space so that each point in a cluster is always closer to at least one point within the cluster than to any point not in the cluster (Guttman 1977). Clustering in that sense is not required for perfect regions. Regions are like countries that cut a continent like Europe into pieces. Malmö/Sweden, for example, is much closer to Copenhagen/Denmark—both are connected by a bridge—than to any large Swedish city, so the Swedish cities do not form a cluster on the European map, but they are all in the same region.

Clusters, moreover, are *formal* constructs, while regions are based on *substantive* thinking that is often expressed via facets. Nevertheless, one can always cluster proximities and then check how the resulting clusters organize the points of an MDS

solution, as demonstrated in Fig. 7.8 (right panel) for the country similarity data. The various contour lines show that cluster analysis found two major clusters, containing the Western and the Communist countries, respectively, plus a two element cluster with Egypt and India, plus a singleton for the Congo. These clusters, when projected onto the diagonal from South-West to North-East, corresponds roughly to the dimension Pro-Western versus Pro-Communist found by Wish (1971). Note, however, that cluster analysis is not particularly robust in the sense that different amalgamation criteria can lead to vastly different clusters.[5] Cluster analysis, therefore, is not a method for "validating" an MDS solution or interpretation, as some writers argue. Rather, cluster analysis typically just leads to groupings of points that tend to surface similarly in MDS solutions.[6]

## 7.11 Poorly Dealing with Disturbing Points

A frequent problem in MDS applications is what to do with points that do not fit into an interpretation. A typical case is a configuration that cannot be partitioned in a theoretically pleasing way because one or a few points are positioned such that they prevent simple partitioning lines. In such cases, one may decide to interpret the solution with overlapping regions, or stick to the partitioning notion and generate rather curvy partitioning lines (as, e.g., in Fig. 6.4). A third solution is to draw a best-possible partitioning system that admits some classification errors, where some points remain in regions to which they do not belong. Probably the most common approach, however, is to eliminate such points from the MDS configuration by "explaining them away" in substantive terms. This method is popular in scaling in general. In scale construction, for example, items that do not fit into a unidimensional structure are systematically eliminated. Reasons why these points do not belong to the universe of the other points are easily found if needed.[7]

A completely different way to deal with disturbing points is asking how much Stress goes up if one shifts these points in space such that simple partitioning becomes possible. The easiest way to answer this question is the following. Take the coordinate matrix **X** of the MDS solution, replace the coordinates values of the disturbing points with "should"-coordinates (i.e., coordinates that put this point into a position where it is not disturbing anymore), and use this modified **X** as a starting configuration for a new MDS analysis, setting the number of iterations to zero. The MDS program then

[5] To generate Fig. 7.8, we used hierarchical single-linkage cluster analysis. Choosing the "average" criterion leads to a solution where the Congo does not remain a singleton, but it is included into one cluster together with Egypt and India.

[6] Cluster analysis identifies cluster structures in the total space of the data: if there are clear clusters in this space, then the major clusters also tend to appear in the plane spanned by the first two principal axes of this space or in an MDS plane.

[7] Guttman (1977) comments on this: "To say that one 'wants to construct' a scale... towards something ... is almost analogous to saying that one 'wants' the world to be flat... To throw away items that do not 'fit' unidimensionally is like throwing away evidence that the world is round" (p. 105).

computes the Stress value for the modified **X**, without changing it. The increment in Stress can be used to evaluate the consequences of moving some selected points so that simple partitionings become possible. If this increment is small, one would probably prefer the solution that allows a substantively simple interpretation over the optimal-Stress solution. The rationale is that it promises to be better replicable, being based on a substantive law of formation, than the solution that represents the given set of data with minimal Stress.

A formally better solution of the problem of ill-placed points is using confirmatory MDS. However, confirmatory MDS with regional restrictions can be quite difficult to formulate and to implement (see Chap. 6). Hence, before trying to do this, a simple shift-and-see approach yields a quick answer that is often sufficient. Some movements of a small number of points normally do not affect the overall Stress very much. This does not mean, however, that one should not study disturbing points more closely in substantive-theoretical ways and also test to what extent their locations replicate with new data.

## 7.12 Scaling Almost Equal Proximities

Proximity data cannot always be represented in a space of low dimensionality. This is true, for example, if the data have a large error component or if they are simply random data. A second instance that is less obvious is data that are essentially constant.

As an example, consider the co-occurrence of 112 words (such as "tribal", "moose", "arctic", and "health") from Sect. 4.5. Using these co-occurrences as proximities in ordinal MDS (with the secondary approach to ties), we obtain the 2-dimensional solution in Fig. 7.10. It looks interesting at first sight, but its Stress value is 0.41, a high value. Moreover, on closer inspection, the points seem to be spread out evenly within a round cloud, which seems artificial, not data-driven. The transformation plot in Fig. 7.11—a data-versus-disparities plot—is conspicuous since it exhibits a regression relation with only very few steps. Looking more closely, one notes that most disparities are numerically very similar, that is, they are all in the interval from 0.93 to 1.00. Turning to the data, one finds that most of the word pairs (99.6 %) do *not* occur together, so their co-occurrence value is equal to zero. These data are all mapped (by using ordinal MDS with the secondary approach to ties) into the same large disparity (= 1). The remaining part of the regression line in Fig. 7.11, therefore, is based on merely 0.4 % of the data. The histogram in Fig. 7.12 confirms this clearly: the overwhelming proportion of the disparities is exactly equal (= 1), except for a few that have either the value 0.93 or 0.96.

If most of the data are equal, then their MDS representation is not informative. Indeed, if all data are equal, Buja et al. (1994) have shown that 2-dimensional (ratio-) MDS leads to points that all lie on concentric circles (Fig. 7.13, right panel), similar to Fig. 7.10. Moreover, the points of this solution can all be interchanged without affecting the Stress. In a 1-dimensional space, a solution as in Fig. 7.13 (left panel)

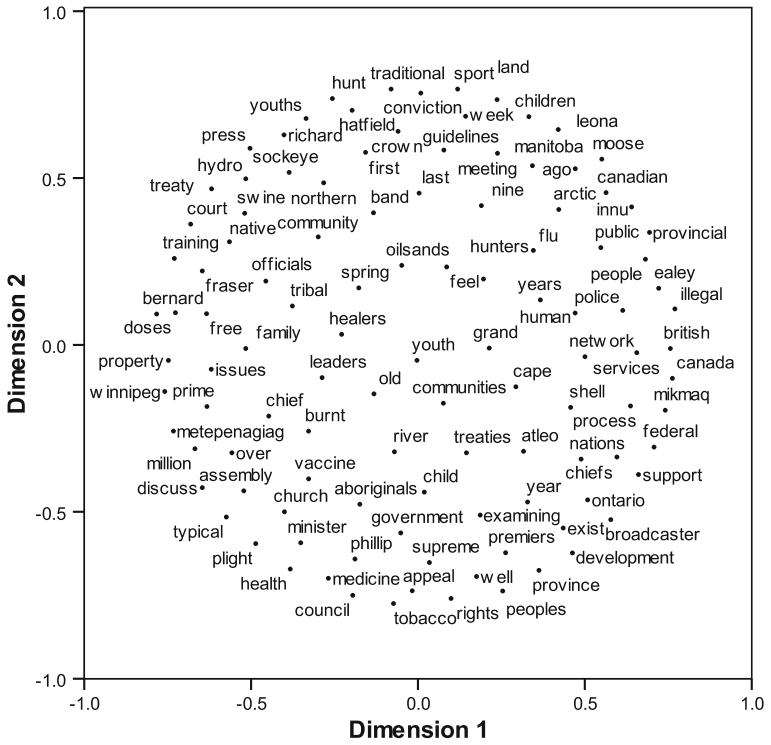


**Fig. 7.10** Ordinal MDS solution (PROXSCAL with Ties=Keep) for the co-occurrences of 112 words with word "aboriginal" in Canadian newspaper articles

is found. In 3-dimensional or higher space, all points are evenly distributed on a hypersphere.

In case of a very small data matrix (and, therefore, very few points) the Stress value can become zero. With only 3 points, a perfect solution is obtained if the points form the corners of an equilateral triangle (similar to Fig. 7.2). In 3 dimensions, 4 points forming the corners of a regular tetrahedron form a perfect solution.

These considerations show that users should keep an eye on the case of almost equal proximities or disparities. In particular, they must look closely at the units of the $Y$ axis of the Shepard diagram: if most of these values are almost equal, then caution is needed. Most computer program choose an origin for the $Y$ axis that magnifies the range of the observed values. If the origin of $Y$ in a transformation plot is zero as in Fig. 7.11, then the almost-equal problem becomes obvious immediately. Also investigate the distribution of the proximities or disparities, preferably in a histogram like in Fig. 7.12. If the histogram shows that the disparities are all close together and much different from zero, then one can expect the 2D solution of concentric circles to occur.

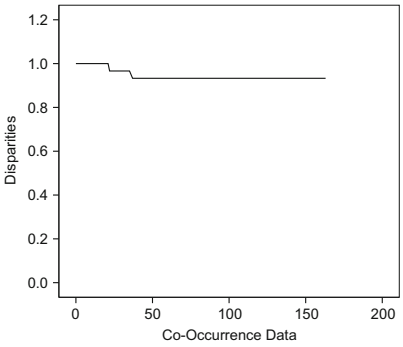


**Fig. 7.11** Transformation plot for solution in Fig. 7.10

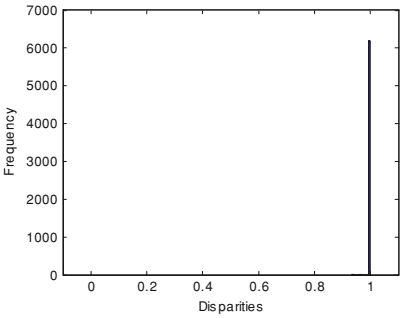


**Fig. 7.12** Histogram of disparities for solution in Fig. 7.10

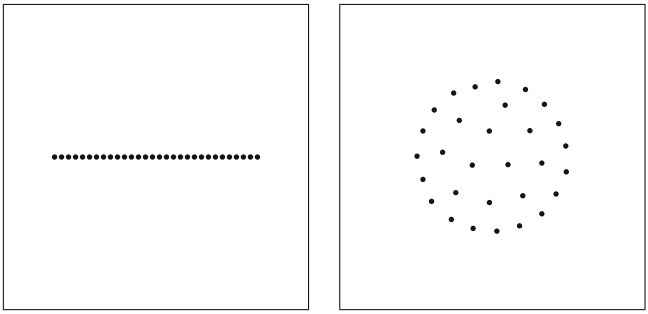

**Fig. 7.13** MDS-solutions for constant dissimilarities ($n = 30$): 1-dimensional solutions (*left*), 2-dimensional solution (*right*)

Another way to diagnose peculiarities in the data is scaling them with different MDS models. In the above, we used ordinal MDS preserving ties (secondary approach). Interval MDS yields almost the same result. However, if ordinal MDS is used with the primary approach to ties—which allows to untie ties in the distances—a radically different solution is obtained, where most of the points cluster in one point, and a few points scatter about this cluster. Stress, moreover, is just 0.09, much smaller than for the other MDS representations. If different MDS models yield such vastly different results, then something is almost always wrong. With well-structured data, different MDS models yield solutions that do not differ much.

## 7.13 Over-Interpreting Dimension Weights

A popular MDS model is INDSCAL. It not only scales a whole battery of proximity matrices in just one computer run, but it also models inter-individual differences by a common space with individual dimension weights. Moreover, the dimensions identified by INDSCAL are uniquely oriented so that it appears that this model does indeed find "the" dimensions of the MDS space.

Unexperienced users of INDSCAL tend to over-interpret these properties. First, they should know that the uniqueness of INDSCAL's dimensions may be quite weak, that is, other dimensions may explain almost as much variance. INDSCAL simply finds the best orientation of the dimension system and also the best weights, but it does not inform the user how much unit weights would explain, for example. Users may feel that if the weights reported by INDSCAL scatter a lot, then they also explain much variance, but this is a fallacy: The scatter of the weights (about the origin) is not related to the explained variance (for an example, see Borg and Lingoes 1978). However, the larger the weights, the better the fit.

Second, the dimension weights are always *dependent* on the *norming* of the common space. This is easy to see from an example. Consider Fig. 5.4, where the common space is scaled so that the sums of the squared projections of the points onto the axes $X$ and $Y$, respectively, are equal. This generates the squarish grid. Weighting the $Y$ dimension of this common space with the factor 2 leads to the individual space 1 in Fig. 5.4; analogously, weighting the $X$ dimension with the factor 2 yields the individual space 2. This seems to suggest that, for person 1, the $Y$ dimension is twice as important as the $X$ dimension, while the opposite is true for person 2. This interpretation of the dimension weight depends, however, on the norming of the common space, and this norming is not data-driven but arbitrarily chosen by the programmer. If the common space is stretched or compressed along its dimensions, then the model fit remains the same, provided one compensates for such transformations by choosing proper dimension weights. Thus, one cannot tell from the INDSCAL weights if person 1 finds dimension $Y$ "twice as important" or, indeed, not even "more important" than $X$, but only that this person finds $Y$ more important than person 2. The dimension weights, therefore, cannot be compared intra-individually over the various dimensions. What can be compared is the *order* of the weights of *different persons for the same dimension*. What can also be compared is the order of the dimension weights that different persons have for different dimensions. For example, person 16 always weights the dimension "Pro-Western versus Pro-Communist" in Fig. 5.5 more than the dimension "Economic development" *compared to person 12*, irrespective of the norming of the common space (Schönemann and Borg 1983). Therefore, for the 2-dimensional case, users sometimes report "flattened weights" that collapse the two weights onto one scale.[8] Such weights make it easier to test hypotheses about differences in the salience of the dimensions for different groups of persons. For

[8] Flattened weights are computed differently. ALSCAL in SPSS first rescales the dimension weights of each person such that they add to 1.00; it then drops dimension 2 and subtracts the mean from each weight for dimension 1. This yields the flattened weights.

example, women with bulimic symptoms, when looking at photographic pictures of other women, pay more attention to body size than to facial expressions; for women without bulimic symptoms, the opposite is true (Viken et al. 2010).

## 7.14 Unevenly Stretching MDS Plots

Some MDS programs produce plots of the configuration **X** where the axes are scaled differently. That is, the units of the *X*- and the *Y*-axis of the plot are not the same which means that the configuration is *stretched* along one of the coordinate axes. The reason for changing the plot's "aspect ratio" is to unclutter the configuration. MDS users sometimes also use dimension-wise stretchings of the plots, or of regions of the plots, to make the configuration as big as possible on their output devices. However, such uneven stretchings almost always lead to misinterpretations, because the more two points lie on a line that is parallel to the stretched axis, the more their distance is actually *smaller* than it *appears* to be. The plot, therefore, can be quite misleading. Hence, users should always check if the configuration plots provided by the MDS program use the same units on the axes. If not, the plots should be redone with *equal units*. To do this in SMACOF, for example, one simply sets `asp=1` in the `plot` command (see p. 100). Users should also make sure that their printing devices are not set to "stretch to page" which, in landscape view, will automatically distort the plots. Such fittings militate against the very purpose of MDS to optimally visualize proximity data.

## 7.15 Summary

MDS users occasionally get confused as a consequence of using key concepts too vaguely. In particular, they subsume all models that somehow map data into a multi-dimensional space under the notion of MDS, or they use the concept of distance all too loosely. Some technical mistakes are also common. For example, not specifying the proper polarity of proximities so that the MDS program uses similarity data as dissimilarity data, or vice versa, leads to scaling problems. Another simple mistake is making MDS programs terminate their iterations too early, or not studying the effects of using different starting configurations. Once aware of these mistakes, they can be easily avoided. Another mistake is overlooking degenerate solutions in ordinal MDS. They can be avoided by using stronger MDS models. A rather frequent mistake is always asking for the meaning of "the" dimensions: Dimensions are but a special case of regions, and other meaningful patterns may also exist in an MDS configuration. Simply discarding disturbing points from an MDS solution is also too mechanical: Sometimes, such points can be shifted without affecting the Stress very much. Then, when comparing different MDS solutions, one should first get rid of meaningless differences via Procrustean transformations. Moreover, data that are almost all equal

can lead to meaningless MDS solutions; and interpreting the results of scaling the INDSCAL model requires much care, because the individual weights may not explain much variance even if they scatter substantially and because the weights themselves are contingent on how one norms the common space. Finally, MDS configuration plots should always be done with equal units on the axes.

## References

Alderfer, C. P. (1972). *Existence, relatedness, and growth*. New York: Free Press.

Borg, I., & Braun, M. (1996). Work values in East and West Germany: Different weights but identical structures. *Journal of Organizational Behavior*, *17*, 541–555.

Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling* (2nd ed.). New York: Springer.

Borg, I., & Lingoes, J. C. (1978). What weights should weights have in individual differences scaling? *Quality and Quantity*, *12*, 223–237.

Borg, I., & Shye, S. (1995). *Facet theory: Form and content*. Newbury Park, CA: Sage.

Buja, A., Logan, B. F., Reeds, J. R. & Shepp, L. A. (1994). Inequalities and positive-definite functions arising from a problem in multidimensional scaling. *The Annals of Statistics*, *22*, 406–438.

Gower, J. C., & Dijksterhuis, G. B. (2004). *Procrustean problems*. New York: Oxford.

Guthrie, J. T. (1973). Models of reading and reading disability. *Journal of Educational Psychology*, *65*, 9–18.

Guttman, L. (1977). What is not what in statistics. *The Statistician*, *26*, 81–107.

Schönemann, P. H., & Borg, I. (1983). Grundlagen der metrischen mehrdimensionalen Skaliermethoden. In H. Feger & J. Bredenkamp (Eds.), *Enzyklopädie der Psychologie: Messen und Testen* (pp. 257–345). Göttingen: Hogrefe.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*, 327–352.

Viken, R. J., Treat, T. A., Nosofsky, R. M., & McFall, R. M. (2002). Modeling individual differences in perceptual and attentional processes related to bulimic symptoms. *Journal of Abnormal Psychology*, *11*, 598–609.

Wish, M. (1971). Individual differences in perceptions and preferences among nations. In C. W. King & D. Tigert (Eds.), *Attitude research reaches new heights* (pp. 312–328). Chicago: American Marketing Association.

# Chapter 8
# MDS Algorithms

**Abstract** Two types of solutions for MDS are discussed. If the proximities are Euclidean distances, classical MDS yields an easy algebraic solution. In most MDS applications, iterative methods are needed, because they admit many types of data and distances. They use a two-phase optimization algorithm, moving the points in MDS space in small steps while holding the data or their transforms fixed, and vice versa, until convergence is reached.

**Keywords** Classical MDS · Iterative MDS algorithm · Disparity · Two-phase algorithm · Rational starting configuration · Majorization · SMACOF

For most MDS models, a best-possible solution $\mathbf{X}$ cannot be found by simply solving a system of equations. The conditions for MDS solutions are so complicated, in general, that they are algebraically untractable. MDS solutions must, therefore, be approximated iteratively, using intelligent search procedures (algorithms) that reduce the Stress by repeatedly moving the points somewhat to new locations and by successively re-scaling the proximities until a Stress-minimum is found.

Algorithms of this kind are not needed if one wants to assume or if one can prove that the dissimilarity data $\delta_{ij}$—possibly derived first from inverting similarity data—are Euclidean distances. In this case, *classical MDS* can be used to find the MDS solution $\mathbf{X}$ analytically.

## 8.1 Classical MDS

Classical MDS—also known as *Torgerson scaling* and as *Torgerson–Gower scaling*— works as follows:

1. Square the dissimilarity data: $\mathbf{\Delta}^{(2)}$.

I. Borg et al., *Applied Multidimensional Scaling*, SpringerBriefs in Statistics,
DOI: 10.1007/978-3-642-31848-1_8, © The Author(s) 2013

2. Convert the squared dissimilarities to scalar products through double centering[1] of $\boldsymbol{\Delta}^{(2)}$: $\mathbf{B}_{\boldsymbol{\Delta}} = -\frac{1}{2}\mathbf{Z}\boldsymbol{\Delta}^{(2)}\mathbf{Z}$, where $\mathbf{Z} = \mathbf{E} - n^{-1}\mathbf{1}\mathbf{1}'$, and where $\mathbf{E}$ is the unit matrix (with all elements in the main diagonal equal to 1, and all others equal to 0) and $\mathbf{1}$ a vector with a 1 in each of its cells.
3. Compute the eigen-decomposition $\mathbf{B}_{\boldsymbol{\Delta}} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}'$.
4. Take the first $m$ eigenvalues greater than 0 ($=\boldsymbol{\Lambda}_+$) and the corresponding first $m$ columns of $\mathbf{Q}$ ($=\mathbf{Q}_+$). The solution of classical MDS is $\mathbf{X} = \mathbf{Q}_+\boldsymbol{\Lambda}_+^{1/2}$.

We demonstrate these steps with a small numerical example:

$$\boldsymbol{\Delta} = \begin{bmatrix} 0 & 4.05 & 8.25 & 5.57 \\ 4.05 & 0 & 2.54 & 2.69 \\ 8.25 & 2.54 & 0 & 2.11 \\ 5.57 & 2.69 & 2.11 & 0 \end{bmatrix}, \text{ which leads to } \boldsymbol{\Delta}^{(2)} = \begin{bmatrix} 0.00 & 16.40 & 68.06 & 31.02 \\ 16.40 & 0.00 & 6.45 & 7.24 \\ 68.06 & 6.45 & 0.00 & 4.45 \\ 31.02 & 7.24 & 4.45 & 0.00 \end{bmatrix}.$$

In the second step we compute

$$\begin{aligned} \mathbf{B}_{\boldsymbol{\Delta}} &= -\frac{1}{2}\mathbf{Z}\boldsymbol{\Delta}^{(2)}\mathbf{Z} \\ &= -\frac{1}{2}\begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} \end{bmatrix}\begin{bmatrix} 0.00 & 16.40 & 68.06 & 31.02 \\ 16.40 & 0.00 & 6.45 & 7.24 \\ 68.06 & 6.45 & 0.00 & 4.45 \\ 31.02 & 7.24 & 4.45 & 0.00 \end{bmatrix}\begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} \end{bmatrix} \\ &= \begin{bmatrix} 20.52 & 1.64 & -18.08 & -4.09 \\ 1.64 & -0.83 & 2.05 & -2.87 \\ -18.08 & 2.05 & 11.39 & 4.63 \\ -4.09 & -2.87 & 4.63 & 2.33 \end{bmatrix}. \end{aligned}$$

In the third step we compute the eigen-decomposition of $\mathbf{B}_{\boldsymbol{\Delta}} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}'$ with

$$\mathbf{Q} = \begin{bmatrix} 0.77 & 0.04 & 0.50 & -0.39 \\ 0.01 & -0.61 & 0.50 & 0.61 \\ -0.61 & -0.19 & 0.50 & -0.59 \\ -0.18 & 0.76 & 0.50 & 0.37 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Lambda} = \begin{bmatrix} 35.71 & 0.00 & 0.00 & 0.00 \\ 0.00 & 3.27 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -5.57 \end{bmatrix}.$$

In the fourth step, this yield the MDS configuration

[1] This means that the centroid of the MDS configuration becomes the origin. The coordinates of $\mathbf{X}$, thus, should sum to 0 in each column of $\mathbf{X}$. This does not carry any consequences for the distances of $\mathbf{X}$, that is, any other point could also serve as the origin. However, one point must be picked as an origin to compute scalar products.

$$\mathbf{X} = \mathbf{Q}_+ \boldsymbol{\Lambda}_+^{1/2} = \begin{bmatrix} 0.77 & 0.04 \\ 0.01 & -0.61 \\ -0.61 & -0.19 \\ -0.18 & 0.76 \end{bmatrix} \begin{bmatrix} 5.98 & 0.00 \\ 0.00 & 1.81 \end{bmatrix} = \begin{bmatrix} 4.62 & 0.07 \\ 0.09 & -1.11 \\ -3.63 & -0.34 \\ -1.08 & 1.38 \end{bmatrix}.$$

To check the goodness of this solution, we compare its distances with the given dissimilarities, $\boldsymbol{\Delta}$. The distances are

$$\mathbf{D} = \begin{bmatrix} 0.00 & 4.68 & 8.26 & 3.60 \\ 4.68 & 0.00 & 5.85 & 2.75 \\ 8.26 & 5.85 & 0.00 & 3.08 \\ 3.80 & 2.75 & 3.08 & 0.00 \end{bmatrix}, \quad \text{so that} \quad \boldsymbol{\Delta} - \mathbf{D} = \begin{bmatrix} 0.00 & -0.63 & -0.01 & 1.97 \\ -0.63 & 0.00 & -3.31 & -0.06 \\ -0.01 & -3.31 & 0.00 & -0.97 \\ 1.77 & -0.06 & -0.97 & 0.00 \end{bmatrix}.$$

In this example, the distances among the points of the MDS configuration constructed by classical MDS are only approximately equal to the given dissimilarity data. The reason for this result is that the dissimilarities in $\boldsymbol{\Delta}$ are *not* Euclidean distances, as classical MDS assumes. Mathematicians would have noticed that in the third step above, because if the dissimilarities are Euclidean distances, then all eigenvalues are non-negative. If negative eigenvalues occur, one may decide to "explain them away" as caused by "error" in the dissimilarities, provided that these negative eigenvalues are relatively small. In the above example, however, this assumption appears hard to justify, because the one negative eigenvalue (=−5.57) is rather large.

Why would one even want to assume that dissimilarity data are Euclidean distances (except for an error component)? The justification must come from the way the data are generated or collected. If persons are asked directly for ratings on pairwise dissimilarities, then it may be plausible to hypothesize that the observed numerical responses are at least distance-like values. Data as in Tables 2.1 and 2.3 could, therefore, be scaled directly using classical MDS. The procedure will show to what extent the data are indeed Euclidean distances.

Correlations as in Table 1.1, however, are definitely not Euclidean distances, but rather scalar products *by construction*. Thus, in this case, one should skip steps 1 and 2 in the above, and begin directly with step 3. This amounts to running a principal component analysis. An alternative approach is to first convert the scalar products to distances. In case of correlations, this conversion is $d_{ij} = \sqrt{2 - 2r_{ij}}$.

In case of larger errors (as in the example above), classical MDS quickly reaches its limits as a useful method. It generates a best-possible solution, but it does so minimizing a criterion known as *Strain* which is not as easily interpretable as Stress. Moreover, in most applications, the data are at best on an interval scale level. Hence, one would not want to interpret the data directly as distances, but rather allow for an optimal re-scaling when mapping them into distances.

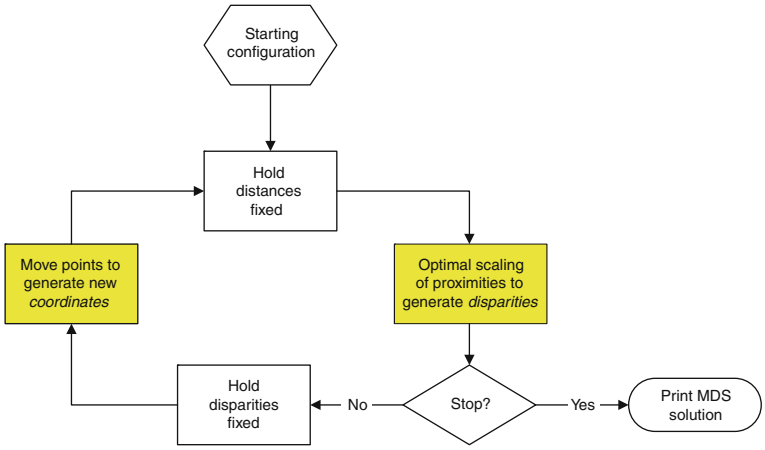


**Fig. 8.1** Principles of an iterative MDS algorithm

## 8.2 Iterative MDS Algorithms

Iterative MDS algorithms are more flexible than classical MDS. They find a Stress-optimal MDS configuration and, in doing so, they re-scale the data optimally within the constraints of their scale level. However, iterative algorithms cannot guarantee to always find the global optimum solution, because their small-step improvements may get stuck in local minima. The user, therefore, should keep an eye on this possibility (see p. 63f for suggestions on how to avoid local minima solutions).

Iterative MDS algorithms proceed in two phases (see Fig. 8.1). In each phase one set of parameters (distances or disparities, respectively) is taken as fixed values, while the other set of arguments is modified in such a way that Stress is reduced:

1. The disparities (i.e., the admissibly transformed proximities) are fixed; the points in MDS space are moved (i.e., $\mathbf{X}_t$ is changed to become $\mathbf{X}_{t+1}$) so that the distances of $\mathbf{X}_{t+1}$ minimize the Stress function.
2. The MDS configuration, $\mathbf{X}$, is fixed; the disparities are re-scaled within the bounds of their scale level so that the Stress function is minimized (*optimal scaling*).

If, after $t$ phases, this ping-pong process does not reduce the Stress value by more than some fixed amount (e.g., 0.0005) the search algorithm is stopped and $\mathbf{X}_t$ is taken as the optimal solution.

Phase 1 amounts to a difficult mathematical problem with $n \cdot m$ unknown parameters, the values of $\mathbf{X}$. To solve it, various optimization algorithms have been developed. The presently best algorithm is the SMACOF procedure (De Leeuw and Heiser 1980; Borg and Groenen 2005), because it guarantees in practical situations

that the iterations will converge to at least a *local* Stress minimum.[2] Other criteria can also be used to assess the quality of MDS algorithms (Basalaj 2001).

Phase 2 poses a relatively easy problem. In interval MDS, one solves the problem via linear regression. It finds the additive and multiplicative coefficients that linearly transform proximities into disparities such that Stress is minimized for the given distances. For other MDS models, appropriate regression procedures are also available (e.g., monotone regression for ordinal MDS).

These issues are purely mathematical ones. Users of MDS need not be concerned with them. They should simply use MDS programs like drivers use their cars: Drivers have to know how to drive, but they do not have to understand the physics of combustion engines. Drivers, however, should also know how to run a car (e.g., making sure that it has enough gas), and MDS users must feed the programs properly and set the right options to get where they want to go, i.e. to obtain optimal solutions.

An important option is picking a good starting configuration. All MDS programs offer a few alternatives that users can try out to see if they all lead to the same solution. PROXSCAL, for example, allows its users to repeat the MDS process with many different *random* starting configurations, or pick a particular *rational* starting configuration (e.g., one that results from using classical MDS),[3] or use an *external* user-constructed starting configuration.

We recommend to always actively influence the choice of the starting configuration rather than leaving it to the MDS program to construct such a configuration internally. Often, a good choice is using a starting configuration constructed on substantive-theoretical grounds. One example is the configuration in Table 6.1 as a starting configuration when scaling the rectangle data in Table 2.3. If such an external configuration can be formulated, one should at least test it out in case the MDS program does not arrive at the expected solution with its internal options.

Depending on the particular MDS program, various "technical" options are always offered to MDS users. These options can strongly impact the final MDS solution, because they often prevent the algorithm from terminating its iterations even though Stress can be further improved. In the GUI window of SYSTAT's MDS program shown in Fig. 1.5, for example, the user can set the maximum number of iterations and define a numerical criterion of convergence. For historical reasons (i.e., to save time and costs), the default values for these parameters are universally set much too defensively in all MDS programs so that the iterations are terminated too early. Users should set these parameters such that the program can do *as many iterations as necessary* to reduce Stress (see Sect. 7.4, p. 67). Computing time is not an issue with modern MDS programs.

---

[2] SMACOF is an acronym for "Scaling by MAjorizing a COmplicated Function" (De Leeuw and Heiser 1980). The optimization method used by SMACOF is called "Majorization" (De Leeuw 1977; Groenen 1993). The basic idea of this method is that a complicated goal function (i.e., Stress within the MDS context) is approximated in each iteration by a less complicated function which is easier to optimize. For more details on how this method is used to solve MDS problems, see De Leeuw and Mair (2009) or Borg and Groenen (2005).

[3] Such options are sometimes called KRUSKAL, GUTTMAN, YOUNG or TORGERSON, depending on their respective inventors or authors (see also Figs. 1.5, 9.8 and Sect. 7.5).

## 8.3 Summary

If the data are Euclidean distances (apart from error), classical MDS is a convenient algebraic method to do MDS. It converts the data to scalar products, and then finds the MDS configuration by eigen-decomposition. Iterative algorithms are more flexible: They allow optimal re-scalings of the data, and different varieties of Minkowski distances, not just Euclidean distances. Such programs begin by defining or using a starting configuration, and then modify it by small point movements reducing its Stress. The distances of this configuration are then used as targets for an optimal re-scaling of the data (thereby generating disparities) within the bounds of the data's scale level. This process of modifying the MDS configuration (with fixed disparities) and re-scaling the disparities (with fixed distances) is repeated until it converges. The presently best algorithm for moving the points is Smacof; re-scaling the data is done by regression.

## References

Basalaj, W. (2001). *Proximity visualisation of abstract data*. Unpublished doctoral dissertation, Cambridge University, U.K.

Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling* (2nd ed.). New York: Springer.

De Leeuw, J. (1977). In J. R, Barra, F. Brodeau, G. Romier & B. van Cutsem (Eds.), *Recent developments in statistics* (pp. 133–145). Amsterdam: North Holland.

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

De Leeuw, J., & Mair, P. (2009). Multidimensional scaling using majorization: SMACOF in R. *Journal of Statistical Software, 31*(3), 1–30.

Groenen, P. J. F. (1993). *The majorization approach to multidimensional scaling: Some problems and extensions*. Unpublished doctoral dissertation, University of Leiden, Rapenburg.

# Chapter 9
# Computer Programs for MDS

**Abstract** Two modern programs for MDS are described: PROXSCAL, an SPSS module, and SMACOF, an R-package. Commands and/or GUI menus are presented and illustrated with practical applications.

**Keywords** PROXSCAL · PREFSCAL · SMACOF · SMACOFSYM · SMACOFINDDIFF · SMACOFRECT · SMACOFCONSTRAINT · ALSCAL · PERMAP

In this chapter we turn to computer programs for MDS. Such programs are contained in all major statistics packages. In SPSS there are even two MDS modules. No single MDS program is generally superior to all others, and none offers all MDS models discussed in this book. Most MDS programs allow the user to do both ordinal MDS and also interval MDS. Some can also handle the INDSCAL model or varieties of this model. Few offer confirmatory MDS that allows the user to impose additional geometric restrictions onto the MDS solution. Only one, PERMAP, offers the possibility to directly interact with the program dynamically.

## 9.1 PROXSCAL

The MDS program that may be accessible to most users and that also offers many MDS models together with technically up-to-date solution algorithms is PROXSCAL. It is one of the two MDS modules in SPSS. PROXSCAL contains all of the popular models (ratio MDS, interval MDS, ordinal MDS; INDSCAL and related models; weights for each proximity; a variety of different starting configurations; numerous options for output, plots, and saving results), but also some forms of confirmatory MDS (using external scales, enforcing axial regions). However, all MDS models in PROXSCAL offer only Euclidean distances; no Shepard plots are generated (only related plots such as transformation plots); and unfolding is cumbersome to run.[1]

[1] For unfolding, we recommend a specialized program, called PREFSCAL, which is also a module of SPSS.

I. Borg et al., *Applied Multidimensional Scaling*, SpringerBriefs in Statistics,
DOI: 10.1007/978-3-642-31848-1_9, © The Author(s) 2013

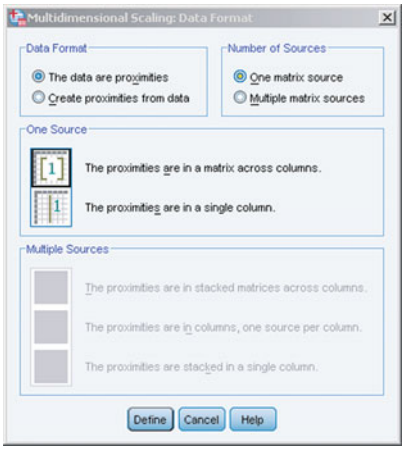


**Fig. 9.1** Starting menu of PROXSCAL

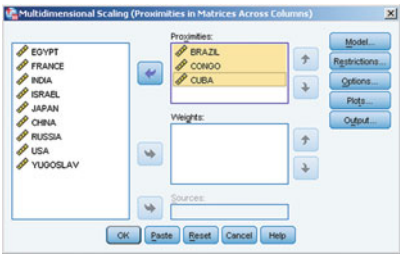


**Fig. 9.2** Cardinal GUI menu of PROXSCAL

The user can interact with PROXSCAL via graphical menus or via commands. Menus are sufficient for most users. Moreover, they may be easier to use for beginners, and they print out the commands for later usage when applications need to be more fine-tuned and better documented. In the following, we look at both ways to use the program.

### *9.1.1* PROXSCAL *by Menus*

The starting menu of PROXSCAL (in SPSS 20) is shown in Fig. 9.1 for the example discussed in Sect. 2.2. The program assumes that the user already imported the data of Table 2.2 into SPSS so that a file with this data matrix exists. This file is to be analyzed with MDS. Hence, no proximities have to be generated within Spss. The user, therefore, checks the button in the upper left-hand corner, informing the program that the data are proximities.

If one begins with the usual "person × variable" data matrix of a social scientist, proximities must first be generated. PROXSCAL offers a few options for doing this if one checks the button "Create...". However, other modules in SPSS are usually better suited for generating proximities (e.g., intercorrelation routines). In this case, one first stores the proximities in some file, and then opens this file for MDS with PROXSCAL.

The remaining options in Fig. 9.1 are relevant only if one has more than just one data set, e.g. in case of INDSCAL modeling or if one has replicated proximities. If so, one can stack the $k$ proximity matrices in an $(k \cdot n) \times n$ matrix as shown in Table 9.1 for a stack of three $3 \times 3$ proximity matrices. In order to keep track of the data,

**Table 9.1** Stacking proximity matrices into one super-matrix

| SourceID | V1 | V2 | V3 |
|---|---|---|---|
| 1 | 0.0 | | |
| 1 | 2.3 | 0.0 | |
| 1 | 3.2 | 1.7 | 0.0 |
| 2 | 0.0 | | |
| 2 | 2.2 | 0.0 | |
| 2 | 2.9 | 2.1 | 0.0 |
| 3 | 0.0 | | |
| 3 | 3.2 | 0.0 | |
| ... | ... | ... | 0.0 |

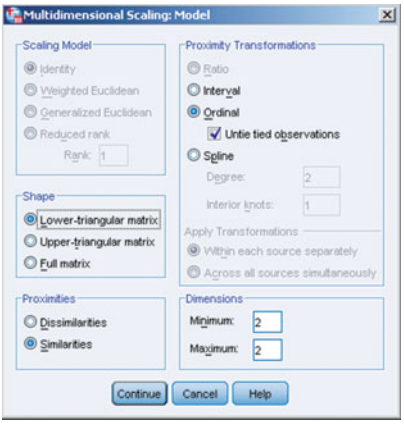


**Fig. 9.3** Window with important model specification and data definitions for an MDS analysis of Table 2.1

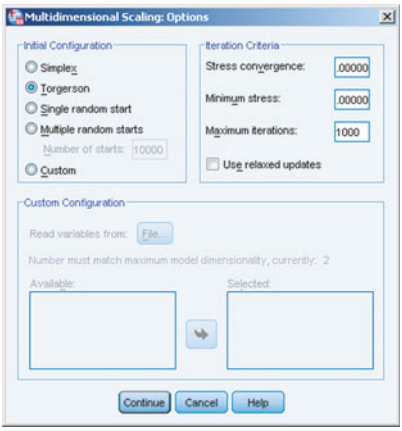


**Fig. 9.4** Important non-default specifications of the options of PROXSCAL

an additional variable is needed (here called "SourceID") that denotes the different matrices.

Figure 9.2 shows the main menu of PROXSCAL. In the upper right-hand corner, you find a set of buttons that call options for running an MDS analysis of the given proximities. The most important ones are subsumed under the Model button. If you check this button, the menu in Fig. 9.3 appears.

For the data of Table 2.2, we have to check in the lower left-hand corner of the menu in Fig. 9.3 that the data are Similarities. (The default setting is Dissimilarities. If you forget to set this properly, an MDS solution is computed that makes no sense and that has a very high Stress value. Sometimes, one notices only then that something must have been misspecified.)

The menu, moreover, offers the user to specify the type of regression that the MDS program should use. For our example data, we specify that we want ordinal MDS, with the primary approach to ties ("untie").

Then, in the lower right-hand corner, we specify the dimensionality of the MDS solutions. The default settings lead to just one 2-dimensional solution. If you want higher dimensionalities as well, simply change Maximum to a higher value (e.g., 6). Note that unidimensional scaling solutions tend to have many local minima. Therefore, it is not recommended to set Maximum to 1 unless precautions are taken against local minima such as multiple random starts.

In the Shape box, we inform the program about the format of the proximity matrix. In our example, we have a lower triangular matrix, as shown in Table 2.1. Note though that PROXSCAL assumes for this specification that the main diagonal exists, as in Table 9.1! Since the values in the main diagonal are not relevant for MDS, these diagonal elements can be filled with any values.

The box in the upper left-hand corner of Fig. 9.3 is relevant only if you have more than 1 proximity matrix. If so, the option Weighted Euclidean yields an INDSCAL solution. In case of replicated data that are to be mapped into one distance each, you choose Identity.

Finally, the options of the PROXSCAL algorithm should be changed, because their defaults often lead to nonoptimal MDS solutions. Figure 9.4 shows how the options need to be set. First, change the initial configuration to Torgerson, that is, the classical scaling solution discussed in Sect. 8.1. Then, use stricter iteration criteria by setting Stress convergence and Minimum stress to 0.000001 or smaller and Maximum iterations to at least 1,000.

Leaving the rest of the buttons in this menu on their default settings, we can return to the cardinal menu in Fig. 9.2 via the "Continue" button. There, we click on "OK", and PROXSCAL will generate an MDS solution.

We now show how to formulate external restrictions on the dimensions of an MDS solution via the PROXSCAL menus. To demonstrate this, we use the rectangle data in Table 2.3, setting all values in the main diagonal to 0. In the cardinal menu in Fig. 9.2, we click on Model to get to the menu that offers options on how the data should be transformed. In this menu (see Fig. 9.3) we inform the program that the data are dissimilarities; that they are stored in a lower triangular matrix; and that we want to run ordinal MDS with the primary approach to ties. Continue brings us back to the cardinal menu.

In the cardinal menu, we click on Restrictions. This brings us to the menu in Fig. 9.5. There, in the center of the window, we click on File and type the name of the SPSS file that contains the external scales (formulated as shown in Table 6.1) into the space to the right of this button. Then, in the box on the left-hand side, we pick the variables that should serve as external scales, that is, "Width" and "Height". Finally, we request in the lower right-hand corner that these scales should be interpreted as ordinal scales and that the secondary approach to ties (keep ties) is to be used by the program.

We also want to use an external starting configuration for the MDS of the rectangle data. The window in Fig. 9.6 shows how to read this into PROXSCAL. We check Custom and write the name of the file with the external starting configuration (as shown in Table 6.1) into the window in the middle of the menu (here:"C:\Documents a..\rectangle_design.sav") that contains the design configuration of the

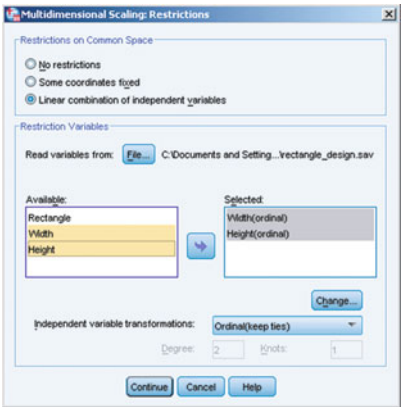


**Fig. 9.5** External scales for confirmatory MDS that yields Fig. 6.1

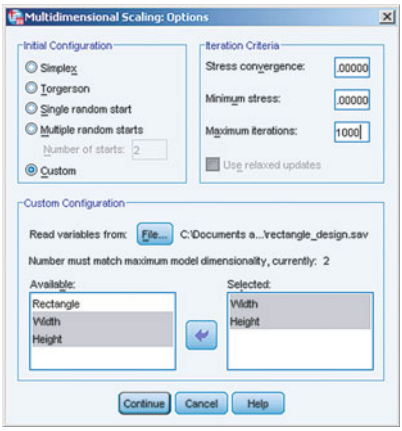


**Fig. 9.6** Reading an external starting configuration into PROXSCAL

rectangles. Its values are the physical coordinates of the rectangles used in the experiment. For the starting configuration, we select the variables "Width" and "Height" for the $X$- and $Y$-coordinates. With these specifications, the program yields the solution shown in Fig. 6.2.

### *9.1.2* PROXSCAL *by Commands*

These options (and more, like the spline transformation) can be set within SPSS syntax. PROXSCAL prints these syntax commands so that beginners can learn the syntax by using the menu. For example, the SPSS syntax that goes with the data of Table 2.2 and that is specified by the GUI menus of the previous subsection is given by the following:

```
PROXSCAL VARIABLES=BRAZIL CONGO CUBA EGYPT FRANCE
INDIA ISRAEL JAPAN CHINA RUSSIA USA YUGOSLAV
  /SHAPE=LOWER
  /INITIAL=TORGERSON
  /TRANSFORMATION=ORDINAL(KEEPTIES)
  /PROXIMITIES=SIMILARITIES
  /ACCELERATION=NONE
  /CRITERIA=DIMENSIONS(2,2) MAXITER(1000) DIFFSTRESS
            (.000001) MINSTRESS(.000001)
  /PRINT=COMMON STRESS
  /PLOT=COMMON.
```

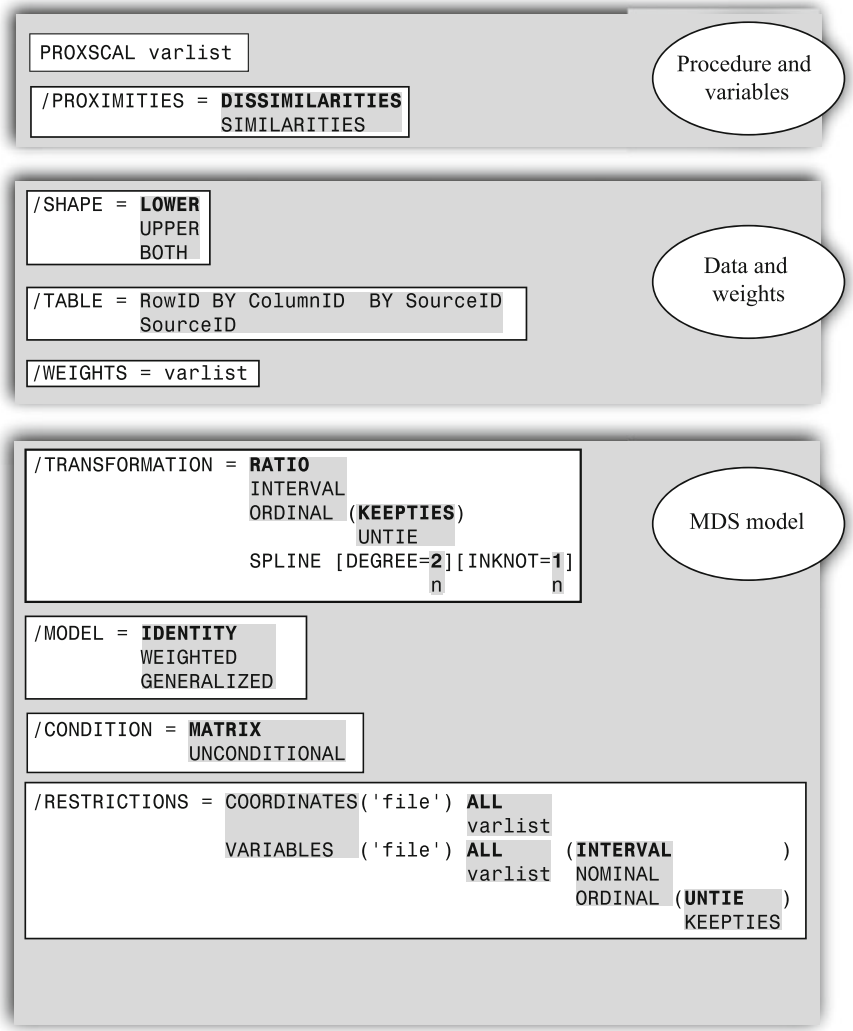


**Fig. 9.7** PROXSCAL subcommands and keywords (1 of 2). All subcommands beginning with "/" are optional: Default settings are printed boldface. Keywords stacked in *grey fields* are alternatives

The subcommands of PROXSCAL are presented in Figs. 9.7 and 9.8. Most keywords are self-explanatory, but they are also described more explicitly below.[2]

[2] Note that PROXSCAL has *two* default presets: (1) If the proximities are dissimilarities, then ratio MDS is the default transformation; for similarities, interval MDS is the default. (2) For ordinal MDS, the secondary approach to ties (KEEPTIES) is preset. Most other MDS programs (like SYSTAT, MDSX, or STATISTICA) use ordinal MDS with the primary approach to ties as their default MDS model.

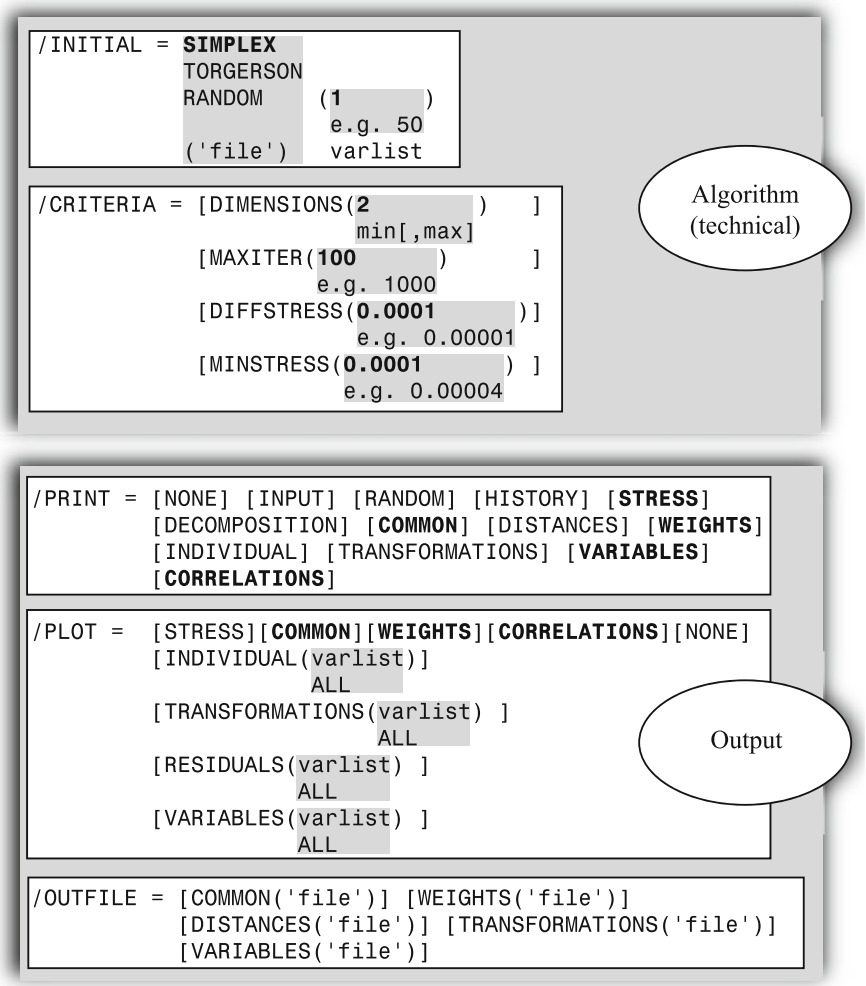


**Fig. 9.8** PROXSCAL subcommands and keywords (2 of 2)

### Procedure and Variables

- PROXSCAL should be followed by a list of variables that indicates where the proximities can be found.
- /PROXIMITIES specifies whether the data are DISSIMILARITIES (default) or SIMILARITIES.

### Data and Weights

- /SHAPE specifies the form of the data matrix. Note that in case of LOWER or UPPER the main diagonal must exist (but it can contain *any* values, including missing values). BOTH takes the symmetric part of the data matrix.
- /TABLE allows the user to specify the proximities $p_{ij}$ in a column. For sorting these values into a proximity matrix, the program needs two extra variables that specify the row index $i$ and the column index $j$ of $p_{ij}$. In case of three-way MDS, an additional third variable is needed to indicate to which matrix $k$ ("Source") the proximity belongs (see Fig. 5.5). The value labels of the row/column/source index variables are used to label the points in the plots.
- /WEIGHTS allows specifying nonnegative weights for the proximities. For example, weights can be used to indicate how much you trust each proximity. If your proximity matrix consists of 15 variables (V1 TO V15, then you specify /WEIGHTS = V1 TO V15. The weights are then found at variables V1 TO V15. If the proximities are in a single column (through the /TABLE subcommand), then the weights should also be in a single column. Thus, the weights take the same form as the proximities.

### MDS Model

- /TRANSFORMATION sets the admissible transformation of the proximities.
- /MODEL specifies how to process more than just one data matrix. In the IDENTITY model, all proximity matrices are represented by a single MDS configuration. This option can be useful if the proximity matrices are mere replications of each other. The WEIGHTED model (INDSCAL) consists of a single common configuration whose columns (dimensions) are weighted differently for each proximity matrix $k$. GENERALIZED computes the IDIOSCAL model that not only includes individual dimension weightings but also individual rotations of the common space before dimension weighting.
- /CONDITION gives the conditionality of the transformations. In case of more than one proximity matrix, MATRIX indicates that every matrix receives its own optimal transformation. Under UNCONDITIONAL there is only a single transformation that is used for all proximity matrices. For example, for UNCONDITIONAL interval MDS, the transformations for all proximity matrices have the same intercept and slope.
- /RESTRICTIONS lets you restrict the coordinates of the MDS configuration. The COORDINATES keyword allows you to specify variables whose non-missing values are used as fixed coordinates. The keyword VARIABLES makes the MDS coordinates a linear combination of the variables you specify. Note that these variables can also be transformed. To really restrict the coordinates it is better to choose only a few variables and have a strict transformation such as interval. With too much freedom (too many variables or free transformations), the restricted solution will not differ much from the unrestricted MDS solution.

### Algorithm (Technical)

- /INITIAL specifies the type of start configuration. SIMPLEX is default and does one iteration in a high dimensional configuration with all distances between the points being the same, followed by a reduction of the space to the dimensionality specified by the maximum-number-of-dimensions criterion. This solution is used to start the iterations. The SIMPLEX choice often leads to local minima with high Stress values. It is better to select TORGERSON that computes a classical MDS solution often leading to solutions with good Stress values. RANDOM(n) computes *N* MDS solutions each starting with a different random start configuration and reports the solution with the lowest Stress. For small MDS matrices (say up to 30 objects) one can easily compute 10,000 random starts.
- /CRITERIA specifies the required dimensionalities and sets the stopping criteria of the iterative algorithm. DIMENSIONS(DMIN,DMAX) specifies the minimal and maximal dimensionality of the MDS solutions. Note that PROXSCAL shows the table of coordinates for the lowest dimensionality. The coordinates for other dimensionalities can be obtained by opening the table and selecting the dimensionality you want. The default of DIFFSTRESS (`.0001`) is may not strict enough to ensure stopping at a local minimum. Therefore, it is is better to set DIFFSTRESS (`.000001`) to make the program stop if the change in Stress of two subsequent iterations is less than `.000001`. MAXITER (`1000`) sets the maximum number of iterations to 1,000. MINSTRESS (`.0001`) stops the algorithm whenever the Stress is below `.0001`.

### Output

- /PRINT offers various print options: INPUT prints the matrix of proximities, RANDOM prints the random number seed and Stress value of each random start (in case you use random starts), HISTORY shows the sequence of Stress values over the iterations, STRESS gives different variants of Stress, DECOMPOSITION prints the decomposition of Stress for every object and every source (can be informative to find outliers), COMMON prints the coordinates of the common space (**X**), DISTANCES prints the distances between the points (one per source), WEIGHTS prints the weights for INDSCAL models, TRANSFORMATIONS prints the matrix of the disparities; if VARIABLES was specified on the RESTRICTIONS subcommand, then VARIABLES prints the transformed variables of the external starting configuration, plus their regression weights, and CORRELATIONS prints the correlations between the external scales and the MDS dimensions.
- /PLOT controls the diagrams of PROXSCAL. The default plots are the MDS configuration (COMMON) and the dimensions weights (WEIGHTS) in case of INDSCAL. Other plots are scree plots of Stress-versus-dimensionality (STRESS), distances-versus-disparities (RESIDUALS), and proximities-versus-disparities (TRANSFORMATION). Shepard-diagrams are (so far) not available in PROXSCAL. You can also request plots of the INDIVIDUAL spaces in case you used individual differences scaling, or

transformation plots of the external VARIABLES specified on the variable list (if you used external variables/scales).
- /OUTFILE allows you to save results to a file. The coordinates are saved by the COMMON keyword, the weights for a weighted or generalized Euclidean three-way MDS by WEIGHTS, the distances by DISTANCE, the pseudo distances or d-hats by TRANSFORMATIONS, and for restricted MDS, the transformed variables by VARIABLES.

## 9.2 The R Package SMACOF

R (R Development Core Team 2011) is a programming language as well as a statistical software environment.[3] R is available for free on Comprehensive R Archive Network (CRAN). The base package implements basic statistical and mathematical methods and functions. It can be extended by various packages that offer additional methodologies.

To install the base package, the following steps need to be carried out:

- Go to http://CRAN.R-project.org.
- Use the link "Download and Install R".
- Specify the operating system (OS) of your computer: R runs under MS Windows, Mac OS, and various Linux distributions.
- Then, follow the remaining download instructions and install R.

R provides efficient handling of vectors and matrices. A key feature of R is that outputs of statistical analyses are stored as R objects such as lists or matrices. The user can access these objects for further processing (very useful, in particular, for simulation studies). R also provides a powerful plot engine that allows for flexible customization of graphical output in publication quality. R is Open Source and issued under the GNU Public License (GPL), so the user has full access to the source code.

In order to work efficiently with R, an appropriate editor is required. There are several good editors available; we suggest RSTUDIO; see http://rstudio.org and Verzani (2011).

We also recommend the R-package RCMDR (Fox 2005) as a GUI for doing basic statistics and data manipulations in the R environment (see below on how to install R packages).

[3] As introductory books we suggest Venables and Smith (2002) (general introduction), Dalgaard (2008) and Everitt and Hothorn (2009) (introductory statistics with R).

### 9.2.1 General Remarks on SMACOF

The SMACOF package (De Leeuw and Mair 2009) is available on CRAN. It implements several MDS models which we introduce in the following sections. After launching the R console, the `smacof` package (as all other R packages as well) can be installed as follows:

```
R> install.packages("smacof")
```

The package installation needs to be done only once, unless you update the R version. Each time the R console is launched, the package needs to be loaded into working memory.

```
R> library("smacof")
```

At this point all functions implemented in `smacof` are available to the user. For a general package overview, the line

```
R> help("smacof")
```

opens the (HTML based) package documentation.

The `smacof` package provides the following MDS methods (including the corresponding function names):

- `smacofSym()`: Simple MDS on a symmetric dissimilarity matrix
- `smacofIndDiff()`: MDS for individual differences scaling (3-way MDS)
- `smacofRect()`: Unfolding models
- `smacofConstraint()`: Confirmatory MDS
- `smacofSphere.primal()` und `smacofSphere.dual()`: Spherical MDS

In general, an R function has various arguments which allow for corresponding parameter settings. Some of them are set to default values. As an example, consider the argument `metric` which is contained in all `smacof` functions. If `metric = TRUE` (default value), interval MDS is computed; if `metric = FALSE`, ordinal MDS is performed.

Basically, all `smacof` functions require dissimilarity matrices as input, except `smacofRect()` (for further details see below). If the data are given as a "person $\times$ variable" matrix (here denoted as `M`), a corresponding dissimilarity matrix can be computed using the `dist(M)` command. Using the `method` argument, different distance measures can be chosen. The default setting is the Euclidean distance.

There are several ways to import data into R. If the data are stored in EXCEL, SPSS, SYSTAT or similar formats, the `foreign` package can be considered which provides various utility functions. For EXCEL files in particular, it is suggested to save the spreadsheet as a csv file and then use the command `read.csv()` to import it into R. This function uses several default settings which the user may have to change depending on the Excel configuration. For instance, the following specification

```
read.csv(file, header = TRUE, sep = ",", ...)
```

implies that the first line contains the variable names and the variables are separated by a comma.

An SPSS file (here called `XYZ.sav`) can be imported directly using `read.spss("XYZ.sav")` from the `foreign` package. If the file is not located in the R working directory, the user can specify a path such as `read.spss("c:/data/XYZ.sav")`.[4]

The results of R functions are typically stored as objects that belong to a certain class. In `smacof`, the resulting objects belong to the class `smacof` and certain sub-classes. For each of these classes, methods for representing the output and for plotting MDS results are provided. As an example for objects of the class `smacof`, there is the `plot.smacof()` method which is called by using the `plot()` command (see example on p. 100). It allows plotting the MDS configuration (argument `plot.type = "confplot"`; set as default), `residuals` (`plot.type = "resplot"`), Shepard diagrams (`plot.type = "Shepard"`), and stress decompositions (`plot.type = "stressplot"`).

The following subsections focus on computing some MDS variants implemented in the `smacof` package.

### *9.2.2 SmacofSym*

Let us start with symmetric SMACOF for simple ordinal MDS. As an example, we use the Wish data of Table 2.1. These data are already contained in the `smacof` package and, therefore, they can be loaded using

```
R> data("wish")
```

If the data are similarity values, as is true for the Wish data, they first need to be transformed into dissimilarity values. The `smacof` package provides a utility function called `sim2diss()` with the `method` argument. For this example, we set `method = 7` which means that all similarities are converted into dissimilarities by subtraction from 7:

```
R> wish.new <- sim2diss(wish, method = 7)     ## convert similarities
R> wish.new                                   ## dissimilarities as input
         BRAZIL CONGO CUBA EGYPT FRANCE INDIA ISRAEL JAPAN CHINA RUSSIA  USA
CONGO      2.17
CUBA       1.72  2.44
EGYPT      3.56  2.00 1.83
FRANCE     2.28  3.00 2.89  2.22
INDIA      2.50  2.17 3.00  1.17   3.56
ISRAEL     3.17  3.67 3.39  2.33   3.00  2.89
JAPAN      3.50  3.61 4.06  3.17   2.78  2.50   2.17
CHINA      4.61  3.00 1.50  2.61   3.33  2.89   4.00  2.83
RUSSIA     3.94  3.61 1.56  2.61   1.94  2.50   2.83  2.39  1.28
USA        1.61  4.61 3.83  3.67   1.06  2.72   1.06  0.94  4.44  2.00
YUGOSLAV   3.83  3.50 1.89  2.72   2.28  3.00   2.56  2.72  1.94  0.33  3.44
```

[4] For Windows user it is important to note that R always requires forward slashes when quoting a path.

This matrix of dissimilarities is assigned as an argument to the function `smacofSym()`.[5]

```
R> res.wish <- smacofSym(delta = wish.new, metric = FALSE)
             ## do MDS
```

The results are stored in the object `res.wish`. Some basic information can be accessed by just typing in the name of the object:

```
R> res.wish                    #Basic Output
```

This command prints out the following information[6]:

```
Call: smacofSym(delta = wish.new, metric = FALSE)

Model: Symmetric SMACOF
Number of objects: 12

Nonmetric stress: 0.0349866
Number of iterations: 59
```

All relevant SMACOF outputs are stored as single objects within the output list. The names of the list elements can be obtained by the `names()` command.

```
R> names(res.wish)              #Output Objects
 [1] "delta"     "obsdiss"   "confdiss"  "conf"      "stress.m"  "stress.nm"
 [7] "spp"       "ndim"      "model"     "niter"     "nobj"      "metric"
[13] "call"
```

The elements are:

- `delta`: Dissimilarity matrix
- `obsdiss`: Proximities (normalized dissimilarity matrix)
- `confdiss`: Proximities computed from the MDS solution
- `conf`: Configuration (coordinates) of the MDS solution ($\mathbf{X}$)
- `stress.m`: Metric stress in case of interval MDS
- `stress.nm`: Non-metric stress in case of ordinal MDS
- `spp`: Stress per point
- `ndim`: Number of dimensions
- `model`: Type of SMACOF model (e.g. `SmacofSym`)
- `niter`: Number of iterations
- `nobj`: Number of objects
- `metric`: The value of the `metric` argument (e.g. `FALSE`)
- `call`: The full `smacof` call

[5] The data do not need to be stored as an object of class `dist`. They can also be provided as a symmetric matrix, alternatively.

[6] Note that SMACOF reports *squared* Stress-1 values. So, `Nonmetric stress: 0.0349866` is actually equal to *Stress-1* $= 0.187$.

As always in R, such list outputs can be accessed using the `$` operator. For example, the fitted dissimilarity matrix can be accessed using

```
R> res.wish$confdiss
```

or, rounded to two decimal digits, by

```
R> round(res.wish$confdiss, 2)

          BRAZIL CONGO CUBA EGYPT FRANCE INDIA ISRAEL JAPAN CHINA RUSSIA  USA
CONGO       0.95
CUBA        1.24  0.70
EGYPT       0.81  0.47 0.44
FRANCE      0.82  1.12 0.88  0.67
INDIA       0.51  0.56 0.74  0.30   0.63
ISRAEL      0.78  1.45 1.35  1.06   0.49  0.89
JAPAN       1.24  1.66 1.31  1.20   0.54  1.16   0.54
CHINA       1.60  1.16 0.47  0.85   1.00  1.12   1.49  1.27
RUSSIA      1.40  1.37 0.79  0.91   0.61  1.05   1.02  0.69  0.60
USA         1.00  1.60 1.41  1.18   0.54  1.05   0.23  0.35  1.48  0.95
YUGOSLAV    1.44  1.33 0.72  0.90   0.67  1.06   1.11  0.80  0.50  0.11  1.05
```

As mentioned above, this dissimilarity matrix is now an R object and can be used for further computations or to produce plots.

The SMACOF package offers several relevant plots. A simple plot of the resulting configuration can be produced using

```
R> plot(res.wish, main = "SMACOF Configuration", xlab = "Dimension 1",
+   ylab = "Dimension 2", xlim = c(-0.9, 0.7), asp = 1,
+   label.conf = list(TRUE, 2, 1), type = "p", pch = 16)
```

R offers numerous arguments for customizing plots (see `help(plot)` and `help(par)`). In the above command, we set a plot title using the `main` argument; specify labels for the $X$- and $Y$-axis using `xlab` and `ylab`, respectively; use `xlim` to specify the range of the $X$-axis of the plot; `asp=1` defines an aspect ratio of 1 which implies that both dimensions are scaled in the same way; and `label.conf` says that the points are to be labeled (`TRUE`), with labels to the left (`=2`) of the points, and in black (`=1`) color. The two remaining arguments of the above `plot()` function are shown in their default settings: `type="p"` requires a plot of points (not `"l"` for lines, for example); and `pch=16` chooses a particular kind of solid point as the symbol for the points in the plot. The resulting configuration plot is shown in Fig. 9.9.

Some important parameters of `smacofSym` models and subsequent MDS variants are the following:

- Number of dimensions: Default is `ndim = 2`; depending on the dimensionality of the solution, this argument can be set correspondingly (from 1 up to the number of variables).
- MDS type: Default is `metric = TRUE`, i.e. an interval MDS is computed; `metric = FALSE` leads to ordinal MDS.
- Weight matrix: Default is `weightmat = NULL`; if the user wants to assign weights, a weight matrix can be provided through this argument.

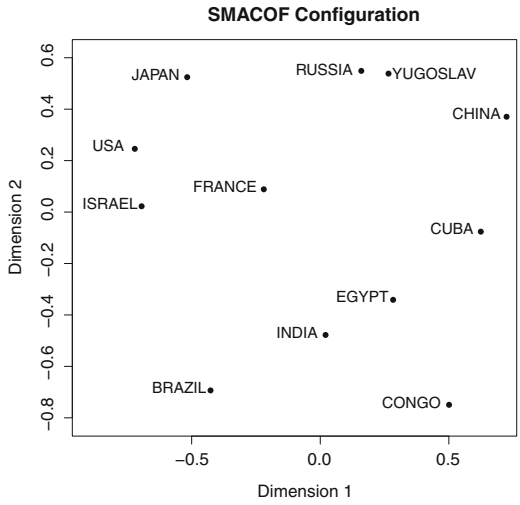


**Fig. 9.9** SMACOF configuration for country similarity data

- Starting configuration: Default is `init = NULL` (Torgerson scaling); through this argument an optional starting configuration can be specified.
- Maximum number of iterations: Default is `itmax = 1000` which can be modified by the user.
- Convergence criterion: Default is `eps = 1e-6`; it can be decreased in order to increase the precision.
- Stress printout: Default `verbose = FALSE`, i.e. only the final stress value will be printed; `verbose = TRUE` prints the stress for each iteration.

Additional arguments are explained in the corresponding help files.

### *9.2.3 SmacofIndDiff*

In this section we focus on MDS modeling of individual differences. The corresponding R function is `smacofIndDiff()`. Using the `constraint` argument, various restrictions can be imposed onto the configurations. Examples are the INDSCAL and the IDIOSCAL models from Sect. 5.5. To illustrate the computation of these models in R, we use the data from Table 7.2, i.e. correlations of 13 working values in former East and West Germany, respectively. In R, the underlying data are organized as a list of length two. Each list element consists of a correlation matrix.

Let us assume that we need to import the data from two separate csv files. Subsequently, they have to be organized as a list. The following steps are required:

```
east <- read.csv("east.csv")
west <- read.csv("west.csv")
EW <- list(east, west)
```

However, these data are already contained in the `smacof` package. They have the following structure:

```
R> data("EW_eng")

R> EW_eng
$east
      int. ind. resp. sens. advc. resp. help usef. other secu. pay spare safe
int.  .47
ind.  .43  .53
sens. .38  .31   .39
advc. .28  .27   .32   .20
resp. .37  .34   .42   .33   .43
help  .29  .23   .38   .38   .19   .37
usef. .28  .25   .38   .44   .25   .39   .48
other .27  .28   .41   .29   .15   .29   .49   .32
secu. .16  .25   .24   .24   .39   .37   .16   .23   .16
pay   .15  .16   .16   .13   .52   .29   .10   .16   .11   .40
spare .21  .15   .09   .08   .27   .21   .14   .18   .10   .18   .27
safe  .28  .26   .25   .33   .34   .35   .26   .30   .19   .38   .29  .25
$west
      int. ind. resp. sens. advc. resp. help usef. other secu. pay spare safe
int.  .51
ind.  .42  .57
sens. .37  .30   .33
advc. .28  .29   .33  .18
resp. .18  .23   .34  .24   .43
help  .20  .19   .31  .33   .17  .32
usef. .20  .17   .28  .40   .18  .37   .56
other .31  .34   .39  .31   .21  .24   .43   .34
secu. .14  .17   .18  .19   .39  .37   .24   .25   .17
pay   .20  .26   .25  .05   .54  .32   .05   .08   .11   .32
spare .25  .22   .13  .09   .19  .30   .13   .18   .19   .16   .30
safe  .32  .31   .23  .37   .25  .20   .25   .23   .24   .33   .16  .23
```

Since these values are correlations, they first need to be converted to dissimilarities. Again, we use the `sim2diss()` function where `method = "corr"` is the default value. Each correlation $r$ is thereby transformed to $\sqrt{1-r}$. We then compute an INDSCAL solution and an IDIOSCAL solution, respectively, for these data:

```
R> eastwest <- lapply(EW_eng, sim2diss)
R> res.indscal <- smacofIndDiff(delta = eastwest, constraint = "indscal")
R> res.idioscal <- smacofIndDiff(delta = eastwest, constraint = "idioscal")
```

For both models we can produce the configuration plots in a single plot device:

```
par(mfrow=c(1,2))            #2 plots, figure 9.10
plot(res.indscal, main = "INDSCAL Configuration",
+     xlab = "Dimension 1", ylab = "Dimension 2", asp = 1,
+     xlim = c(-1, 1), ylim = c(-1, 1), type = "p", pch = 16,
+     label.conf = list(TRUE, 1, 1))
plot(res.idioscal, main = "IDIOSCAL Configuration",
+     xlab = "Dimension 1", ylab = "Dimension 2", asp = 1,
+     xlim = c(-1, 1), ylim = c(-1, 1), type = "p", pch = 16,
+     label.conf = list(TRUE, 1, 1))
```

The argument `xlim` scales the $X$-axis, `ylim` the $Y$-axis. The resulting plots (the configurations are highly similar for both models) are shown in Fig. 9.10.

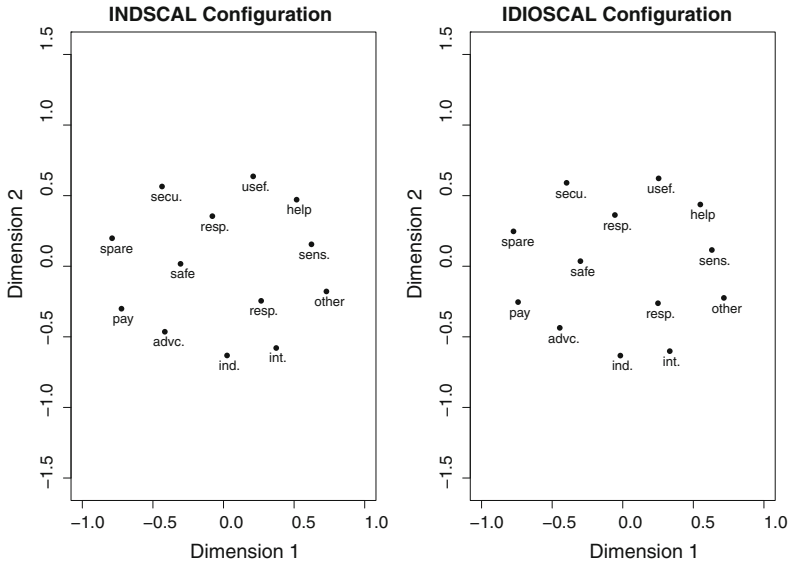


**Fig. 9.10** INDSCAL and IDIOSCAL configurations for work values

### *9.2.4 SmacofRect*

Unfolding models can be computed by means of the function `smacofRect()` (rectangular SMACOF). As mentioned above, the input data for unfolding is not a dissimilarity matrix but rather a $n \times k$ matrix of preferences. To give an example, we use the artificial data from Table 5.1.

```
R> data("partypref")
R> partypref
   A B C D
1  1 3 2 4
2  4 2 3 1
3  4 1 2 3
4  4 1 3 2
5  3 2 1 4
```

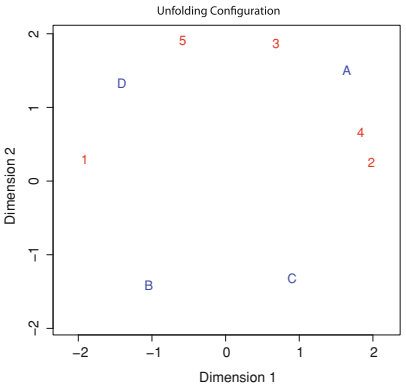


**Fig. 9.11** Unfolding solution for party-preferences data

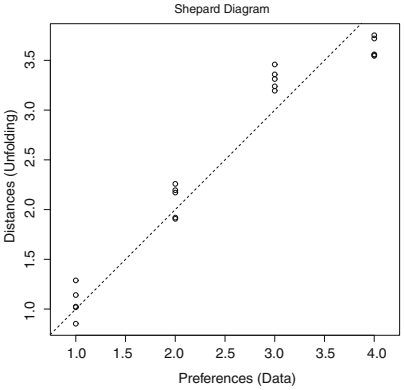


**Fig. 9.12** Shepard diagram of party-preferences unfolding solution

Note that for this SMACOF variant the data must be stored as a "matrix" and not, as above, as an "object" of class `dist`. For the party-preferences data, we need to take into account that the larger a value, the larger the preference for a party. The function `smacofRect()` interprets preference values in the opposite way, i.e., 1 for the strongest preference and, in this example, 4 for the weakest. Therefore, we first transform the data matrix as follows:

```
R> partypref_rev <- 5 - partypref
```

After this conversion we compute the unfolding model. As graphic output, we plot the joint configuration of persons and objects in a common space (Fig. 9.11). Furthermore, a Shepard diagram is produced (Fig. 9.12).

```
R> res.rect <- smacofRect(delta = partypref_rev)
R> par(mfrow = c(1, 2))
##fig 9.11
plot(res.rect, joint = TRUE, main = "Unfolding Configuration",
+  xlab = "Dimension 1", ylab = "Dimension 2", asp = 1)
##fig 9.12
plot(res.rect, plot.type = "Shepard", main = "Shepard Diagram",
+  xlab = "Preferences (Data)", ylab = "Distances (Unfolding)", asp = 1)
```

### 9.2.5 *SmacofConstraint*

Finally, we present an application of confirmatory MDS. We use the rectangle data from Table 2.3 with the corresponding external scales from Table 6.1. We restrict the **C** matrix to be diagonal, using the argument `constraint` which allows for several restrictions.[7]

[7] Note that SMACOF reports *squared* Stress-1 values. So, `Metric stress: 0.03265856` is actually equal to *Stress-1* = 0.181.

```
R> data("rectangles")
R> data("rect_constr")
R> res.constr <- smacofConstraint(rectangles, external = rect_constr,
+         constraint = "diagonal", metric = TRUE)
R> res.constr
```

```
Call: smacofConstraint(delta = rectangles, constraint
    = "diagonal", external = rect_constr, metric = FALSE)

Model: SMACOF constraint
Number of objects: 16

Metric stress: 0.03265856
Number of iterations: 13
```

Plots for the MDS configuration (forced into an orthogonal grid of points) and for the Stress per point (SPP) values of all rectangles, respectively, can be produced by the following commands:

```
par(mfrow = c(1, 2))
plot(res.constr, main = "Metric MDS (orthog.)", xlab = "Dimension 1",
+  ylab = "Dimension 2", asp = 1)
plot(res.constr, plot.type = "stressplot", main = "Stress per point",
+  xlab = "Rectangle", ylab = "Stress Contribution (\%)")
```

### *9.2.6 Final Remarks*

The sections above cover only a fraction of SMACOF's functions and options. Especially for confirmatory MDS, SMACOF allows for highly flexible specifications of constraints. Details can be found in the corresponding help files.

For all MDS variants, solutions with higher dimensionality can be computed. For 3D solutions, dynamic 3D configuration plots are available. The package also does MDS on the surface of a sphere using `smacofSphere.primal()` and `smacofSphere.dual()`, an option that is not that relevant for general data analysis.

Further technical details, more comprehensive descriptions of functions, options, and arguments, as well as additional examples can be found in De Leeuw and Mair (2009). In the following section we quote a few additional R packages for MDS computations.

## 9.3 Other Programs for MDS

PROXSCAL and SMACOF are not the only programs for MDS. All major statistics packages offer MDS modules. Examples are SYSTAT, STATA, SAS or XLSTAT, an add-in for EXCEL. Within R, the `MASS` package offers functions for classical MDS.

Other packages (such as `vegan` (Oksanen et al. 2007), `labdsv` (Roberts 2006), `ecodist` (Goslee and Urban 2007), and `ggobi` in R (Maechler 2005)) have built-in functions for ordinal and interval MDS. MDS for individual differences can also be done with `SensoMineR` (Husson and Le 2007). SPSS offers even a second MDS module, called ALSCAL, a program that is older than PROXSCAL.

Not all of these MDS programs offer different MDS models. This makes it impossible for the users to scale their data with different MDS models in order to check the effects of such choices (e.g., in case of degeneracy). Also, with some of the MDS programs, it is not clear whether they always converge to local Stress optima. If convergence is not guaranteed, then the MDS program's algorithm must be stopped after $k$ iterations by using ad-hoc criteria (e.g., if $k$ is greater than 50, say, or if Stress is not being reduced by more than 0.005; see Fig. 1.5), but it cannot be forced into a local minimum by setting very demanding stopping criteria.

Large mathematics packages such as MATLAB also offer MDS, although only in statistics modules that have to be bought additionally. However, they allow users to program certain MDS analyses themselves. For example, in MATLAB, Procrustean problems can be solved with a few commands, and publication-ready graphics can be generated and modified interactively in WYSIWYG.

Besides such commercial software for MDS, there also exist some freeware programs. A particularly interesting example is PERMAP, which is also the only truly interactive MDS program. In PERMAP, one can, for example, use the mouse to move single points on the computer screen to a different location: The program then immediately responds by dynamically shifting the points of this "starting configuration" into optimal positions. This is great for actually seeing how the iterative processes works. It also allows one to easily check if manually re-locating single points always ends up with the same final configuration.

For further information on these and other MDS programs, including detailed descriptions and comparisons, see Borg and Groenen (2005). Data sets for MDS applications and web addresses for MDS programs can be found on Patrick Groenen's website, http://people.few.eur.nl/groenen/.

## References

Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling* (2nd ed.). New York: Springer.

Dalgaard, P. (2008). *Introductory statistics with R* (2nd ed.). New York: Springer.

De Leeuw, J., & Mair, P. (2009). Multidimensional scaling using majorization: SMACOF in R. *Journal of Statistical Software, 31*(3), 1–30.

Everitt, B. S., & Hothorn, T. (2009). *A handbook of statistical analyses using R*. Boca Raton, FL: Chapman & Hall.

Fox, J. (2005). The R commander: A basic-statistics graphical user interface for R. *Journal of Statistical Software, 14*, 1–42.

Goslee, S., & Urban, D. (2007). Ecodist: Dissimilarity-based functions for ecological analysis (R package version 1.1.3).

Husson, F. & Le, S. (2007). Sensominer: Sensory data analysis with R (R package version 1.07).

Maechler, M. (2005). Xgobi: Interface to the xgobi and xgvis programs for graphical data analysis (R package version 1.2-13).
Oksanen, J., Kindt, R., Legendre, P., O'Hara, B., Henry, M., & Stevens, H. (2007). Vegan: Community ecology package (R package version 1.8-8).
R Development Core Team. (2011). R: A language and environment for statistical computing. Vienna. (ISBN 3-900051-07-0).
Roberts, D. W. (2006). Labdsv: Laboratory for dynamic synthetic vegephenomenology. (R package version 1.2-2).
Venables, W. N., & Smith, D. N. (2002). *An introduction to R*. Bristol, UK: Network Theory Ltd.
Verzani, J. (2011). *Getting started with RStudio: An integrated development environment for R*. Sebastopol, CA: O'Reilly Media.

# Author Index


**A**
Alderfer, C. P., 66, 71

**B**
Basalaj, W., 85
Bilsky, W., 32
Borg, I., 9, 16, 29, 30, 32, 42, 44, 46, 49, 53, 57, 65, 71, 72, 78, 84, 85, 106
Braun, M., 66, 71
Buja, A., 75
Busing, F. M. T. A., 46

**C**
Carroll, J. D., 42, 46
Chang, J. J., 42
Cox, M. A. A., 32
Cox, T. F., 32
Coxon, A. P. M., 31

**D**
Dalgaard, P., 96
De Leeuw, J., 84, 85, 90
Dichtl, E., 52
Dijksterhuis, G. B., 66
Domoney, D. W., 28

**E**
England, G., 31
Everitt, B. S., 96

**F**
Fox, J., 96

**G**
Garner, W. R., 14
Glushko, R. J., 30
Goslee, S., 106
Gower, J. C., 31, 32, 66
Graef, J., 23, 29
Green, P. E., 29
Groenen, P. J. F., 30, 32, 44, 46, 49, 56, 67, 84, 85, 90, 106
Guthrie, J. T., 63
Guttman, L., 16, 17, 68, 73, 74

**H**
Heiser, W. J., 46, 84


I. Borg et al., *Applied Multidimensional Scaling*, SpringerBriefs in Statistics,
DOI: 10.1007/978-3-642-31848-1, © The Author(s) 2013

**H** (*cont.*)
Horan, C. B., 43
Hothorn, T., 96
Husson, F., 106

**J**
Jones, C. L., 31

**K**
Kruskal, J. B., 10, 43

**L**
Le, S., 106
Leutner, D., 16
Levy, S., 16, 17
Lingoes, J. C., 44, 53, 57, 78
Liu, C., 7

**M**
Maechler, M., 106
Mair, P., 85, 97, 105

**N**
Nosofsky, R. M., 86

**O**
Ogilvie, J. C., 27
Oksanen, J., 106

**P**
Palmeri, T. J., 81

**R**
R Development Core Team, 96
Restle, F., 29
Roberts, D. W., 106
Rothkopf, E. Z., 54
Ruiz-Quintanilla, S. A., 31

**S**
Schönemann, P. H., 42, 44, 78
Shye, S., 72
Smith, D. N., 96
Spence, I., 23, 28, 29
Staufenbiel, T., 29, 46

**T**
Thurstone, L. L., 30
Torgerson, W. S., 13, 38
Tversky, A., 44, 60

**U**
Urban, D., 106

**V**
Van der Lans, I., 56
Venables, W. N., 96
Verzani, J., 104
Viken, R. J., 79

**W**
Wind, Y., 29
Wish, M., 11, 43, 74

# Subject Index


**A**
Almost equal proximities, 75
ALSCAL, 106
Asymmetric proximities, 40
Axial partition, 54
Axial restriction, 54

**C**
Centroid configuration, 44
City-block distance, 13
Classical MDS, 81
CMDA, 53
Co-occurrence data, 31
Common space, 42
Computer programs for MDS, 2, 4, 105
Confirmatory MDS, 56
Convergence, 85, 106
Coordinate axes, 11
Coordinate matrix, 10

**D**
Data
  asymmetric, 41
  correlations of crimes, 1
  similarity of countries, 10
  similarity of rectangles, 16
  similarity of work values, 65
  unfolding, 45
Data smoothing, 70
Design configuration, 16
Dimension
  interpretation of, 4, 71
  salience of, 43
Disparities, 22
Distance
  as the crow flies, 15, 39
  axioms of, 12
  Euclidean, 12, 39, 83
  Minkowski, 15
Distance formula
  as a model of judgment, 12
  Minkowski, 13
Disturbing points, 75
Dominance metric, 13

**E**
Euclidean distance, 13
External scale, 71

**F**
Facet
  nominal, 72
  ordered, 72
  unordered, 72
Facet diagram, 72
Feature model, 29
Fit index, 21
Flattened weights, 78

**G**
Geometry
  curved, 39
  Euclidean, 14
  natural, 14
Gravity model, 32, 34

**I**
Ideal points, 45


I. Borg et al., *Applied Multidimensional Scaling*, SpringerBriefs in Statistics,
DOI: 10.1007/978-3-642-31848-1, © The Author(s) 2013

**I** (*cont.*)
Incidence matrix, 31
Individual space, 42
Internal scale, 70
Interpretation
  dimensional, 11, 73
  of clusters, 73
  of directions, 71
  of regions, 71
  regions vs. clusters, 73
Intra-dimensional difference, 16
Iterations, 2, 61

**J**
Jaccard index, 32

**L**
Local minimum, 61
Loss function, 21

**M**
MatLab, 106
MDS
  asymmetric, 40
  confirmatory, 49
  exploratory, 49
  for exploring data structures, 9
  for testing structural hypotheses, 16
  for visualization of data, 7
  individual differences, 42
  interval, 38
  metric, 37
  non-metric, 37
  ordinal, 37
  purpose of, 7
  ratio, 39
  three-mode, 43
  weak confirmatory, 50
MDS algorithms
  iterative, 84
MDS model
  Idioscal, 42
  Indscal, 42
  and scale level of data, 37
  dimensional salience, 43
  dimensional weighting, 42
  drift vector, 40
  metric, 38
  strong, 39, 65
  subjective metrics, 42
MDS models, 47
MDS solution
  degenerate, 39, 63
  goodness of, 21
  over-fitted, 39
Metric, 12
Minkowski distance, 13
Missing data in MDS, 28
Monotone regression, 21, 38

**N**
NewMDSX, 44
Noise in data, 70

**O**
Optimal scaling, 84
Over-interpretation
  of dimension weights, 78

**P**
Partition
  axial, 72
  modular, 72
  polar, 72
Partitioning, 17
Permap, 87, 106
Pindis, 44
Prefscal, 87
Procrustean transformations, 66
Proximities
  derived, 29
  direct, 27
  direction of, 66
  from co-occurrence data, 31
  from conversions, 30
  replicated, 39
Proxscal, 2, 51, 61, 87, 95
Pseudo data, 53
Psychological map, 11

**R**
R, 96
Radex, 73
Random noise in data, 28
Residuals, 22
Rotation, 11

**S**
Sas, 105
Shepard diagram, 21, 22, 68

Similarity
judgments of, 13
Similarity transformation, 68
Simple matching coefficient, 32
SMACOF, 84, 85, 96, 105
Space
Euclidean, 12
metric, 12
multi-dimensional, 13
SPP, 68
SPSS, 106
Starting configuration
random, 61, 85
rational, 61, 85
STATA, 105
Stress
and dimensionality, 24
and MDS model, 24
and missing data, 24
and noise in data, 24
and outliers, 25
and ties, 24
evaluating, 23, 26, 68
normalized, 25
S-Stress, 25
Stress-1, 25
total, 25
Stress per point, 69
Subject space, 43
Systat, 2, 61, 105

**T**
Termination criteria, 61
Thurstone's LCJ, 30
Ties in MDS, 54
Ties in ordinal MDS, 24
Torgerson scaling, 81
Torgerson-Gower scaling, 81
Transformation plot, 75
Triangle inequality, 12

**U**
Unfolding
model, 45
multi-dimensional, 46
multiple, 46

**X**
XLSTAT, 105