MULTISCALE MANUAL
(Extended Version)

J. O. Ramsay
McGill University

March 21, 1997

Contents

List of Tables

List of Figures

Chapter 1

Introduction, Installation and Operation

Throughout this manual it will be necessary to display material as it actually is entered into the computer or as it is output by MULTISCALE. Typewriter font is used for this purpose, a sample of which is as follows:

```
This is an example of typewriter font.
```

Some material displayed in this way is set aside as a Figure, and referred to by the number of the figure.

This manual presupposes a certain level of knowledge of both statistics and analytic geometry. Readers with very limited backgrounds in these areas should consider consulting references such as Schiffman, Reynolds and Young (1981) or Kruskal and Wish (1978).

Those already familiar with multidimensional scaling and interested in moving as quickly as possible to some basic analyses may want to scan through Sections 2.1, 2.2, and 2.9 in the next chapter, read Chapter 6, and browse through Chapter 7 before attempting to set up an analysis. They can return to a more in depth reading of these and other chapters subsequently.

1.1 Installation

MULTISCALE is distributed on a set of floppy diskettes (usually 3 1/2 inch high density), and to install the MS/DOS personal computer version it is sufficient to merely copy these diskettes into the appropriate directory using the DOS COPY or XCOPY command. For example, the following two DOS commands will create a directory in the hard disk drive c: called MULTISCL.DIR and copy a single distribution diskette in drive a: into that directory:

```
mkdir multiscl.dir
xcopy a: c:
```

After this step is completed, one can usually proceed immediately to the next section which explains how to run the MS/DOS version of MULTISCALE.

It will not be necessary to compile the program in order to execute the program. Note, however, that some distributed versions may require a math coprocessor. Most newer MS/DOS machines come equipped with a math coprocessor, and if one is not present, it would be highly desirable to install one, since MULTISCALE performs rather lengthy calculations for many problems.

1.1.1 HALO Versions:

Some distributed versions of MULTISCALE use the HALO Professional Graphics library to produce graphics output on the computer screen, a plotter using the HP-GL plotting language, a dot-matrix printer, or a laser printer processing the Adobe Postscript language. Use of such a version presupposes that a license has been obtained from Media Cybernetics to use this software. The HALO package should be obtained from Media Cybernetics before attempting to use MULTISCALE if the HALO version is the one being installed. The following HALO files should be accessible to the program either by being included in the directory containing MULTISCALE or having the directory appear in the DOS PATH command.

ahd102 .fnt
ahd104 .fnt
ahd106 .fnt
ahd201 .fnt
ahdbw8 .prt
ahdibme .dsp
ahdibmg .dsp
ahdibmv .dsp
ahdmsfl .krn

In addition, if HP-GL or Postscript plotting commands are to be produced, the following two HALO files must also be available:

ahdhp .exe
ahdps .exe

Other HALO files may also be necessary for certain types of computer and plotting hardware. The manuals accompanying the HALO package should be consulted for further details.

1.1.2 Systems other than MS/DOS:

If the program is to be run on a computer with a FORTRAN 77 compiler in an operating system other than MS/DOS, or even if it is to be used on MS/DOS systems, but it is desired to use a compiler other than Microsoft FORTRAN Version 5.0 or higher, than the program will have to be compiled from the source code. Moreover, the main or driver routine, in file MSCLMAIN.FOR, will have to be altered in order to use the program. Instructions on how to alter the program are including in the comment section of this file, and are as follows:

```
C  FOR THE UNIX VERSION, (1) SET LOGICAL DOSWRD TO .FALSE.,
C                        (2) COMMENT OUT THE DIMENSION STATEMENT MAKING
C                            INTEGER ARRAY Z ALLOCATABLE
C                            IN LINE 30
C                        (3) REMOVE COMMENT CHARACTER FROM DIMENSION
C                            STATEMENT IN LINE 31.  THE SIZE OF THE
C                            ARRAY Z MAY BE INCREASED IF NEEDED.
C                        (4) COMMENT OUT LINE 268 CALLING
C                            MICROSOFT FORTRAN NARGS STATEMENT
```

```
C                              (5) COMMENT OUT LINE 772 CALLING
C                                  MICROSOFT FORTRAN ALLOCATE STATEMENT
```

Although these instructions refer to the version for the Unix operating system, this version should run on most systems.

1.2 Executing MULTISCALE

The steps involved in carrying out a MULTISCALE analysis are as follows:

1. Use an editor to set up a file containing the data to be analyzed and information required by MULTISCALE to perform the analysis. An example is given in Chapter 6, and a complete discussion of the setup of a MULTISCALE input file is in Chapter 7.

2. Save this file with an appropriate file name. If the extension part of the file is `.msl` then invoking the program is particularly simple, as discussed below. However, any file name can be used.

3. Invoke MULTISCALE as described below.

4. MULTISCALE produces an output file that can be printed, editted, saved, or processed in any way desired. If graphics output to be processed on plotting equipment using the HP-GL plotting language of Hewlett-Packard plotters or the Adobe Postscript language of laser printers, another file will also be produced which may need further processing or directing the appropriate output port.

1.2.1 Invoking MULTISCALE in MS/DOS Systems

If step 1 above has been completed, the second step requires executing the program. There are two ways to do this:

Interactive: Simply enter the command

```
multiscl
```

and the program will prompt for an input file name and for an output file name. If, in addition, HP-GL or Postscript plotting commands are to be output, the program will prompt for a file name for storing these commands.

Processing `.msl` Files: If the input file is saved with the extension `.msl`, then the program can be invoked with the name of the stem of the file name. For example, the file set up for the data described in Chapter 6 is including on the distribution diskette as `funseekr.msl`. To analyze these data, enter the command

```
multiscl funseekr
```

and the program will look for this input file. Rather than prompting for the output file (and plotting file) the program will output the results to file `funseekr.out`.

On MS/DOS systems MULTISCALE uses the Microsoft FORTRAN ALLOCATE function to allocate the amount of memory needed for the problem at hand at execution time. On some problems the amount of available RAM memory available may not be sufficient, and an error message will result. In such cases it may be possible to reboot the system eliminating "terminate and stay resident (TSR)" software, often used for communications and notebook programs, to free up additional memory.

If the version calling HALO graphics routines is used, the program will attempt to load a font into RAM. At this point the program also may fail due to insufficient memory. Aside from the advice given above, one may also try graphics output with the PLOt-HPGl command (which will also produce screen graphics and can be processed to produce Postscript plots) since this uses a font requiring substantially less memory.

Further information on processing graphics output is given in Chapter 8.

1.2.2 Invoking MULTISCALE in Systems Other than MS/DOS

With the modifications indicated above to the source code of the driver or main routine, and after compilation and linking, MULTISCALE expects the input file to be the standard input file (usually file number 5 in FORTRAN languages) and the output to be the standard output file (usually file number 6). Operating systems vary in terms how these files are identified to the program. On Unix and many other systems, the appropriate command would be

```
multiscl <funseekr.msl >funseekr.out
```

assuming that the executable file had been given the name `multiscl`.

There is no provision for graphics output in non-MS/DOS versions of the program. The author can be consulted about possible versions for specific systems such as the Sun Sparcstation series.

Chapter 2

Modelling Dissimilarity Data

The purpose of this chapter is to introduce the essentials of multidimensional scaling to the beginner. An overview of the chapter is as follows:

- In Section 2.1 we discuss the characteristics of measures of dissimilarity. These closely parallel the mathematical properties of distance, and the goal of multidimensional scaling can be simply stated as the approximation of dissimilarities by distances. Some commonly used measures of dissimilarity are mentioned in Section 2.2.

- MULTISCALE works with Euclidean or straight-line distances, and Section 2.3 of this chapter delves into the properties of these distances, including questions about coordinate systems and metrics.

- The fit of distance to dissimilarity will generally improve if some degree of order-preserving or monotone transformation of one or the other is allowed. In Section 2.4 various transformation possibilities are discussed.

- Most sets of data have sources of variation which cannot be approximated by distances even after transformation; possible characteristics of this noise, residual, or error variation are discussed in Sections 2.5 and 2.6.

- MULTISCALE allows the user to impose linear constraint conditions on the coordinates of the points representing the stimuli, a topic discussed in Section 2.7.

- In the final section the multidimensional scaling problem is summarized compactly in mathematical notation.

2.1 Properties of Dissimilarity and Distance

Multidimensional scaling is based on an assumed relationship between the psychological concept of *dissimilarity* and the mathematical concept of *distance.* There are certain features that these two ideas have in common. Dissimilarity is the subjective unlikeness of two objects for a particular perceiver. The term *difference* is often used in casual communication, but because it also can mean something like the arithmetic difference between two numbers, we shall stick to the longer but less ambiguous term. The concept *similarity* has an inverse relation to dissimilarity: two very similar objects have little dissimilarity. The term *proximity* is often used instead of similarity. The relations between dissimilarity and similarity or proximity will be discussed more carefully subsequently.

The close connection between the intuitive or subjective characteristics of dissimilarity and the mathematical properties of distance which form the logical basis for multidimensional scaling. At the most basic level, both are properties of pairs of objects: dissimilarity describes a pair of stimuli and distance a pair of points. Even when we refer to a single thing as "very different" we usually have in mind a standard or reference stimulus to which it is being compared. The two concepts share the following features:

Origin: A judgment of no dissimilarity is applied to two identical stimuli. Likewise a distance of zero is assigned to two coincident points.

Positivity: When two stimuli are not identical in some relevant way, we to assume that they have a positive amount of dissimilarity. Likewise the distance between two noncoincident points is always positive.

Symmetry: Usually the order in which two stimuli are specified has little or no effect on their perceived dissimilarity, especially when the stimuli have the same level of generality: the dissimilarity of a Chevrolet and a Cadillac doesn't change when the names are reversed. Some problems may arise when the dissimilarity of a Chevrolet and a "typical French automobile" is judged, since the latter concept may depend on whether or not the former was considered first. Symmetry is also an essential property of distance, and for two points i and j the distances d_{ij} and d_{ji} are equal.

Consistency: The most important characteristic of dissimilarity is a requirement that three or more judgments of dissimilarity be consistent with one another in the following sense: Two stimuli each judged very similar to any third cannot be seen to be mutually very dissimilar. Otherwise something changed during the judgments: perhaps the criteria for judging were not the same for all judgments, or perhaps the judge was not able to judge consistently. Distances, too, obey an analogous internal consistency principle called the *triangle inequality.* In considering points i, j, and k it is always true that

$$d_{ij} \leq d_{ik} + d_{jk}.$$

This inequality says the same thing as the statement above about dissimilarity judgments since d_{ij} cannot be large if both d_{ik} and d_{jk} are small.

A supplementary property of dissimilarity, not essential in general but is necessary to use MULTISCALE, is the property of *magnitude.* Dissimilarities must be quantified at some level that reasonably approximates a continuous scale of measurement.

The goal, then, of multidimensional scaling is to generate a set of distances which correspond to or approximate a set of dissimilarities. Consider the simple example in Figure 2.1. Of the three faces presented at the top of the figure, the first and third are much more similar than either the first and second or the second and third. This relation can be represented in the display at the bottom of the figure where the second face has been moved away from the other two so that its distance from each of them corresponds its dissimilarities with respect to them.

A rather more challenging example is provided in Figure 2.2. Here a particular subject judged every one of the 105 possible dissimilarities among a set of 15 forms of recreation. He recorded his judgments by choosing among 25 categories presented to him for each pair presented as follows:

```
Museum and Hockey
```

Figure 2.1: The three faces above are rearranged below so that the distances among them correspond to the perceptual dissimilarities among them.

```
- - - - - - - - - - - - - - - - - - - - - - - - -
Very                                         Very
Similar                                 Different
```

These categories were subsequently numbered from 1 to 25, and the position of the chosen category used as a measure or index of his perceived similarity for that pair. The order of pairs was carefully randomized, but Figure 2.2 contains the judgments sorted into tabular form. Note that, because of the assumed symmetry of judgments and because dissimilarities for identical pairs were assumed to be zero, only the lower right triangle excluding the diagonal of the table is presented.

Figure 2.3 displays a set of points arranged on a plane so that the distance between any two points corresponds as closely as possible to the corresponding entry in Figure 2.2. For example, it can be seen that the points corresponding to hockey and ballet are far apart in the display and that the corresponding index of dissimilarity is 25. On the other hand, the points for ballet and theatre are close together and the corresponding dissimilarity value is only 6. Just how one achieves this arrangement is, of course, what multidimensional scaling in general and MULTISCALE in particular is all about.

2.2 Some Types of Dissimilarity Measures

Before discussing the mathematics of distance in more detail, it is worth reviewing some of the ways that dissimilarity measures arise in experimental and field situations, and what some of their characteristics are. In many cases, the measures available are actually similarity or proximity measures, and in such cases a preliminary transformation is necessary to convert them to dissimilarity measures for analysis by MULTISCALE. The following list of dissimilarity or similarity measures is not at all exhaustive, but mentions those commonly used.

Direct Ratings: With human subjects who can be counted upon to give reasonably careful judgments, some form of direct rating of dissimilarity is natural and efficient. The most popular choice is probably the ordered category scale such as displayed in the previous chapter and in

Figure 2.2: Ratings of Dissimilarity for 15 Recreations

	Concert	Museum	Theatre	Movie	TV	Confernce	Reading	Hockey	Ballet	Debate	Fash sho	Doc film	Exhibitn	Win shop
Concert														
Museum	16													
Theatre	3	18												
Movie	12	12	11											
Watch TV	16	21	16	2										
Conference	20	10	19	15	12									
Reading	15	12	13	9	19	6								
Watch hockey	21	23	23	19	7	22	20							
Ballet	7	10	6	18	19	25	15	25						
Political debate	19	22	25	22	14	8	22	23	25					
Fashion show	9	7	13	15	12	19	20	22	8	25				
Documentary film	22	16	16	19	13	7	13	15	23	13	25			
Exhibition	7	3	13	12	21	13	10	22	13	12	7	18		
Window shopping	21	22	22	12	23	21	18	18	21	22	9	22	12	
Restaurant	8	8	7	9	21	21	12	22	5	25	9	23	10	8

Figure 2.3: Fifteen forms of recreation as represented by a MULTISCALE analysis of the recreation dissimilarity data.

Figure 2.2. One of the extreme categories can be labelled "very similar" while the other might be labelled "very different." It is then a simple matter to number the categories so that the numbers become indices of dissimilarity. In cases where the indices from 1 to n are in similarity order, so that 1 really means "very different", it is usually sufficient to subtract the indices from $n+1$ to produce corresponding dissimilarity indices. Note that it is wise in general to use a large number of categories. It has been observed in practice and shown statistically (Ramsay 1973) that reducing a potentially continuous variable to less than 7 categories results in a substantial loss of precision in estimating a location on the scale. It should also be noted that subjects vary considerably in their ability or desire to make judgments in a graduated manner. Some subjects never use more than two or three of the categories provided. Their data inevitably contain less information and, in extreme cases, may pose serious computational problems. These problems notwithstanding, direct ratings can be obtained rapidly and cheaply from a wide range of individuals.

Normal adult subjects can make about 100 such judgments per hour, although fatigue and boredom become serious problems beyond an hour. The wise investigator encourages the subject to respond carefully and to take his time. Some preliminary exposure to the task is often important as well. Takane (1981) has developed a technique specifically appropriate when the number of categories is small (say less than 7).

Other forms of direct rating include techniques such as ratio production (Ramsay, 1968), magnitude estimation, placing check marks on a continuous line, and the use of physical response devices.

Rankings of Dissimilarity: A relatively little used but potentially useful form of data are the rankings of a stimulus in terms of its dissimilarity with respect to each of a number of other stimuli. Takane and Carroll (1982) have developed an approach to data of this sort.

Confusion or Transition Frequencies: In studies involving animals or in psychophysical studies one can often collect the number or proportion of times a stimulus is associated with a particular response. These are actually proximity measures, and a well known example is the Rothkopf (1957) morse code confusion data analyzed by a number of authors

(Kruskal, 1964; Shepard, 1962). Among the possible transformations which can convert such data to dissimilarity indices are the reciprocal of the square root of the frequencies, and one minus the confusion proportions. Asymmetry is usually apparent in such data, and any index will have a correlation with other indices even when the approximating distances are subtracted out. Shepard (1957) and Nakatani (1972) have techniques specifically appropriate to such data.

Co-occurrence Frequencies: Closely related to confusion frequencies, co-occurrence frequencies are the number of associations where the upper limit on such associations is not known or experimentally controlled. For example, one might count the number of times residents of one city move to another city within a fixed period. Again asymmetry is common in such data, and multidimensional scaling as realized in MULTISCALE can only hope to approximate the symmetric components of variation. Again some order-reversing transformation to dissimilarity form is necessary.

A variant on such data arises when each subject is asked to sort the stimuli into a number of piles or groupings. The frequency with which two stimuli occur in the same grouping then is a measure of similarity. The number of groups may or may not be left to the subject to determine. The advantage of such data is that a very large number of stimuli can be judged at one time.

Dissimilarities from Multivariate Observations: When each stimulus has associated with it a variety of measurements, it may seem natural to compute a coefficient of distance between pairs by viewing each measure as defining a dimension or axis in a Euclidean space. Unfortunately, the problem of choosing a metric for evaluating distance (see the next chapter for more details) is seldom easy to resolve, and there are better techniques such as principal components analysis better suited to representing the objects in a space of reduced dimensionality.

Table 2.1 summarizes the notation that has been used to this point.

Table 2.1: Symbols Introduced in this Section

Symbol	Meaning
I	Number of stimuli or objects
i, j	Indices of particular stimuli
R	Number of subjects or replications
r	Index of a particular subject
$d_i j$	Dissimilarity observed for stimuli i and j

2.3 Types of Distances and Metrics

The distance between two points can be defined in many ways that satisfy the origin, positivity, symmetry, and triangle inequality conditions. For example, assigning a distance of one to all points which are not coincident is a perfectly valid rule and corresponds to binary same-different dissimilarity judgements. However, multidimensional scaling requires a more discriminating measure, and most programs are based on the familiar "straight line" or Euclidean distance measure. From a numerical point of view the definition and computation of Euclidean distance requires a *coordinate system.* This is a rule for locating points in space. The coordinate system is in fact a function which assigns to each point M numbers, called the coordinates of the point. The number M is called the *dimensionality* of the space. It represents the amount of information required to locate each point in space. Euclidean distance then requires another function which takes a pair of M-tuples and from them computes a single nonnegative number having the distance properties. What makes the distance Euclidean as opposed to something else is the way in which Euclidean distances vary or do not vary with respect to certain transformations of the points. These additional features will be reviewed below after we have considered some possible coordinate systems.

It is important to understand both the role of a coordinate system and that many alternative coordinate systems are possible for a given configuration of points. A coordinate system can function simply as a convenient numerical procedure for locating points in space and for computing the distances among them. That is, once the points are located and distances computed, the coordinate system can be discarded or ignored. For example, a multidimensional scaling program such as MULTISCALE uses a particular coordinate system during the course of its computations, but in most applications this coordinate system is best discarded when the final results are computed.

A coordinate system may also have a substantive interpretation and thus form part of the descriptive system used to explain the results. In selecting an interpretive coordinate system it is helpful to be aware of some of the possibilities. Three of these are as follows.

Orthogonal Cartesian Coordinates: This is the most familiar system and is illustrated in Figure 2.4. One begins by choosing M directions in space which are mutually at right angles or *orthogonal.* In addition an arbitrary point in space is specified as the origin of the system. From this origin one can draw lines or axes which go in these mutually orthogonal directions. These reference lines are called *dimensions.* Each point is located by computing the location of the perpendicular projection of the point on each axis. The number of such axes defines the dimensionality of the space. Of course, it is difficult to plot points with respect to more than two dimensions, and very difficult to even visualize points when more than three dimensions are involved. However, this limitation in our capacity for visual imagery has nothing to do with mathematics, and in principle any number M of dimensions is possible.

The distance d_{ij} between two points i and j is particularly simple to compute in this system. If the points are located by coordinates $(x_{i1}, \ldots, x_{iM})$ and $(x_{j1}, \ldots, x_{jM})$, respectively, then distance is computed according to the equation

$$d_{ij}^2 = \sum_{m=1}^{M} (x_{im} - x_{jm})^2. \tag{2.1}$$

According to this rule we use the "root-sum-square" procedure for combining the distances $|x_{im} - x_{jm}|$ between points with respect to each dimension.

It will be very useful to modify the distance formula (2.1) so as to permit different weights or saliences to be applied to different dimensions. We

Figure 2.4: An orthogonal Cartesian Coordinate system for determining the locations of two points.

do this as follows:

$$d_{ij}^2 = \sum_{m=1}^{M} w_m (x_{im} - x_{jm})^2. \tag{2.2}$$

The coefficient w_m defines the sensitivity to variation along axis or dimension m. When w_m is large, a fixed difference $(x_{im} - x_{jm})$ has a greater impact on the size of d_{ij}^2 in (2.2) than if this weighting coefficient is small.

The set of coefficients w_m is referred to as the *metric* of the Cartesian coordinate system. When we are using these differential weights, we shall say that we are using the *diagonal metric* for distance.

Polar or Spherical coordinates: Points can also be located by choosing a single fixed direction and then specifying one or more angles of rotation from this direction and a distance from the origin. Specifically, the polar coordinate system is the two dimensional system in which a point is located in terms of number of degrees from the horizontal and the displacement or radius from the origin. Figure 2.5 shows an example of how points representing 21 standard colors in the Munsell system can be located by either an orthogonal Cartesian system or a polar coordinate system. In this instance, the polar coordinate system is the more natural from an interpretive point of view since hue can be defined in terms of angle and saturation in terms of distance from the origin. More generally, a spherical coordinate system for M dimensions requires $M-1$ angles and a single radius. Although for most purposes a Cartesian coordinate system is computationally more convenient, a spherical system does have one important advantage: a rotation of a configuration of points about the origin is simply a matter of adding constants to one or more of the angles.

Hybrid systems are possible: one plane in an M-dimensional system can be represented in polar coordinates and the remaining $M-2$ dimensions in Cartesion coordinates. Such systems are called cylindrical. There are also elliptical coordinate systems which are analagous to oblique Cartesian coordinates.

What defines a distance measure as Euclidean is not the coordinate system, in spite of the popular tendency to confuse Euclidean distance with

Figure 2.5: A representation of 21 colours in the Munsell system by Cartesian and polar coordinate systems.

orthogonal Cartesian coordinates. Rather, Euclidean distance is defined by an invariance with respect to two kinds of transformations of the points and coordinate system: translations or changes of origin and rotations. Both of these transformations leave the angles between points and dimensions unchanged. A distance measure is not Euclidean if it is not invariant with respect to these transformations, or if it is invariant with respect to other transformations.

2.4 Individualized Metrics

The most important reason for considering the diagonal metric distance model (2.2) is the possibility that subjects may generate dissimilarities which are best fit by allowing the metric to vary from subject to subject. This naturally implies that distances for any particular pair of points will vary from subject to subject according to the values of the weights w_m. MULTISCALE permits this possibility.

Let us consider more closely how to interpret individualized metrics. Figure 2.6 contains a two-dimensional configuration in terms of its coordinates with respect to the identity metric matrix. That is, distance between points is to be computed using equation (2.1). Also in Table 2.6 are two other metrics, both of which are diagonal.

Any direction in space can be interpreted as a point of view with respect to the stimuli. The extent of the spread of the points in this direction defines the variation of the stimuli from that point of view. In particular, a direction that is used as a dimension for locating points corresponds to just such a point of view. According to this interpretation, the square root of the elements in the diagonal of a metric matrix defines the sensitivity to variation from this point of view. This is so because a particular coordinate difference defined with respect to a standard reference coordinate system is multiplied by this coefficient in computing distance within this individual's metric. If, as is the case for the first dimension for the first person, the coefficient is larger than unity, variation in this direction contributes more heavily or tends to increase distance relative to the standard metric. Conversely, a small value in the diagonal corresponding to a dimension results in less contribution by variation in this direction.

We can see this in Table 2.6 if we examine the distances between points

Figure 2.6: Coordinates for Four Points and Two Diagonal Metrics

Configuration

	Dim 1	Dim 2
Point 1	-1	1
Point 2	1	1
Point 3	1	-1
Point 4	-1	-1

Metric 1

Dim 1	Dim 2
4	.25

Metric 2

Dim 1	Dim 2
.25	4

1 and 2, which vary only with respect to the first dimension. In the identity metric the distance is 2 units. But for individual 1 the distance is 4 units since he is twice as sensitive ($w_1 = 4$) to variation in this direction, while for individual 2 the distance is only 1 unit since he is half as sensitive ($w_2 = .25$). On the other hand, the relative sensitivities to the second dimension are reversed, and the distances between points 2 and 3 are 1 and 4, respectively.

We can graphically depict each coordinate system by plotting a line for each dimension which extends a unit distance out from the origin. If a metric is diagonal, these lines will be orthogonal to each other. The length of a line is inversely proportional to the corresponding diagonal entry in the metric matrix; a unit distance for sensitive directions being smaller than for less sensitive directions. The first two figures in Figure 2.4 show the relation between metric weights and this graphical representation for the two sets of metric weights in Figure 2.6. (Disregard the final figure in Figure 2.4).

The possibility of allowing the metric to vary from subject to subject makes psychological sense and may enrich the interpretation of results as well as substantially improve the fit of data. Wish and Carroll (1974) discuss a number of examples where variation in diagonal metrics led to interesting insights.

We saw in Section 1.3 that the coordinate system was defined only to

Figure 2.7: A representation of three sets of metric weights by drawing axes of unit length, with an angular orientation corresponding to the off-diagonal entries.

within a translation and rotation. Therefore MULTISCALE must impose additional constraints on the coordinate system used to compute distances. These constraints are

$$\sum_{i=1}^{I} x_{im} = 0, \;\; m = 1, \ldots, M$$

$$\textstyle\sum_{i=1}^{I} x_{im} x_{in} = 0, m \neq n. \tag{2.3}$$

The first equation implies that the origin for the coordinate system is at the centroid or center of the configuration. The second equation implies that the configuration is in *principal axis* orientation; that is, the first dimension will have the greatest possible variability in its coordinates, the second the greatest possible variability given the first, and so on. This second constraint cannot be applied when the metric is diagonal, however, since distances computed with this model are not invariant under rotations.

In addition to constraining the coordinates by equations (2.3) it is also necessary to constrain the dimension weights w_{rm} since multiplying these by a constant could be compensated for by dividing the coordinates by the square root of that constant. An additional constraint is also necessary to define the across-subject average of these diagonal entries. Thus MULTISCALE imposes the following two constraints:

$$\sum_{m=1}^{M} w_{rm}^2 = M, \;\; r = 1, \ldots, R,$$

$$\textstyle\sum_{r=1}^{R} w_{rm}^2 = R, \;\; m = 1, \ldots, M. \tag{2.4}$$

If the weights are arranged in an R by M matrix $\mathbf{W}$, these equations imply that both the row and column average squares are one.

In order for the distance properties to hold, it is necessary that the weights w_{rm} be positive. MULTISCALE achieves this by setting a lower bound of approximately 0.01 on these diagonal entries.

The mathematical concepts used introduced in this section are summarized in Table 2.2.

2.5 Some Transformations of Dissimilarity

As we have seen, many different indices of dissimilarity are possible. By contrast, Euclidean distance is very rigidly defined, even given the latitude

Table 2.2: More Symbols Introduced in this Section on Coordinate Systems

Symbol	Meaning
M	Number of dimensions for computing distance
m, n	Indices of particular dimensions
x_{im}	Coordinate of point i for dimension m
$\mathbf{X}$	I by M matrix of coordinates x_{im}
x_i	Row i of $\mathbf{X}$
w_m	Weight for dimension m
w_{rm}	Weight for dimension m and subject r

in choice of metric. Often in practice a particular dissimilarity measure and the best-fitting distances are related curvilinearly. Figure 2.8 shows a plot for the dissimilarity data in Figure 2.2 against distances generated from the configuration displayed in Figure 2.3. This chapter examines ways in which the fit can be "tuned" by estimating a transformation of the dissimilarity data to provide a more linear relationship to distance.

When a transformation improves the fit of distances to dissimilarities, the nature or degree of curvilinearity of this transformations usually varies from subject to subject. Thus, separate transformations must be estimated for each subject in the analysis.

The transformation problem is often considerably simpler after applying a logarithmic transformation to both the dissimilarities and the distances. Since the logarithmic transformation is an order-preserving or monotone transformation, it is as valid in principle to approximate log dissimilarity on the basis of log distance as it is to work with the original scales. Moreover, as mentioned in the next chapter, the logarithmic transformation also often simplifies other aspects of the fitting problem by making the error variances more nearly constant. MULTISCALE permits the approximation to take place in either the original scale or in the logarithmic scale. When we discuss various transformations it will be assumed that the logarithmic transformation may already have been made.

At this point we must introduce appropriate notation to distinguish between the dissimilarity data and the computed distances. Let d_{ijr} be the

Figure 2.8: Dissimilarity plotted against distance for a MULTISCALE analysis of the recreation data.

dissimilarity measure for stimuli i and j produced by subject r. Let I denote the total number of stimuli and R the total number of subjects or replications. It is not necessary to assume that we have an observation for every possible pair (i, j) or subject r, although obviously there must be at least a few observations for each stimulus and subject. By contrast, let us indicate a computed distance corresponding to dissimilarity d_{ijr} as $\hat{d}_{ijr}$. The "hat" reminds us that $\hat{d}_{ijr}$ is a mathematically generated as opposed to a "real world" quantity. This notation is also used in statistics to indicate something that is an approximation to something else, as distance is in this case with respect to dissimilarity.

The transformation to be estimated will be denoted by $f_r(d)$, with the subscript r indicating that it will be estimated independently or uniquely for each subject r. The transformation problem then is to estimate function f_r for each subject r so that

$$\begin{array}{ll} \text{either} & f_r(d_{ijr}) \approx \hat{d}_{ijr} \\ \text{or} & f_r(\log d_{ijr}) \approx \log \hat{d}_{ijr} \ , \end{array} \tag{2.5}$$

where the symbol $\approx$ mean "approximates".

In MULTISCALE three levels of estimated transformation are possible. Since the default condition in the program is one in which the logarithmic transformation is applied first, these levels are named in terms of the transformation implied for dissimilarity when the transformation is in fact applied to the logarithm of dissimilarity. Thus,

- adding a constant to $\log d_{ijr}$ amounts to multiplying d_{ijr} by a constant, and this is referred to as a *scale* transformation, and
- multiplying $\log d_{ijr}$ by a constant amounts to taking d_{ijr} to the corresponding power, and this is referred to as a *power* transformation.

Scale: In this case

$$f_r(\log d_{ijr}) = \log d_{ijr} + v_r \tag{2.6}$$

or

$$f_r(d_{ijr}) = d_{ijr} + v_r \ .$$

The constant v_r estimated separately for each subject changes the location or origin of the log dissimilarity data. This changes the scale or unit of measurement for the dissimilarity data. Thus, each dissimilarity observation d_{ijr} is multiplied by the quantity $\exp(v_r)$, and this quantity is termed the *regression coefficient* for the dissimilarities for the r^{th} subject.

Power:

$$f_r(\log d_{ijr}) = p_r \log d_{ijr} + v_r \tag{2.7}$$

or

$$f_r(d_{ijr}) = p_r d_{ijr} + v_r \ .$$

The estimated transformation amounts to changing both the scale and location of the log dissimilarities. The effect is equivalent to taking d_{ijr} to the power p_r and multiplying it by the regression coefficient $\exp(v_r)$. Thus p_r is referred to as the exponent or power when the dissimilarity data is first log-transformed.

Because this transformation is very easy to estimate, and because it captures at least some of the nonlinearity typically observed between dissimilarity and distance, it is the default transformation in MULTISCALE. Most analyses should begin by using the power transformation.

Spline: A considerably more powerful and sophisticated transformation is possible using monotone splines. A detailed discussion of monotone splines is deferred to the final section in this chapter.

Figure 2.9 shows a very simple example of a spline transformation. These are produced by joining polynomials smoothly at junction points called *knots.* In the case shown the polynomials are quadratic. The two quadratic functions in this case are constrained to join so that they have one level of derivative. In Figure 2.9a the derivative is shown. Since the transformation must be monotone, the derivative is always positive, but note that it changes direction abruptly at the knot (its own derivative is discontinuous at the knot). We say that the derivative of the transformation is a positive piece-wise linear function, and this implies that the transformation itself is a piecewise quadratic function. This is shown in Figure 2.9b. Figure 2.9c displays the transformation at the level of the dissimilarity data.

Figure 2.9: An example of a monotone spline transformation $f(\log d)$. The leftmost figure displays the first derivative of the function f. The central figure displays f itself The rightmost figure displays the corresponding transformation of dissimilarity.

In general the flexibility of a spline function is determined by the degree of the polynomials and by the number of knots. In MULTISCALE the degree is always two (only quadratic functions are used) but the location and number of knots can be controlled by the user. By default the program uses a single knot and positions it so that the greatest flexibility is obtained in the upper range of the scale. Note that the number of parameters to estimate is two more than the number of knots. The estimation of these parameters is somewhat more complicated and requires considerably more computation time than the two parameters in the power transformation case. Although spline transformations often dramatically improve the fit, they should be reserved for the final stages of the analysis when other basic decisions have been made.

Finally, the constant v_r is automatically estimated in addition to the monotone spline transformation, so that the mathematical form of the transformation is

$$f_r(\log d_{ijr}) = s_r(\log d_{ijr}) + v_r \tag{2.8}$$

or

$$f_r(d_{ijr}) = s_r(d_{ijr}) + v_r \ ,$$

where $s_r(\cdot)$ is a monotone spline function.

For all three levels of transformation the coefficients v_r are estimated for each subject. To define these uniquely, MULTISCALE applies the location constraint

$$\sum_{r=1}^{R} v_r = 0 \ . \tag{2.9}$$

The three levels of transformation provide considerable flexibility combined with parsimony in terms of number of parameters to be estimated. A number of programs exist such as KYST (Kruskal, Young, & Seery, 1973) and ALSCAL (Takane, Young, & de Leeuw, 1977) which use step functions as transformations. They have acquired the somewhat misleading adjective of "nonmetric" even though they fit Euclidean distances to transformed dissimilarity data. Since step functions are neither smooth nor defined in terms of a fixed number of explicit parameters, there are serious problems assessing the quality of the final fit and interpreting the resulting transformations.

Table 2.3: More Symbols Introduced in this Section on Transformations

Symbol	Meaning
d^*	A distance or model value
$f_r(d)$	The value of a transformation of dissimilarity d
$f_r(\log d)$	The value of a transformation of log dissimilarity
v_r	A change of origin for log dissimilarity for subject r, or e^{v_r} is a change of scale for dissimilarity
p_r	A change of scale for log dissimilarity for subject r, or a power for dissimilarity
$s_r(d)$	A monotone spline transformation of dissimilarity for subject r

With monotone spline functions MULTISCALE provides all the flexibility most users will need while avoiding these difficulties.

The symbols introduced in this section on transformations are in Table 2.3.

2.6 Sources of Variation in the Data

Even with the best transformations the distances will not fit the transformed dissimilarities perfectly. The factors that make a residual, defined mathematically as

$$e_{ijr} = f_r(\log d_{ijr}) - \log \hat{d}_{ijr} \ , \tag{2.10}$$

something other than zero are many and complex. There are two reasons why we should ponder these sources of variation in residuals or errors. First, taking various components of variation into account can further tune the analysis, enabling more precise estimates and better statements of the precision of estimation. Secondly, a summary of the components of variation can add an important aspect to the interpretation of the data.

The basic problem, then, is to further analyze the variance of e_{ijr}, denoted by $\mathrm{Var}(e_{ijr})$. We do this by splitting this quantity into simpler components which are specific to subjects and specific to stimuli and pairs. This is done in MULTISCALE by considering the variance component model

$$\mathrm{Var}(e_{ijr}) = \sigma_r^2 \; . \tag{2.11}$$

We shall refer to the quantity σ_r as the *within-subject standard error.* Permitting this component to vary from subject to subject implies that the variation in a residual or the size of the typical deviation from best-fitting distance depends on the subject. In short, some subjects will tend on the whole to produce dissimilarity judgments deviating relatively little from the fitted distances, while others will tend to produce relative larger deviations. This is a generally reasonable assumption and is usually obvious from an inspection of the residuals following an analysis. Thus the subject-specific variance component σ_r^2 is an index of the relative lack of precision (from the point of view of the model) of a certain subject's data and thus has a variety of practical uses. For example, a relatively large value might indicate that a subject should be removed from the sample for closer individual scrutiny and analysis. A relatively large value might indicate a radically idiosyncratic view of the stimuli, a failure to understand the instructions, a change of point of view during the ratings, or a failure to judge dissimilarity appropriately.

MULTISCALE allows the user to assume either that the σ_r's are variable and therefore to be estimated along with other aspects of the fit or that they are all equal to a common value σ. Since the assumption of σ_r variable is realistic, and since it costs very little to estimate them, this is the default choice in MULTISCALE. However, even if the σ_r's are specified as constant, MULTISCALE will provide after-the-fact or post-hoc estimates of them conditional on all other parameter estimates.

In summary, MULTISCALE defaults to the variance component model

$$\mathrm{Var}(e_{ijr}) = \sigma_r^2 \; . \tag{2.12}$$

The average of the within-subject variances σ_r^2 is an over-all index of the variability of the data. This quantity is referred to in the program and the manual as the global error variance and its square root as the *global standard error.*

2.7 The Distribution of Errors

MULTISCALE estimates distances by the statistical procedure *maximum likelihood estimation.* A more detailed discussion this technique is provided in the next chapter. However, what is important here is that maximum likelihood estimation requires the user to state explicitly how the errors are distributed in the hypothetical population of possible judgments on a specific pair. That is, the statistical model is one in which a hypothetical sequence of independent judgments of a given pair by a given subject would be a sample from a population having a specified distribution. For most designs any subject gives only one judgment per pair, so the sample size is only one. Nevertheless, each residual is supposed to have the same characteristics except for a possible change in location and dispersion, so that all the residuals can be taken as providing information about this assumed distribution.

Figure 2.10 displays in three different ways the distribution of residuals for the data in Figure 2.2. In Figure 2.10a we see dissimilarity plotted against distance for these data. An examination of this plot indicates that large distances are associated with larger residuals than small distances. This is natural enough: when two nearly identical stimuli are presented, the range of plausible dissimilarity judgments is bounded below by zero, so the range is generally tighter than when two very different stimuli are presented. This finding is a very general one in psychophysics, where stimuli eliciting large judgments will also produce large variability. This is the celebrated Weber's Law. We can remove this effect by plotting dissimilarity against distance on the logarithmic scale. Figure 2.10b shows that the variability in such a plot tends to be much more homogeneous. Figure 2.10c provides a very useful alternative way to plot the residuals. This is called a *q-q or quantile plot,* and consists in plotting the normalized residuals

$$z_{ijr} = [f_r(\log d_{ijr}) - \log \hat{d}_{ijr}]/\sigma_r \ , \tag{2.13}$$

where σ_r is the estimated standard deviation of the residuals on the log scale, against the associated quantiles or points marking off equal areas for the standard normal distribution. The quantiles are, in a sense, idealized standard normal deviates, and by plotting the actual z_{ijr}'s against them, we can assess the extent to which the z_{ijr}'s act like standard normal deviates. The plot in Figure 2.10c indicates that the log dissimilarities are distributed fairly normally around the log distances. The logarithmic transformation played the important role of stabilizing variance. Of course the variance component models discussed in the previous chapter may also play some role in variance stabilization as well.

Figure 2.10: Three plots of the relationship between dissimilarity and distance for the data in Figure 1. The rightmost figure is a quantile or q-q plot of the ordered residuals against quantiles of the standard normal distribution.

No one assumption about the distribution of residuals is going to work for all sets of data. Consequently MULTISCALE permits four possible distribution assumptions in the hopes of accommodating a wider range of possible data. These are discussed below.

Lognormal distribution: If the log transformation of a random variable has a normal distribution, we say that the random variable has a lognormal distribution. This is approximately what is happening in Figure 2.10. Figure 2.11 displays the probability density functions corresponding to various locations or central tendencies for a fixed spread. A comparison of these indicates that the lognormal distribution has the following characteristics:

1. Positivity: only positive random variables can be lognormally distributed.
2. Spread proportional to location: roughly speaking, the spread or dispersion increases with the location of the distribution.
3. Skewness: the distribution has a longer tail in the positive direction than in the negative direction.

As already indicated, dissimilarity data frequently display the first two characteristics. The third is more problematical, especially since the response continuum for direct ratings usually has an upper bound. However, since the lognormal distribution is also very convenient to work with from a computational point of view, and since the absence of the long positive tail in practice does not greatly affect the results, MULTISCALE assumes the lognormal distribution by default.

Normal distribution: For some sets of data the distribution of dissimilarity about distance appears to be more symmetric than is consistent with a lognormal distribution. The familiar normal distribution is perfectly symmetric, although having the unrealistic feature of always assigning positive probability to negative values of dissimilarity. The usual form of the normal distribution also has the more serious flaw of stating that error variance does not depend on the mean.

Further remarks on the question of which distribution to select are made in the next chapter in the discussion of the statistical basis for MULTISCALE. It can be noted here, however, that one may view the selection

Figure 2.11: Two lognormal distributions plotted in terms of their probability density functions. Both have a scale value of 0.2 but the left distribution has a location parameter of 1/3 and the right a location parameter of 1.

Table 2.4: More Symbols Introduced in this Section on Variation

Symbol	Meaning
e_{ijr}	A residual or error for stimulus pair (i,j) and subject r
σ_r	Within-subject standard error for subject r
σ	Global standard error for all the data

of the distribution in an entirely nonstatistical way. In fact choice of the lognormal distributon results in MULTISCALE computing a least squares approximation in the log scale, and choice of thenormal distribution results in a least squares approximation in the original scale. This nonstatistical interpretation of what MULTISCALE does determines how one will interpret the results, especially when comparing fits using different models or assessing precision of estimation.

Table 2.4 offers a summary of the mathematical notation used in the discussion of error variation and distribution.

2.8 Mathematical Summary

Let I indicate the number of stimuli and R indicate the number of subjects. For stimulus pair (i,j) and subject r the dissimilarity judgment or index will be denoted by d_{ijr} and the corresponding distance by $\hat{d}_{ijr}$. It will be assumed that all dissimilarity observations are positive.

The most general distance model fit by MULTISCALE is

$$\hat{d}_{ijr} = \sum_{m}^{M} w_{rm}(x_{im} - x_{jm})^2 \ ,$$

where the number of dimensions is indicated by M. There are two special cases of this model fit by MULTISCALE:

1. The weights w_{rm} are unrestricted, except for being required to be non-negative, called the *diagonal metric* model, and

2. The weights w_{rm} are equal to one for all subjects, called the *identity metric* model. This last case is the default one for MULTISCALE.

MULTISCALE computes distances $\hat{d}_{ijr}$ so as to provide optimal fit of either

1. $\log \hat{d}_{ijr}$ to $\log d_{ijr}$ or

2. $\hat{d}_{ijr}$ to d_{ijr}.

The first case is the default one. In addition to computing the best-fitting distances, the program also computes a monotone transformation $f_r(\cdot)$ for each subject which is applied to either $\log d_{ijr}$ or to d_{ijr} according the choice above. Thus, in case (1) what is actually optimized is the fit of $\log \hat{d}_{ijr}$ to $f_r(\log d_{ijr})$ with both the distances and the monotone transformations being estimated from the data. The corresponding situations hold for case (2).

There are three levels of monotone transformation that are possible in MULTISCALE. These are presented below as applied to the case of fitting log dissimilarity.

Scale transformation:

$$f_r(\log d_{ijr}) = \log d_{ijr} + v_r \ .$$

The coefficient v_r has the effect of changing the unit of measurement for the d_{ijr}'s for subject r and is estimated separately for each subject.

Power transformation:

$$f_r(\log d_{ijr}) = p_r \log d_{ijr} + v_r \ .$$

The transformation is equivalent to taking the d_{ijr}'s for subject r to a power p_r and altering their unit of measurement.

Spline transformation:

$$f_r(\log d_{ijr}) = s_r(\log d_{ijr}) + v_r \ ,$$

where the function $s_r(\cdot)$ is a monotone spline function. Such functions are constructed by joining monotone increasing quadratic functions end to end so that the polynomials join smoothly. The values of $\log d_{ijr}$ at which these junctions take place are called *knots.* They can either be set by the user or left to MULTISCALE to choose. MULTISCALE chooses only a single point of junction by default. Further discussion of the spline transformation option can be found in Appendix 1.

The manner in which $\log \hat{d}_{ijr}$ is fit to $f_r(\log d_{ijr})$ or $\hat{d}_{ijr}$ to $f_r(d_{ijr})$ is determined by assumptions made about the distributions of the residuals, defined in the log transformation case as

$$e_{ijr} = f_r(\log d_{ijr}) - \log \hat{d}_{ijr} \ .$$

These are specified in two ways: first, by indicating the nature of the variance of the e_{ijr}'s in a hypothetical population of independently replicated judgments, and second, by specifying the type of distribution that these have. The most general formula for the variance of the residuals used in MULTISCALE is

$$\text{Var}(e_{ijr}) = \sigma_r \ .$$

This states that error variance is a subject-specific component σ_r^2. The subject-specific component can either be unrestricted and thus estimated uniquely for each subject, or may be required to be constant across subjects. The former case is the default one. The square root of the average of the subject-specific variances is referred to in the program as the *global standard error.*

The actual distribution of errors can be chosen to be one of the following:

lognormal: This says that the values of e_{ijr} defined in terms of $\log d_{ijr}$ and $\log \hat{d}_{ijr}$ are normally distributed with mean zero and variance determined as above. This is the default choice.

normal: In this case the residuals $e_{ijr} = f_r(d_{ijr}) - \hat{d}_{ijr}$ are assumed distributed with mean zero and variance

$$\text{Var}(e_{ijr}) = \sigma_r^2 \ .$$

The interaction between the three types of transformation and the two types of distribution options are summarized in Table 2.5.

Table 2.5: Transformations under the Two Distribution Options

Transformation	Lognormal Distn.	Normal Distn.
Scale	$f_r(\log d_{ijr}) = \log d_{ijr} + v_r$	$f_r(d_{ijr}) = d_{ijr} + v_r$
Power	$f_r(\log d_{ijr}) = p_r \log d_{ijr} + v_r$	$f_r(d_{ijr}) = p_r d_{ijr} + v_r$
Spline	$f_r(\log d_{ijr}) = s_r(\log d_{ijr}) + v_r$	$f_r(d_{ijr}) = s_r(d_{ijr}) + v_r$

2.9 Monotone Spline Transformations

In this section the class of monotone spline transformations are described from a mathematical point of view. Monotone splines are discussed in greater detail in Ramsay (1988) and Winsberg and Ramsay (1982), and more general references on splines are de Boor (1978) and Schumaker (1981).

In the most general terms a spline function is simply a set of polynomials joined end to end. The points at which they join are called *knots*. Between knots the spline is a polynomial and as a rule the degree of all polynomial segments are fixed in advance and identical. The interest and applicability of spline functions depends on their behavior at the knots themselves. Here they are generally required to exhibit a degree of smoothness which is determined by the number of derivatives that exist at the knot. If the polynomials merely join, they may not have any derivatives at the knot. On the other hand, they have have up to one less than the degree of the polynomials if the two segments are not part of the same polynomial. Figure 2.12 illustrates a spline function composed of three quadratic or degree two polynomial segments. Notice that at the first junction point there are no derivatives, but that at the second junction point the two polynomials have the same slope and therefore that the spline function has one derivative. The degree plus one of each polynomial segment is known as the *order* of the spline, and will be indicated by k. The number of polynomial segments less one will be indicated by N. Thus, if $N = 0$, the spline function reduces to a single polynomial segment.

A *knot sequence* $t_1, \ldots, t_{N+2k}$ is an integral part of the definition of a particular spline function. This sequence must obey certain rules. For the purposes of understanding MULTISCALE these rules can be given in the following more specialized version:

1. The first k values are equal and strictly less than the remaining values,

Figure 2.12: A piecewise quadratic function (solid line) constructed by joining three quadratic functions at the positions indicated by vertical dashed lines.

2. the last k values are equal and strictly greater than all previous values,
3. and values between the first and last k are in strictly increasing order.

These rules ensure that there are $N+1$ nonempty intervals between knots.

There are a number of ways of mathematically defining a spline function given a knot sequence. Each of these involves taking a linear combination of $N+k$ primitive spline functions. What differs from one scheme to another is how these primitive spline functions are defined. One of the most useful schemes from a practical point of view is the use of B-splines (short for basis splines). We shall denote the n^{th} B- spline of order k by $B_{nk}(x)$. These special spline functions have the following properties:

1. The n^{th} B-spline is zero except between t_n and t_{n+k} where it is positive,
2. In any interval there are only k B-splines which are positive. The remainder are zero.
3. The n^{th} B-spline of order 1 has the value one between knots t_n and t_{n+1} and is zero elsewhere.
4. The value of the n^{th} B-spline of order k for argument x lying between knots t_n and t_{n+k} is given in terms of the B-splines of order $k-1$ by the recursion formula

$$B_{nk}(x) = B_{n,k-1}(x)\frac{(x-t_n)}{(t_{n+k}-t_n)} + B_{n+1,k-1}(x)\frac{(t_{n+k}-x)}{(t_{n+k}-t_{n+1})}$$

Properties (c) and (d) permit us to compute the value of an B-spline function at any point relatively cheaply. Figure 2.13 displays some B- spline functions of order 2 and 3.

It turns out that any spline function joining with maximal smoothness at the knots (with $k-2$ derivatives) can be expressed as a weighted sum of B-spline functions. Thus, a spline function $f(x)$ can be represented as

$$f(x) = \sum_{n=1}^{N+K} c_n B_{nk}(x) \ .$$

Because of property (1) of B-spline functions this function will always be nonnegative provided that all the coefficients c_n are nonnegative. Note, too,

Figure 2.13: Two sets of B-spline functions. B-splines of order 2 are piece-wise linear and B-splines of order 3 are piece-wise quadratic

that because of property (2) only k of the B- spline functions need to be evaluated at any value of x.

We may now move easily to the definition of a family of *monotone* spline functions. When a spline function $f(x)$ is nonnegative, its integral

$$s(x) = \int_{t_k}^{x} f(u)du$$

will be a monotonically increasing function. Expressing $s(x)$ in terms of the B-splines, we have that

$$\begin{aligned} s(x) &= \int_{t_k}^{x} c_n B_{nk}(u)du \\ &= \sum_{n=1}^{N+K} c_n \int_{t_k}^{x} c_n B_{nk}(u)du \ . \end{aligned} \tag{2.14}$$

Monotonicity depends on all of the coefficients c_n being either positive or zero. Moreover, the integral of a B-spline function is not at all difficult to compute since, after all, a B-spline is still a polynomial between knots and the integral of a polynomial of degree $k-1$ is a polynomial of degree k. In fact, the integral of a B-spline function of order k can be simply expressed as a linear combination of B-splines of order $k+1$. Thus we have in equation (2.14) a family of monotonic splines which are easy to compute given coefficients c_n.

The shape of the monotone spline defined by (2.14) depends on the coefficients c_n. MULTISCALE computes these so as to maximize log likelihood while keeping them nonnegative. In statistical work it seems best in most situations to work with the lowest possible order of splines, and MULTISCALE employs order two splines. An order two spline is actually a piecewise linear function or a polygonal line. Its integral is a piecewise quadratic function. Thus, the monotonic spline computed by MULTISCALE are piecewise polynomials of degree two joining at knots so as to have a single derivative. The number of coefficients required is then equal to the numberof intervals plus one. The knot sequence will consist of two equal knots at the beginning of the interval over which the transformation is to be performed, some number (MULTISCALE requires at least one) of *interior knots*, and two final knots at the upper end of the transformation domain. Finally, note that the number of knots will be two more than the number of coefficients, which in turn will be two more than the number of interior knots. The default situation in MULTISCALE is one in which there is a single interior knot, three coefficients to compute, and five knots in all.

Chapter 3

Auxiliary Variables

While dissimilarity data yield rich information about how the subjects perceive the stimuli, they provide rather uncertain information about the way in which subjects value or prefer some stimuli over others. The auxiliary variables pairwise preference or direct ratings can provide this complementary information, and MULTISCALE provides a means of linking dissimilarity with one or the other of these auxiliary data by means of a common geometric model.

In this chapter we first discuss the data themselves, and then move to describe the two types of models which MULTISCALE can fit to them.

3.1 The Auxiliary Data

3.1.1 Pairwise Preferences

Although the term *preference* is used in a wide variety of ways in the behavioral sciences, we shall be quite restrictive in this manual. Preference refers to a judgement specific to a *pair* of stimuli in which

- the stimulus which is valued more is indicated,
- the degree to which it is valued more is also indicated, and
- if two stimuli are equally valued, the judgment is zero.

If we code such a judgment for stimulus pair (i, j) and subject r as a number p_{ijr}, then it is natural to use a *signed* number such that if

- $p_{ijr} > 0$, then stimulus i is valued more highly than j,
- $p_{ijr} < 0$, then stimulus i is valued less highly than j,
- $p_{ijr} = 0$, then stimulus i and j are equally valued, and
- $|p_{ijr}|$ indicates the degree to which one is valued over the other.

Such a judgment might be made using the following rating scale:

```
ice                                                        apple
cream                                                      pie

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1 1 1 1 1 1
 5 4 3 2 1 0                                       0 1 2 3 4 5
```

The subject is invited to place a check mark or "X" in one of the 31 blanks on this scale. The numbers positioned below the each blank indicate the value that p_{ijr} would assume if the mark is placed in that blank. Normally these coding values would not be actually presented to the subject.

For example, if the response is

```
ice                                                        apple
cream                                                      pie
                               X
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

then this subject values the two deserts equally, has no preference for one over the other, and would receive $p_{ijr} = 0$. On the other hand, if

```
ice                                                        apple
cream                                                      pie
         X
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

then the subject has a fairly strong preference for ice cream over apple pie, and would receive an observation $p_{ijr} = -11$, meaning that the preference for apple pie over icecream is -11 units, or that ice cream is actually preferred. Finally,

```
ice                                                         apple
cream                                                       pie
                                     X
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

indicates a slight preference for apple, and is coded $p_{ijr} = 3$.

Pairwise preference data may be collect in ways other than the use of a rating scale of this nature. MULTISCALE does assume, however, that the observations are signed numbers whose magnitude indicates the degree of preference, and for which zero has the special significance of no preference. Hence MULTISCALE assumes that pairwise preferences are on the *ratio scale* of measurement.

3.1.2 Direct Ratings

The stimuli can also be evaluated or rated one by one rather than as pairs. Such ratings are very common, and can be carried out in many different ways. All that MULTISCALE assumes in this case is that the bigger the number used to code the judgment, the more that stimulus is preferred. Such a judgment can be indicated as u_{ij}.

Here is an example of a direct rating scale:

```
                X
apple pie    - - - - - - - - - - - - - - -


                                X
ice cream    - - - - - - - - - - - - - - -
```

Each stimulus has 15 categories, and in this case, $u_{ir} = 3$ for apple pie and $u_{ir} = 11$ for ice cream, indicating that ice cream is more highly valued than apple pie.

A code of zero has no special signifcance for direct ratings, and MULTISCALE assumes an *interval scale* for these data.

For direct rating data, but not for pairwise preference data, it is possible to use ratings or judgments with respect to multiple attributes at the same time. That is, we might be judging possible sweets with respect to being a dessert, a late evening snack, or a replacement for lunch. When multiple attributes are involved, we use the notation u_{kir} to indicate the evaluation by subject r of stimulus i with respect to attribute k, where $k = 1, \ldots, K$.

When the number of stimuli I is large (15 or more, for example), the number of pairs to judge with respect to pairwise preference is also large, and it seems unlikely that studies involving pairwise preference data would consider more than one attribute at a time.

Although direct ratings involve less time to collect, preferences have the advantage of providing more information per stimulus, and hence of potentially providing more accurate assessments of subject utilities.

3.2 Models for Auxiliary Variables

For either pairwise preferences or direct ratings MULTISCALE proposes two geometrical models: the ideal direction and the ideal point models.

3.2.1 Ideal Direction Model

This is the model most often used in market research and other areas of psychometrics. According to this model, there is a direction in the space used to represent the stimuli such that the further a point is along this direction, the more highly the subject values the stimulus.

Consider Figure 3.1 in which the ten forms of recreation introduced in Chapter 2 in Figures 2.2 and 2.3 are again represented as points. This time, however, each subject is also displayed in the figure as an arrow. This arrow indicates that subject's direction of greatest utility. Points far along in this direction are estimated to be highly valued by this subject, and points by contrast which are in the opposite direction are little or even negatively values. We see in this display that subject 13 is the only one with a strong positive value for television, fashion shows and window shopping. Presumably the same subject has little taste for or even strongly dislikes attending conferences or reading.

This model is expressed algebraically for direct ratings on a single attribute as follows:

$$u_{ir} = \sum_{m=1}^{M} y_{rm} x_{im} + v_r \tag{3.1}$$

In this equation

Figure 3.1: The ideal direction representation of the recreation dissimilarities and pairwise preferences.

- x_{im} is the coordinate of the point representing stimulus i on dimension m, and is primarily determined by the dissimilarity data,
- y_{rm} is the value placed by subject r on variation along dimension m, and
- v_r is merely a constant or intercept that permits direct ratings to be on an interval scale.

Coefficient y_{rm} determines by its size how closely the ideal direction will line up with dimension m in the coordinate system. Because of the structure of the model and this role of y_{rm}, the model is often called the *scalar product* model.

MULTISCALE assumes implicitly that pairwise preferences and direct ratings are related very simply by the following *difference* model:

$$p_{ijr} = u_{ir} - u_{jr} \ . \tag{3.2}$$

This model expresses algebraically the significance of zero preference, since this implies that the two utilities are equal.

Of course, both types of data are not present at the same time, but it follows from the difference model (3.2) and the ideal direction direct rating model (3.1) that the model for pairwise preferences will be

$$p_{ijr} = \sum_{m=1}^{M} y_{rm}(x_{im} - x_{jm}) \ . \tag{3.3}$$

Thus either set of auxiliary data are used by MULTISCALE to estimate the dimension utilities y_{rm}. While these auxiliary data do provide some information about the point coordinates x_{im} as well, this is very limited relative to that available in the dissimilarities.

Finally, when multiple direct ratings are involved, the model (3.1) requires an extra subscript for the rating variable, and becomes

$$u_{kir} = \sum_{m=1}^{M} y_{rmk} x_{im} + v_{rk} \tag{3.4}$$

In this equation y_{rkm} is now the utility or value for dimension m for subject r with respect to property or attribute k. Constant v_{rk} is an intercept term specific to a subject and property.

3.2.2 Ideal Point Model

It can also be seen as natural to represent subjects as points in the same manner as stimuli. In the ideal point model subject points are positioned close to stimuli which are highly valued, and far from stimuli which have little or negative value.

Figure 3.2 shows a display of this type. We can see that subjects are indicated in this display as points, just as stimuli are. The closer a subject is to a given stimulus, the more that stimulus is preferred, and for pairs of points the ideal point model implies that a subject will prefer the stimulus which is closer. For example, we now see that subject 13 is very close to television, while at the same time time being more distant from fashion shows and window shopping. Also, note that 1, 2, and 15 are now portrayed as having little preference for conferences, in contrast to what was indicated in the ideal direction model, but that they are seen as liking reading. A weakness of the ideal direction model was that any direction favoring reading would also favor conference going, but the ideal point model is able to circumvent this difficulty.

Since distance and dissimilarities are on ratio scales with fixed origins, while direct ratings are on an interval scale as a rule, we first take the logarithm of distance to put it into interval scale form, and then use a linear transformation to yield the model

$$u_{ir} = p_r \log(1/d_{ir}) + v_r = -p_r \log d_{ir} + v_r \tag{3.5}$$

where d_{ir} is the distance from subject r to point i computed by either

identity metric:

$$d_{ir}^2 = \sum_m^M (y_{rm} - x_{im})^2$$

or

diagonal metric:

$$d_{ir}^2 = \sum_m^M w_{rm}(y_{rm} - x_{im})^2$$

Figure 3.2: The ideal point representation of the recreation dissimilarities and pairwise preferences.

Correspondingly, the model for pairwise preference data is, using the difference model (3.2),

$$p_{ijr} = (-p_r \log d_{ir}) - (-p_r \log d_{jr}) = p_r \log d_{jr}/d_{ir} \ . \tag{3.6}$$

In effect, the ideal direction model can be viewed as a special case of the ideal point model. The latter behaves more and more like the former as a subject's ideal point gets further and further away from the configuration of points, in which case it basically defines a direction of preference. Moreover, the ideal point model has one more parameter per subject (p_r) than the ideal direction model.

Chapter 4

Some Important Statistical Concepts

MULTISCALE approaches the multidimensional scaling problem from a statistical point of view. Although this point of view is not essential for the use of MULTISCALE, it is important to understand a number of statistical terms and concepts that appear throughout this manual and on the output. Several important results can often be obtained with MULTISCALE which are not possible with nonstatistical programs. The advantages of a statistical approach are broadly speaking three:

- Improved precision of estimation: Taking explicit account of the nature of random variation in the data leads to "best possible" estimates
- Hypothesis tests: Assuming the account of random variation is reasonable, it is possible to test a wide variety of hypotheses about the data. Usually this takes the form of comparing two or more models by calculating an index of improvement of fit which can be assessed with familiar statistical tables.
- Estimates of precision of estimation: Not only are parameter estimates optimal, but the precision with which corresponding population values are estimated can be assessed numerically and graphically. This gives the investigator an idea of how much information the data have provided with respect to these parameters.

In this chapter a number of statistical notions such as maximum likelihood estimation and asymptotic chi squared tests are introduced. Readers already familiar with these ideas may want to either skim or skip this discussion. A more technical account of these matters may also be found in Ramsay (1977, 1978, 1982).

4.1 Maximum Likelihood Estimation

Any process of fitting a model to data must employ a criterion to assess the fit. Those parameter estimates which optimize the criterion are then computed. The optimized value of the fitting criterion is then evaluated in various ways. The best known example of this process is probably the use of summed squared errors from a regression line as a fitting criterion and the computation of the slope and intercept parameters which minimize this, a process known as least squares estimation. The final error sum of squares can then be assessed in a variety of ways, especially when an alternative value is available.

MULTISCALE is based on the principle of *maximum likelihood estimation.* As the term implies, a fitting criterion called the likelihood is maximized by determining the parameter values yielding its highest possible value. Except for a scale factor, likelihood is essentially the probability of obtaining the data given the parameter values. The parameters include everthing that must be estimated: the Cartesian coordinates of the points, the individual metric matrices if required, parameters determining the estimated data transformations $f_r(d)$, and the variance component parameters. For any specific set of values for these parameters, the likelihood of the data can be determined. MULTISCALE finds those parameter values which maximize likelihood.

The maximum likelihood process is based on the following rationale: If the data are "typical", that is, a true random sample from the population, then their probability with respect to the population parameter values will be relatively high. Thus the parameter values that actually maximize this probability define a pseudo population for which the data are the most typical possible. Since the data are also assumed to be very typical with respect to the actual population, the parameter values in the population and those maximizing the data probability ought to be fairly similar. In fact, when an arbitrarily large amount of data are available, the maximum likelihood estimates and the population parameters are arbitrarily close to one another, a property known as consistency. Moreover, maximum likelihood estihation leads to the best possible estimates in the sense that when large sample sizes are involved no other estimation process provides estimates which are consistently closer to the population values. For large samples maximum likelihood estimates are distributed normally with a mean equal to the population values and a variance-covariance matrix that can be easily estimated.

Any useful appreciation of maximum likelihood estimation must include an understanding of what it does not imply. Most importantly, all the desirable properties of maximum likelihood estimation relate to large, not small, samples; that is, it is asymptotically optimal. In fact there are many situations where in small samples maximum likelihood estimates have rather undesirable properties. For example, dividing the sum of squared deviations from the sample mean by the sample size is the maximum likelihood estimate of variance, but this estimate tends to be too small (biassed) in small samples, and dividing by one less than sample size is preferable (unbiassed.) Nevertheless, any estimation procedure which fails to be competitive for large sample sizes is less likely to be interesting for small samples, unless nonstatistical considerations such as economy are critical.

A second important caveat about maximum likelihood estimation is that its optimal characteristics become irrelevant if the sample cannot be reasonably considered to come from the hypothesized population. The requirement of explicitly stating the nature of random variation in maximum likelihood estimation is both a strength and a weakness. If one wishes to use the advantages that this process makes possible it becomes very important to critically examine the actual behavior of the data in various ways. MULTISCALE has a number of such diagnostic procedures available for the serious data analyst. Again, however, even a nonstatistical approach should involve serious considerion of how the data are actually distributed. For example, the use of a fitting process which places heaviest emphasis on large dissimilarities when these are the noisiest parts of the data is ill-advised in most applications.

From a numerical point of view, it is much more natural to maximize the logarithm of the likelihood than the likelihood itself. Since the logarithmic transformation preserves order or is monotone, it follows that any parameter values which maximize the likelihood will also maximize the log likelihood and vice versa. Moreover, the log likelihood has a number of uses which are outlined below.

4.2 The Independence Assumption

An important statistical assumption incorporated into MULTISCALE is independence of residuals. This means that the size and direction of the lack of fit, error, or residual for any judgment or datum does not depend on that for any other datum. This is a rather strong assumption and needs some special comment. Most well-known statistical procedures such as analysis of variance and multiple regression analysis also involve the independence assumption. In fact it is often not completely justified. For example, if the data are direct ratings, subjects' memory of previous judgments can affect the current judgment, so that if one suspects that one overrated the previous dissimilarity one may compensate by underrating the present one. Dependencies of this sort are usually relatively mild, however. Much more serious are dependencies due to the data being derived from other data or due to mathematical constraints. For example, confusion frequencies are interdependent because the frequencies in any row of the data matrix must sum to the total number of presentations. Similar remarks pertain to dissimilarity indices derived from combining various univariate measures.

The effects of violation of the independence assumption are complex and difficult to assess. Most immediately affected are estimates of variance, which will be under- or over-estimated depending on the nature of the dependencies. A goodness-of-fit measure such as the likelihood or error sum of squares is usually directly related to variance estimates and therefore is similarly affected. However, estimates of what one might call structural parameters such as coordinates of points are not usually very sensitive to violations of independence.

With rating scale data, the dependencies are usually not serious enough to warrant concern. In fact, one of the important reasons for "tuning" the fit by the use of spline transformations and individual metrics is to remove such dependencies. However, with derived dissimilarity indices, it is probably better avoid the statistical statements based on a MULTISCALE analysis, and to adopt a nonstatistical point of view in interpreting the results.

4.3 Log Likelihood and Least Squares

An important practical implication of the independence assumption is that the log likelihood for the total set of observations is simply the sum of the log likelihoods for each individual observation. It is also a sum of the log likelihoods for the data for each subject. For further discussion of how the log likelihood is defined for the various distributions and types of data transformation consult Ramsay (1982). However, it may be informative to see how log likelihood is described mathematically in the default case of the lognormal distribution with estimated power transformations and only subject-specific variance components. If we denote the log likelihood for subject r by L_r , then

$$L_r = -(M_r/2)\log\sigma_r^2 - 0.5\sum_{ij} e_{ijr}^2/\sigma_r^2 + M_r\log p_r \ , \tag{4.1}$$

where

$$e_{ijr} = p_r \log d_{ijr} + v_r - \log d^*_{ijr},$$

and where M_r is the total number of judgments for subject r. This can be expressed more simply if one substitutes in (4.1) the maximum likelihood estimate of $\hat{\sigma}_r^2$, given by

$$\hat{\sigma}_r^2 = M_r^{-1}\sum_i\sum_j e_{ijr}^2. \tag{4.2}$$

Then (4.1) reduces to

$$L_r = -(M_r/2)(\log\hat{\sigma}_r^2 + 1) + M_r\log p_r.$$

Since $\hat{\sigma}_r^2$ is simply a multiple of the error sum of squares in the log scale for subject r, the log likelihood can be seen to be very simply related to this quantity. The final term, called the Jacobian term, corrects for the effect of estimating the data transformation. Thus, $-L_r$ is a function to be minimized which can be easily understood in relation to least squares fitting criteria commonly used in other programs.

The log likelihood for the total sample is then computed by summing log likelihoods across subjects:

$$L = \sum_{r=1}^{R} L_r \tag{4.3}$$

The log likelihood cannot be interpreted by itself. It may be either positive or negative, and its size will depend on the number of observations and the number of parameters in the model for fitting the data. Under some circumstances two log likelihoods can be compared, and this is discussed in section 2 of this chapter.

4.4 AIC and BIC Measures of Fit

Fitting criteria like log likelihood and least squares have the practical problem that their value depends on the number of parameters being estimated; the more parameters one uses to define the model, the larger log likelihood usually is. Two log likelihoods can be compared only when the model producing one is a specialization or subset of the model yielding the other. However, the log likelihood can be modified slightly to approximately correct for number of parameters so that two models can be compared at a descriptive level even though they involve differing numbers of parameters. The *AIC statistic* is one such modification described by Akaike (1974). If L denotes the log likelihood, then the AIC statistic is

$$AIC = -2L + 2N_p \ , \tag{4.4}$$

where N_p is the number of mathematically independent parameters being estimated. Another modification proposed by Schwarz (1978) is the BIC statistic:

$$BIC = -2L + N_p \log N_o \ , \tag{4.5}$$

where N_o is the total number of observations. Obviously the smaller both statistics are, the better the fit of the model, since they are related to the negative of the log likelihood. Both involve adding a correction term to the negative of the log likelihood. The model yielding the smallest value of one or the other statistics is to be preferred. BIC is usually more conservative in the sense of favoring models having smaller numbers of parameters than those favored by AIC, although both yield similar conclusions in most situations. Both are intended for descriptive use only; they cannot be used in hypothesis testing.

4.5 Comparing Models or Hypothesis Testing

As introduced in the previous section, maximum likelihood estimates have very desirable properties when the sample is large and the statistical assumptions are justified. Under these circumstances, the log likelihood can be used in a simple way to test hypotheses concerning which of two models provides the better fit to the data.

This can be done when one of the models is a special case of the other, when the simpler model is derived by fixing or constraining the parameters of the more general model in some way. We say in this case that the simple model is *nested* within the more general model.

Here are some useful examples of nested multidimensional scaling models:

- the comparison between a distance model using M dimensions and one using $M-1$ dimensions. The former reduces to the latter if all the coordinates with respect to the M^{th} dimension are set to zero.
- the identity metric model is a special case of the diagonal metric model in which all weights w_{rm} are equal to one.
- The power transform of data reduces to the scale transform when the exponent p_r becomes one.
- The power transform model in turn is a special case of the spline transform model.
- Yet another example arises when one wishes to ask whether two or more groups differ with respect to the configuration generating the respective distances. The general model involves estimating all parameters uniquely for each group and then simply summing the log likelihoods. This corresponds to the hypothesis that the data in any group have nothing to do with the data in any other group. The special case arises when all the data are combined together and fit by one common model.

Let us denote the log likelihood for the more general model by L_Ω and that for the simpler or more specific special case by L_ω. Then the quantity

$$\chi^2 = 2(L_\Omega - L_\omega) \tag{4.6}$$

provides a convenient test criterion. When sample size is large and the statistical assumptions are justified, and under the null hypothesis that the simpler model describes the population, this criterion has a chi squared distribution. The degrees of freedom of this distribution are simply the difference between the number of parameters for the general model and the number of parameters for the simpler model. One may then consult a table of the critical values of chi squared to determine whether the computed value is significant at some level. Significance constitutes a statistical argument that the more general model adds something substantial to the fit of the data.

Of course, many things, some of which are hard to detect in practice, can perturb this principle. The wise user neither bases his decision about which model to use solely on such a test nor takes overly seriously a value which barely achieves a modest level of significance. Ramsay (1980b) describes in more detail how this criterion behaves in the specific context of testing dimensionality.

Since one of the assumptions underlying the use of (4.6) is that of a "large" sample, it is important to explain what is meant by this. Large refers to the excess of data over parameters. One may have a large number of observations but use up most of the degrees of freedom for error in a very extravagant model (full metrics, spline transformations, high dimensional distances, complicated variance component models, etc.). Then the large number of observations is nevertheless a small sample. A useful rule of thumb in a wide range of applications is that the data-to-parameter ratio should be at least ten. Models such as multidimensional scaling can proliferate parameters very rapidly and the sensible investigator is cautious in this matter. Again consult Ramsay (1978, 1982) for more details.

Even when one feels uncomfortable about the interpretation of (4.6) as a chi squared variable, it is possible to use it in a descriptive way to compare the degree of improvement of various extensions of a baseline model. We shall provide examples of this later. The AIC and BIC statistics are also useful in this regard, and experience has shown that AIC is particular helpful. In general, experience indicates that hypothesis tests tend to favor more complex models than most users of the results of the analysis want to contemplate, whereas choosing the model which yields the minimum AIC criterion usually results in a model of modest complexity.

The critical value of a chi squared variable depends on its degrees of freedom. A statistic which can be used in the same way can be computed using the following transformation of chi squared to a standard normal deviate (Wilson & Hilferty, 1934):

$$z = [(\chi^2/k)^{1/3} - 1 + q]/\sqrt{q} \ , \tag{4.7}$$

where $q = 2/(9k)$ and k is the number of degrees of freedom. The advantage of this transformation is that z does not depend on the number of degrees of freedom and thus one can compare via the respect z-values two or more ways in which fit has been improved.

4.6 Assessing Precision of Estimation

As noted above, when sample size and reality of the statistical assumptions permit, it is possible to estimate the variance-covariance matrix for parameter estimates. The matrix summarizes the nature of the variation of these estimates from sample to sample and provides an indication of how far wrong the estimates are likely to be.

The variance-covariance matrix is estimated by computing the negative inverse of the expected value of the matrix of second derivatives of the log likelihood function. This matrix is not difficult to compute in general, and is used in other ways by MULTISCALE. However, when the number of parameters is large, as would be the case when data transformations are estimated for each subject and there are many subjects, this would be a very large matrix indeed and the computation of the inverse would be unduly expensive. Thus, MULTISCALE uses only the matrix associated with the coordinate estimates. It therefore produces a *conditional* variance-covariance matrix; that is, the estimated variance-covariance matrix in effect assumes all other parameters fixed rather than estimated. This is obviously a compromise, and the user of MULTISCALE should realize that this process tends to produce estimates of sampling variance which are somewhat smaller than is realistic. Nevertheless, when substantial numbers of subjects are involved, the underestimation is not serious, and Ramsay (1980b) describes a correction that can offset this tendency.

One way in which the variance-covariance matrix of the coordinate estimates can be used is to provide *confidence ellipsoids* surrounding each point indicating a region within which the population point is included with a specified level of probability. If the reader wishes to turn ahead to Chapter 6, an example of such a display can be seen.

Chapter 5

How MULTISCALE Works

This chapter presents the general strategies adopted by MULTISCALE to compute maximum likelihood estimates. Computing these estimates is not a simple matter, and it is essential to understand some aspects of how this is done to determine if the computation has reached a successful conclusion and to modify or control the computation when necessary.

The computational problems posed by multidimensional scaling derive from two facts: there is no direct or analytic solution for the coordinates yielding the best distances, and in a typical application a very large number of parameters must be estimated. In response to this MULTISCALE in common with most other multidimensional scaling programs does two things: first, it proceeds to a final solution incrementally or iteratively by successively improving an initial set of parameter values; and secondly it separates the parameters into groups and improves each group in turn leaving the others fixed. In this way it carves the computational problem into manageable portions and proceeds by modest steps to the final desired result.

The parameters in a MULTISCALE analysis are grouped as follows:

1. Configuration $\mathbf{X}$ or Coordinates x_{im}: These are the IM coordinates, one for each point and each dimension, from which the distances are computed.

2. Metric weights w_{rm}: There are M per subject.

3. Transformation parameters: These are the parameters for each subject determining the dissimilarity transformations. In the scale transforma-

tion case they are the coefficients v_r, in the power case they are the coefficients v_r and and p_r, and in the spline case they are v_r plus the coefficients determining the monotone spline transformations.

4. Variance parameters: In the subject-specific variance case they are the variances σ_r^2 and in the constant variance option it is the global standard error σ.

We can define a *main iteration* as a cycle in which the parameters in each block are updated. For example, a main iteration in MULTISCALE may consist in first computing better point coordinates, then improving the parameters defining the dissimilarity transformations while holding the coordinates fixed, and then finishing by improving the variance components parameters while fixing the coordinates and transformation parameters. The parameter blocks are processed in the order in which they are shown above. A typical MULTISCALE analysis will carry out 20 to 30 of these main iterations.

The entire computational process begins with *initial values* for each block of parameters computed by procedures that provide reasonable first guesses. For example, when required, MULTISCALE begins by setting each metric matrix to the identity matrix. Coordinates are initially estimated using the procedure described by Torgerson (1958). These starting values can be said to be the consequence of iteration 0. Starting values can also be set by the user and may come from previous MULTISCALE analyses or analyses by other programs. MULTISCALE provides convenient procedures whereby results from one analysis can be passed on to the next analysis as starting values.

Within each main iteration, any block of parameters can be updated one or more times. The number of iterations of a particular block of parameters within a main iteration is termed that block's *secondary iterations.* For example, within any one main iteration one may choose to iterate four times on the coordinates while holding all other parameters fixed. Of course, multiple secondary iterations can be expected to decrease the total number of main iterations required. The effectiveness of the trade-off will depend on the particular data and model involved. MULTISCALE defaults to a single updating for each parameter block.

A critical part of any iterative process is determining when to stop it. This introduces the problem of defining *convergence criteria.* Convergence is reached when any iterative process, main or secondary, satisfies a numerical test. There are many numerical tests possible. For example, one may decide to terminate an iterative process when the parameters being updated do not change by more than a certain amount from one iteration to the next. Alternatively, one may examine the function being maximized or minimized. For example, if the log likelihood function is not improved by more than a certain amount from one iteration to the next, one may decide to quit. Another feature of the function that can be tested is its rate of increase or slope. If, at any point, the function is not increasing very rapidly as a function of the parameters being updated, the iterations can be terminated.

In MULTISCALE convergence criteria are based on the behavior of the log likelihood or a function derived from it rather than on the behavior of parameters. The user can control the various criteria used or can accept the default criteria built into MULTISCALE. The criteria used in each iterative process are described below.

Convergence for Main Iterations: The change in log likelihood from one iteration to the next is tested. As mentioned in the previous chapter, the primary application of the log likelihood is in terms of computing differences between two values and then assessing twice this difference against tabled values of the chi-squared variable. One does not really need much accuracy in this test; as a rule it suffices to know a chi-squared variable with anything more than 5 degrees of freedom (almost always the case in multidimensional scaling) to the nearest whole number. Consequently MULTISCALE defaults to terminating the main iterations when the log likelihood does not change by more than 0.05 from one iteration to the next. It is often practical to accept results well short of reaching this criterion, however, especially in more exploratory phases of the investigation. The user can override this criterion if desired.

Convergence for Configuration Iterations: In the lognormal distribution option, the log likelihood depends on the coordinates x_{im} only through the error sum of squares,

$$\mathrm{SSE} = \sum_r^R \sum_i^I \sum_j^I (f_r(\log d_{ijr}) - \log d^*_{ijr})^2 = \sum_r^R \sum_i^I \sum_j^I e^2_{ijr} \ .$$

This must be minimized in order to maximize the log likelihood. It is simple to work out the criterion for SSE which is equivalent to the criterion of a change of .05 in log likelihihood. MULTISCALE uses this equivalent SSE criterion in assessing on iteration ν whether the relative change $(SSE^{(\nu-1)} - SSE^{\nu})/SSE^{(\nu-1)}$ is sufficiently small. An analagous criterion is applied in the normal distribution case.

Convergence for Metric Weight Iterations: The log likelihood depends on the metric weights w_{rm} for subject r only on the error sum of squares for that subject. The main iteration criterion of .05 (or whatever the user supplies) then determines the criterion for relative change in this subject's error sum of squares.

Convergence for Transformation Parameters: The process here is iterative only in the spline transformation case. In that case the relative change in the contribution to the log likelihood of the r^{th} subject is tested.

Convergence for Subject-specific Variance Components: This is not an issue since these can be solved for exactly.

The convergence criterion for each parameter block or secondary iteration can be determined by the user. All convergence criteria are displayed at the beginning of the analysis.

We now come to the updating process itself for any particular parameter block. MULTISCALE always improves parameters by seeking to decrease the relevant function. This implies that when log likelihood is to be increased with respect to a parameter block, MULTISCALE seeks to decrease the negative of log likellihood.

There are many possible ways of modifying parameters so that the relevant function is decreased. In general, however, an updating process must take two things into account. The first is the rate of decrease of the function for a particular set of parameter values. The second is the nature of the infinitismal change in parameters that will decrease the function the most rapidly. These are assessed by the *gradient* of the function, or the set of first partial derivatives of the function with respect to each of the parameters. The rate of decrease can be assessed in various ways, of which two are:

1. gradient length: If the gradient has elements $g_1, ..., g_N$, where N is the

number of parameters in the block, then this is given by

$$\mathrm{GL} = \sum_n g_n^2$$

2. relative gradient length: If the parameters themselves are $y_1, ..., y_N$, then this is

$$\mathrm{RGL} = \sum_n (g_n^2)(y_n^2)/N \ .$$

   This has the practical advantage of not depending on the scale or number of parameters in the way that the simple gradient length does. When MULTISCALE refers to gradient length, it is in fact using this measure.

One procedure for updating parameters is simply to subtract from each parameter a constant times the corresponding gradient element. This technique, known as the *gradient method* or the *method of steepest descent* in effect is a movement in parameter space in the direction of the most rapid decrease.

Although the gradient method can be used optionally in MULTISCALE, it is generally better to do something more sophisticated. This involves not only taking into account the rate and direction of decrease, but also the rate of change of this direction itself. This requires some use of the second partial derivatives with respect to the parameters in the block. If there are N parameters in a block, there are N first partial derivatives and N^2 second partial derivatives. The latter can be assembled in a square symmetric matrix called the Hessian. In fact, it turns out that the average or expected value of the Hessian serves just as well and is much cheaper to compute. If we indicate the expected Hessian by E and the gradient vector by g, then the direction of increment δ is determined by solving the equation

$$E\delta = -g \tag{5.1}$$

For some parameters, most notably the configuration matrix, E will not be of full rank. In such cases MULTISCALE uses the minimum norm solution for δ. The process of using (5.1) to compute the best increment is known as the *scoring method.* It has been proven to be very effective in a wide range of problems.

The primary argument against the scoring method is that it can require substantial computer time and memory to compute the increment vector δ. Thus, when the configuration matrix has a very large number of elements (perhaps more than 100) it may be desirable to revert to the gradient method, which will in general require substantially more iterations but cost less per iteration and use less memory.

Once MULTISCALE has determined the appropriate increment vector or search direction in parameter space, it then attempts to minimize the function by moving from the current point in this direction. This is the *line search* step. MULTISCALE first checks that the function is changing sufficiently rapidly along this line. If not, it terminates the iteration. If the initial slope, called the directional derivative, is sufficiently large, it then uses a sophisticated procedure for locating the minimum of the function on this line. This is in turn an iterative procedure, and these iterations can be called *tertiary iterations.* MULTISCALE permits a maximum of 5 of these. Convergence usually takes place in two or three tertiary iterations, and the history of these iterations is not normally displayed. Occasionally, however, there will be a failure to attain the optimum in 5 iterations, and then a warning message will be output and the history of the line search iterations displayed. As a rule this is not serious and the program will continue unimpeded to the final result.

Chapter 6

An Introductory MULTISCALE Analysis

The next two chapters discuss in detail how to set up a MULTISCALE analysis and how to control and interpret the results. It is often helpful, however, to see what a simple job looks like in terms of its input and output before considering more complex possibilities. This chapter presents the job setup and output for the data in Figure 2.2. A simple two-dimensional fit employing the various default options of MULTISCALE is used.

Table 2 presents the data in Table 1 as set up for a MULTISCALE run. The reader may recall that a single subject rated all 105 possible pairs of 15 forms of recreation using a category rating scale containing 25 categories. The job displayed in Figure 6.1 does not have any system control statements.

The first thing to note is that some lines begin with the "at" sign (@). This special symbol tells MULTISCALE that an input block is beginning. Immediately following the symbol @ is the name of the block. Block names and all other names or keywords in a job are identified by the program using only the first three letters. Aside from this restriction, the block name can be as long as desired and spelled in any way. In this manual we shall name blocks by capitalizing the first three letters to remind you of this fact. The input blocks ocurring in this job are

1. The TITle block which gives the title of the job.
2. The PARameters block which specifies the dimensions of the problem and the type of model to be fit. In this example MULTISCALE is told

Figure 6.1: Recreation Data Set Up for MULTISCALE Analysis

```
@title;
Judgments of 15 Recreations by Subject FUNSEEKER
@parameters nstim=15, ndim=2, nsub=1;
@disdata;
(14F3.0)
 16
  3 18
 12 12 11
 16 21 16  2
 20 10 19 15 12
 15 12 13  9 19  6
 21 23 23 19  7 22 20
  7 10  6 18 19 25 15 25
 19 22 25 22 14  8 22 23 25
  9  7 13 15 12 19 20 22  8 25
 22 16 16 19 13  7 13 15 23 13 25
  7  3 13 12 21 13 10 22 13 12  7 18
 21 22 22 12 23 21 18 18 21 22  9 22 12
  8  8  7  9 21 21 12 22  5 25  9 23 10  8
@stimlabels;
(56A1)
CONCERT MUSEUM  THEATRE MOVIE   TV      CONFRNCEREADING
HOCKEY  BALLET  DEBATE  FASH SHODOC FILMEXHIBITNWINSHOP
RESTAURT
@subjlabel;
(8A1)
FUNSEEKR
@compute itmax=50;
```

that there are 15 stimuli and one subject and that the fit is to be in two dimensions. The EJECT item instructs MULTISCALE to skip to a new page before each output section.

3. The DISdata block in which the data to be analyzed are input. In this example the data are in the form of the lower triangle of the dissimilarity matrix excluding the diagonal and are preceded by a FORTRAN format specification.

4. The STImulus labels block containing 8-character labels for the stimuli, seven of them per line. The labels are preceded by a FORTRAN format specification.

5. The SUBject labels block containing an 8-character label for the subject.

6. The COMpute block in which instructions for computation are given. In this example the maximum number of main iterations is set to 50.

These blocks can occur in any order with the following two exceptions:

- PARameters block must appear either first, or following the TITle block if the latter appears first,

- and the last block must be the COMpute block.

Within a block information is input in array form and/or using keywords and logical items. A keyword is a string followed by an equal sign such as NSTIM above. Again only the first three letters are significant for MULTISCALE. A logical item is a string ocurring without an equal sign. Keywords and logical items can occur in any order and are separated from each other and the block name by blanks or commas. The occurrence of a semicolon or a slash (/) signals the end of the keywords and logical items in a block.

Arrays are input following keywords and logical items. They may be input in a variety of ways. In this example the common procedure of giving a FORTRAN format specification is employed, but array information can also be input in free format or binary form. Note that character arrays are input one character at a time. In the following chapter other possible ways in which dissimilarity data can be arranged (and rearranged) are described.

The output from a MULTISCALE job is also arranged into sections. There are 18 possible output sections, but most jobs will not generate all of these. MULTISCALE permits considerable control over the output, including the possibility of supressing certain sections or directing them to special output files. Output can also be made to be left-justified and a maximum of 80 columns wide for terminal users. The output which the example in Table 6.1 generates is given in Appendix B. After printing out the title and analysis number, Section 1 specifies the model chosen and the dimensions of the problem along with initial or starting values of various parameters. Section 2 contains a detailed history of the computation, iteration by iteration, showing within each iteration the secondary iterations performed for various parameter estimates. Section 4 presents global results for the job including the final log likelihood, AIC, and BIC statistics. Also included are the same statistics for a simpler "benchmark" model in which the data are fit by a single constant along with a chi-squared test of the improvement of fit that the distance model affords. Section 5 presents the final configuration in three ways: in floating point form, in rounded integer form, and in polar coordinates. Section 6 displays the fitted interpoint distances in rounded integers. Section 8 displays post-hoc estimates of the stimulus standard error weights α_i computed on the basis of the final configuration. Section 10 indicates the final exponent p_r for the subject, but omits the value of v_r since this is automatically set to one for data from a single subject. Section 12 provides the final correlation between the log dissimilarity judgments and the log distances. Section 13 indicates the estimated standard errors and correlations of estimate for the coordinates. Sections 15 and 16 plot the final configuration in two ways: first by plotting the points for each dimension separately, and then plotting the points for each pair of dimensions. The final configuration is also displayed in Figure 2.3..

In this introductory analysis heavy use has been made of the default values of various options in MULTISCALE. Among these are the use of the lognormal distribution, the identity metric for distance (this is necessary when only one subject is involved), the power transformation option, and the supression of variance component options. The possible options and their default values are described in the next chapter.

Chapter 7

MULTISCALE Input

This chapter presents a detailed discussion of how to set up a MULTISCALE analysis. It begins with two examples of job setups, and proceeds to describe each possible section of a input file for MULTISCALE.

In the personal computer version of MULTISCALE, the name of the input file is requested by the program when it is invoked. In the mainframe version the input file is FORTRAN file number 5, the usual default file in most computer systems.

7.1 Dissimilarity Analysis for Recreation Data

Figure 7.1 displays an example of an analysis of some dissimilarity judgments on pairs among 15 forms of recreation. The number of subjects is 10 in this case.

This example is designed to show how to set up MULTISCALE in a typical situation in which only dissimilarity data are to be analyzed. The data are dissimilarity judgements for recreations, but this time by 15 subjects, and only with respect to the first 10 stimuli presented in Figure 2.2.

A detailed discussion of what each line in this job means follows below. Before launching into this, however, we provide a second example.

Figure 7.1: Job Setup for the Analysis of Recreation Data (Dissimilarities Only)

```
@TITLE;
             10 RECREATIONS

@PAR NSTIM=10,  NDIM=2, NSUB=15;

@DISDATA VECTOR;
(15F3.0)
  8 23  2 24 18 25  7 20 25  8 12 23 25  2 12   1
 21  3 24  4  9  5 11 10 25 20 20 12 22 17 22
 25 25  8 25 21 25 25  6 12 10 11 22 18 25  1
  1  2  9 11 11 11  5 13  8 14 10 10 10 16 15   2
  1  4  8 12  3 12  5 16 13 14 10  6  7 15 18
 19 17 23 13 18 19 20 14 25 16 25 16 19 19  8

       ...   and so on  ...

 16 17  6 18 21 21  8 24 19  8 10  8 20  2  5  15
  7  7 16  2  7  4  3 23 19 19  9 15 23 10  5
 17 18 10  6  9 23 17 14 22 17  3  6  4 20  3

@STIMLAB;
(64A1)
READING TV      HOCKEY  BALLET  ARTMUSM CONCERT THEATRE CONFEREN
WINDOWSHFASHSHOW

@SUBJLAB;
(64A1)
AYOTTE  BAKER   BAXTER  CORONEL DEMPSEY FOREY   HABIB   HANLEY
KIRKWOODLIGHT   MARR    PERREAULPOWIOZNYSHAKIN  TARINI

@COMPUTE ITMAX=50;
```

7.2 Dissimilarity and Preference Analysis for Recreation Data

In this example, each subject has also given a judgment of pairwise preference for each pair of stimuli.

Note that pairwise preferences are signed numbers, unlike dissimilarities, which are positive.

In this analysis the metric for distance is the diagonal metric. The model for preferences is the ideal point model, set by the `AMO=POI` command. Unlike the previous example, neither the dissimilarities nore the preverences require re-ordering.

7.3 The Block Structure of MULTISCALE INPUT

In these two examples, note that input to MULTISCALE is organized into input *blocks*. Each block begins with a name starting with the @ symbol. In both examples, the initial block defines a title for the analysis, and the second block gives essential information about the dimensions of the data (number of stimuli, subjects, dimensions, and etc.) and types of models. The actual data are input in subsequent blocks, and other blocks are used for inputting stimulus and subject labels.

MULTISCALE recognizes 19 types of input blocks. These are given in Table 7.1. In the setup of an analysis, MULTISCALE only uses the first three letters to identify a block or other keyword items, and consequently the first three letters are capitalized in the block name.

Each block begins with an initial portion beginning with an @ and terminating with a semicolon (;). Following this initial portion there may or may not be further records containing additional information. On some operating systems the symbol @ may be reserved for special purposes. The initial portion of a block may also be terminated by a slash (/).

For the most part these input blocks may occur in any order. Blank lines can also be inserted between blocks, although no more than two blank lines can appear within a block.

Figure 7.2: Job Setup for the Analysis of Recreation Data Involving Both Dissimilarities and Pairwise Preferences

```
@TITLE;
             10 RECREATIONS

@PAR NSTIM=10,  NDIM=2, NSUB=15, AUX=PREF, AMO=POINT, METRIC=DIAGONAL;

@DISDATA VECTOR;
(15F3.0)
  8 23  2 24 18 25  7 20 25  8 12 23 25  2 12     1
 21  3 24  4  9  5 11 10 25 20 20 12 22 17 22
 25 25  8 25 21 25 25  6 12 10 11 22 18 25  1
  1  2  9 11 11 11  5 13  8 14 10 10 10 16 15     2
  1  4  8 12  3 12  5 16 13 14 10  6  7 15 18
 19 17 23 13 18 19 20 14 25 16 25 16 19 19  8

       ...   and so on  ...

@PRFDATA VECTOR;
(15F5.0)
   10  -12    0  -12  -11    8   12  -12    9   -1  -12  -12    5   12   -8    1
    5  -10   12   11   11   -2  -12  -11    9    2    0    9  -10   -8    0
   12   10    3   -1  -10    0    0  -12  -12   -1    1   -8   12   -6    8
    4  -12   12  -12   -7    0  -12    0   -9    4   -8    0   -3   -5   10    2
  -12    5   -2   -6   -1    2  -12   10    7    0   12    1   10  -12    3
   -6    4    3    0    7    0   -4  -12    0   -8    2  -12    8  -10   12

       ...   and so on  ...

@STIMLAB;
(64A1)
READING TV      HOCKEY  BALLET  ARTMUSM CONCERT THEATRE CONFEREN
WINDOWSHFASHSHOW

@SUBJLAB;
(64A1)
AYOTTE  BAKER   BAXTER  CORONEL DEMPSEY FOREY   HABIB   HANLEY
KIRKWOODLIGHT   MARR    PERREAULPOWIOZNYSHAKIN  TARINI

@COMPUTE ITMAX=50, ITXMAX=1, ITWMAX=1, ITYMAX=1;
```

Table 7.1: Blocks of Parameters and Data in a MULTISCALE Analysis

Block Name	Contents
PARameter	Parameters defining size of data and type of model
TITle	Title for analysis
DISdata	Dissimilarity input data
PRFdata	Preference input data
RATdata	Direct rating input data
OLDdata	Dissimilarity data analyzed previously
TRAnsformation	Type of *a priori* transformation of data
STImlabels	Labels for stimuli
SUBlabels	Labels for subjects
CONfiguration	Initial coordinates for configuration $\mathbf{X}$
IDEalpoint	Initial ideal points or directions
METric	Initial metric weights w_{rm}
DORder	Indices for re-ordering dissimilarity data
PORder	Indices for re-ordering preference data
KNOt	Knot values determining spline transformations
SPLine	Initial spline parameters or regression coeffs.
COMpute	Parameters controlling computation
WTDiss	Dissimilarity standard errors for each subject
WTPref	Preference or rating std. errors for each subject

The two exceptions to the arbitrariness of block order are the PARAMETERS block, which must either be first or come second immediately after the TITLE block, and the COMPUTE block which must be the last block before analysis is to take place. Thus, all the blocks beginning with the initial PARAMETERS (or TITLE) block and ending with the first COMPUTE block define a single run or analysis of the data. Multiple runs are achieved by subsequent blocks, each run always being defined by an initial PARAMETERS (or TITLE) block and the next COMPUTE block. In the example above a second run has been defined by the last three blocks.

7.4 Within-block Structure

Within any block there is an initial portion containing a number of pieces of information in "keyword" format, and terminating with a semicolon or slash. These will be referred to as block *items*.

The first item in a block must be separated from the block name by one or more blanks or commas. Items within the initial portion are also separated by one or more commas or blanks. No more than 50 consecutive blanks or commas is permitted. When more than one line is required, an item must not be broken up by the end of the line, since the end of a line is also taken as separating two items. As many cards as required may be used. Blank lines (to an upper limit of three) may be inserted between blocks. If the semicolon or slash ending this initial portion is omitted, the program will usually recover appropriately but will issue a warning message.

There are four types of items within any block. In the description of each type below, capital letters will be used to indicate a specific item. Some computer systems differentiate between upper and lower case characters, and on such systems these items will actually be typed in lower case when entering information into the computer.

logical: These items are a single word, which can be abbreviated to their three leading characters. MULTISCALE II uses only these three characters in any case to identify a logical item. An example would be NARROW in the first PARAMETERS block, which could have been abbreviated to NAR. Logical items must be separated from both a previous item and a following item by one or more blanks or commas (blanks and commas are treated the same).

keyword: These precede an equal sign (=), and identify the role in MULTISCALE of the following value. An example is NSTIM in the first PARAMETERS block. Keywords can also be abbreviated to their three leading characters. They must be separated from the previous item by one or more blanks or commas.

literal: These are strings of characters like logical items. However, they either follow an equal sign (and hence a keyword item) when occurring within the initial statement of a block, or occur in an array (such as a set of labels). They may be abbreviated to their first three characters. An example of a literal item would be DIAGONAL in the second PARAMETERS block. Literal items must be separated from a subsequent item by one or more blanks or commas.

numerical: These are either integer or real numbers. Some numbers follow a keyword, as does the number of stimuli, 15, in the first PARAMETERS block, and others are entered as an array, as are the observations in the NEWDATA block.

Within any block the information in the initial portion can appear in any order. In some cases there will be no information, in which case the block name is followed by a semicolon or slash.

Following the initial statement, sets of literals or numbers may be entered as arrays. The observations in the NEWDATA block, the stimulus labels in the STIMLABELS block, and the re-ordering indices in the ORDER block are examples of arrays. There are three ways to input arrays:

fixed format: A FORTRAN format specification precedes the array, and the array is then input under the control of this specification. This is the usual mode of entering arrays in statistical programs.

free format: No format specification is supplied and array items need only be separated by commas or blanks, and as many items as required are input. While free format input is convenient and occasionally essential, it does impose the constraint that no items other than those to be entered can be in the input stream. Since items must be separated by one or more blanks or a comma, neither the observations nor the reordering indices in the format in which they occur in Figure 7.1 could be entered in free format style because some values are not separated in this way.

binary: Occasionally items have been written in binary or unformatted form on disk or tape by a previous program. Such files can be input to MULTISCALE by using the binary option.

If an array is to be input, it begins on a new line following the line containing the terminating semicolon. This is what happens in the NEWDATA, STIMLABELS, and ORDER blocks in the example above. Thus, these blocks are terminated by reading the last required item in the array. In the case of arrays of literal items input under free format, the array must be terminated by a semicolon or slash.

More than one analysis can be carried out within any MULTISCALE run. In the above example two analyses are to be carried out, the first with the IDENTITY metric model, and the second with the DIAGONAL metric model. Each analysis commences after the COMPUTE block has been input.

Most of the items entered with keywords have default values that will be suitable for many situations. For example, the default distance model is the IDENTITY option, and hence the item METRIC=IDENTITY does not need to appear in the first PARAMETERS block. When multiple analyses are asked for in a MULTISCALE run, the values set or defaulted in a previous analysis apply automatically unless specifically overridden. It is for this reason that items like NSTIM=15 do not reappear in the second PARAMETERS block. Stimulus and subject labels are also passed automatically from one analysis to another. However, the parameters entered in the COMPUTE block are exceptions to this rule, and will revert to their default values unless explicitly set on a subsequent run.

Each of the input blocks is described in greater detail in the following sections. A block is only supplied in the input stream if it supplies needed information. For each input block the items that can be input within the block are specified along with any default values. Only the first three letters of each block name are critical, and the remaining letters can be anything you wish or omitted entirely. To emphasize this, only the first three letters of block names and keywords are capitalized below. Remember that the first line of each block begins in column 1 with @ followed by at least the first three characters of the block name.

Table 7.2: Keyword Items Accompanying the PARameters Block

Keyword	Possible Values	Default Value	Function
NSTimuli	> 1	0	Number of stimuli or points
NDImensions	> 0	2	Number of dimensions
NSUbjects	> 0	1	Number of subjects or replications
NRAtings	> 0	1	Number of direct rating variables
NKNots	≥ 0	0	Number of interior knots for spline transformations
TP1	≥ 0	1	First scratch file number
TP2	≥ 0	2	Second scratch file number
TP3	≥ 0	3	Third scratch file number
ISEed	< 0		Initial seed value for random number generator

7.4.1 PARameters Block:

The parameters which define the dimensions of the problem are specified here along with the parameters which determine the distance model, variance component and transformation options to be used, and most aspects of input and output control. The items that can be input in this block are presented in the Tables 7.2, 7.3, and 7.4. Tables 7.2 and 7.3 present keyword items and specifies the permitted range of values. If a value outside the specified range is entered, an error message and termination of the run will usually result. To emphasize that only the first three characters of an item are required, these are capitalized. It should be noted, however, that items are input in lower case on systems differentiating between upper and lower case.

An example of a PARameters block is as follows:

```
@PARAMETERS  NSUB = 6, NSTIM = 10, TRANFORMATION = SPLINE, NOASYM,
   METRIC = DIAGONAL, DISTN = NORMAL;
```

In this block MULTISCALE is informed that there are 10 stimuli judged by 6 subjects. The dissimilarity data will be transformed by monotone splines. The diagonal metric is to be used, and the assumed distribution-for errors is normal. The logical item NOASYM instructs the program to

Table 7.3: Keyword Items Accompanying the PARameters Block (continued)

Keyword	Possible Values	Default Value	Function
PTHresh	any	99.0	Bound on preference abs. values
SDDiss	> 0.0	0.0	Initial value for global standard error for dissimilarities
SDPref	> 0.0	0.0	Initial value for global standard error for preferences or ratings
PRObability	$> 0, < 1$	0.95	Confidence level for confidence regions for points
METric	IDEntity, DIagonal	IDEntity	Distance model option
AUXiliary	NONe, RATing, PREference	NONe	Type of auxiliary variable
AMOdel	DIRection, POInt	DIRection	Type of model for auxiliary variable
TRAnsform	SCAle, POWer, SPLine	POWer	Transformation option
SUVariance	CONstant, SUBwise	SUBwise	Variance component model for subjects
DIStributn	LOGnormal, NORmal	LOGnormal	Distribution option
PLOt	NONe,	NONe	Type of confidence ellipse plot:
	HPGl,		Hewlett-Packard HPGL commands
	POStscript		Postscript commands
	PRInter		On-line dot-matrix printer
	SCReen		Only the computer screen

Table 7.4: Logical Items for PARameters Block

Logical Item	Option Set when Item is Specified
DCOmplete	Complete matrix of dissimilarities for each subject rather than just lower triangle
PCOmplete	Complete matrix of preferences for each subject rather than just lower triangle
DEBug	Complete output for each iteration for debugging purposes
NARrow	Output confined to 80 columns
EJEct	Page eject before each output section
GRAdient	Optimization by gradient method rather than scoring method
LIStdata	Input data matrices to be displayed
NARrow	Output confined to 80 columns
NOPlots	Suppress plots of configuration and of dimension saliences
NOStats	Suppress various statistics for subjects
NOAsympt	Suppress asymptotic variance estimates
NODist	Suppress table of interpoint distances
QPLot	Plot of normalized residuals against quantiles of normal distribution for each subject
RECover	Recover initial parameter values from file ITP1
SAVe	Save configuration, weight matrices, and all other estimated parameters on file ITP1
HIStory	Print out results for each iteration.
TABles	Table of transformed dissimilarities, distances, and normalized residuals for each subject
DPLot	Transformed dissimilarity plotted against distance for each subject
PPLot	Prefernce plotted against predicted preference for each subject
TPLot	Transformed dissimilarity plotted against dissimilarity for each subject
QPLot	Plot of normalized residuals against quantiles of normal distribution for each subject
WIDe	Output in 120 columns and center
RANdom	Generate random data

not compute or display asymptotic standard errors for the coordinates. Implicit in this block is the back that the analysis will be in only two dimensions, since the default value for the NDIM item is 2.

Here is another example of a PARameters block, which instructs MULTISCALE to also analyze the auxiliary pairwise preference data, using an ideal point model. The final configuration and ideal points are to be plotted along with confidence ellipses on a dot matrix printer.

```
@Parameters  nstim = 10, ndim=2, nsub = 15, auxil = pref, amodel = point,
   plot=print;
```

7.4.2 TITle Block

Only the title is input in this block. There is a single keyword item for this block: LINes. The numerical item following this keyword specifies the number of lines of title to be input. The maximum possible value is 5 and the default value is 1. The title may contain any characters useful for identifying an analysis. Note that this block may occur either before or after the PARameters block. This is the only block that can precede the PARameters block.

An example of a title block is as follows:

```
@TITLE  LINES=2;
THIS IS AN EXAMPLE OF A JOB TITLE AND MAY CONTAIN
ANYTHING EXCEPT SPECIAL CHARACTERS
```

7.4.3 DISdata Block:

The dissimilarity observations to be analyzed are input here. The format of the data is specified within this block prior to the occurrence of the data themselves. The dissimilarity judgments for each subject in turn are input in this block. Thus MULTISCALE expects an array of dissimilarity observations for each subject. These arrays may be of the following types:

lower triangle: Observations for each unordered pair are input excluding pairs in which a stimulus occurs with itself. If there are I stimuli, this implies that $I(I-1)/2$ dissimilarities are input for each subject. If there are observations missing, these are entered as zeros or negative numbers. This type of array is the default type for MULTISCALE and no special instructions are necessary if the data are in this form.

complete matrix: Observations for every ordered pair are input, implying $I(I-1)$ values for each subject. MULTISCALE assumes that the observations for pair (i,j) and pair (j,i) are independent of each other. Thus, this option should not be used if only the unordered pairs were actually presented. This option is enabled by the logical item COMplete in the PARameters block.

diagonal present: By default MULTISCALE assumes that the observations corresponding to the pairing of a stimulus with itself are not present in the array. This can be overriden for either lower triangle or complete arrays by the logical item DIAgonal in this block. These observations may be anything since MULTISCALE does not make use of them. One of the uses of the DIAgonal option is when the complete matrix is to be input even though the matrix is symmetric. The DIAgonal option without the COMplete logical item in the PARameters block will cause MULTISCALE to read all I rows of the matrix but only use the lower triangle excluding the diagonal.

Dissimilarity observations where are read in as zero or negative numbers are treated by MULTISCALE as missing data.

However the array is defined, it can be read either as a matrix or as an unbroken set of numbers. By default MULTISCALE assumes the observations arranged in a matrix. In this case and assuming lower triangle input (the default assumption for MULTISCALE) a single number is read from the first row, two from the second, three from the third, and so on until all observations are input.

It is often more efficient to enter the observations without arranging them into a matrix. In this case the data are entered as an unbroken vector of $I(I-1)/2$ elements in the case of lower triangle input. The number of observations per line is then either specified by the FORTRAN format specification or is arbitrary if free format entry is used. This option is enabled by the VECtor logical item in the NEWdata block. When data are entered in this way, a rearrangement of the data may be desired, and the ORDer block described below is used to enter the necessary reordering information. Unless the observations are reordered, they are assumed to be in the order

$$d_{21}, d_{3,1}, d_{32}, d_{41}, d_{42}, d_{43}, \ldots$$

and so on until $d_{I,I-1}$; that is, in the same order as they would have if entered as a matrix.

In the DISdata block and all other blocks containing numerical arrays the observations in a given line of data can be formatted in three ways:

1. fixed format: In this case a FORTRAN format specification beginning in parentheses is entered before the first line of data. MULTISCALE assumes fixed format input by default, but it may also be enabled by the statement FORmat=FIXed.

2. free format: In this case numbers are separated by blanks, commas, or the end of a line. There may be as many numbers per line as is consistent with this rule. MULTISCALE simply continues reading new lines until all the observations that it expects are entered. Since the end of a line is also separates numbers, a number may not begin on one line and continue on the next. MULTISCALE cannot enter more than 400 numbers at a time in free format. This restriction will never cause problems if the data are in matrix form since MULTISCALE reads one row of the matrix at a time. However, the VECtor option could result in this restriction being violated for large arrays. An error message would result. Free format input is enabled by the statement FORmat=FREe in this block.

3. binary format: In cases where the observations have been placed on disk or tape in binary format it is necessary to specify FORmat=BINary in this block. No format specification is necessary in this case.

Finally the user can control which file the array is read from by specifying DATfil=n. MULTISCALE will assume the array is part of the normal input stream on file ITP5 otherwise. In the case of fixed format, the format specification is also assumed to reside on file n as the first line. Of course the user will have had to instruct the operating system to ready file n for reading before executing MULTISCALE. It will usually be necessary to include the

Table 7.5: Keyword Items Accompanying the DISdata Block

Keyword	Possible Values	Default Values	Comments
FORmat	FIXed FREe BINary	FIXed	FIXed implies fixed format input. The array must then be preceded by a FORTRAN format specification. FREe implies free format input. A list-directed read of data results. BINary implies numbers are to be input in binary format.

Table 7.6: Logical Items for the DISDATA Block

Logical Item	Option Set when Item is Specified
VECtor	Dissimilarities are in a single vector instead of a lower triangular matrix.
DIAgonal	The diagonal entries of the dissimilarity matrices are present.

logical item REWind to ensure that MULTISCALE reads from the beginning of file n.

A summary of the keyword and logical items in the NEWdata block are given in Tables 7.5 and 7.6. These items are also applicable to the input of numerical arrays in other blocks.

If no items occur (that is, the block name is followed by a semicolon or slash), then the program will assume the data are to be input under format control with each row of the lower triangle of a dissimilarity matrix beginning a new line.

In the following example a single matrix is read in under format control using the default conditions. Only the lower triangle of the matrix is input, and since the diagonal entries are not input, only seven lines are required although there are eight stimuli. Note that the format specification causes MULTISCALE to bypass the row numbers.

```
@NEWDATA;
(5X,7F2.0)
```

```
2   2
3   6 4
4   3 2 6
5   3 2 3 2
6   1 2 3 3 2
7   2 4 2 2 3 3
8   2 3 3 1 2 2 3
```

In the following example the observations are to be read in free and vector format.

```
@NEWDATA FORMAT=FREE;
2 6 4 3 2 6 3 2 3 2 1 2 3 3 2 2 4 2 2 3 3 2 3 3 1 2 2 3
```

7.4.4 PRFdata Block:

The pairwise preference observations to be analyzed as the auxiliary variable are input here. The format of the data is specified within this block prior to the occurrence of the data themselves. The preference judgments for each subject in turn are input in this block. Thus MULTISCALE expects an array of preference observations for each subject. These arrays may be of the same types as those for the DISdata block.

The main difference between pairwise preferences and dissimilarities are that the latter are assumed to be signed numbers; some will be positive, and some will be negative.

MULTISCALE uses a threshold or bound to determine which preference values are to be treated as missing. By default, if the absolute value of an observation exceeds 99.0, it is treated as missing. This threshold value can be modified by using the PTHresh keyword in the PARameters block

7.4.5 RATdata Block

Direct rating data are input in this block. For direct rating data, each subject provides a rating or judgment separately on each stimulus. In addition, MULTISCALE permits ratings of this nature with respect to multiple rating variables or attributes. This contrasts with pairwise preference data, where a subject provides a single judgment with respect to each stimulus *pair*, and only with respect to a single variable or attribute.

The direct ratings are input as R matrices, each with I rows and K columns, where I and R are, as before, the number of stimuli and subjects, respectively, and K is the number of rating variables. In applications where only a single rating variable is involved ($K = 1$), the I ratings associated with a subject can be entered as one or more rows rather than necessarily as I lines of data. The essential thing to remember is that MULTISCALE expects the rating variable index to vary within the stimulus index, which in turn varies within the subject index.

Otherwise direct rating data follow the same rules as for dissimilarity or preference data, except of course that the data cannot be in triangular format since they are of necessity in the form of complete matrices.

7.4.6 OLDdata Block

No information is input in this block, which is simply an instruction to re-analyze the previously input set of data. Thus, this block appears simply as the following example:

```
@OLDDATA;
```

7.4.7 TRAnsformation Block

It often happens that it is desirable or necessary to transform the data prior to analysis by MULTISCALE. For example, if a set of category ratings have been recorded in such a way that they are indices of similarity, it would be necessary to first reflect them. MULTISCALE permits the input data to be transformed by the following function:

$$f(d) = \min\{ad^b - c, e\} \tag{7.1}$$

With this formula the reflection of similarity ratings could be achieved by using $a = -1, b = 1, c = -10$, and $e = 10$. Another example is provided by the desire to use a preliminary power transformation along with the SCAle estimated transformation option. In this case, one would use something like $a = 1, b = 1.5, c = 0, e = 1e20$. Finally, it might be desirable to convert dissimilarity values of zero, which would be treated as missing data by MULTISCALE, to some small positive constant. This could be achieved by $a = 1, b = 1, c = 0$, and $e = 0.5$.

Table 7.7: Keyword Items Accompanying the TRAnsformation Block

Keyword	Possible Values	Default Value	Comments
ATRan	any	1.0	Multiplier a in (7.1)
BTRan	any	1.0	Exponent b in (7.1)
CTRan	any	0.0	Constant c in (7.1)
ETRan	> 0	1E20	Limit e in (7.1)

The keywords and their associated default values are given in Table 7.7. An example of a TRAnsformation block is:

```
@TRANSFORMATION  BTR=1.5;
```

7.4.8 STImlabels Block

In this block the labels for the stimuli are input as a literal array. Although a label may be as long as desired, only the first eight characters will be used. If this block does not occur, the label fields in the output are left blank. Note also that labels must be input under format control if blanks or commas are to be part of any label. If read in under format control, exactly eight characters will be read per label, with each character occupying an entire word. This implies that the format statement should be of the form (mA1), where m specifies the number of characters per line. The rules for input of labels under free format are slightly more restrictive than those for free format input of numerical arrays since blanks may be part of a label. Thus labels must be separated by commas and the final label must be followed by a semicolon or slash.

The items in the initial statement in this block are the same as those in the NEWdata block. For a subsequent analysis of the same data, these labels (and the subject labels) will be automatically in force and do not need to be reinput. An example of this input block would be

```
@STIMLABELS;
(40A1)
STIM 1  STIM 2  STIM 3  STIM 4  STIM 5
```

7.4.9 SUBlabels Block

Subject labels are input here in the same manner as stimulus labels. They are also passed from one analysis to another. An example of input of subject labels with free format is

```
@SUBLABELS  FORMAT=FREE;
SYLVIE,MARY LOU,PIERRE,J RAMSAY;
```

7.4.10 CONfiguration Block:

This block is included when the user decides to input an initial configuration. While the program will generate its own initial configuration if one is not input, it is often useful to provide one. For example, when large amounts of data are missing (a missing datum is indicated by a value less than or equal to zero or by simple nonoccurrence in INDex mode), the generation of an initial configuration by the program may be either impossible or produce results worse than could be achieved by an intelligent guess. Another situation requiring an initial configuration arises when a previous analysis terminated before achieving convergence, and it is desired to restart the iterative process beginning where the previous run left off. Finally, when estimating a configuration that is subject to constraints, it may be desirable to begin computation with an initial estimate which satisfies the constraints since MULTISCALE does not take these into account in generating its own initial estimate.

The configuration matrix has as many rows as stimuli and as many columns as dimensions. The keywords available are the same as those described in the NEWdata block, although VECtor and DIAgonal are not relevant here. An example is:

```
@CONFIGURATION;
(2X,2F6.1)
 1   1.1  -2.3
 2  -0.2   5.2
 3  10.0  -1.7
 and so on
```

A special case arises when the RECover logical item is included in the PARameters block. This causes MULTISCALE to look for the initial configuration on file TP1 on the assumption that it has been placed there by a previous analysis in which the SAVe logical item was present. In this case it is still necessary to have the CONfiguration block, but no keywords or logical items are necessary. MULTISCALE will have placed the configuration along with an initial FORTRAN format specification on this file. However, since other arrays as well as the configuration will also reside on file TP1, it is necessary to input each of them in the appropriate order by including the appropriate input block in the same order. The order in which arrays are deposited in file TP1 under the SAVe option corresponds to the order in which they are printed out by MULTISCALE. The RECover option is in effect equivalent to `@CONFIGURATION DATFILE=m;` where the integer m is the number of file TP1 (default value 1).

7.4.11 METric Block

Each subject's initial metric matrix is input here. For the DIAgonal metric model, there is a weight for each dimension. The metric matrices are input as a single array having as many rows as subjects and as many columns as weights. In the DIAgonal metric model the rth row contains the diagonal weights

$$w_{r1}, w_{r2}, \ldots, w_{rM} \ .$$

The keywords available here are the same as those in the NEWdata block, with VECtor and DIAgonal being irrelevant in this block.

7.4.12 IDEalpnt Block

In this block either the coordinates of the ideal points or the coefficients defining the ideal directions are input.

There is a row or line for each subject, and in each such row a coordinate or coefficient is required for each dimension. For direct rating data only, a final additional value is required for each subject specifying the constant term on the model.

This block will usually be used to provide good initial values for the iterative estimation of ideal point coordinates, rather than for ideal direction coefficients, since the optimal values for the latter are computed exactly and noniteratively by the program.

The options available in this block are the same as those available for the CONfiguration and METric blocks.

7.4.13 DORder Block

Pairs of stimuli are usually presented to subjects in randomized order in order to minimize dependencies in judgments. In such cases the labor involved in sorting the dissimilarity observations into the order

$$d_{21}, d_{31}, \ldots, d_{I,I-1}$$

for each subject may be formidable and introduce possible errors. For this reason MULTISCALE permits the observations to be input directly from the questionnaire and then can reorder them internally. The example at the beginning of the chapter is a case in point. In order to reorder observations, it is necessary to number the entries in the lower triangle of the dissimilarity matrix sequentially (the full matrix is numbered rowwise if the COMplete option is used). The sequential number for the entry in the i^{th} row and j^{th} column is given by the formula $(i-1)(i-2)/2 + j$. The reordering index corresponding to each observation is then the sequential number of the cell in the matrix in which it belongs. In the example, the first observation entered whose value is 18 belongs in the 4th cell. This cell corresponds to the 3rd row and 1st column of the dissimilarity matrix.

When the COMplete option is specified, sequential numbering is of the complete matrix excluding the diagonal or not according to whether the DIAgonal option is present. The DIAgonal option with lower triangle input means that the sequential numbering is of the lower triangle including the diagonal, and the appropriate formula in this case is $i(i-1)/2 + j$.

The keywords are the same as those in the NEWdata block. Refer to the example in Table 7.1 for an illustration of the ORDer block.

7.4.14 PORder Block

Re-ordering indices for the pairwise preference data are input here if desired in the same way as for dissimilarity data, described in the DISdata block above. The only additional feature is that if an index is negative in sign, the sign of the corresponding preference observation is reversed.

7.4.15 KNOt Block

The interior knots for the monotone spline transformations are input here. The program automatically determines the knots at the ends of the transformation interval, and the interior knots as well if they are not input in this block. Interior knots are input in free format, and there can be a maximum of 10 of them. The same interior knots are used for all subjects. The final knot value is followed by a semicolon. Note that when the LOGnormal distribution option is in force (MULTISCALE defaults to this) the monotone spline transformation is applied to the logarithm of the dissimilarity data, so that the knots must be chosen with respect to the log scale. The knot values must also be sequenced in ascending order. Note that the knots are not input as an array, but in free format within the initial portion for this block. Thus, there is no need to specify a format.

An example of a KNOt block is:

```
@KNOTS  1.0, 2.0, 3.5, 5.0;
```

7.4.16 SPLine Block

The coefficients determining the monotone spline transformation for each subject are computed iteratively, and require initial values. Unless set otherwise, they are all given the initial value of one. Initial values can be set in this block. The matrix of spline coefficient initial values has as many rows as number of subjects and as many columns as number of interior knots plus two. That is, if there is only one interior knot (the default condition), there will be three coefficients per subject to input. Input is under the control of the same keywords as used in the NEWdata block.

7.4.17 WTDiss Block

The default model for the standard errors for both dissimilarities and subjects permits the standard error to vary from one subject to another. Occasionally it can be useful to specify these values for the initial calculations by MULTISCALE. This is especially so when random data are to be generated having varying standard errors, perhaps so as to simulate some model estimated from actual data.

In this block the NSUb standard errors for dissimilarities are input. These can appear organized as one or more rows of values, or as a column with one number per row. One is required for each subject.

An example of a WTDiss block is as follows:

```
@WTDISS;
(15f5.1)
  0.3  0.2  0.9  0.8  1.1  0.2  0.8  0.7  0.3  0.2  0.8  0.5  0.4  0.4  0.1
```

7.4.18 WTPref Block

In this block the standard errors for preferences or direct ratings are input, as in the WTDiss block above.

7.4.19 COMpute Block

This block is both an instruction to begin analysis of the data and an input of various parameters controlling the course of the computation. Therefore, it must be the final input block in an analysis. This block must either end the job deck, or be followed by a TITle block or a PARameters block describing a new analysis. The keywords that may be specified are listed in Table 7.8. In the table, some of the default values are described as "variable." This means that the program computes them internally according to some procedure, and the values may vary from analysis to analysis. As mentioned above, parameters set in this block are not passed to a subsequent analysis of the same data, and revert to their default values unless explicitly set for each analysis. The roles of the parameters in this block are described in greater detail in Section 3.

An example of a COMpute block is as follows:

```
@COMPUTE  ITMAX=10, ITXMAX=4, ITWMAX=4, CONV=0.1;
```

Table 7.8: Keyword Items Accompanying the COMpute Block

Keyword	Possible Values	Default Value	Comments
ITMax	≥ 0	30	Number of main iterations. Within each main iteration the configuration (X), the metric matrices (W), the ideal points (Y), and the spline coeffs. (P) are modified a number of times.
ITXmax	≥ 0	1	Number of configuration iterations within each main iteration
ITWmax	≥ 0	1	Number of metric matrix iterations
ITYmax	≥ 0	1	Number of ideal point iterations
ITPmax	≥ 0	1	Number of spline coefficient iterations
CONv	≥ 0	0.05	Convergence crit. for log likelihood
XCOnv	≥ 0	variable	Convergence crit. for configuration
WCOnv	≥ 0	variable	Convergence crit. for metric matrices
YCOnv	≥ 0	variable	Convergence crit. for ideal points
PCOnv	≥ 0	variable	Convergence crit. for spline coeffs.

Chapter 8

MULTISCALE Output

Output from the personal computer version of MULTISCALE is directed to a file named by the user when the program is invoked. MULTISCALE prompts for file names for input and output. In the mainframe version, however, output is automatically directed to FORTRAN file number 6, which is the usual output file for printed output.

The results from the analysis of data by MULTISCALE are organized into separate sections. Each section is numbered allowing easy reference to a particular type of result. Because of the many options available in MULTISCALE, no one analysis is apt to produce all sections. For example, one section displays the coefficients for the monotone spline transformations, and will appear only if the TRA=SPLine phrase appears in the PARameters input block. Also, some sections are apt to be very long and thus are present only if specifically requested. An example would be the plot of transformed dissimilarity versus distance for each subject.

Output from MULTISCALE can be in one of two styles. The default style assumes 80 characters per line and the output is arranged with that line length in view. It is also possible to specify that output will be contained within 132 characters per line and centered, as one would wish for line printer output from mainframe computers. This style results when the WIDe logical item is specified in the PARameters block, while narrow page style is forced by the WIDe item. An optional page eject can be performed before each section by including the EJEct logical item in the PARameters block.

Each output section is described below and an example of output in that section is provided. In most cases the output displayed results from the analysis of the data and job presented in Figure 7.1 in the previous chapter, except that some lines are eliminated for purposes of compression. Appendix C contains an example of wide line centered output. The output examples presented in this chapter are intended to be representative, but the actual material displayed for any particular analysis will vary depending on the options requested and the data analyzed.

8.1 Problem Definition and Input Quantities

In this section the requested analysis options are indicated, as well as the basic dimensions of the problem. Default values of various quantities are also displayed where their values would not be obvious. For example, the various convergence criteria relevant to the run are displayed along with the initial configuration matrix. If missing data are detected, the number of observations treated as missing is output. The input data matrices are displayed if the LIStdata logical item was present in the PARameters block

An example of output appearing in Section 1 for the job set up in Figure 7.1 is shown below.

```
MULTISCALE ANALYSIS NUMBER  1

     RECREATION DATA

SECTION  1.  PROBLEM DEFINITION AND INPUT QUANTITIES

ANALYSIS NUMBER  ...........................       1
NUMBER OF SCALED OBJECTS OR STIMULI  .......      10
NUMBER OF DIMENSIONS  ......................       2
NUMBER OF SUBJECTS OR REPLICATIONS  ........      15
MAXIMUM NUMBER OF MAIN ITERATIONS  .........      50
MAXIMUM NUMBER OF CONFIGURATION ITERATIONS         1
CONVERGENCE CRITERION FOR MAIN ITERATIONS       .50E-01
CONVERGENCE CRITERION FOR CONFIGURATION  ...    .15E-03
INDIVIDUAL METRIC MATRIX  ..................   IDENTITY
ESTIMATED DATA TRANSFORMATION TYPE  ........   POWER
```

```
ASSUMED ERROR DISTRIBUTION IS LOGNORMAL.
```

8.2 History of Computation

Here results for each main and secondary iteration are displayed. The results in this section include the current error sum of squares resulting from the fit to the data transformations defined at that iteration, the log likelihood, the standard error, and the gradient length with respect to the configuration.

Within any main iteration there are secondary iterations for the configuration X, the metric weights w_{rm}, the monotone spline transformations s_r, and the ideal point coordinates y_{rm}. The maximum number of each of these secondary iterations can be specified, or left to its default value of one. If zero iterations is specified, the quantity in question is not changed during the course of the analysis. Various error conditions are flagged.

Within any given secondary iteration, there are a number of tertiary iterations as the program searches for a minimum of a loss function along a specified line of search. These tertiary iterations are not displayed unless the DEBug option is specified. However if convergence is not achieved for these iterations (maximum number 5), then an error message will result and the history of this line search will be output to special file, called mpda.err, and a message that output to this file has taken place is printed out. The program will continue however and the occasional failure to achieve convergence in line searches is not serious. Warning messages also result if the rank of any matrix of expected second derivatives is less than expected by the program. For the most part, these messages can be ignored since recovery strategies within the program result in unimpeded progress towards the final solution. If many occur, however, the DEBug option should be requested and some expert advice sought.

Normally the history of secondary iterations is not displayed since this would greatly increase the amount of output. Complete information on the progress of all iterations can be displayed by specifying the DEBug logical item in the parameters block.

An abbreviated example of output in Section 2 for the Figure 7.1 job is as follows:

```
SECTION  2.  HISTORY OF COMPUTATION

MULTISCALE ANALYSIS NUMBER 1 IS IN PROGRESS.

ITER.  LOG LIKEL.  DISS. ERROR SS    GRAD. LNGTH.

   0   -1661.914         549.469       61.259
   1   -1559.740         428.615       10.290
   2   -1541.793         432.034        2.636
   3   -1538.121         451.816        1.485
   4   -1536.251         471.554         .810
   5   -1535.014         490.753         .796
   6   -1534.170         508.881         .734
   7   -1533.526         526.555         .780
   8   -1532.988         543.237         .862
   9   -1532.505         559.776         .932
  10   -1532.079         575.622         .837
  11   -1531.708         591.378         .784
  12   -1531.420         606.530         .594
  13   -1531.193         621.481         .519
  14   -1531.023         635.866         .389
  15   -1530.889         649.971         .329
  16   -1530.786         663.603         .252
  17   -1530.702         676.933         .210
  18   -1530.634         689.866         .164
  19   -1530.577         702.516         .135
  20   -1530.529         714.839         .107
```

8.3 Post-Mortem Display of Iterations

As a summary of the progress of the iterations toward a solution, a post-mortem of the iterative process is optionally displayed. This shows the log likelihood and the gradient length with respect to the configuration on each main iteration. These quantities are also plotted against iteration number.

8.4 Some Useful Summary Results

After successful convergence of the main iterations has been achieved, or the maximum number of main iterations have been completed, the final results are displayed. These include the final log likelihood, the AIC statistic, the BIC statistic, an unbiassed estimate of standard error, the number of parameters used to fit the data, and the number of degrees of freedom for error. The final convergence status is also indicated. The AIC and BIC statistics are useful for comparing models where there is not a nesting relationship between them, and are described in Akaike (1974) and Schwarz (1978).

In addition to the final results based on the options chosen, corresponding results are displayed when all the entries in a matrix are fit by a single constant. This best-fitting constant is estimated separately for each subject. A solution of this sort can be naturally thought of as a zero dimensional solution and provides a useful reference model with which the distance model can be compared. An asymptotic chi-squared test of the improvement in fit is computed. If this chi-squared value is not significant, one should conclude that the multidimensional scaling model used is not accounting for any structure in the data.

Section 4 output for the recreation data in Figure 7.1 is as follows:

```
SECTION  4.  SOME USEFUL SUMMARY RESULTS

LOG LIKELIHOOD  ............................   -1530.529
2 X LOG LIKELIHOOD  ........................   -3061.058
AIC STATISTIC  .............................    3181.058
BIC STATISTIC  .............................    3451.941
NUMBER OF PARAMETERS  ......................      60
NUMBER OF DEGREES OF FREEDOM FOR ERROR  ....     615
DISS MAX. LIKELIHOOD STD. ERROR ESTIMATE ...       1.029

CONVERGENCE WAS ACHIEVED AFTER  20  ITERATIONS.

THE SAME RESULTS FOR ZERO DIMENSIONS ARE:

LOG LIKELIHOOD  ............................   -1714.026
2 X LOG LIKELIHOOD  ........................   -3428.051
```

```
AIC STATISTIC ............................   3458.051
BIC STATISTIC ............................   3525.772
NUMBER OF PARAMETERS .....................     15
NUMBER OF DEGREES OF FREEDOM FOR ERROR ....   660
DISS MAX. LIKELIHOOD STD. ERROR ESTIMATE ...      .703
CHI SQUARE FOR COMPARING TWO FITS .........   366.993
NUMBER OF DEGREES OF FREEDOM FOR CHI SQUARE    45
PROBABILITY OF EXCEEDING THIS CHI SQUARE ...      .000
```

8.5 Final Configuration Estimate:

The final configuration matrix is output in three ways: first, in floating point form with three decimal places, second, in rounded integer format, and third, in polar coordinate form. The location of the decimal place may be shifted prior to these displays in order to have about two digits to the left of the decimal for most values. A message is output if the decimal place is shifted.

An example is:

```
SECTION  5.  FINAL CONFIGURATION ESTIMATE

 IN ALL OF THE ENTRIES TABLED BELOW THE DECIMAL POINT
 HAS BEEN MOVED  1  PLACES TO THE LEFT.

                   1          2

  1  READING      -.498   -15.324
  2  TV          22.382      .741
  3  HOCKEY      37.695   -10.949
  4  BALLET     -21.651     4.524
  5  ARTMUSM     -6.076     2.970
  6  CONCERT    -19.110    -2.936
  7  THEATRE    -17.370     3.505
  8  CONFEREN    -8.611   -25.733
  9  WINDOWSH    10.093    21.851
 10  FASHSHOW     3.145    21.352
```

```
RESULTS IN POLAR COORDINATE FORM:
    FIRST VALUE IS DISTANCE FROM THE ORIGIN,
    OTHER VALUES ARE ANGLES IN DEGREES FROM FIRST DIMENSION

                         1

  1  READING     15.3   268
  2  TV          22.4     2
  3  HOCKEY      39.3   -16
  4  BALLET      22.1   168
  5  ARTMUSM      6.8   153
  6  CONCERT     19.3   189
  7  THEATRE     17.7   168
  8  CONFEREN    27.1   251
  9  WINDOWSH    24.1    65
 10  FASHSHOW    21.6    82


STD. DEVS. AND VARIANCES FOR EACH DIMENSION:

   STD. DEV.  VARIANCE  % VARIANCE
 1       18        328         62
 2       14        200         37
```

8.6 Interpoint Distances

The distances between points for an identity metric matrix are displayed in shifted integer format. For the sake of brevity an example is omitted.

8.7 Final Metric Weight Estimates

In this section, which occurs only if either the DIAGONAL or FULL metric options is requested, the metric matrix for each subject is displayed. For the DIAGONAL option, it consists of weights for each dimension. The following sample comes from the analysis of the data in Figure 7.1 using the

METric=DIAgonal option

```
SECTION  7.  FINAL METRIC MATRIX ESTIMATES

                  1         2

  1             1.043     0.955
  2             0.799     1.167
  3             1.186     0.770
  4             1.125     0.857
  5             0.729     1.212
  6             0.819     1.153
  7             0.752     1.198
  8             1.413     0.062
  9             1.336     0.465
 10             0.252     1.392

RESULTS IN POLAR COORDINATE FORM:

                         1

  1             1.4    42
  2             1.4    56
  3             1.4    33
  4             1.4    37
  5             1.4    59
  6             1.4    55
  7             1.4    58
  8             1.4     3
  9             1.4    19
 10             1.4    80
```

8.8 Ideal Direction or Point Coordinates for Ratings

If auxiliary variable observations are included, and if these are direct ratings, this section contains the quantities y_{rm} defining either the ideal directions or the ideal points. The appearance of this section is very similar to that of the following section, for which an example is provided.

8.9 Ideal Direction or Point Coordinates for Preferences

If auxiliary variable observations are included, and if these are pairwise preferences, this section contains the quantities y_{rm} defining either the ideal directions or the ideal points. The following example of this section appears in the output from the analysis shown in Figure 7.2.

```
SECTION  9.  FINAL IDEAL POINTS FOR PREFERENCES

                       1         2


  1  AYOTTE      -4.396  -282.357
  2  BAKER     -236.787  -520.021
  3  BAXTER    -102.735  -151.278
  4  CORONEL   -242.697  -393.319
  5  DEMPSEY   -157.069  -310.241
  6  FOREY     -204.277   710.018
  7  HABIB     -123.583  -202.575
  8  HANLEY    -210.343  -293.516
  9  KIRKWOOD  -209.373   -17.333
 10  LIGHT     -334.437  -116.424
 11  MARR       -52.655  -141.439
 12  PERREAUL  -104.441  -161.448
 13  POWIOZNY  1625.267    26.680
 14  SHAKIN     -84.531  -175.525
 15  TARINI      27.795  -462.040
```

8.10 Standard Error Weight Estimates

These weights are post-hoc estimates. The weights have a lower limit of zero and an average of one. They indicate the relative variability of judgments specific to each stimulus and are helpful in identifying stimuli for which perceptions vary considerably from subject to subject.

The results in this section for the demonstration job are

```
SECTION 10.  POST-HOC STANDARD ERROR WEIGHT ESTIMATES

    INPUT ORDER                SIZE ORDER

  1  READING   1.55        3  HOCKEY     .15
  2  TV        1.56        8  CONFEREN   .42
  3  HOCKEY     .15        4  BALLET     .53
  4  BALLET     .53        9  WINDOWSH   .54
  5  ARTMUSM   1.63        6  CONCERT    .67
  6  CONCERT    .67       10  FASHSHOW  1.14
  7  THEATRE   1.81        1  READING   1.55
  8  CONFEREN   .42        2  TV        1.56
  9  WINDOWSH   .54        5  ARTMUSM   1.63
 10  FASHSHOW  1.14        7  THEATRE   1.81
```

8.11 Final Regression Coefficients and Exponents for Dissimilarities

If the default power transformation option was used, a regression coefficient v_r and an exponent p_r are displayed for each subject by order of subject and by order of size. If TRAnsform=SCAle or TRAnsform=SPLine options were selected, only regression coefficients are displayed. The demonstration job produces the following results

```
SECTION 11.  FINAL CONSTANTS AND REGRESSION COEFFS.
                 FOR LOG DISSIMILARITY

                  CONSTANT   REGRESSION       REGRESSION
                             COEFFICIENT COEFFICIENT BY SIZE

  1  AYOTTE         2.89          1.06       12  PERREAUL   .95
  2  BAKER         -2.02          3.27        7  HABIB      .98
  3  BAXTER         2.38          1.34        1  AYOTTE    1.06
  4  CORONEL      -10.01          5.21       15  TARINI    1.07
  5  DEMPSEY        1.62          1.61        9  KIRKWOOD  1.15
  6  FOREY          1.91          1.57        8  HANLEY    1.15
  7  HABIB          3.10           .98       14  SHAKIN    1.22
  8  HANLEY         2.72          1.15        3  BAXTER    1.34
  9  KIRKWOOD       2.72          1.15       10  LIGHT     1.35
 10  LIGHT          1.88          1.35        6  FOREY     1.57
 11  MARR         -16.45          7.05        5  DEMPSEY   1.61
 12  PERREAUL       3.59           .95       13  POWIOZNY  2.32
 13  POWIOZNY       -.25          2.32        2  BAKER     3.27
 14  SHAKIN         2.78          1.22        4  CORONEL   5.21
 15  TARINI         3.13          1.07       11  MARR      7.05

 MEAN                              2.09
 STANDARD DEVIATION                1.78
```

8.12 Final Monotone Spline Coefficients

If the TRAnsform=SPLine option is selected, the coefficients defining the monotone spline transformation for each subject are output. These can be used as input to another program to produce various displays or analyses of these transformations.

8.13 Final Regression Coefficients and Exponents for Dissimilarities

If auxiliary variables are used, there are also regression coefficients p_r and possibly constants v_r estimated for these variables, as well as for the dissimilarity data. These are output here.

8.14 Within-Subject Standard Error Estimates and Multiple Correlations for Dissimilarities

The two main goodness of fit indices for the dissimilarity data computed for each subject are output here. Both should be viewed as conditional on the dissimilarity transformations estimated. These are helpful in identifying subjects whose judgments are especially badly fit by the model chosen. Both quantities are also output by order of size.

An example is

```
SECTION 14.  WITHIN-SUBJ. DISS. STD. ERROR ESTIMATES
                   & MULTIPLE CORRS.

                  WITHIN-SUBJECT            WITHIN-SUBJECT
                  STANDARD ERROR          MULTIPLE CORRELATION
                UNORDERED    ORDERED      UNORDERED    ORDERED

   1  AYOTTE       .58    14      .43           .72     2      .10
   2  BAKER       2.41    15      .47           .10     4      .32
   3  BAXTER       .74     9      .54           .63    13      .40
   4  CORONEL     1.57     1      .58           .32     6      .43
   5  DEMPSEY      .60     7      .59           .71    11      .59
   6  FOREY       1.23     5      .60           .43    10      .59
   7  HABIB        .59    12      .61           .71     3      .63
   8  HANLEY       .62     8      .62           .69     8      .69
   9  KIRKWOOD     .54     3      .74           .74    12      .70
  10  LIGHT        .81    10      .81           .59     5      .71
```

```
11  MARR         .83  11    .83        .59   7     .71
12  PERREAUL     .61   6   1.23        .70   1     .72
13  POWIOZNY    1.31  13   1.31        .40   9     .74
14  SHAKIN       .43   4   1.57        .81  15     .79
15  TARINI       .47   2   2.41        .79  14     .81
```

8.15 Within-Subject Standard Error Estimates and Multiple Correlations for Preferences

If preference auxiliary data are included in the analysis, the same goodness of fit measures computed above for dissimilarity are also computed for preferences and displayed here. No such measures are displayed for direct rating data because the amount of information for each subject is usually not sufficient to justify computing these values.

8.16 Standard Errors of Estimate of Coordinates

Asymptotic standard error estimates for each coordinate in the configuration are displayed here. Note that these are computed based on the somewhat unrealistic assumption that all other parameters such as those defining the transformations are not estimated from the data. Thus, they should be treated as lower bounds for what the actual standard errors are likely to be. This is especially so when individualized metrics or SPLine transformations are used. However, they are corrected to some extent based on simulation results with small sample sizes using an IDEntity metric and POWer transformation (see Ramsay, 1980, for more details) and are probably fairly reasonable in that case.

In the sample job the results for this section are

```
SECTION 15.  STANDARD ERRORS OF ESTIMATE OF COORDINATES

STIMULUS  1  READING
```

```
COORDINATES:            -4.97665  -153.24130
STANDARD ERRORS:        12.79587    13.30638
CORRELATION MATRIX FOR COORDINATE ESTIMATES
    1     1.000      .056
    2      .056     1.000

       ... and so on ...

STIMULUS 10  FASHSHOW
COORDINATES:            31.45437   213.52430
STANDARD ERRORS:        16.03794    14.85428
CORRELATION MATRIX FOR COORDINATE ESTIMATES
    1     1.000     -.072
    2     -.072     1.000
```

8.17 Plots of the Configuration for Each Pair of Dimensions

In this section the configuration is plotted for each pair of dimensions. Each point is plotted using the plotting symbol found in the legend; that is the first I letters of the alphabet. In each plot the vertical and horizontal ranges and scales are the same. Axes are not explicitly displayed since as discussed in the first chapter the coordinate system or orientation of the orthogonal Cartesian coordinates used by MULTISCALE to compute distances should not be considered as necessarily the right system for interpretive purposes. Nevertheless, the configuration is displayed in principal axis orientation. This implies that the first dimension is that direction in which the points vary the most, the second the direction of highest variation orthogonal to the first, and so on. It does happen fairly frequently that this orientation is helpful in interpreting the results, especially when a larger number of dimensions is involved.

The configuration for the recreation data in the following display has been clipped slightly for display here.

```
SECTION 17.  PLOTS OF THE CONFIGURATION FOR EACH PAIR OF DIMENSIONS
           LEGEND

PLOTTING  STIMULUS
 SYMBOL
    A       1  READING
    B       2  TV
    C       3  HOCKEY
    D       4  BALLET
    E       5  ARTMUSM
    F       6  CONCERT
    G       7  THEATRE
    H       8  CONFEREN
    I       9  WINDOWSH
    J      10  FASHSHOW
MULTIPLE POINTS        #
```

```
  -280.0 -210.0 -140.0  -70.0     .0    70.0  140.0  210.0  280.0  350.0
       .      .      .      .      .      .      .      .      .      .
 247.1
 229.4                                J      I
 211.8
 194.1
 176.5
 158.8
 141.2
 123.5
 105.9
  88.2
  70.6
  52.9       D
  35.3           G          E
  17.6                                                   B
    .0
 -17.6         F
 -35.3
 -52.9
 -70.6
 -88.2
-105.9                                                                C
-123.5
-141.2                            A
-158.8
-176.5
-194.1
-211.8
-229.4
-247.1                    H
-264.7
      .      .      .      .      .      .      .      .      .      .
  -280.0 -210.0 -140.0  -70.0     .0    70.0  140.0  210.0  280.0  350.0
        CONFIGURATION PLOT FOR DIM. 2 (VERTICAL AXIS) AGAINST DIM. 1
```

8.18 Plots of the Dimension Weights for Each Pair of Dimensions

If the METric=DIAgonal option was chosen, the dimension weights are plotted here, each subject being identified by a letter. A legend precedes the plots. The following example is from the analysis of the Figure 7.1 data using the diagonal metric option.

```
SECTION 18.  PLOTS OF THE DIMENSION WEIGHTS
             FOR EACH PAIR OF DIMENSIONS

          LEGEND

PLOTTING  SUBJECT
 SYMBOL

    A       1  AYOTTE
    B       2  BAKER
    C       3  BAXTER
    D       4  CORONEL
    E       5  DEMPSEY
    F       6  FOREY
    G       7  HABIB
    H       8  HANLEY
    I       9  KIRKWOOD
    J      10  LIGHT
    K      11  MARR
    L      12  PERREAUL
    M      13  POWIOZNY
    N      14  SHAKIN
    O      15  TARINI
MULTIPLE POINTS        #
```

```
   .0     .2     .4     .6     .8    1.0    1.2    1.4    1.6
    .      .      .      .      .      .      .      .      .
1.5
1.5  M
1.4    BD        A                                 . .
1.4                                               .
1.3                        #                   . .
1.2                            I              .
1.2                                        ..
1.1                                L     .
1.1                                     .
1.0                                  ..
 .9                                .
 .9                              ..
 .8                            .
 .8                           .
 .7                        ..                  N
 .6                      .                      #
 .6                   . .
 .5                     .
 .5                  ..
 .4                 .
 .4               .                                    #
 .3            . .
 .2           .                                        K
 .2        ..
 .1      .                                              F
 .1     .
 .0
    .      .      .      .      .      .      .      .      .
   .0     .2     .4     .6     .8    1.0    1.2    1.4    1.6
DIMENSION WEIGHTS PLOT FOR DIM. 2 (VERTICAL AXIS) AGAINST DIM. 1
```

8.19 Within-Subject Plots and Residual Analyses for Dissimilarities

For a more fine-grained analysis of each subject's data, some within-subject results can be displayed. These are:

1. A table of transformed dissimilarities, original dissimilarities, computed distances, and normalized residuals. This results if the TABles logical item occurs.

2. A plot of transformed dissimilarities against fitted distances. This results if the DPLot logical item occurs.

3. A plot of transformed dissimilarities against original dissimilarities. This occurs if the TPLot logical item occurs.

4. A quantile plot consisting of a plot of ordered normalized residuals against the quantiles of the standard normal distribution. Such plots have proven to be very useful for detecting outliers and departures from the distributional assumptions. This results if the QPLot logical item occurs.

These results require a page per subject each, so that a request for them will generate a considerable amount of output when many subjects are involved.

Examples of these plots can be found in Appendix B.

8.20 Within-Subject Plots and Residual Analyses for Preferences

If preference auxiliary data are included, the same plots as in the previous section can also be displayed, with the exception that there is no plot of transformed preference versus preference (the equivalent of the DTPlot logical item). These plots are enabled by including the PPLot logical item in the PARameters block.

8.21 Graphics Output (HALO Version)

The personal computer version of MULTISCALE can produce graphics on the following devices:

- the screen of the computer
- plotting devices using the HP-GL plotting language used on Hewlett-Packard plotting equipment such as the HP 7475 series plotters
- laser printers using the Postscript language
- dot matrix printers

These plots are enabled using the PLOt keyword in the PARameters block. Note: use of these graphics displays requires the HALO graphics subroutine and software package. A license to use the software must be obtained from Media Cybernetics before it can be used.

Further comments on each of these types of displays are given below:

PLOt=SCReen If the graphics display is only on the screen, then there is nothing further to do. The program prompts the user when it is ready to begin plotting, and when ready, the user presses any key to begin plotting. After prompting for various features of the plot, such as whether stimulus labels and confidence ellipsoids are to be displayed, the results are plotted for each pair of dimensions. The program waits for a key to be pressed before going on to the next pair of dimensions.

PLOt=HPGl If the graphics display is to be on a device processing the HP-GL plotting languaged developed by Hewlett-Packard for its series of pen plotters, such as the HP 7475A, then the program produces an additional file for each pair of dimensions. These files are given names with the same stem as the input file and with extensions of the form `.pmn` where "m" is the vertical dimension number and "n" is the horizontal dimension number. Thus, if there are only two dimensions in the analysis of the `funseekr.msl` file, then a single file named `funseekr.p21` will be produced.

Each of these files, called "display files" in the HALO documentation, must then be processed by the program `AHDHP`. The purpose of this

HALO system program is to produce HP-GL commands. The program can either output the commands to a serial port for direct control of the plotter, or can store than in another file for subsequent output to the serial port. Complete details on how to use this program are available in the HALO documentation. However, the following command outputs HP-GL commands directly to the serial port, configured for 9600 baud no parity, single stop bit, eight data bit transmission:

```
ahdhp funseekr.p21 -H1
```

The plotting device is assumed to be an HP 7475A plotter. The following command, on the other hand, outputs the HP-GL commands to file `funseekr.hp`:

```
ahdhp funseekr.p21 -H1 -P0,10365,7962 -Ofunseekr.hp
```

PLOt=POSt If the output is to be displayed on a laser printer processing the Adobe Postscript language, this option can be used. Just as for HP-GL commands, a display file is produced which must be processed in a subsequent step by the HALO program AHDPS. The following command, for example, will direct output from this program to file `funseekr.ps`:

```
ahdps funseekr.p21 -P0,1222,1582 -K0,0,2 -Ofunseekr.ps
```

PLOt=PRInter With this command the graphics output will appear on the screen and will be printed through the parallel port to a dot-matrix printer. The printer driver incorporated into the program should work for most printers, but HALO has other drivers for specific printers. The program prompts for a printer driver file, so that an alternative can be substituted during execution. This option is nice for getting a cheap, reasonably rapid hard copy of the results where high resolution is not important.

PLOt=SCReen This command causes the graphics output to appear on the computer's display only.

Actually, both the HP-GL and Postscript options produce display files either of which can be processed to produce plots of the other kind. The only difference is that the Postscript option uses a slightly more attractive font which, however, also requires more RAM memory. The font used in the HP-GL version requires a minimum of plotting commands, and is more appropriate for the slower pen plotting equipment.

When the program attempts to load a font into RAM, it may fail due to insufficient memory. Aside from rebooting the system eliminating unnecessary memory resident software, one may also try graphics output with the PLOt-HPGl command (which will also produce screen graphics and can be processed to produce Postscript plots) since this uses a font requiring substantially less memory.

Chapter 9

The Emotions Data: A Sample Analysis

This chapter presents a typical MULTISCALE analysis. The data are ratings on a 9-category rating scale of dissimilarities among 14 emotions. The ten subjects were participants in a workshop on multidimensional scaling. They received the following instructions:

```
This questionnaire is concerned with the similarity and
dissimilarity of various emotions with respect to your experience
of them.  Think about these 14 emotions, and tick those you have
experienced at least once.

Affectionate ___
Afraid       ___
Angry        ___
Despising    ___
Eager        ___
Fascinated   ___
Guilty       ___
Happy        ___
Panicky      ___
Passionate   ___
Rejected     ___
Sad          ___
```

```
Satisfied    ___
Surprised    ___

You will now be presented with a series of emotions taken two at
a time from the list.  For each of these pairs, your task is to
decide how similar or how different those two emotions are in
your experience and then to indicate your judgment using the
nine-point scale next to it.  For instance, if you decided that
the two emotions had nothing in common, you would indicate this
by circling a 9 on the scale (Very Different).  If on the other
hand, you felt the emotions were very much alike, then circle a 1
on the scale (Very Similar).  The scale is provided to allow you
to indicate the degree of similarity- dissimilarity you feel
exists between the two emotions, so please try to use the full
range of the scale in making your judgments.  If you felt that
two emotions are different but not too different, use a 6 or 7;
if they are similar but not very similar use 3 or 4. The
important thing is that you make your judgments in the way that
seems right for you, for your experience of the emotions.
```

A typical pair in the questionnaire was then presented as follows:

```
1.                     Eager - Satisfied
           Very Similar                Very Different
               1   2   3   4   5   6   7   8   9
```

The data resulting from this study are presented in Figure 9.1, which also includes the appropriate job setup for the first analysis to be performed.

The strategy in the analysis is to begin with a minimal plausible model, referred to as the *baseline model*, and then to extend it in various ways. The baseline model will depend on prior experience and intuitions associated with the problem. In many situations, however, this can be the default model for MULTISCALE. This provides a two- dimensional fit using the identity metric, power transformations, subject-specific variance components, and the lognormal distribution assumption.

It can be helpful to use the one-character codes in Table 9.2 in recording the results of various extensions of the baseline model.

Table 9.1: Job Setup and Data for Emotions Study

```
@TITLE  LINES=2;
      EMOTIONS FOR MEMBERS OF MDS WORKSHOP
      BASELINE MODEL
@PARAMETERS  NSTIM=14, NDIM=2, NSUB=10, EJECT;
@NEWDATA  VECTOR;
(11X,50F1.0)
SUBJECT  1 77794913693342162967197899399642966996399694199999
           399249738936922499769152439699799999921413
SUBJECT  2 34562513362451232524299779198939857598769884387797
           674638756957344488679292334299778892 55422
SUBJECT  3 44452313444332213525298678478533757394579584177479
           658437877984364293469593343397289872 43212
SUBJECT  4 44574218325342227562477789599866949894858378188788
           895747688887774688779253753678479864 65513
SUBJECT  5 54833424483962446543158558688646988488468883178877
           882279762868144289449553743697349972 44424
SUBJECT  6 34293314344249423562494539899532834583179742199248
           882219762954221299339491322193789481 22211
SUBJECT  7 22132513223241122331198599998824828583379145198959
           594187879623573287229292386389199393 12311
SUBJECT  8 43344533435342453333298768799737889774768576497777
           793347477776673775448793443377578863 44456
SUBJECT  9 66736815633462322736197689398438476494378493299867
           347429578923795389349382736499499999 34612
SUBJECT 10 65233824546343333556287887399335588796579585998888
           597438987968488486579382474489989984 45328
@STIMLABS FORMAT=FREE;
SATISFY,FASCINAT,SURPRISD,EAGER,HAPPY,PASSION,AFFECTN,DESPISE,
PANICKY,AFRAID,GUILTY,SAD,ANGRY,REJECTED;
@COMPUTE;
```

Table 9.2: Character Codes Designating Models

Character	Meaning
n	an integer designating the number of dimensions. If absent 2 is implied.
D	diagonal metric
C	no subject-specific variance components γ_r
S	scale transformation
M	monotone spline transformation
N	normal distribution

One can append these codes to a name for the data to indicate the nature of the extension involved. For example, if we designate these data by "EMO", then EMO3DM designates analysis using three dimensions, a diagonal metric, and monotone spline transformations.

It is also helpful to compile summaries of each analysis into a table in which the major aspects of each can be quickly retrieved. Table 9.3 is an example of how this might be done. It contains the results of a number of extensions of the baseline model as well as the baseline results.

Each line of this table contains an identification of the analysis, the final value of the log likelihood function, the AIC statistic, the BIC statistic, the global standard error, the number of independent parameters estimated, the number of iterations, whether or not convergence was attained, and a measure of execution time. The last quantity will depend on the computer installation involved, and the values in this table are in terms of McGill Computing Center execution units.

The value of each extension can be assessed in various ways. The decrease in the AIC and BIC statistics provide some indication. A more statistical basis is offered by considering the chi squared statistic (4.6 discussed in Section 2 along with its normalized version (4.7). These values are tabulated in Table 9.4. Associated with each extension is its degrees of freedom, defined as the number of of additional parameters introduced by the extension. Also in Table 9.4 are the standard normal deviates corresponding to each chi squared value using transformation (4.7).

Table 9.3: Summary Table for MULTISCALE Analyses of Emotions Data

Analysis	Results								Comments
	Log L	AIC	BIC	Sigma	No. Par	No. Its	Con ver	Ex. Tim	
Baseline	-986.0	2078	2333	.642	53	14	Y	32	
3	-963.9	2056	2364	.457	64	10	Y	29	
4	-938.4	2025	2381	.334	74	21	Y	57	
D	-978.0	2086	2399	.656	65	11	Y	45	
M	-787.2	1718	2070	.560	73	50	N	187	

Table 9.4: Assessments of Various Extensions of the Baseline Model

Extension	Degrees of Freedom	AIC Decrease	BIC Decrease	Chi Squared	Z
3	11	22	-31	44	4.3
4	21	53	-48	95	6.5
D	12	-8	-66	16	0.9
M	20	360	260	398	16.3

On the basis of Table 9.4 some conclusions can be made concerning the best model for the data. Since both the AIC statistic and the BIC statistic are best when they are small, one looks for the largest decrease in each. For both this occurs when monotone spline transformations are fit for each subject. The chi squared statistic or the normalized version of it in the final column tells the same story. The indications for other possible extensions are mixed, however. The BIC tends in general to favor models with fewer parameters and this shows up in terms of the spline transformation being the only single extension that improves the baseline model. Both the AIC and the chi squared statistics are more liberal and both indicate that increasing dimensionality will improve the fit as well as using either pair-wise or stimulus-wise variance components. All statistics indicate that little is to be gained for these data by going beyond the identity metric.

The final representation chosen used monotone spline transformations. One might have used three or more dimensions in addition, but for simplicity of interpretation only two were used. The final row of Table 9.3 contains the results for these models. In Table 9.3 it should be noted that MULTISCALE did not converge in 30 iterations and that the log likelihood was still decreasing substantially. This is typical; fitting monotone spline transformations usually results in rather slow convergence and long computation times. The final log likelihood resulting from allowing iterations to proceed to convergence would probably be substantially higher than the value in Table 9.3. Nevertheless, the estimates of the various parameters in the final model should be satisfactory for most purposes after only 30 iterations.

The complete output from the final analysis is given in Appendix B. The results for each output section are discussed below.

9.1 Section 1.

The various parameters and computation specifications are displayed here. Most of these are the default values. Thus by default only one secondary iteration per main iteration was allowed for the configuration, standard error weights, and monotone spline transformations. A single interior knot was permitted and the value that MULTISCALE used is displayed below the initial configuration. The various default convergence criteria are also indicated.

9.2 Section 3.

The post-mortem display found in this section provides a useful compact summary of the progress to convergence. As the table indicates, the log likelihood was still increasing by about 0.5 at the 30th iteration. The plot of loglikelihood versus iterations shows that it was at least in the final phases of maximization, and one can estimate that the true maximum would have been in the vicinity of -770. The final plot shows the relation between log gradient length and iterations, and it is typical that this relation is usually not very smooth.

9.3 Section 4.

A summary of final results is contained here. Both an unbiassed and a maximum likelihood estimate of global standard error are provided. The former is corrected for the number of parameters involved in the fit and thus somewhat more useful in general. As indicated the ratio of number of degrees of freedom for error to number of parameters is about 10. This is a fairly healthy ratio for descriptive purposes and also permits us to take reasonably seriously the various hypothesis tests carried out in Table 9.4. Unlike the Section 4 results displayed in Chapter VI here we have no results for a benchmark model, because when monotone spline transformations are fit MULTISCALE as currently designed cannot fit either the zero-dimensional or additive model. The fit of the baseline model was superior, however, to either benchmark and thus this comparison is not really needed since this model in turn is much superior to the baseline model.

9.4 Section 5.

The final configuration estimate is recorded here in numerical terms with the Cartesian coordinate system results displayed in both floating point and rounded integer form. For purposes of having a fixed level of precision in the tables MULTISCALE has multiplied all coordinates by ten before the display. The first dimension accounts for 85% of the variation among the points and thus is highly dominant.

9.5 Section 6.

The final matrix of interpoint distances is displayed in rounded integer form, again with entries multiplied by ten before display.

9.6 Section 8.

The final standard error weight estimates σ_r are displayed here. They vary widely about one and this is the reason that this extension of the baseline model was important. The emotion "satisfied" was one for which there was considerable concensus among subjects and served as a sort of reference point among the stimuli. By contrast subjects disagreed strongly with respect to the place of "passionate" and "surprised" among the other emotions.

9.7 Section 10.

The final regression coefficients have little interpretive value, but are displayed in this section for completeness.

9.8 Section 11.

This section actually is displayed before Section 10 for computational reasons and contains the final B-spline coefficients. If the transformations were linear, these would all be about one for each subject. The value of any coefficient can be roughly interpreted as the size of the derivative of the transformation at that knot value. Most transformations have much higher derivatives at the final knot value than at the previous two, indicating a postively accelerated function. The actual transformations are displayed in Figure 9.1. It can be seen from both the knot coefficients and the figure that subject 8 required a somewhat different transformation than the others. Only his transformation could be approximated at all well by a power function, which is why the use of the monotone spline extension made such a large contribution to the fit.

Figure 9.1: Final monotone spline tranformation for emotions data. The single knot is indicated by the vertical dashed line.

9.9 Section 12.

The final within-subject statistics show that the model fit all subjects' data reasonably well. The values of the multiple correlation, which have a median of 0.71, are typical of fairly experienced and careful subjects.

9.10 Section 13.

The standard errors of estimate for coordinates are measures of the lack of precision in coordinate estimates. When compared to the coordinates themselves in Section 5 one can arrive at some impression of how well they are estimated. A better impression is given in the figure presented below.

9.11 Sections 15 and 16.

These sections graphically display the final configuration. For only two dimensions the display in Section l6 is obviously much more useful, although the line plots do show how much more variation there is in the first dimension than in the second. The positive emotions are to be found to the right in the configuration plot and the negative ones to the left. The more active emotions are typically in the lower part of the plot and the passive ones in the upper part.

Figure 9.2 displays the final configuration with each point surrounded by an ellipse indicating 95% confidence regions for the location of the population point. This gives us an idea of how well the locations of the points are defined by the amount of data that we have available.

Figure 9.2: Final configuration for the emotions data. The ellipses give 95% confidence regions for the location of the true points.

Chapter 10

The Recreation Data with Preferences: A Sample Analysis

In this chapter a sample analysis is given showing how to incorporate auxiliary preference data. The setup of the data is illustrated in Figure 7.2.

The baseline model for these data is an ideal direction model for preferences using two dimensions and the identity metric. This baseline model is used for both the lognormal and normal distribution assumptions. Table 10.1 indicates the results of trying various models in terms of the goodness of fit statistics. Table 10.2 compares various pairs of models using the chi square criterion and the transform of chi square to a standard normal variate using (4.7).

Here are some conclusions:

1. The normal distribution produced better results than the lognormal distribution for all models.

2. Clearly we need three dimensions. Whether we should go to four could be argued. The improvement in fit is significant, but not nearly as much so as for other upgrades of the model.

3. The ideal point model also does better than the ideal direction model for the identity metric, but not for the diagonal metric.

4. Going to a diagonal metric does improve the fit significantly, and is probably worth doing.

5. A minimal model for these data seems to be a three-dimensional diagonal metric model using the normal distribution. This is also the model which gives the minimum of the AIC criterion, which is a popular model-selection rule.

   Figures 10.1, 10.2, and 10.3 show the final solution for the three-dimensional diagonal metric ideal direction model using the normal distribution assumption.

Figure 10.1: Final configuration for the recreation data for dimensions 1 and 2. The ellipses give 95% confidence regions for the location of the true points.

Figure 10.2: Final configuration for the recreation data for dimensions 1 and 3. The ellipses give 95% confidence regions for the location of the true points.

Figure 10.3: Final configuration for the recreation data for dimensions 2 and 3. The ellipses give 95% confidence regions for the location of the true points.

References

Akaike, H. (1974) A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19,* 716-723.

de Boor, C. (1978) *A Practical Guide to Splines.* New York: Springer-Verlag.

Fiacco, A. V. and McCormick, G. P. (1968) *Nonlinear programming: Sequential Unconstrained Minimization Techniques.* New York: Wiley.

Kruskal, J. B. (1964) Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika, 29,* 1-27.

Kruskal, J. B., Young, F. W., and Seery, J. B. (1973) How to use KYST, a very flexible program to do multidimensional scaling and unfolding. Bell Laboratories (unpublished).

Kruskal, J. B. and Wish, M. (1978) *Multidimensional Scaling.* Beverly Hills, CA: Sage Publications.

Nakatani, L. H. (1972) Confusion choice model for multidimensional psychophysics. *Journal of Mathematical Psychology, 9,* 104-127.

Ramsay, J. O. (1968) Economical method of analyzing perceived color differences. *Journal of the Optical Society of America, 58,* 19-22.

Ramsay, J. O. (1973) The effect of number of categories in rating scales on precision of estimation of scale values. *Psychometrika, 38,* 513-532.

Ramsay, J. O. (1977) Maximum likelihood estimation in multidimensional scaling. *Psychometrika, 42,* 241-266.

Ramsay, J. O. (1978) Confidence regions for multidimensional scaling analysis. *Psychometrika, 43,* 145-160.

Ramsay, J. O. (1980a) The joint analysis of direct ratings, pairwise preferences, and dissimilarities. *Psychometrika, 45,* 149-165.

Ramsay, J. O. (1980b) Some small sample results for maximum likelihood estimation in multidimensional scaling. *Psychometrika, 45,* 139-144.

Ramsay, J. O. (1982) Some statistical approaches to multidimensional scaling data (With discussion). *Journal of the Royal Statistical Society, Series A, 145,* 285-312.

Ramsay, J. O. (1988) Monotone regression splines in action. *Statistical Science, 3,* 425-441.

Ramsay, J. O. (1990) MATFIT: A FORTRAN subroutine for comparing two matrices in a subspace. *Psychometrika, 55,* 551-553.

Ramsay, J. O., Styan, G. P. H., and ten Berge, J. (1984) Matrix correlation.. *Psychometrika, 49,* 403-423.

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

Schiffman, S., Reynolds, M., & Young, F. (1981) *Introduction to Multidimensional Scaling.* New York: Academic Press.

Schumaker, L. (1981) *Spline Functions: Basic Theory.* New York: Wiley.

Schwarz, G. (1978) Estimating the dimension of a model. *Annals of Statistics, 6,* 461-464.

Shepard, R. N. (1957) Stimulus and response generalization: a stochastic model relating generalization to distance in a psychological space. *Psychometrika, 22,* 325-345.

Shepard, R. N. (1962) Analysis of proximities: Multidimensional scaling with an unknown distance function. I and II. *Psychometrika, 27,* 125-140, 219-246.

Takane, Y. (1981) Multidimensional successive categories scaling: a maximum likelihood method. *Psychometrika, 46,* 9-27.

Takane, Y. and Carroll, J. D. (1981) Nonmetric maximum likelihood multidimensional scaling from directional rankings of similarities. *Psychometrika, 46,* In press.

Takane, Y., Young, F. W. and de Leeuw, J. (1977) Nonmetric individual differences multidimensional scaling: an alternating least squares method with optimal scaling features. *Psychometrika, 42,* 7-67.

Torgerson, W. (1958) *Theory and Methods of Scaling.* New York: Wiley.

Wilson, E. B. and Hilferty, M. M. (1931) The distribution of chi square. *Proceedings of the National Academy of Science, 17,* 684-688.

Winsberg, S. and Ramsay, J. O. (1982) Monotone splines for dimension reduction. Submitted for publication.

Wish, M. and Carroll, J. D. (1974) Applications of individual differences scaling to studies of human perception and judgment. Pp. 449-491 in E. C. Carterette and M. P. Friedman (Eds.) *Handbook of Perception, Vol. 2,* New York: Academic Press.

Table 10.1: Summary Table for MULTISCALE Analyses of Recreation Data

Analysis	Distn.	Results							
		Log L	AIC	BIC	Sigma Diss.	Sigma Pref.	No. Par	No. Its	Con ver
Baseline	Lognor.	-3090.6	6361	6830	1.091	6.318	90	22	Y
P		-3073.7	6357	6904	0.999	6.122	105	24	Y
3		-3043.0	6310	6893	.761	6.056	112	18	Y
4		-3013.8	6294	6986	.732	5.844	133	31	Y
D		-3053.8	6318	6864	1.019	6.374	105	50	N
D3		-2968.9	6224	6968	0.592	6.174	143	34	Y
P3		-3030.7	6315	6977	0.716	5.897	127	16	Y
P4		-3013.4	6323	7094	0.675	5.831	148	25	Y
PD3		-2979.2	6274	7097	0.566	6.034	158	31	Y
Baseline	Normal	-3004.6	6189	6658	5.884	6.384	90	19	Y
P		-2975.5	6161	6708	5.770	6.142	105	26	Y
3		-2946.6	6117	6700	5.718	6.046	112	18	Y
4		-2915.7	6097	6790	5.665	5.819	133	20	Y
D		-2981.9	6174	6720	5.083	6.410	105	12	Y
D3		-2899.1	6084	6829	4.850	6.135	143	12	Y
P3		-2935.0	6124	6785	5.603	5.930	127	17	Y
P4		-2919.4	6135	6906	5.658	5.859	148	24	Y
PD3		-2891.8	6100	6922	4.882	5.989	158	13	Y

Table 10.2: Assessments of Various Extensions of the Baseline Model

Extension	Distn.	AIC Decrease	BIC Decrease	Degrees of Freedom	Chi Squared	Z
P	Lognorm.	4	-74	15	34	6.1
3		51	-63	22	94	16.5
4		65	-56	43	154	18.7
3 vs 4		16	-93	21	38	4.8
D		43	-34	15	74	15.7
D vs D3		94	-104	38	170	22.5
P	Normal	27	-50	15	58	12.1
3		72	-42	22	116	20.3
4		92	-132	43	178	22.0
3 vs 4		20	-10	21	62	10.4
D		15	-62	15	45	9.0
D vs D3		90	-109	38	166	21.9
PD vs PD3		-16	-97	15	15	0.1