3 Scaling: The Basic Non-Metric Distance Model

The journey of a thousand miles begins with a single step.

LAO TZE

3.1 Ordinal Rescaling: Introduction

In using MDS, the user must pay attention to three things:

the data, which give empirical information on how the objects or stimuli relate to each other;

the model, which provides a set of assumptions in terms of which the data will be interpreted; and

the transformation, which is the rescaling which may legitimately be performed on the data to bring them into closer conformity to the model. This is usually referred to as the 'level of measurement' of the data.

The data

In the basic MDS model (frequently called 'smallest space analysis' or 'non-metric distance scaling') the data take the form of a square, symmetric 2-way table, whose entries indicate how similar or how dissimilar any two points are. (To avoid unnecessary repetition we shall assume that the data are dissimilarities, unless otherwise indicated). By convention, the entry in the ith row and jth column of the table is denoted δ_{ij}, and gives the value of the dissimilarity measure between object i and object j. Because the data are symmetric ($\delta_{ij} = \delta_{ji}$) and each object is considered to be identical to itself, the diagonal entries are ignored, and only one half of the matrix, usually the triangle below the diagonal, is presented (see Table 1.1 as an example).

The model

The model used in basic MDS is the simple Euclidean distance model described in section 2.1. In terms of this model, the data δ_{ij} will be interpreted as being 'distance-like'; not as actual distances, but as approximate or distorted estimates of distance. The aim of the MDS analysis is to turn such data into a set of genuine Euclidean distances. The solution (also called the 'final configuration') consists of an arrangement of points in a small number of dimensions, located so that the distance between the points matches the dissimilarities between the objects as closely as possible.

Column:	(1)	(2)	(3)	(4)	(5)	(6)
			DATA		DISTANCES	
	No.	Pair	Data	Rank	Real*	(Est.) Scaled**
		(i, j)	$\delta(i, j)$	$\rho(i, j)$	$d(i, j)$	$d(i, j)$
	1	(8, 2)	0.932	1	5.830	2.411
	2	(8, 1)	0.901	2	5.656	2.270
	3	(6, 1)	0.899	3	5.385	2.206
	4	(7, 6)	0.833	4	5.099	2.206
	5	(7, 3)	0.752	5 =	5.000	2.145
	6	(6, 2)	0.752		5.000	2.118
	7	(7, 2)	0.730	7 =	4.123	1.827
	8	(8, 7)	0.730		4.123	1.739
	9	(3, 1)	0.712	9 =	4.000	1.632
	10	(8, 3)	0.712		4.000	1.558
	11	(8, 4)	0.634	11	3.605	1.557
	12	(6, 4)	0.541		3.162	1.429
	13	(5, 2)	0.541	12 =	3.162	1.377
	14	(3, 2)	0.541		3.162	1.367
	15	(6, 5)	0.530	15 =	3.000	1.320
	16	(7, 1)	0.530		3.000	1.308
	17	(7, 4)	0.521		2.828	1.264
	18	(5, 3)	0.521	17 =	2.828	1.221
	19	(8, 5)	0.521		2.828	1.148
	20	(5, 1)	0.521		2.828	1.130
	21	(4, 3)	0.364		2.236	0.972
	22	(7, 5)	0.364		2.236	0.941
	23	(8, 6)	0.364	21 =	2.236	0.895
	24	(6, 3)	0.364		2.236	0.882
	25	(4, 2)	0.364		2.236	0.866
	26	(4, 1)	0.364		2.236	0.779
	27	(2, 1)	0.211	27	1.414	0.586
	28	(5, 4)	0.007	28	1.000	0.532

*Strong monotonicity, secondary approach to ties
**Weak monotonicity, primary approach to ties

Table 3.1 *Ordinal rescaling of the data from Table 1.1*

The transformation

The ordinal or monotonic* transformation used in non-metric MDS assumes that only the rank order of the entries in the data matrix contains significant information. Consequently the distances of the solution should, as far as possible, be in the same rank order as the original data. For this reason, non-metric MDS is sometimes referred to as 'ordinal rescaling analysis' (Sibson 1972).

The purpose of the basic non-metric MDS procedure, then, is to find a configuration of points whose distances reflect as closely as possible the rank order of the data. This

*A *monotone* (or monotonic) *increasing* quantity is one which never decreases. (a *monotone decreasing* quantity is one which never increases). Hence, a monotonic transformation of data preserves their order, and in this text the terms 'monotonic' and 'ordinal' are used interchangeably.

is done by trying to find an ordinal rescaling of the data which transforms them into Euclidean distances.*

3.1.1 Perfect ordinal rescaling

To illustrate ordinal rescaling, let us return to the data originally presented in Table 1.1. First, the 28 entries of the data matrix are sorted into order. Column 2 of Table 3.1 gives the (column, row) location of each entry in the matrix, and column 3 gives the actual dissimilarity value (the data). Thus the highest dissimilarity, between object 8 (receiving) and object 2 (rape), has a value of 0.932 and the lowest dissimilarity, between object 5 (libel) and object 4 (perjury), has a value of 0.007. The rank number of each data entry is given in column 4. Note that there is a goodly number of tied data values, including six with the same value of 0.364. In all, only 14 distinct values appear in the data.

In column 5, the original data have been ordinally rescaled into a set of Euclidean distances which correspond to the two-dimensional configuration presented in Figure 3.1. (How the ordinal rescaling was obtained and how the configuration was produced need not concern us at this point. It is only important to see that a configuration has been obtained whose distances are a perfect rescaling of the original data).

The Shepard diagram

It is always instructive to look at the shape of the ordinal transformation function. This is done by producing a 'Shepard diagram' (named after Shepard's seminal paper of 1962), where the data dissimilarities and the distances of the solution are first plotted against each other, and then the ordinal transformation is depicted by joining the points in an upward direction, as is done in Figure 3.2. Since we are dealing with interpoint distances rather than co-ordinates, there will be $p(p-1)/2$ values contained in the Shepard diagram; with eight points, as here, there are 28 such values. For instance, the bottom left hand point is the one corresponding to (5, 4), whose dissimilarity value is 0.007, and the corresponding distance value is 1.000. The next point up corresponds to (2, 1), with $\delta_{21} = 0.211$ and $d_{21} = 1.414$, and there then follows a point representing the six entries whose data value is 0.364 and whose distance value is 2.236. In all, there are clearly 14 distinct data and distance values. When the points are joined, it can be seen that the transformation function between the data dissimilarities and the solution distances is perfectly monotone, i.e. always moves upwards and to the right. (If the data were similarities, the direction of transformation function would be downward and to the right).

Strong and weak monotonicity

This particular rescaling function illustrates a *strong* (or *strict*) monotonic relationship between the data and the distances, defined in the following way:

Strong monotonicity: Whenever $\delta_{ij} < \delta_{kl}$ then $d_{ij} < d_{kl}$

*The exposition of the basic MDS model in the following sections is primarily based upon the Shepard (1962) and Kruskal (1964a, b) procedure for non-metric MDS, supplemented by the work of Guttman (1968) and Lingoes and Roskam (1973).

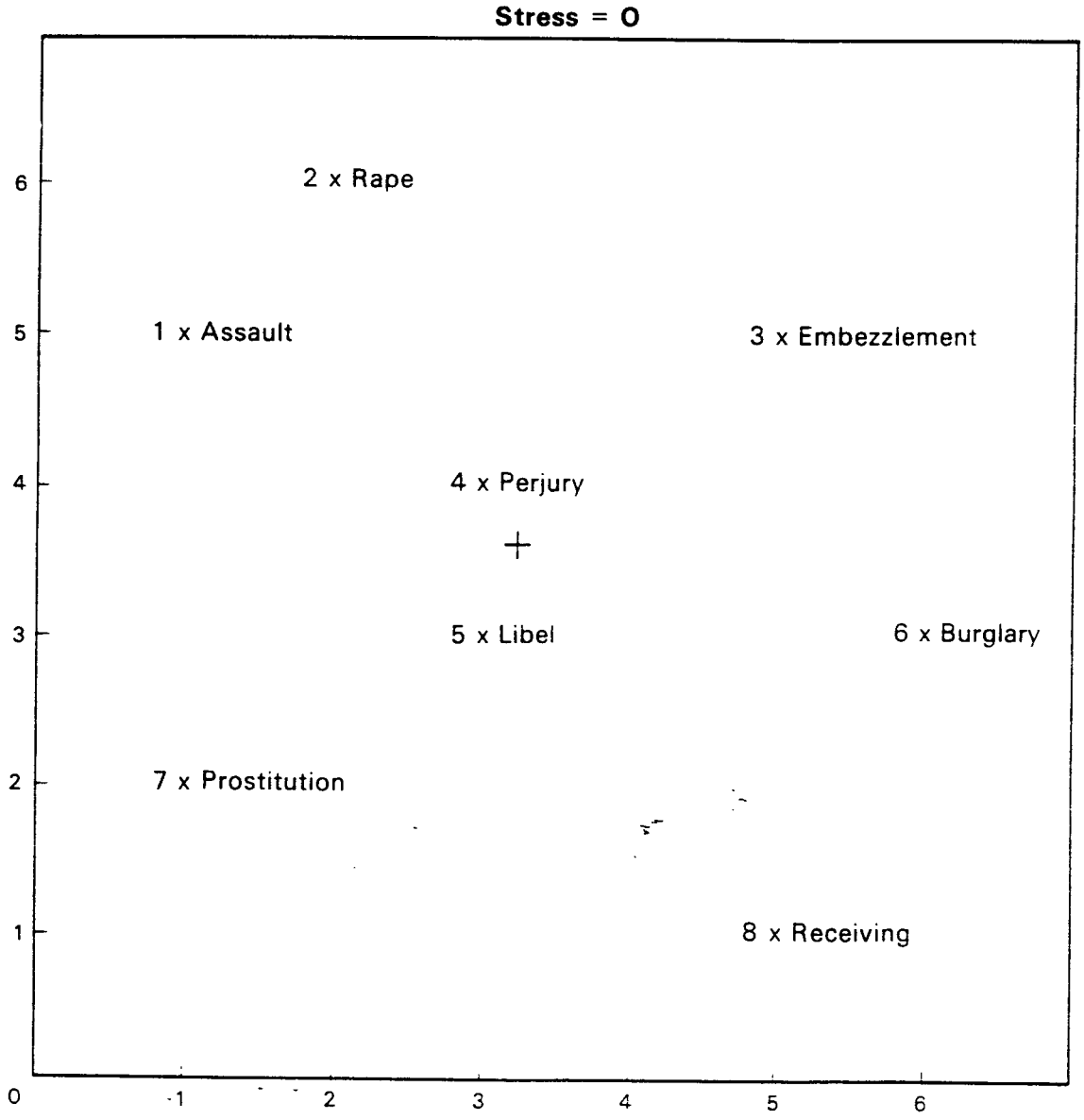


Figure 3.1 *Two-dimensional configuration generating distances of Table 3.1*

That is, if one datum is less than another then the corresponding distances must be in the same order.

A less restrictive requirement often encountered in scaling is that a *weak* monotonic relationship holds between data and distances. Weak monotonicity only requires that no *inversions* in order should occur between the data and distance. That is, that if $\delta_{ij} < \delta_{kl}$ then it should never be the case that $d_{ij} > d_{kl}$. However, in the case of weak monotonicity note that d_{ij} may equal d_{kl}, even when $\delta_{ij} < \delta_{kl}$.

Weak monotonicity: Whenever $\delta_{ij} < \delta_{kl}$ then $d_{ij} \leqslant d_{kl}$

In this case, the transformation function moves upward (even vertically upward) but it may never move downwards. Figure 3.6 gives an example of a weak monotonic transformation of the same data.

3.1.2 An illustrative example: Scottish mileages

People often need convincing that it is really possible to derive distance information purely from the rank order of pairwise dissimilarities. A further, less artificial, example should persuade doubters. (In addition, the following example

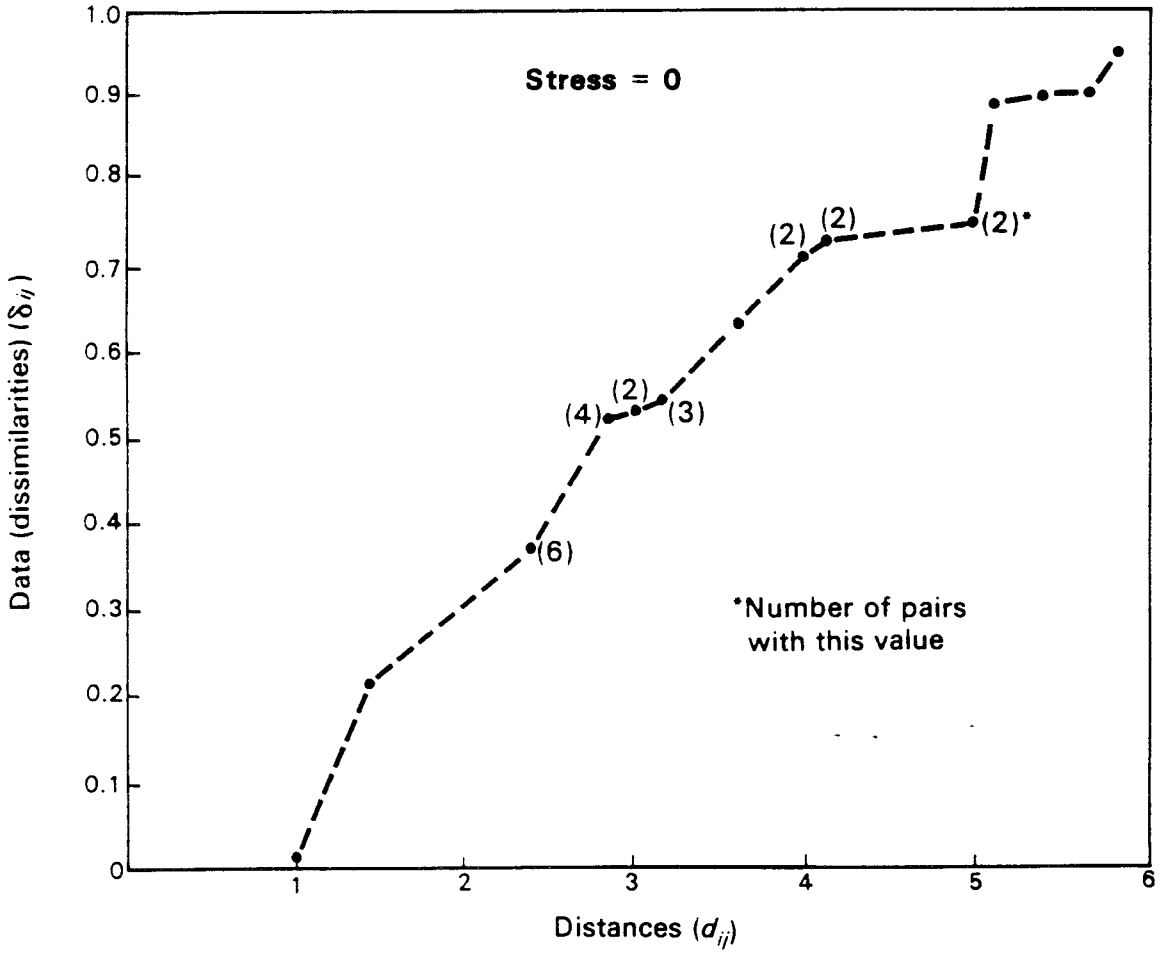
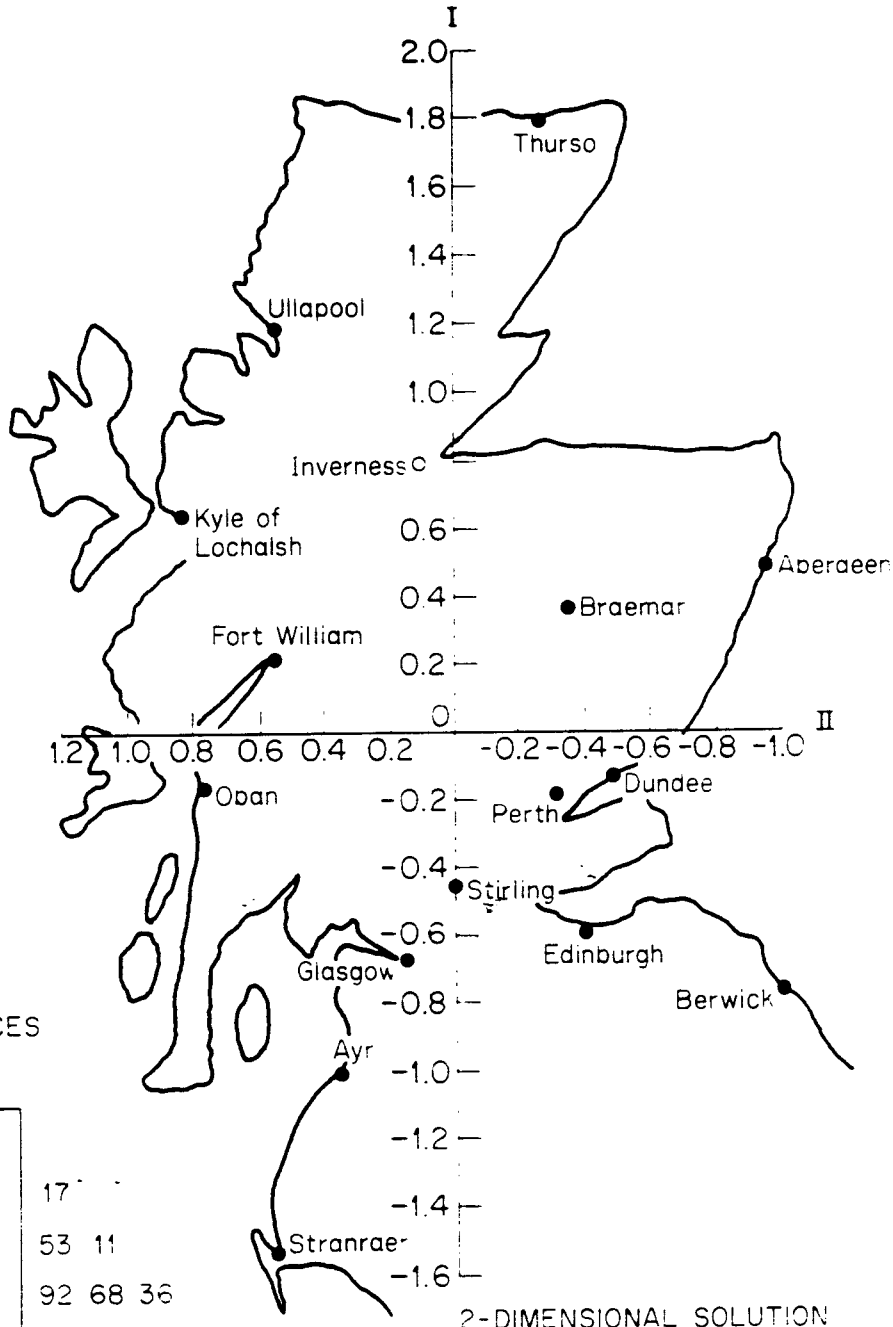


Figure 3.2 *Strong monotone rescaling of data*

involves a monotonic transformation with a regular or smooth shape, and serves to introduce some further ideas of 'badness of fit').

Sixteen towns on the mainland of Scotland were chosen, their distances were measured on a map with a ruler, and a small degree of error (inaccuracy) was added to the distances. The 120 distances were then reduced to rank order, and the rank numbers used as data. These are presented in the bottom left hand corner of Figure 3.3. (There are 92 distinct values, so the number of tied data values is much less than in the previous example).

These data were submitted to MINISSA, the program in the MDS(X) series implementing the basic model, and the configuration presented in Figure 3.3 was produced. (The map outline is drawn in freehand, and the dimensions were rotated counterclockwise through 90° to give the northern orientation to the configuration.) Obviously, it is an excellent recovery of the original configuration of 16 towns. The corresponding Shepard diagram is presented in Figure 3.4. In this example, the monotonic 'line' has not been drawn in, because the smooth, regular shape of the relationship is clear simply by looking at the pattern of the points. The relationship between ranks and distances is linear in the main range (say between 0.5 and 2.00 along the distance axis), but over the entire range the relationship is



NON-METRIC:
RANK-ORDER OF
PAIRWISE DISTANCES

Towns:															
1 Berwick															
2 Edinburgh	17														
3 Glasgow	53	11													
4 Stranraer	92	68	36												
5 Ayr	70	30	4	11											
6 Perth	34	7	19	83	45										
7 Dundee	27	8	29	93	58	1									
8 Aberdeen	63	56	83	115	103	35	24								
9 Stirling	43	4	2	58	21	3	14	66							
10 Oban	99	57	26	72	36	43	63	98	29						
11 Ft. William	96	60	39	89	62	41	49	79	33	6					
12 Inverness	100	75	75	112	97	45	49	48	60	52	22				
13 Kyle of Lochalsh	111	89	78	107	89	72	81	94	70	26	9	23			
14 Braemar	67	36	51	106	80	15	10	19	30	65	30	15	54		
15 Ullapool	114	105	101	119	109	85	87	86	88	68	40	16	17	55	
16 Thurso	117	113	115	120	119	103	102	77	110	107	95	47	62	74	42
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 3.3 *Data (rank of mileages) and solution of Scottish distances.*

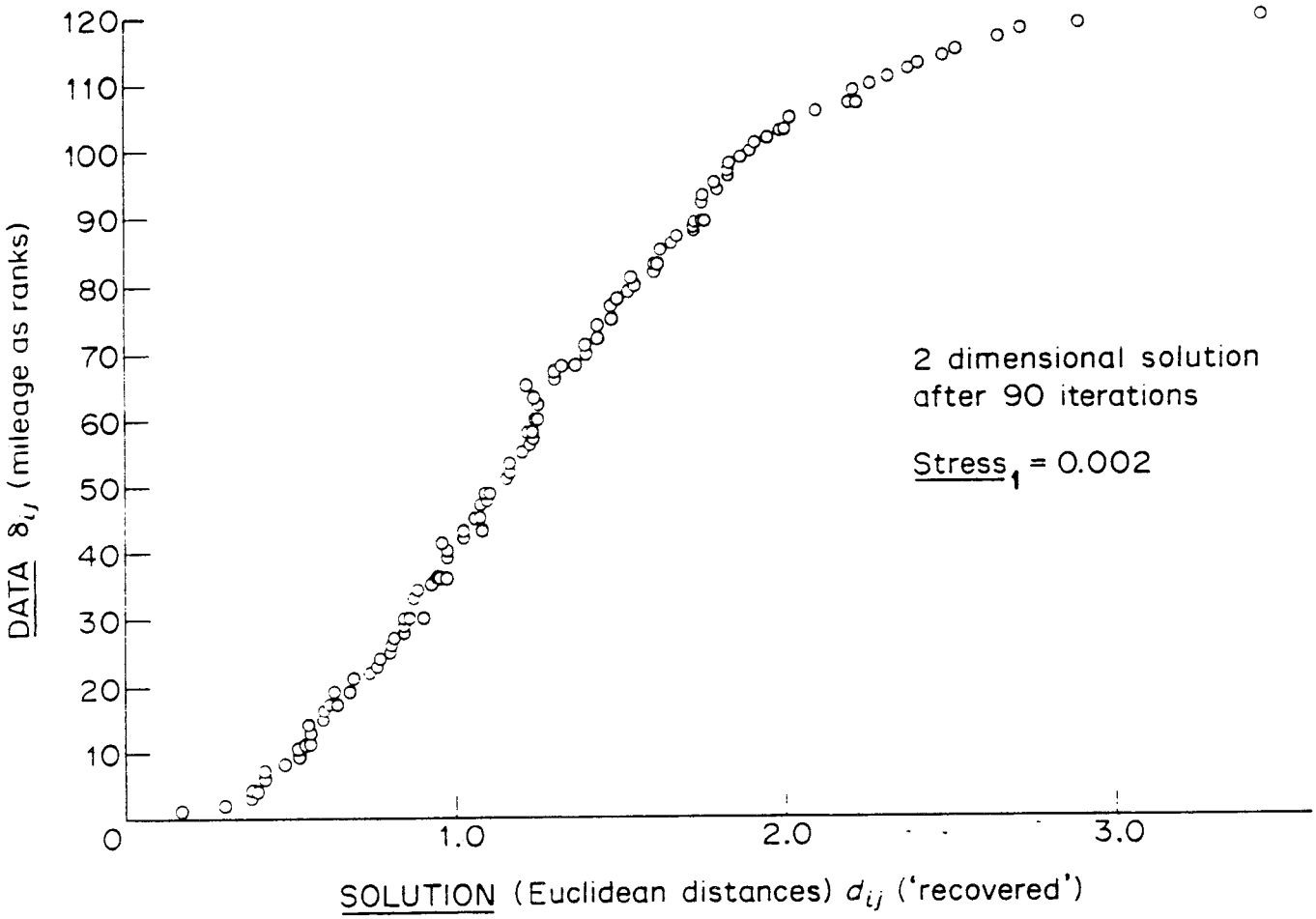


Figure 3.4 *Shepard diagram: Scottish distances*

close to an S-shaped (logistic) curve, where the initial smallest values increase slowly, and the final largest values decrease slowly.

But notice that a few rank mileages are not well fit—if a line were to be drawn through all points, it would occasionally have to move *downwards* and to the right to accommodate all the points, contrary to the requirement of monotonicity. True, there are very few such instances, but they are sufficient to show that, even in the case of slightly imperfect error-prone data, it will not always be possible to define a perfect monotonic rescaling. Instead we shall have to talk about fitting or estimating a monotonic 'line', about errors (or departures from a monotonic relationship), and these will be used to define *stress* as an overall measure of fit.

3.2 Monotone Regression: The basic ideas

In using non-metric MDS a perfect ordinal rescaling of the data into distances is usually not possible. What is sought is as good a rescaling as can be achieved. Later, in section 3.4, it will be seen that this involves finding a *series* of configurations in which the interpoint distances come more and more closely into conformity with the data. For the present it simplifies matters to concentrate upon how well one particular configuration (or, strictly, its distances) matches the data. To do so it is crucial to grasp the important, but basically simple, ideas involved in what is termed 'monotonic (or ordinal, or isotonic) regression'. To illustrate the main ideas, we return to the previous example of rescaling the data of Table 3.1, which is illustrated in Figures 3.1 and 3.2.

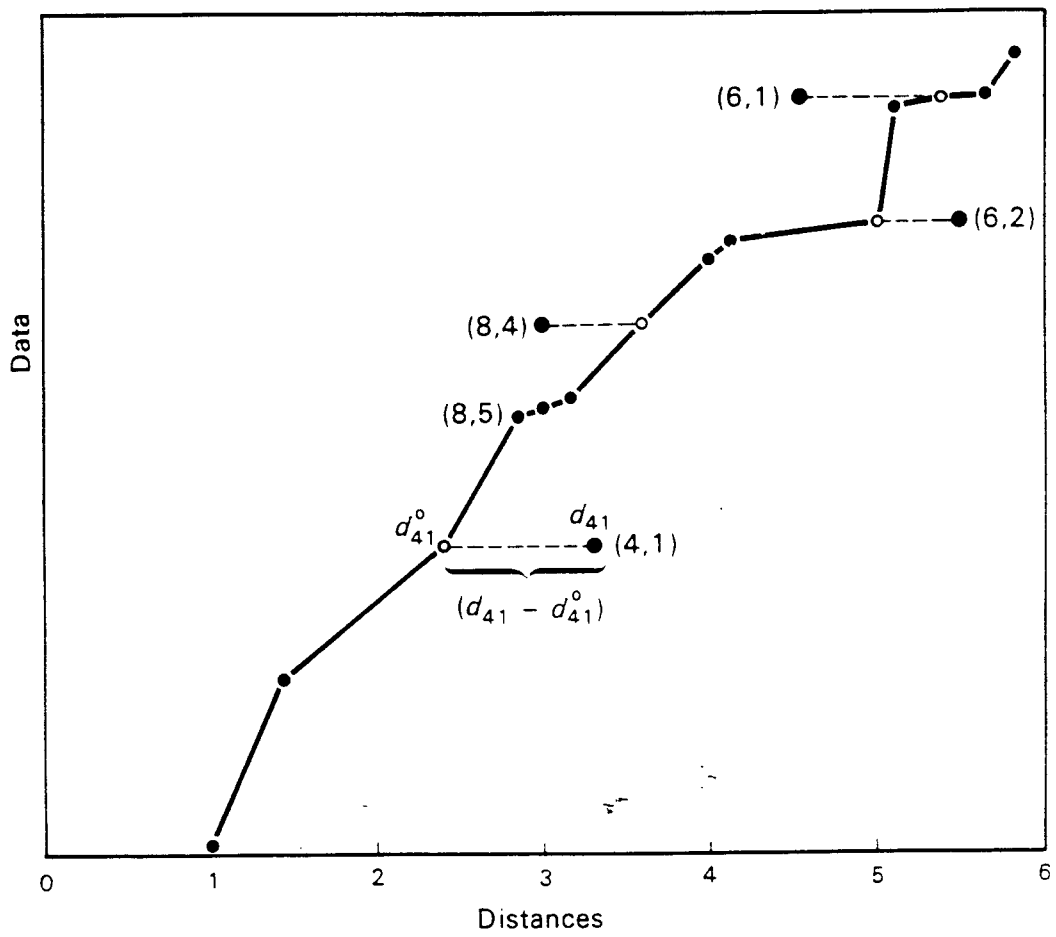


Figure 3.5 *Imperfect monotone rescaling of data*

Suppose points 1, 4 and 6 in the final configuration of this earlier example were moved slightly: the changed configuration would not fit the data so well, and when the Shepard diagram was constructed it would have the characteristic form of Figure 3.5. The distances between the objects (4, 1), (8, 4), (6, 2) and (6, 1) have now changed, and in such a way that the data and the distances can no longer be put into a perfect monotone relationship. For instance, the data indicate that $\delta_{41} < \delta_{85}$, whereas now $d_{41} > d_{85}$. This inversion in ordering means that it is impossible to construct a monotone 'line' through all the (black) points on the Shepard diagram. We are in a quandary: we must either follow the counsel of perfection, and declare that properly speaking an ordinal rescaling is not possible, or recognise the fallibility of the data, and construct as good an ordinal rescaling as possible. In actual fact, it is the second option which is usually followed. But to do so, we shall need to know how to construct 'as good a rescaling as possible', and we shall need a precise definition of what is meant by the 'fallibility' of the data.

This can be accomplished by means of a process of monotone regression. This involves the calculation of a new set of 'distances', often called 'pseudo-distances' (since they are not *actual* distances corresponding to any real configuration nor, indeed, need they obey the triangle inequality), or 'fitted distances' or 'disparities'. These three terms are used interchangeably and are denoted: d^0_{jk}. At this point, it is not important to know how they are calculated—that can wait until section 3.5.2. But it is important to know what the properties are and how they serve to measure the extent to which a given configuration fits the data.

The fitting values are ratio-level quantities defined as *'distances' which would preserve perfect monotonicity with the data.* Once calculated for each pair of points (j, k), the discrepancy between the actual distance (d_{jk}) and that disparity (d^0_{jk}), which would give a perfect monotonic solution (i.e., $d_{jk} - d^0_{jk}$), serves as a basis for measuring how far the distances of the configuration depart from those 'pseudo-distances' necessary to keep perfect order with the data. Note that these differences between the distances and the fitting values are calculated along the *distance* axis. The reason for this is that we have committed ourselves to regarding the data as ordinal and any arithmetic operations involving them (in this case, subtraction) are illegitimate. In Figure 3.5 the disparities are denoted by white circles, in the cases where they differ from the actual distances.

3.2.1 Two types of fitting value

In non-metric MDS, two types of fitting quantities (disparities) are frequently used—one of which measures the deviation of the distances of the configuration from *weak* monotonicity, and the other which measures deviation from *strong* monotonicity. Both of these types are used in many MDS(X) programs, and it is important to see how they differ, and that strong monotonicity is usually bound to produce worse fit than weak monotonicity.

(i) *Weak monotonicity* (Kruskal 1964a)

In this case the pseudo-distances which are fitted in the monotone regression are denoted $\hat{d}_{jk}$ ('*d*-hat'), and are required to be *weakly* monotone with the data:

Weak mon: Whenever $\delta_{ij} < \delta_{kl}$ then $\hat{d}_{ij} \leqslant \hat{d}_{kl}$

That is, weak monotonicity allows unequal data to be fitted by equal disparities. When this happens it shows up on a Shepard diagram in the form of vertical lines in the monotone transformation function. In addition the disparity values, $\hat{d}_{jk}$ have the useful property of being as close as possible to the corresponding distances. This means that, over all the points of the configuration, the sum of the squared differences between the distances and the corresponding disparities is as small as possible, i.e.

$$\sum_{\substack{\text{all pairs} \\ (j,k)}} (d_{jk} - \hat{d}_{jk})^2 = \text{minimum}$$

In brief, Kruskal's fitting quantities, also referred to as BFMF (best-fitting monotone function estimates), are required to be both weakly monotone with the data and a least-squares fit to the actual distances.

(ii) *Strong monotonicity* (Guttman 1968)

In this case the pseudo-distances fitted in monotone regression are denoted d^*_{jk} ('*d*-star'), and are required to be *strongly* monotone with the data. They are often referred to as 'rank images' estimates:

Strong mon: Whenever $\delta_{ij} < \delta_{kl}$ then $d^*_{ij} < d^*_{kl}$

It should be noted that strong monotonicity does *not* allow unequal data to be fitted by equal disparities. In this case no vertical segments will appear on the Shepard diagram. Consequently, if the criterion of strong monotonicity is chosen, more discrepancies will occur between the data and the distances of the solution than if weak monotonicity is chosen. Moreover, it is not required that the d^* values be as close as possible to the actual distances, so the difference $(d_{jk} - d^*_{jk})$ will usually be larger than $(d_{jk} - \hat{d}_{jk})$.

As a result of these differences in definition, any overall measure of badness-of-fit between a particular configuration and a set of data is bound to be higher (i.e. worse-fit) when measured in terms of departure from Guttman's strong monotonicity requirement than when measured from Kruskal's weak monotonicity requirement.

In using MDS programs it is important to pay attention to which form of monotonic regression is being used. A full technical comparison of the differences and similarities is contained in Lingoes and Roskam (1973) and in Young (1973).†

3.2.2 Ties in the data

In a set of research data, the values will not normally be distinct: at least some values will be the same. The question arises: should equal dissimilarities be fit by equal disparities?

Two main answers have been given to the question, referred to as the 'primary' and the 'secondary' approach to ties:

Primary approach

Primary approach to ties: If $\delta_{ij} = \delta_{kl}$ then d^0_{ij} may, or may not, equal d^0_{kl}

This indulgent approach treats ties as indeterminate and allows fitting values either to preserve the equality or replace it by an inequality. In fact, the tie will be broken if in so doing the goodness of fit is improved. In a Shepard diagram, the primary approach to ties shows up characteristically in the form of horizontal straight lines in the monotone function (since identical dissimilarity values are allowed to be represented by different distance values). Figure 3.6 presents a perfect weak monotonic rescaling of the data of Table 3.1, *using the primary approach to ties*. Compare this with the strong monotonic rescaling in Figure 3.2. The configuration recovered by the two scalings is however, virtually identical.

Secondary approach

Secondary approach to ties: Whenever $\delta_{ij} = \delta_{kl}$ then $d^0_{ij} = d^0_{kl}$

On the other hand, in the secondary approach, ties in the data are required to be retained in the fitting values. Consequently, if the actual distances do not preserve

†Young shows that Kruskal's weak monotone and Guttman's rank image transformations mark the extremes of a continuum of a bounded, one-parameter, family of possible monotonic transformations. Although other possible variants have some desirable properties, they are not used extensively in MDS programs and they all seem able to recover metric information with about equal proficiency.

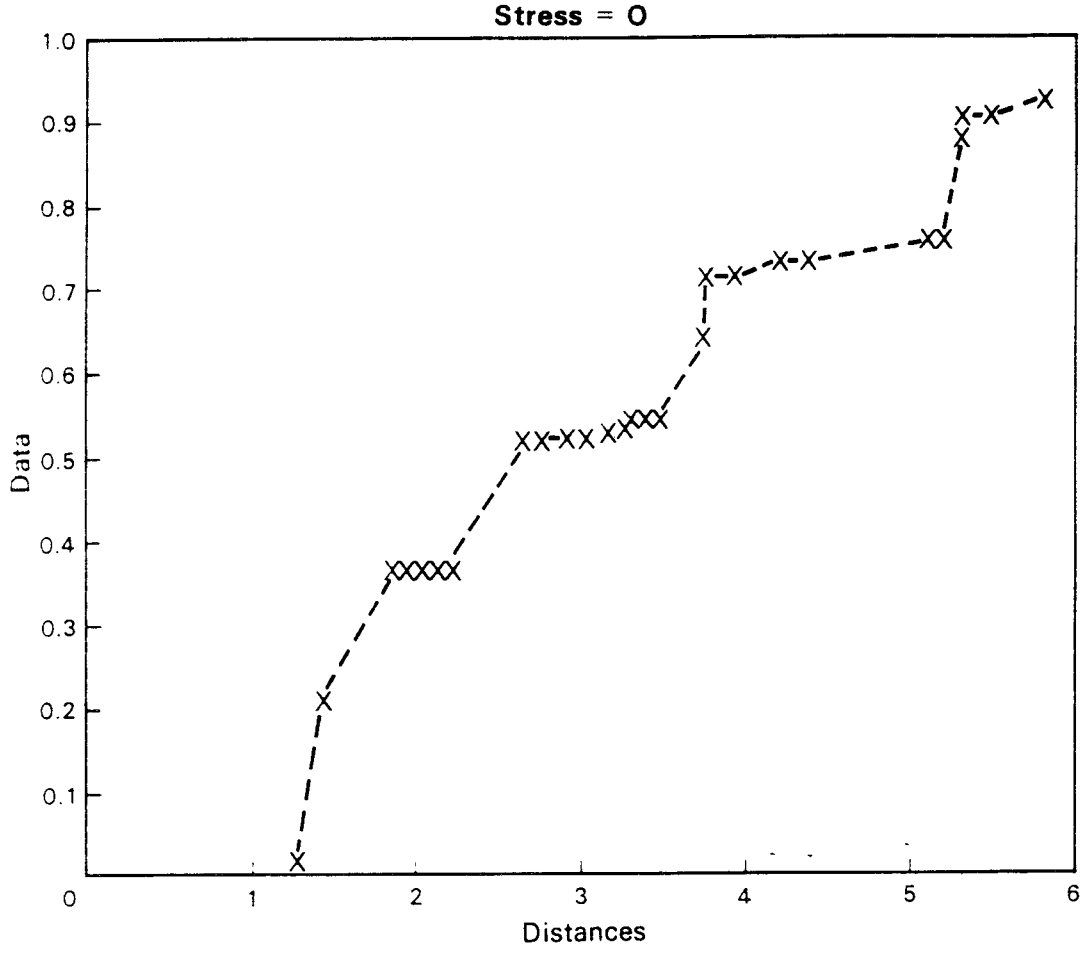


Figure 3.6 *Weak monotone rescaling of data*

every equality in the data, each infraction will be counted as a deviation from monotonicity. In effect, in the case of the secondary approach tied data are treated as being genuinely equivalent. (Note from Table 3.1 that Figure 3.2 is a perfect strong rescaling, using the secondary approach to ties.)

In general, the primary approach to ties should be used in preference to the secondary approach, especially if there is a fairly large number of distinct values in the data. Kendall (1971b, p. 313 et seq.) shows that adoption of the secondary approach can badly misrepresent the structure present. However, if there is only a small number of distinct dissimilarity values (as, for instance, when the data are ratings of similarity from a scale containing a very limited number of ordered categories) then allowing the program the additional indulgence of fitting equivalent category values by disparities in any order may destroy virtually all information.

Generally, MDS programs use the primary approach to tied data in obtaining a solution and, in the MDS(X) series, MINISSA(N), among others, offers the user the choice of primary or secondary approach.

The decision as to what values count as the same is far from trivial. Often data input will consist of numerical association coefficients, such as correlations, which have been calculated to several decimal places of accuracy. So long as two values differ—even if only in the final place—a non-metric program will treat them as distinct and attempt to find corresponding distance values which are also distinct. Such spurious exactness can be avoided by including coefficient values only up to

the desired level of exactness (e.g. by rewriting an INPUT FORMAT of: 10F8.5 as, say, 10(F5.2, 3X) in order to keep only two significant decimal places). Another alternative is to employ the parameter EPSILON in MINISSA(N) or the parameters TIEUP and/or TIEDOWN in SSA(M); in either case values will be treated as identical if they differ by less than a specified amount (see also 6.1.1).

3.2.3 Preservation of order information

Taken together, the monotonicity criterion (weak *vs* strong) and the approach to ties (primary *vs* secondary) produce somewhat different effects on the preservation of ordinal information (order inequalities and equal values) in the data, and the monotonic function has a slightly different form in each of the four cases. These alternatives are presented as a typology in Table 3.2.

Type I: weak monotonicity

This is the most commonly employed option, and the most indulgent one, since it allows maximum flexibility in rescaling the data ordinally. At best it can recover structure which is obscured by a good deal of error (cf. Kendall 1971b); at worst it can destroy virtually all significant information if it ties data which ought not to be equal and unties data which ought to be ordered.

Characteristically, the monotone regression function in this case is very 'steppy', with a number of both vertical segments (due to weak monotonicity allowing different data values to have the identical fitting value), and horizontal segments (due to the primary approach allowing identical data values to be fit by different differences). See Figure 3.6 for an example of this, although the only effect of weak monotonicity occurs in the single small vertical segment in the top right-hand corner. On the other hand, there are a number of obvious instances of the effect of primary ties (horizontal segments). In general, type I monotone regression functions are 'upward non-decreasing', and often include a number of right-angle steps.

Type II: semi-weak monotonicity

No horizontal segments appear on the function, since they are excluded by the secondary approach to ties. But vertical segments will usually occur. It is a combination which should be used when the user ascribes greater importance to tied information in the data than to the order information.

Type III: semi-strong monotonicity

In this case, no vertical segments occur on the function, since strong monotonicity precludes them. But the function will normally contain horizontal 'plateaux', indicating the operation of the primary tying option. This combination is used surprisingly infrequently, given that it seeks to preserve the significant order information, but gives the freedom to treat ties as indeterminate.

Type IV: strong monotonicity

This is the most restrictive option, where the function is strictly increasing, but can never be a step function. It should be used when the data are believed (or known) to be virtually error-free. An example appears in Figures 3.2 and 3.5. It can also serve as a salutary reminder of what a rigorous and uncompromising interpretation of ordinal measurement actually involves when applied to real data.

MONOTONICITY CRITERION	TYING OPTION CHOSEN: Primary (Indeterminate)	Secondary (Equivalence)
Weak (*Kruskal's* $\hat{d}$)	I WEAK *May equate unequal data* May untie tied data	II SEMI-WEAK *May equate unequal data* Preserves ties
Strong (*Guttman's* d^*)	III SEMI-STRONG *Preserves strict inequalities* ($>$, $<$) May untie tied data	IV STRONG *Preserves strict inequalities* ($>$, $<$) Preserves ties

Table 3.2 *Preservation of ordinal information in monotone regression*

Note This table and terminology are based upon Roskam (1969, pp. 9–11) and is reproduced with permission.

since every infraction will count as evidence against the hypothesis that the data can be perfectly rescaled ordinally into Euclidean distances.

Note that the question of how to calculate the fitting values has so far been ignored, but is taken up in 3.5.2. Monotonic regression may also be performed on quantities other than distances: it provides a general procedure for comparing the ordinal rescaling of data into a corresponding set of quantities defined by *any* sort of model (e.g., factor or scalar-product models, additive models, etc.; see Chapter 5).

3.3 Goodness/Badness of Fit: Stress and Alienation

We have seen that the difference between a particular distance and its corresponding 'pseudo-distance' $(d_{jk} - d^0_{jk})$ serves as an index of how badly the distance between j and k in the solution configuration departs from the value required to preserve an ordinal relation with the data. If there is no inversion in the required ordering then the difference will be zero. Alternatively, the difference can be looked on as the residual from monotone regression, i.e. an index of the difference between the solution distance and (an ordinal rescaling of) the data.

A simple *overall* measure of how the distances in a configuration ordinally fit the data can be constructed by squaring the differences between the actual distances in the configuration and the 'distances' fitted by monotone regression, and then sum them. MDS almost universally adopts the habit of using a *badness*-of-fit measure—the higher the index, the worse the fit—to assess the fit between the solution and the data. This basic index, called variously raw stress (Kruskal), raw phi (Guttman, Lingoes, Roskam), or stressform 0 has the same form as the 'residual sum of squares' in other types of regression, except that in this case it measures the residuals from monotonic regression. We shall refer to it normally as raw stress.

$$S_0 = \begin{cases} \text{Stressform 0} \\ \textit{Raw stress} \\ \text{Raw phi} \end{cases} = \sum_{\substack{\text{all pairs} \\ (j,k)}} (d_{jk} - d^0_{jk})^2$$

By convention, if the fitting quantities are Kruskal's $\hat{d}_{jk}$, then S_0 is referred to as raw stress, and if Guttman's d^*_{jk} are used, then it is called raw phi. In any event, for the same configuration, raw phi based on rank images will normally be higher than raw stress based on Kruskal's BFMF quantities, because of the strong monotonicity requirement. That is:

$$S_0(d^*) \geqslant S_0(\hat{d})$$

Raw stress is unfortunately a very unsatisfactory measure of fit for MDS solutions. The reason is that configurations which are identical in all but size will have different values of raw stress. But it is not the actual numerical distances (or co-ordinates) of an MDS configuration which are important or significant, but only the *relative* distances. For instance, doubling or halving the scale of the configuration is usually considered simply an irrelevance. We are only concerned with obtaining a configuration of points which is unique up to the uniform stretching or shrinking of the axes (or distances) by *any* constant, which is simply another way of saying that distances are at the ratio level of measurement. But unfortunately, if a configuration is shrunk uniformly by a constant, k, then the raw stress value shrinks by a value of k^2. That being so, if raw stress is used as an index of fit, it will always be possible to get a better fit simply by scaling down the size of the configuration! This is obviously an undesirable state of affairs, but the remedy is simple. To prevent it happening, raw stress can be divided by a factor which takes the size of the configuration into account, which has the effect of giving the same stress value to all configurations which differ only in size. A number of such 'normalising' or 'scale' factors have been proposed (see Kruskal and Carroll 1969, Roskam 1975, Lingoes and Roskam 1973). One family (the 'stress' indexes) stems largely from the Bell Laboratories group, and another family from the Guttman-Lingoes-Roskam group. Since programs from both sources are included in the MDS(X) series, the interrelations of these various measures are discussed in Appendix A3.1. For expository purposes, it will be sufficient to discuss the two most commonly used versions of normalised stress, each of which is widely used.

3.3.1 Normalised forms of stress

By normalising raw stress, it is possible to compare configurations by making stress independent of the size or scale of the configuration, and norming its value between 0 (perfect fit) and 1 (worst possible fit). The two most commonly used normalising factors are:

NF 1:

$$\sum_{(j,k)} d_{jk}^2 \quad \text{(the sum of the squared distances)}$$

This removes dependence on the scale of the distances. Its value will equal p^2, where p is the number of stimuli, if the configuration is centred and standardised (which means that the sum of co-ordinates of each dimension is zero and the sum of the squared co-ordinates is p) as is normally the case in MDS solutions.

NF 2:

$$\sum_{(j,k)} (d_{jk} - \bar{d})^2 \quad \text{(the sum of the squared differences between the distances and their average, } \bar{d})$$

This factor represents the variation of the distances about their mean and should always be used when data are conditional, such as ratings or rank orders (see 5.6.2), since it helps prevent a situation where a subject's rank values are fitted by the same disparity value (see Kruskal 1965).

Basically, normalised stress measures take the form:

RAW STRESS / NORMALISING FACTOR

but usually the square root of this ratio is taken, which has the effect of deflating the size of the index and making it sensitive to relatively small improvements when a configuration is coming close to being a perfect fit to the data (cf. Roskam 1968, pp. 34–5).

$$S_1 = \begin{cases} \text{Stressform 1} & \\ \textit{Stress}_1 & = \sqrt{(\text{Raw Stress}/\text{NF 1})} \\ \sqrt{(2 \times \text{phi})} & \end{cases}$$

$$= \sqrt{\left\{ \sum \left(d_{jk} - d^0_{jk} \right)^2 \Big/ \sum d^2_{jk} \right\}}$$

3.3.1.1 Properties of stress

Several important properties of stress_1 become more obvious if we consider its squared values, S_1^2:

(i) when S_1 (and hence S_1^2) is zero, then there is perfect ordinal fit, and all fitted 'distances' will equal the actual distances,

$$d^0_{jk} = d_{jk}, \qquad \text{for all } (j, k)$$

(ii) the maximum value of stress_1 is more difficult to determine, but S_1^2 can be shown (Lingoes and Roskam 1973, p. 12) to reach a maximum of $(1 - 2/p)$, which implies that S_1 approaches 1 as an upper limit as p, the number of stimuli, gets larger (as $p = 8, 16, 32, 64$: S_1 (max) $= 0.87, 0.94, 0.97, 0.98$).

(iii) S_1^2 can be re-expressed as

$$S_1^2 = 1 - \left(\sum d^0_{jk} \Big/ \sum d_{jk} \right)$$

which is equivalent to the proportion of residual variance from monotone regression.

$$S_2 = \begin{cases} \text{Stressform 2} & \\ \textit{Stress}_2 & = \sqrt{(\text{Raw Stress})/\text{NF 2}} \end{cases}$$

$$= \sqrt{\left\{ \sum \left(d_{jk} - d^0_{jk} \right)^2 \Big/ \sum \left(d_{jk} - \bar{d} \right)^2 \right\}}$$

Stress_2 can be interpreted as being the variation between the distances and the disparities as a fraction of the variation of distance round their mean. It has the same properties as stress_1 when zero, but its maximum value is difficult to

determine. In general $stress_2$ will be larger than $stress_1$, often twice as large, for the same configuration. Although $stress_2$ should always be chosen in preference to $stress_1$, for conditional data it makes little practical difference which is used to monitor an MDS solution in the case of the basic model.

So far we have dealt with a situation in which we have a set of data and a configuration. We have sought to measure how closely the distances between the points in that configuration are to a monotonic rescaling of the data, and the measure stress, which performs this task, has been described. We now go on to see how stress can be used to indicate not only how well a particular configuration captures the information in the data but also how an imperfectly fitting configuration can be improved to fit the data.

3.4 Finding the Best Configuration

The next question is: how does non-metric MDS actually work? Given a set of data, how does one find a configuration of points in Euclidean space where the rank-order of the distances best matches the rank order of the data?

3.4.1 Data as constraints on the solution

In principle, it ought to be possible to find a solution analytically. The rank order of the data imposes a set of constraints on where the points can be positioned in a configuration if it is going to conform to the data. As the number of points increases, information on the rank order of the dissimilarities begins to constrain the location of the points in the configuration so much that the distances to all intents and

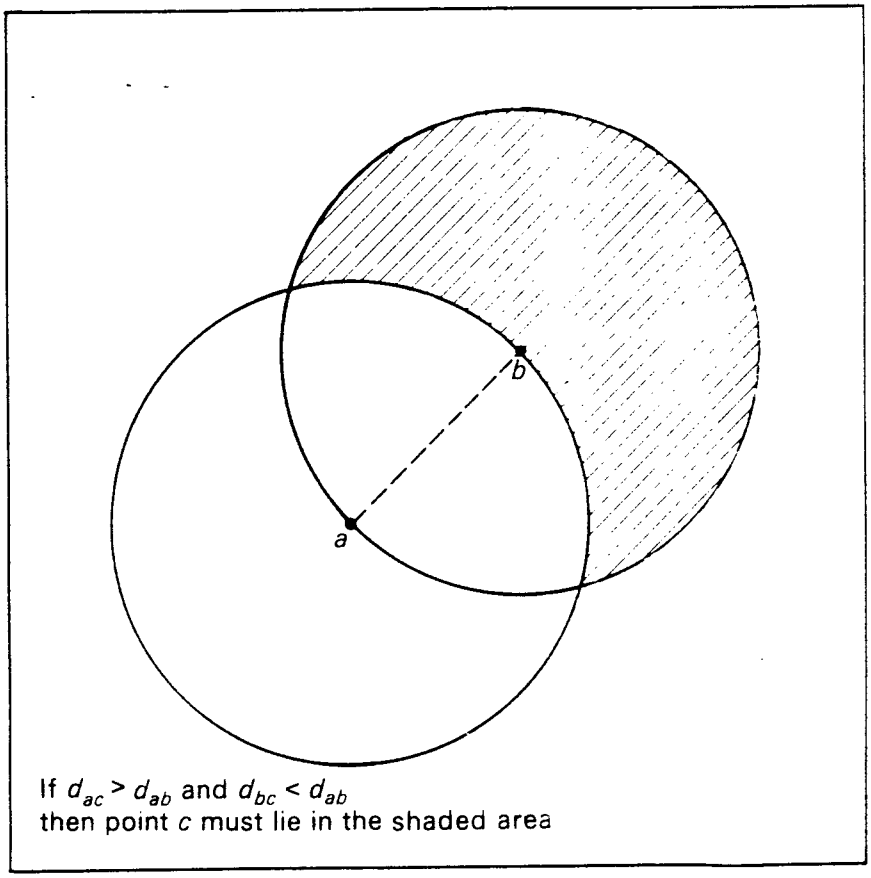


Figure 3.7 *Constraints on positioning a point*

purposes become fixed. (This point was also made by Abelson and Tukey (1959) and by Shepard (1962b, p. 238 et seq.)).

Actually, the position of the point never becomes 'fixed', but rather becomes constrained within a smaller and smaller region of the space. Consider a two-dimensional plane. Suppose there are two points a and b, then a third point, c, whose distance from a is greater than that between a and b (i.e. $d_{ac} > d_{ab}$), must lie outside of the circle whose centre is a and whose radius is d_{ab} (see Figure 3.7). If we also know that d_{bc} is less than d_{ab} then, in addition, c must lie *within* the circle whose centre is b and whose radius, again, is d_{ab}. Thus c must lie in the shaded area in Figure 3.7, but may be located anywhere within it.

It should be clear that as the number of points increases and the number of such inequality constraints also increases, then the area within which a point may be positioned in accordance with these constraints becomes smaller and smaller (and indeed may not exist). This region within which a point must lie, bounded by the inequality constraints implied by the data, is known as an *isotonic region*, since any point within the region equally satisfies the constraints. As Shepard puts it:

> Actually, though, if non-metric constraints are imposed in sufficient number, they begin to act like metric constraints. In the case of a purely ordinal scale, the non-metric constraints are relatively few and, consequently, the points on the scale can be moved about quite extensively without violating the inequalities (i.e. without interchanging any two points). As these same points are forced to satisfy more and more inequalities on the interpoint distances as well, however, the spacing tightens up until any but very small perturbations of the points will usually violate one or more of the inequalities.*
>
> (Shepard 1966, p. 288)

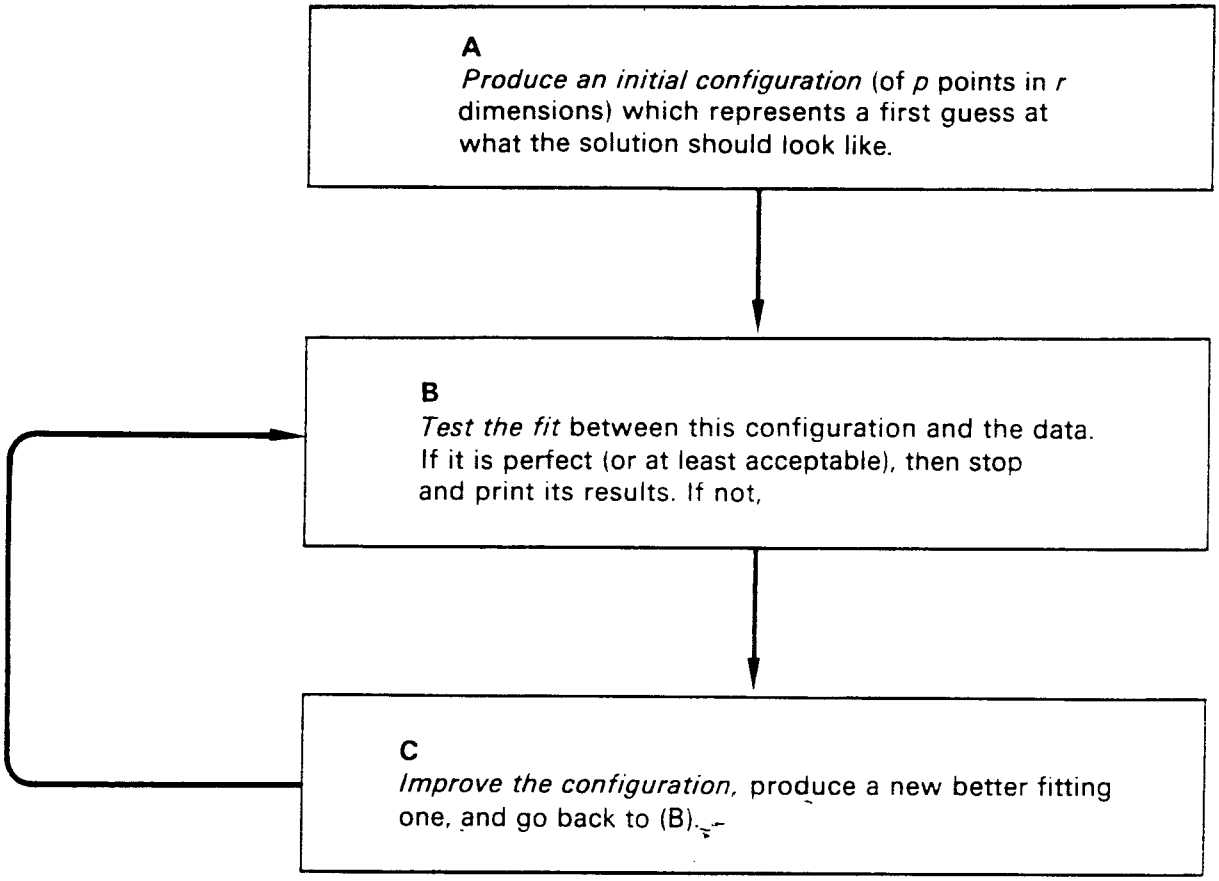
The imposition of more and more constraints by increasing the number of points hence means that the solution becomes more fixed or determinate. What is really happening is that the rank order of the $p(p-1)/2$ dissimilarity coefficients in the data is being 'distilled' or concentrated into the very much smaller $(p \times r)$ co-ordinates needed to define the solution configuration, and the number of data increases much faster than the number of co-ordinates, so long as the dimensionality is small.

3.4.2 The solution as an iterative process

How is the solution configuration obtained? It is done by a process which is surprisingly simple in form, though sometimes technically complex in detail. The user begins by choosing the number of dimensions r in which she wants a solution to be obtained. Roughly speaking, there should be at least twice as many data as parameters needed to specify the configuration, i.e. $\{p(p-1)/2\} > 2(p \times r)$, so the choice of dimensionality of the solution should be made with this in mind. Other factors are also relevant, and are considered below in section 3.7.

The basic outline of the process used to obtain a solution is as follows:

*Suppes and Winet (1955) had already established that for the *unidimensional* case a *complete* ordering of all pairwise distances on a closed interval establishes the representation of these distances up to a multiplicative constant—i.e. a complete 'ordered metric scale' is equivalent in the limit to a *ratio* scale. (Strictly, this only applies for the limiting case of an infinite number of points). This also confirms the similar conjecture by Abelson and Tukey (1959).



The process of moving round the cycle from C to B, producing somewhat improved configurations each time, is termed an 'iterative procedure', each cycle being an iteration. Before the advent of electronic computers, such procedures were simply not feasible—it is not unusual for there to be 50–100 iterations before a satisfactory solution is obtained.

All non-metric MDS programs follow this same basic procedure, but there are considerable differences in the details of the process, many of them now of purely historical interest. Very similar procedures had also been developed independently in Japan by Hayashi (1968) termed 'quantification scaling', and in France by Benzécri (1964) who had produced 'l'analyse des correspondances'.

3 5 General Outline of MINISSA

The next step is to outline the general iterative procedure in a little more detail, but in a non-technical manner. The overall flow is illustrated in Figure 3.8.

Let us begin by expanding on the 3-step process to include the major steps followed in all the basic programs.

3.5.1 The initial configuration

(i)* **Create an initial configuration** which will provide a good estimate of the final solution. This will reduce the number of iterations, and lower the probability of finishing with a sub-optimal solution (or 'local minimum', see section 3.5.4). In the case of MINISSA, this is done by keeping the *r* most important principal components of a matrix which is based upon the rank-order of the dissimilarities (see Appendix A3.2 for details).

*The roman numerals follow the steps of the sequence summarised in Figure 3.8.

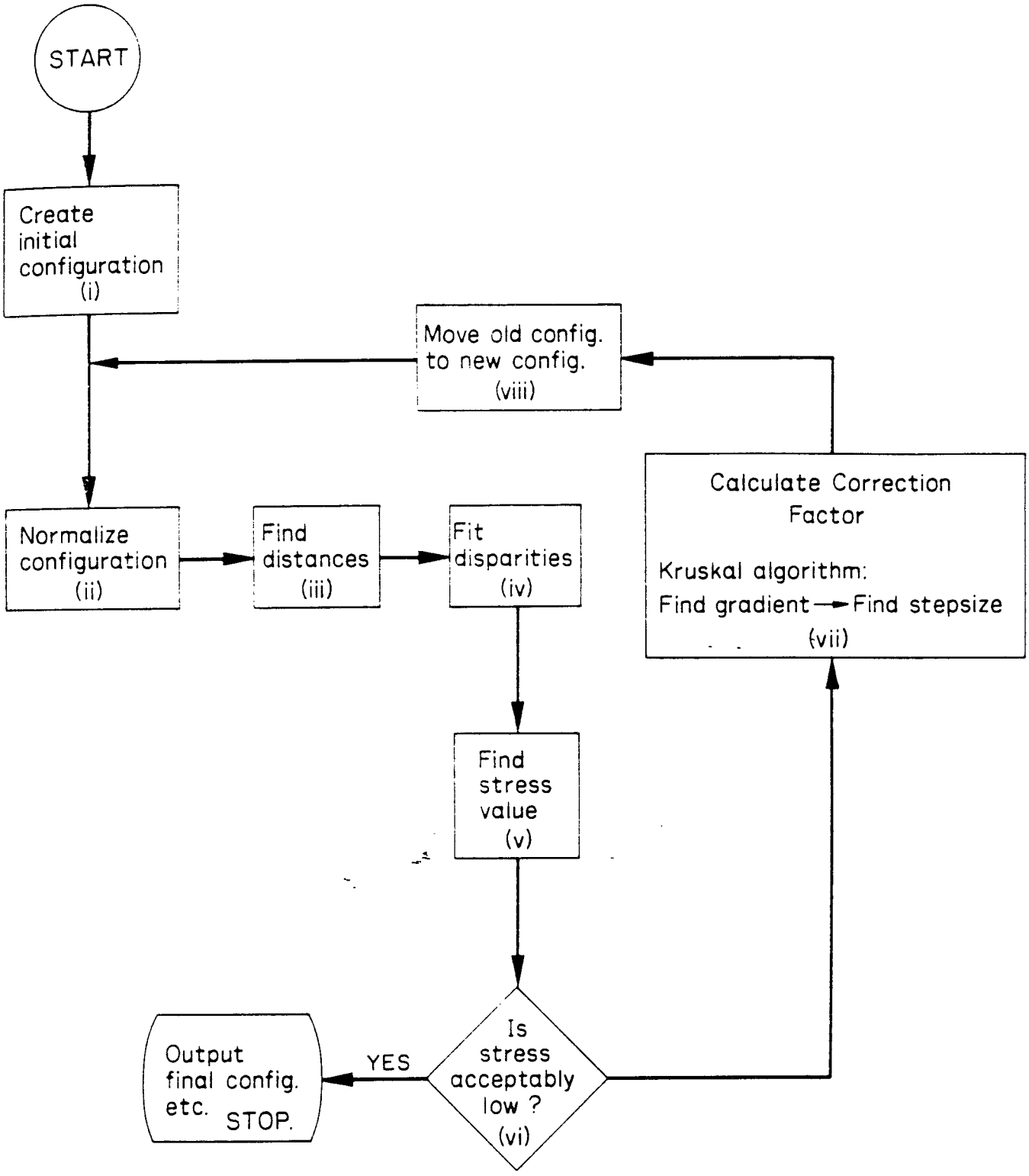


Figure 3.8 *Summary of the iterative process*

(ii) **Normalise this configuration** so that the origin is at the centroid (centre of gravity) of the stimulus point locations, and the configuration has constant dispersion (i.e. the sum of squares of the co-ordinates equals the number of points). This step is not always necessary, but is relevant and important if raw stress or phi is being used to measure badness of fit.

3.5.2 Comparing the current configuration with the data

The aim is to produce a configuration whose distances match the rank order of the

I Original data

Stimulus no.	1	2	3	4	5
1	-				
2	3	-			
3	6	4	-		
4	10	5	7	-	
5	2	8	9	1	-

												Table Row
II Sorted into order												
(column, row) indices	(5, 4)	(5, 1)	(2, 1)	(3, 2)	(4, 2)	(3, 1)	(4, 3)	(5, 2)	(5, 3)	(4, 1)	← (j, k)	1
Data (δ_{jk})	1	2	3	4	5	6	7	8	9	10	← δ_{jk}	2
Distances in *current* configuration (d_{jk})	3	6	3	5	8	10	13	11	9	15	← d_{jk}	3
III Calculation of disparities												
(a) MONOTONE REGRESSION step: 1	3											4
2		$4\frac{1}{2}$	$4\frac{1}{2}$									5
3				5	8	10						6
4							11	11	11	15		7
5												8
Disparities $(\hat{d}_{jk})$	3	$(4\frac{1}{2}$	$4\frac{1}{2})$	5	8	10	(11	11	11)	15	$\hat{d}_{jk}$	9
(b) RANK IMAGES (Permutation of d_{jk}) Disparities (d^*_{jk})	3	3	5	6	8	9	10	11	13	15	← d^*_{jk}	10
IV Squared differences												
(a) $(d_{jk} - \hat{d}_{jk})^2$	0	9/4	9/4	0	0	0	4	0	4	0	(*Sum*): 12.5	11
(b) $(d_{jk} - d^*_{jk})$	0	9	4	1	0	1	9	0	16	0	(*Sum*): 40.0	12

V Badness of Fit Measures

Raw Stress	$(\hat{d})$	$= \sum (d_{jk} - \hat{d}_{jk})^2$	$= 12.5$	13
Raw Phi/Stress	(d^*)	$\sum (d_{jk} - d^*_{jk})^2$	$= 40.0$	14
Stress_1	$(\hat{d})$	$\sqrt{(\text{Raw Stress/NF 1})}$	$= 0.1221$	15
Stress_1	(d^*)		$= 0.2184$	16
Stress_2	$(\hat{d})$	$= \sqrt{(\text{Raw Stress/NF 2})}$	$= 0.2886$	17
Stress_2	(d^*)		$= 0.5162$	18

NF 1 $= \sum d^2_{jk} = 839$; NF 2 $= \sum (d_{jk} - \bar{d})^2 = 150.1$

Table 2.2 Calculation of disparities

input data as closely as possible. How well does our current configuration, i.e. at the current iteration, match the data? The answer to this question involves three things:

calculating the distances between the points in the current configuration;
comparing these distances with the data (by calculating disparity values); and
assessing how badly the configuration departs from perfect ordinal fit to the data, i.e. by calculating stress.

Each point is now taken up in turn, in (iii), (iv) and (v) below, and illustrated by reference to a simple example shown in Table 3.3. Let us suppose the data dissimilarities are those given in section I ('Original Data') of Table 3.3. The first step consists of sorting the dissimilarities into ascending order (from the lowest to the highest value), keeping track of the column and row reference of each datum. The sorted data values are given in row 2 of Table 3.3, and the table (column and row) indices of each datum are noted in row 1. It should be stressed that *this information remains fixed throughout the iterative procedure.*

(iii) **Find distances.** Each time a new configuration is produced, a new set of distances is calculated, according to the usual Euclidean distance formula:

$$d_{jk} = \sqrt{\left\{\sum \left(x_{ja} - x_{ka}\right)^2\right\}}$$

These current distance values are then slotted into the same position as their corresponding data value, as in row 3. For instance, the distance between points 5 and 1, namely 6, is inserted into the second position which corresponds to the data dissimilarity between stimuli 5 and 1.†

(iv) **Fit disparities.** Now we are in a position to compare the solution (the distances in the current configuration) with the data, which is done by first calculating disparities (d^0_{jk}), which are to be monotonic with the data. As we have seen, it is possible to calculate either Kruskal's weak monotonic $\hat{d}_{jk}$ values or Guttman's strong monotonic d^*_{jk} rank image values. MINISSA and its cognates calculate both forms of disparities. Guttman's rank-image method has been found to be useful in avoiding sub-optimal solutions especially at the start of the process, but later in the program a switch is made to Kruskal's weak monotone regression, which provides a smoother, 'finer-honed' approach to obtaining an acceptable solution.

(a) *Monotone Regression* (Kruskal's $\hat{d}$ disparities)
The procedure of finding the Kruskal $\hat{d}$ values uses the *weak* monotonicity criterion that if $\delta_{ij} < \delta_{kl}$ then the corresponding $\hat{d}$ values should be in the same order, but are permitted to be *equal* without the tie counting as an infraction of monotonicity.

In brief, monotone regression consists of working consecutively through the distance values, checking whether they are in the same order as the data. However, when an inversion appears, i.e. where one or more distance values *decreases*, then a 'block' is formed by taking the offending value and the preceding one. These are

†The reader will have noticed that the distance values are not genuine distances at all, but numbers chosen to simplify the arithmetic of the example.

averaged until monotonicity is restored between blocks. The sequence is illustrated in rows 4 through 8 of Table 3.3. The sequence consists of repeatedly comparing the distances (row 3) and the data (row 2). It is useful to follow the example in detail.

1 The first two distances are compared to the data, and are found to be in correct (increasing) order, but the third distance *decreases* to the value of 3. At this point, we go back one position (to the second distance), treat the distances corresponding to (5, 1) and (2, 1) as tied, and average them to $4\frac{1}{2}$ (row 5). Now we have three disparities which are weakly monotonic with the data, viz

data	... 1	2	3	
disparities	... 3	$(4\frac{1}{2}$	$4\frac{1}{2})$	(cf. row 5)

2 The subsequent four distances increase, but then the eighth (and ninth) values decrease, so once more we backtrack to the last distance which *is* in order—the seventh—and average. The sequence is now

data	... 6	7	8
disparities	... 10	12	12

So far so good.

3 But then the ninth value decreases, so the backup process continues, still averaging distances until finally a block is formed whose average *does* preserve order, with respect to both the block below it and the one above it. (This is what Kruskal (1964b, pp. 40–1) refers to as a block being 'down-satisfied' and 'up-statisfied' with respect to monotonicity):

data	... 6	7	8	9	10	
disparities	... 10	(11	11	11)	15	(cf. row 9)

The monotone regression procedure of fitting a set of disparities to the data is now complete (row 9). There are seven blocks in all, and a set of disparities has been produced which are now perfectly *weakly* monotonic with the data (compare rows 2 and 9).

(b) *Rank Images* (Guttman's d^* disparities)

In Guttman's approach, we seek a set of quantities which are *strongly* monotonic with the data and once again use the distances in the current configuration as a starting point for calculating disparities. Guttman's procedure consists simply of taking the *current* set of distances (row 3), sorting them into order, and using them as the rank-image fitting quantities, d^* (row 10).

Notice that in the example chosen, all data values are distinct. This will rarely happen in practice. The calculation of disparities is modified slightly when ties occur in the data, depending upon whether the primary or secondary approach to ties is chosen by the user. The process of calculation is described in detail in Roskam (1975, p. 12 bis). Note also that in this example there are two equal distances (i.e. those between points 4 and 5 and 1 and 2). These two values become the fitting values corresponding to the pairs (4, 5) and (1, 2), and thus in the Shepard diagram would show up as a vertical segment *even though* strong monotonicity is being sought.

(v) **Find stress value.** We are now in a position to assess how well the current configuration fits the data. This is done by calculating the extent to which the actual distances diverge from the distances-made-to-conform-to-monotonicity, i.e. from the disparities. Since the sum of all the differences $(d_{jk} - d^0_{jk})$ will be zero, they are first squared.† Row 11 presents the squared differences based on $\hat{d}$ (formed by subtracting row 9 from row 3 entries and squaring), and row 12 presents those based on d^* (subtracting row 10 from row 3 and squaring). Whichever fitting procedure is used, the overall picture is very similar: four of the ten data are fit perfectly, and in both cases the main distances contributing to the badness-of-fit are between 5 and 3, 4 and 3, 5 and 1, and 2 and 1. Clearly, points 5, 3 and 1 are especially badly positioned in the current configuration, and will need to be moved to achieve a better-fitting configuration.

The overall badness-of-fit is now calculated by summing the squared differences to form raw stress (based either on $\hat{d}$ (row 13) or on d^* (row 14)). Note that rank image fitting produces higher stress values than monotone regression fitting, because strong monotonicity is the more stringent criterion and is not a least-squares fit to the data.‡

To compare configurations, a normalised version of stress is necessary, and these are presented in rows 15 to 18. Stress_1 $(\hat{d})$ is the measure most commonly reported in the literature, and is to be preferred at least on the grounds that we have more information on its properties and distribution than for any other measure.

(vi) **Is stress acceptably low?** There are several grounds for terminating the iterative procedure:

(a) if the stress value is zero;
(b) if the stress value is 'acceptably close' to zero;
(c) if the improvement in stress since the last iteration is so little that it does not seem worth continuing.

In the first instance, a perfectly fitting configuration has been obtained, and a perfect rescaling achieved.

In the second instance, a number of guidelines have been given as to what value of stress counts as 'acceptable'. (see especially Kruskal (1964b, p. 32) and Roskam (1975, p. 16)). The justification for these values is obscure, and even as rules of thumb they should be treated with considerable caution. A rather different approach to assessing stress values is the so-called 'Monte Carlo' simulation approach, which is discussed in greater detail below in section 3.7.1.

A third criterion for terminating the iterative process is simply that there has been so little improvement in the last few iterations that it is scarcely worth continuing. A good example is provided in Figure 3.9, which charts the progress of just over 200 iterations in reducing stress.

Clearly, there is a dramatic decrease in stress in the first few iterations: from 0.334 at the start to 0.158 at iteration 5, and improvement continues until just after

†When using rank images, the differences will not necessarily sum to zero, but the squaring convention is employed to keep comparability with BFMF estimates.

‡Raw stress, phi, based on d^* rank image fitting (whimsically christened 'soft squeeze' by Guttman) is used to monitor the first stage of MINISSA (and related programs). Stress_1 based on monotone regression (called 'hard squeeze' by Guttman) is used to monitor the second stage of MINISSA.

the 100th iteration. Beyond that point, there is virtually no improvement for the next 100 iterations; it would have saved time and expense to have stopped at the point where the improvement between iterations had become negligible.†

3.5.3 Improving the configuration

Having calculated the disparities and the measure of overall fit between the present configuration and the data, we now want to produce a *new* configuration whose distances will approximate more closely to the data. Put slightly differently, we want to move the points in the current configuration in such a way as to decrease the stress value.

As we have seen above, by looking at the differences between the current set of distances and disparities, $(d_{jk} - d^0_{jk})$, we can tell:

1 *which are the greatest discrepancies* (by the absolute value of the difference); and

2 *which points are involved* (by referring to the row and column references of these values).

The conclusions will depend in part upon which disparity values are used. For example, in Table 3.3,

(a) using monotone regression (row 11), the two greatest differences have the value 2, referring to the pairs (4, 3) and (5, 3);

(b) using rank images (row 12), the greatest difference is (5, 3), with a value of 4, followed by (5, 1) and (4, 3) with a value of 3.

But we can also infer two further pieces of information about each pair of points:

3 *in which direction to move the points to produce a better fit;* and

4 *how far to move them.*

(vii) **Calculate correction factor.** A full and accurate explanation of how this information is obtained involves differential calculus, and will not be discussed here.‡ But it is possible to get a perfectly adequate grasp of the basic process by a little simplification, drawing on Gleason (1969, pp. 6–8) and Spence (1978, p. 192). Let us concentrate upon one point, say point 5 in the present example and its relationship to each other point in the current configuration.

First consider point 5 with respect to point 3. Now imagine a line drawn in the configuration to connect points 5 and 3 (its length will be $d_{53} = 9$). How well does the current positioning of points 5 and 3 correspond to the data? If it were a perfect correspondence, then the difference $(d_{53} - \hat{d}_{53})$ would equal zero. As it is, the difference $(9 - 11) = -2$. This tells us that in order to improve the fit point 5 should be moved *away* from point 3 so as to increase the distance, and that this

†In the MDS(X) series, control of the number of iterations is given by the ITERATIONS command (MINISSA, SSAM) which sets the minimum number of iterations before a test is made. From then on, a check is made of each iteration to see whether improvement in stress has been more than a given amount. If not, the iterations are terminated.

‡The method of 'steepest descent' or 'negative gradients', which is used to move the configuration is discussed Kruskal (1964b, pp. 30–9). The methods used in MINISSA are reviewed in the MDS(X) documentation, and a full technical discussion is contained in Lingoes and Roskam pp. 12–33).

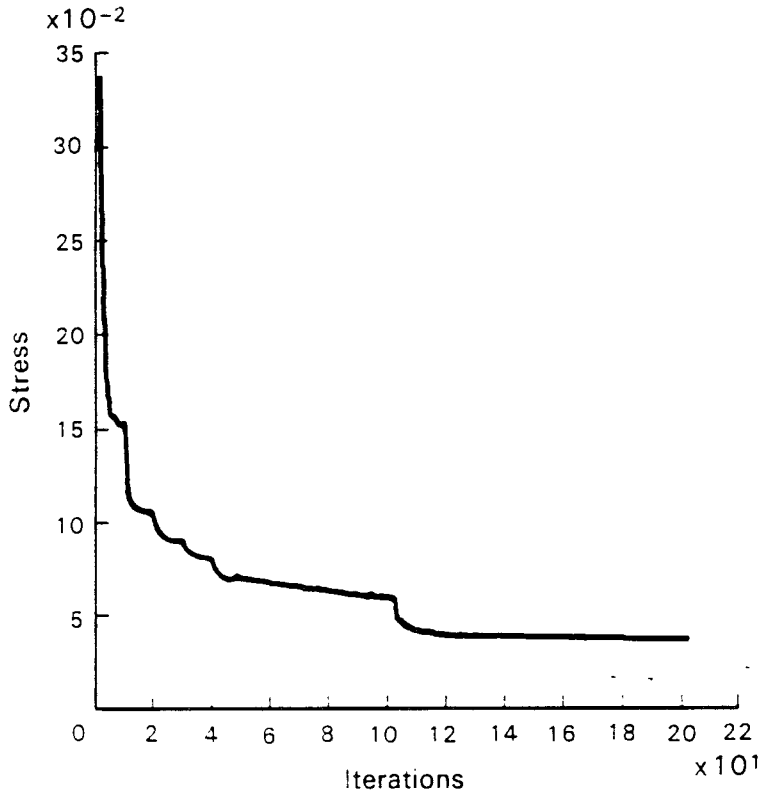


Figure 3.9 *Stress by iteration*

distance should be increased by 2 units if the difference is to be zero. (In general, if $d_{jk} < d^0_{jk}$, then the difference is negative, which indicates that the points should be moved away from each other.)

Now consider point 5 and point 1. The difference $(d_{51} - \hat{d}_{51}) = (6 - 4\frac{1}{2}) = +1\frac{1}{2}$. This tells us that point 1 should be moved *towards* point 5 (to reduce the distance, and lower the difference), and should be moved $1\frac{1}{2}$ units if the difference is to be zero. (In general, if $d_{jk} > d^0_{jk}$, then the difference is positive which indicates that the points should be moved towards each other.)

Finally, consider point 5 and its relation to points 2 and 4. In this case, the difference is zero, indicating that the fit is perfect and they should not be moved.

Usually the formula used in MDS programs to improve the position of a point j with respect to another point k takes the form

NEW POSITION of j = OLD POSITION of j + CORRECTION FACTOR

specifically

$$x_j^{\text{new}} = x_j^{\text{old}} + \left(1 - \frac{d^0_{jk}}{d_{jk}}\right)\left(x_j - x_k\right)$$

A little arithmetic reduces the formula to a particularly simple form†:

$$\overset{\text{new}}{x_j} = \overset{\text{old}}{x_j} + (d_{jk} - d^0_{jk})$$

If there are p points, then for each there will be $(p - 1)$ correction factors, pushing and pulling point p in various directions and with different degrees of force. In this instance, point 5 is being pushed away from point 3 by 2 units, and towards point 1 by $1\frac{1}{2}$ units. The actual move is bound to be a compromise between these various forces, and the greatest discrepancies will obviously tend to dominate the movement of the points.

When all the discrepancies are considered simultaneously, it is necessary to rewrite the correction formula to take into account the forces from *all* the points (represented in the summation), and consider the location of the point p on each dimension, a.

General correction formula:

$$\overset{\text{new}}{x_{ja}} = \overset{\text{old}}{x_{ja}} + \alpha \sum \left(1 - \frac{d^0_{jk}}{d_{jk}}\right)\left(x_{ka} - x_{ja}\right)$$

The only new quantity to appear is α (alpha) the 'step-size' which in Kruskal's version represents the overall amount by which the points are moved. The technicalities are complex,‡ and all the user need know is that longer step-sizes are usually taken in the earlier stages of the process, and when a program is minimising in terms of rank-image disparities, whereas smaller steps are taken towards the end of the process (and when stress is being minimised by reference to Kruskal's weak monotonic disparities). In fact, MINISSA uses a hybrid approach, starting by minimising raw phi stress in terms of Guttman's Rank Images ('soft squeeze'), and then switching to minimising stress_1 in terms of Kruskal's weak monotone disparities ('hard squeeze') later in the process.

---

†The first formula takes the form: $\overset{\text{new}}{x_j} = \overset{\text{old}}{x_j} -$ (correction factor). Now examine the correction factor further. Since we are restricting attention to the line joining x_j and x_k, then $(x_j - x_k)$ is simply d_{jk}. Thus the correction factor reduces to

$$\left(1 - \frac{d^0_{jk}}{d_{jk}}\right)d_{jk}.$$

Putting the term in the brackets over a common denominator, gives

$$\frac{(d_{jk} - d^0_{jk})}{d_{jk}}\, d_{jk}.$$

and cancelling, the correction factor simplifies to

$$(d_{jk} - d^0_{jk}).$$

Hence, the formula now takes on the simple second form:

$$\overset{\text{new}}{x_j} = \overset{\text{old}}{x_j} - (d_{jk} - d^0_{jk})$$

‡The basic reference is Kruskal (1964b, p. 121), which is further expanded and discussed in Lingoes and Roskam (1973, pp. 13–16). The Guttman-Lingoes procedure is a somewhat different correction procedure, described in Lingoes and Roskam (1973, pp. 22–9) which provides for different step sizes for each point.

3.5.4 Local minima and related problems

Another analogy, used to explain what is happening during the process of moving the configuration to one that fits the data better, is geographical. Suppose our data refer to 20 stimuli and we seek a solution in three-dimensions. We are now asked to think of *all possible* three-dimensional configurations or solutions—good, bad or indifferent—and pay attention to the stress value of each one. Obviously, what we are looking for is that one configuration whose stress is lowest. One way to think about the problem is to simplify matters and imagine the solution space like a 'rolling terrain, with hills and valleys' (Kruskal 1964b, and Kruskal and Wish 1978, pp. 27–8). They continue:

> Each point of the terrain corresponds to an entire configuration (*not* to a point *in* the configuration). Each point of the terrain can be described by three co-ordinates—the altitude, and the two location co-ordinates, North-South and East-West. The locations co-ordinates are analogous to all the co-ordinates of all the points of the configuration. (Of course, a configuration with p points in r dimensional space has $p \times r$ co-ordinates, and $p \times r$ is far greater than two, so the analogy does not convey the full richness and difficulty of the situation). The altitude is analogous to the objective function, that is, the altitude is the stress. (quoted, with slight notational changes, from Kruskal and Wish 1978, p. 27, with permission)

The real problem is that the terrain—the hills of high stress and valleys of low stress—is unknown territory, and we have no means of knowing before the event even what its general features are like, so we are in the position of a blindfolded parachutist dropped from a plane on a dark night (Kruskal and Wish's analogy) or a climber lost in the mist. It is not in fact quite as bad as this—there is a way to locate a sensible starting point, by choosing an initial configuration which gets us fairly close to the point where the stress is lowest. Nonetheless, the imagery is apt. To move from the present position, a configuration of relatively high stress at the earlier stages of the iterative process, we need to know two things: in what direction the ground is sloping downwards; how large a step to take in that direction.

In the first case, we can detect the general direction by calculating the negative gradient, which tells us in what direction to move each point of the current configuration if we want to lower stress; it gives rise to the correction factor formula discussed above in 3.5.3. In the second case, we need to adopt a strategy which avoids the extremes of foolhardiness and over-cautiousness—if we move too far in the general direction of improvement, we might in fact overshoot the actual minimum, and if we move in very small steps we are going to consume an enormous amount of time getting virtually nowhere.

How does the climber know that the minimum (valley floor) has been reached? The answer is, where the gradient is zero; that is, where stress increases in every direction. (This state of affairs is actually tested for at each iteration, and provides yet another criterion for terminating the iterative process.) But unfortunately there is no guarantee, even if a valley floor is reached, that it is actually the lowest point on the terrain—it may simply be a local dip. The situation is illustrated in Figure 3.10. Indeed, there is no sure way of knowing whether a particular configuration is actually the one which best fits the data, i.e. that a 'global minimum' has been reached. But there are, at least, ways of guarding against entrapment in a local

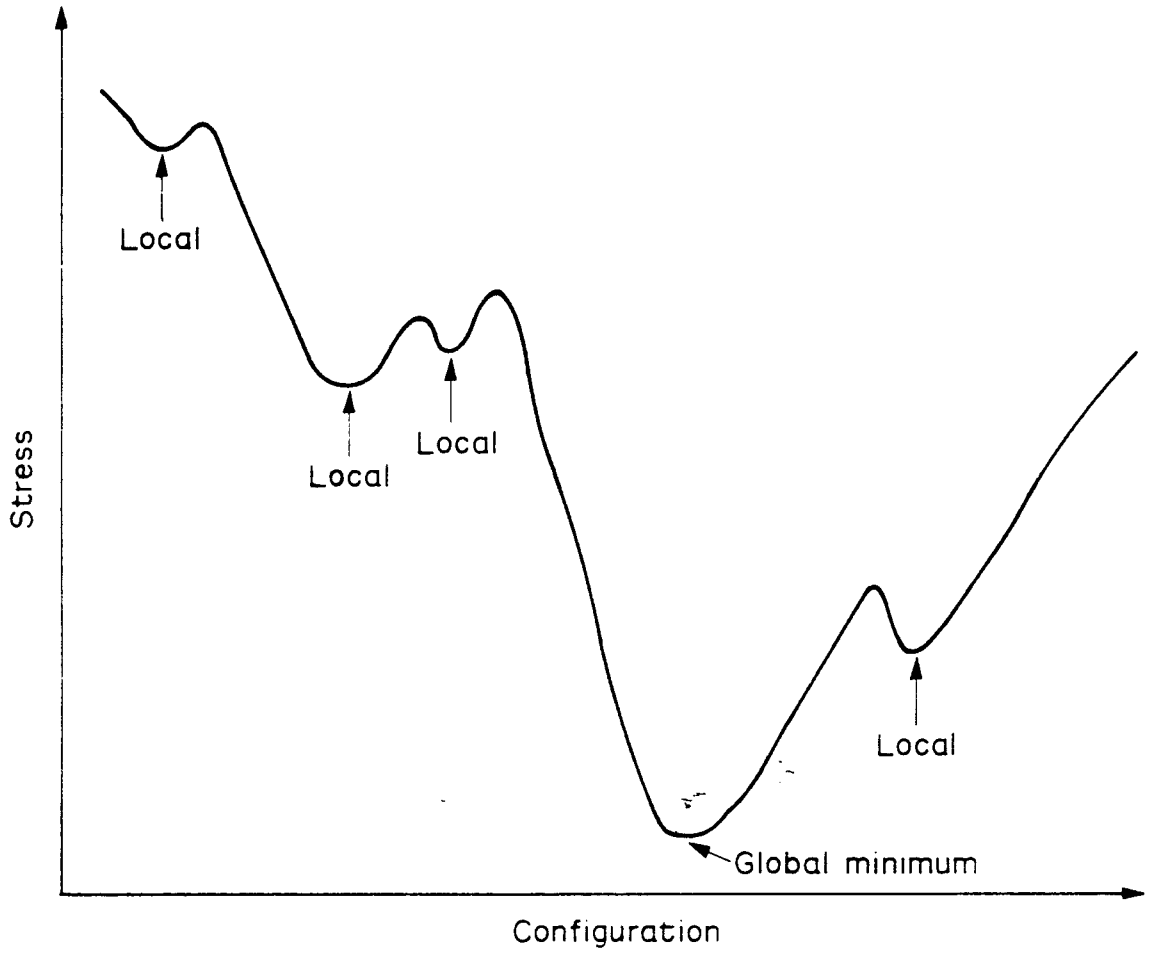


Figure 3.10 *Local and global minima*

minimum, and fairly reliable ways of detecting a local minimum:

(a) Reliable modern computer programs avoid starting from an *arbitrary* initial configuration, and produce an initial configuration which is likely to be fairly close to the global minimum (see Appendix A3.2). This is like using what information we have about the terrain to position our climber close to where we think the lowest valley is.

(b) Programs use more than one minimising procedure to capitalise on the strengths of each approach. (Thus MINISSA uses the somewhat erratic but rapid Guttman technique at the start and then switches to the smoother Kruskal technique when the earlier phase shows no signs of systematic improvement.)

(c) It is always sensible to obtain MDS solutions in a number of different dimensions (e.g. in five through one dimensions, for reasons discussed in section 3.7.1 below). The stress value should *decrease* as the number of dimensions increases. If it actually increases then that solution is bound to be a local minimum.

(d) The best safeguard against local minima, and one which is strongly recommended, is to use *several* different starting configurations (implemented in MINISSA by using the READ CONFIG command) and see whether they produce markedly different results. (Do not rely upon just looking at the resulting configurations: remember that a reflected and/or rotated configuration can *look* very different to the original one, though be identical to it in the sense of being a legitimate similarity transformation. If in doubt it is safest to use PINDIS to compare configurations; see 7.3.2 below).

(viii) **Move the old configuration to a new positioning of points.** When all the correction factors and the step size have been calculated, the general correction formula is used, and every point is moved into greater conformity with the data. Thus a new configuration has then been obtained and the iterative cycle is complete.

3.5.5 The final configuration

In the basic model, and indeed in most MDS models, the final configuration is rotated to principal components before being output. This rotation does not in any way change the pattern of points or the relative distances between them, but principal components (or 'principal axes') provide a framework of reference axes which possesses some useful statistical properties. It may be helpful to recapitulate the method of principal components.

Given a set of points located in a multidimensional space (in this case the final configuration) the method of principal components first finds the line (axis dimension) through the configuration which has maximum variation, i.e. along which the co-ordinates of the points are maximally spread or differentiated. This line is termed the first 'principal component' or 'principal axis' of the space. Following this a second axis is found which is orthogonal to the first axis, i.e. which is statistically independent of the first component, in the sense that the correlation of the co-ordinates of the points on the two components is zero, and also explains the maximum amount of the remaining variation. This is the second principal component. The process continues in this manner—identifying axes which are orthogonal to those already found and which explain maximum amounts of remaining variation—until the final components are normally explaining trivially small proportions of the total variation. In this sense, principal components is often viewed as a way of orienting the configuration so that variation is concentrated into as few dimensions as possible.

The actual amount of variation represented by a particular component is given by the size of its latent root (also called the eigenvalue) which can be thought of as a standard deviation measuring the dispersion of the objects along that dimension (and indicated by the sigma value at the foot of each column of the final configuration printout in some MDS(X) programs). Comparison of the values of sigma is often instructive: the more equal they are, the more circular (or spherical in three-dimensions) the pattern of points in the final configuration is. Conversely, the more unequal they are, the more ellipse-like the pattern is. If any sigma values are close to zero, this signals the fact that there is virtually no variation on the dimension concerned—i.e. that the dimensionality chosen for analysis is unnecessarily high.

3.6 Assessing the Solution

We now want to illustrate the process of finding a 'best solution' to a set of data using non-metric MDS. Information from the process is used to help decide how good a solution has been obtained and diagnose inadequacies in it. This will be done by using a genuine set of data, as opposed to one chosen for purely illustrative purposes.

	CA	SST	GM	BM	ST	SW	C	AD	CPR	MOR	PL	MPN	BCK	PST	UMO	PM	CE	PHT	BSL	RCK	AP	A	RED	PO	GEO	SMG	TDH	TDR	ESG	JN	LT	BC
CA																																
SST	25																															
GM	3	2																														
EM	3	2	9																													
ST	43	29	2	2																												
SW	13	40	6	3	17																											
C	2	1	61	11	3	5																										
AD	2	4	21	21	2	12	19																									
CPR	27	17	8	5	32	18	6	4																								
MOR	26	43	3	2	20	39	3	7	12																							
PL	4	2	58	10	3	6	64	17	7	4																						
MPN	10	22	13	5	9	39	12	20	19	25	13																					
BCK	22	16	10	8	23	26	9	11	33	11	10	22																				
PST	15	54	3	2	20	47	2	8	21	36	3	28	27																			
UMO	1	1	17	39	2	3	16	22	3	3	15	8	7	2																		
PM	6	16	10	12	4	29	7	25	17	17	8	27	29	21	7																	
CE	45	27	8	1	34	14	7	1	26	25	9	12	8	16	3	7																
PHT	5	4	16	11	7	8	17	8	15	6	18	16	12	6	5	14	7															
BSL	1	1	15	42	2	1	20	20	2	1	19	6	5	1	63	6	3	5														
RCK	4	2	36	28	2	9	37	20	13	4	35	16	16	5	18	18	3	20	20													
AP	26	14	10	3	15	11	8	9	22	17	9	9	8	8	4	11	31	9	3	7												
A	10	8	7	6	12	12	7	3	14	15	7	7	9	9	3	9	10	35	3	11	10											
RED	5	3	35	12	3	6	33	32	10	4	33	13	11	5	9	20	5	18	11	28	23	7										
PO	2	5	13	40	1	6	12	35	3	4	11	8	12	7	34	25	1	8	35	22	4	3	18									
GEO	37	26	5	0	37	13	6	2	24	26	8	12	7	16	1	7	46	9	3	5	25	13	8	1								
SMG	25	18	3	7	26	15	4	3	25	15	3	11	26	14	2	8	14	14	2	6	11	20	6	6	14							
TDH	2	1	16	38	3	2	18	19	4	1	19	9	7	1	56	8	2	8	57	20	8	3	17	34	3	2						
TDR	2	1	18	35	3	5	14	44	5	1	14	7	9	3	25	15	1	11	26	23	13	8	32	36	3	6	25					
ESG	33	28	3	0	21	20	2	4	13	35	4	25	5	22	1	7	34	3	1	3	30	10	5	2	34	10	2	1				
JN	11	17	9	3	19	19	8	4	17	19	9	12	16	19	3	15	14	37	3	10	15	39	12	4	18	24	4	3	10			
LT	12	8	23	7	10	16	23	15	35	10	21	27	21	11	8	20	19	14	6	19	20	8	18	4	18	11	6	8	13	12		
BC	2	2	14	39	3	3	12	33	2	2	12	4	9	3	37	15	1	6	38	18	9	4	23	51	2	5	38	44	1	3	5	

Table 3.4 *Co-occurrence frequencies of sortings of occupational titles*

3.6.1 An illustration of the iterative procedure in non-metric MDS

In a study (Coxon and Jones 1978a, p. 42 et seq.) of the natural groupings which people use to classify occupations, a group of 71 individuals were asked to sort a set of 32 occupational titles into as many or as few groups as they wished. A measure of pairwise similarity between the occupations was defined as the frequency with which two occupations were sorted into the same group: the greater the number of subjects who put a pair of occupations together, the more similar they are defined to be. (This co-occurrence measure is *M1*, discussed earlier in 2.2.3.3). The frequency matrix is presented in Table 3.4. The data are analysed by the basic non-metric distance model in two dimensions, using $stress_2$ and Kruskal's weak monotonicity.

In order to dramatise the process of improvement, a deliberately poor starting configuration was chosen ($stress_2 = 0.986$). The iterative process is now examined at the 2nd, 5th, 10th and 23rd iteration to see what progress is made.

Iterations 0–2 (Fig. 3.11)

Figure 3.11a shows the moves made in the positioning of the points from the initial configuration (iteration 0) to the second iteration. Note that there is very little change in positioning, but that the step size increases. By iteration 2, the monotone fitting function (Figure 3.11b) is beginning to descend from the top left to the bottom right of the Shepard diagram (since similarities are inversely related to distances). The transformation function is little better than a straight vertical line, indicating that a very large number of data values are being fitted by the same disparity value.

Iterations 2–5 (Fig. 3.12)

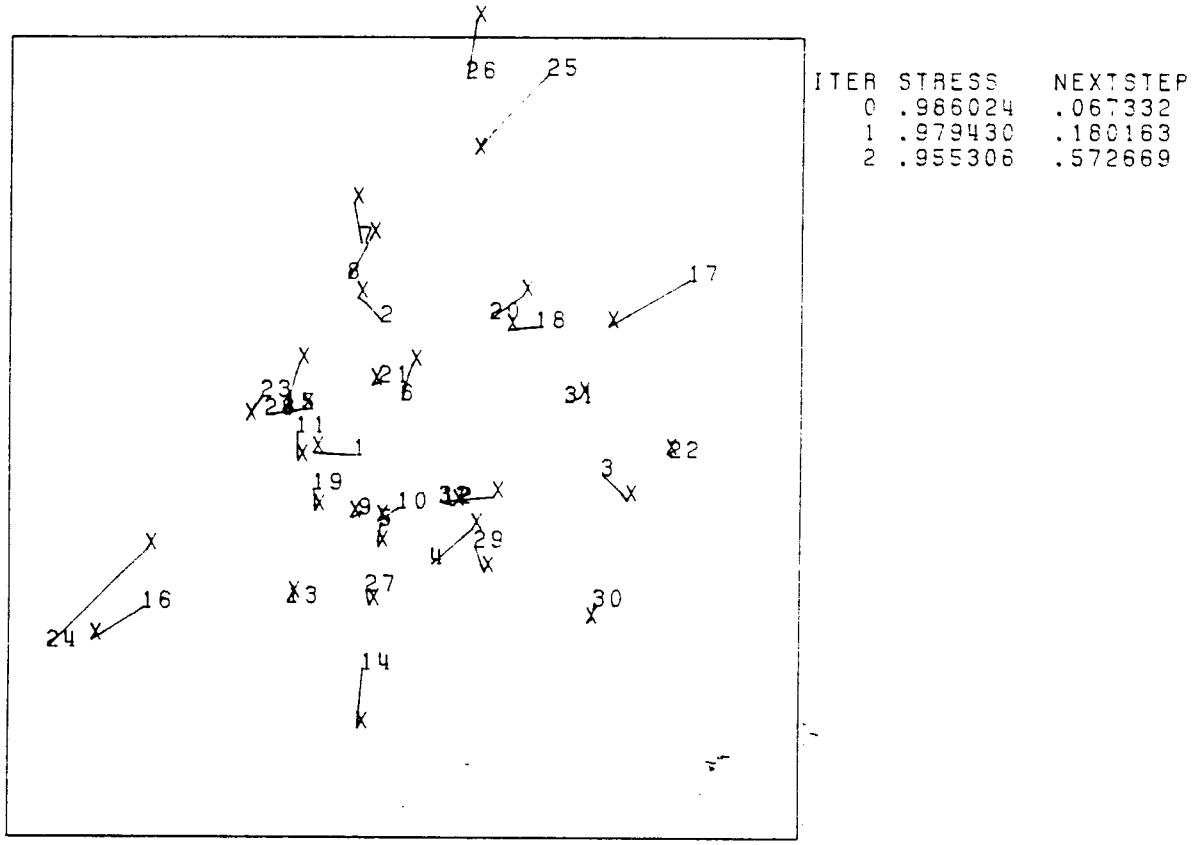
Clearly, the positioning of the points is changing fast (Figure 3.12a) and improving rapidly, with a reduction in the size of $stress_2$ by a third. Note that larger step sizes are occurring. By iteration 5, the monotone fitting function (Figure 3.12b) is taking on its characteristic 'steppy' form and the larger number of steps at the bottom right hand of the function shows that the improvement is concentrated principally in the positioning of the smaller similarity values.

Iterations 5–10 (Fig. 3.13)

There is a dramatic improvement from iterations 5 to 6 (Figure 3.13a) but beyond here it tails off somewhat, and the step size decreases fairly drastically. In terms of improvement, the point of diminishing returns is setting in. The monotone function is now fitting the highest and lowest similarities well, though the middling data values are still not well fit. Note that monotone function is taking on a recognisably convex form (cf. Shepard 1974, p. 400), suggesting that the co-occurrence frequency data might well be related to distance in a more regular (polynomial) manner.

Iterations 10–23 (Fig. 3.14)

The improvements are now very slight indeed: $stress_2$ decreases by the same amount between iterations 9 and 11 as between 11 and 23. Also, the step size becomes smaller and smaller, till the process finishes at iteration 23. The Shepard diagram (Figure 3.14b) for iteration 23 is virtually identical to that for iteration 10, indicating some slight improvement.



F gure 3.11a *Iterations 0–2: moves in positioning of points*

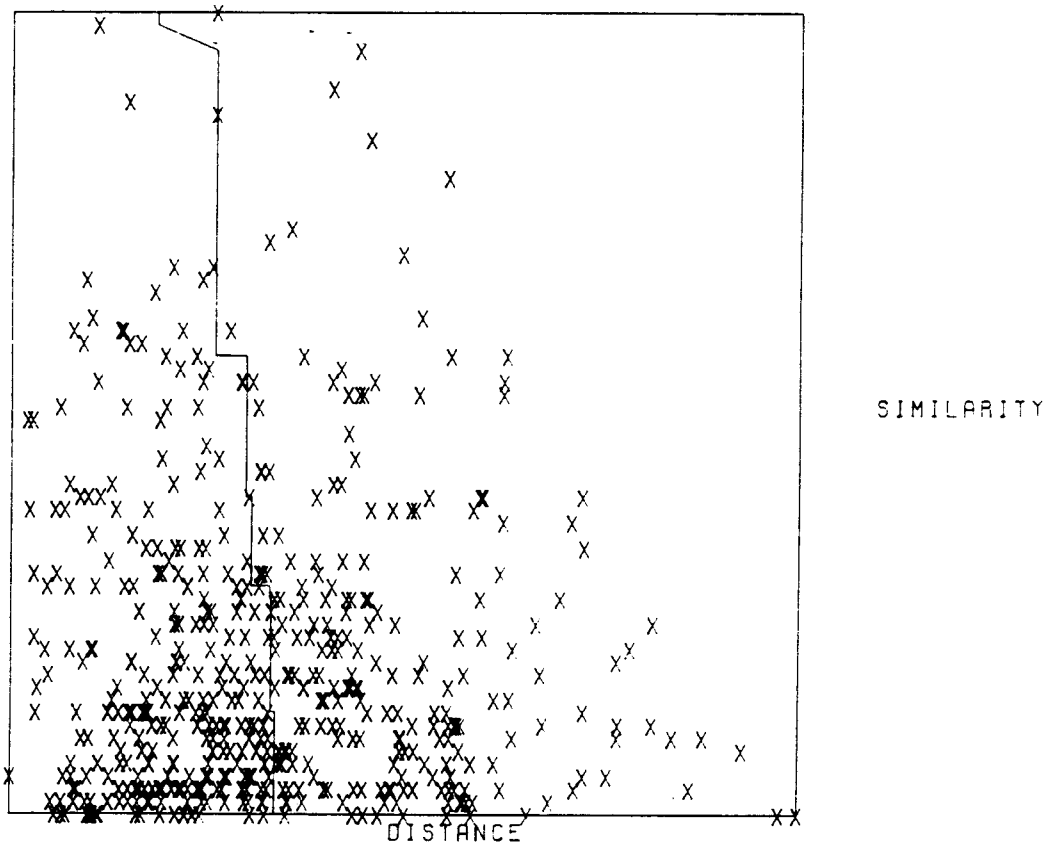


Figure 3.11b *Shepard diagram at iteration 2*

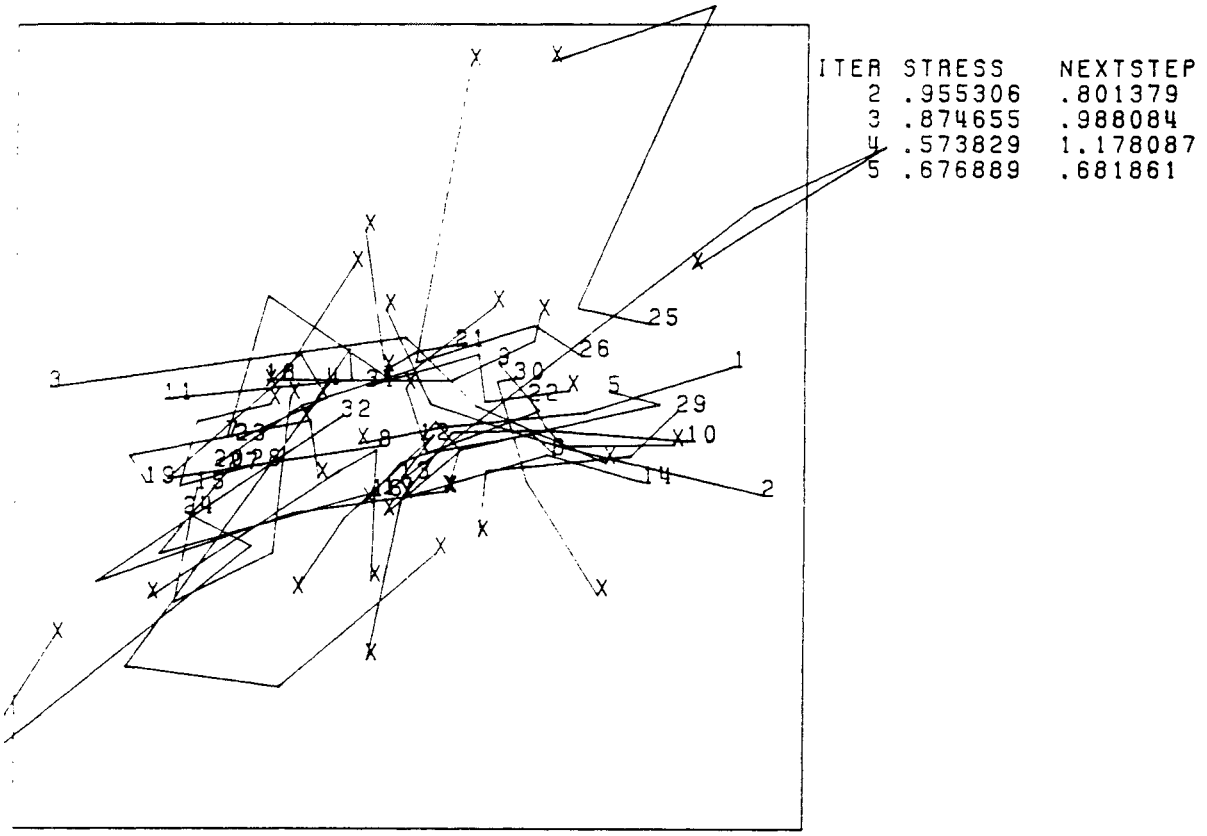


Figure 3.12a *Iterations 2-5: moves in positioning of points*

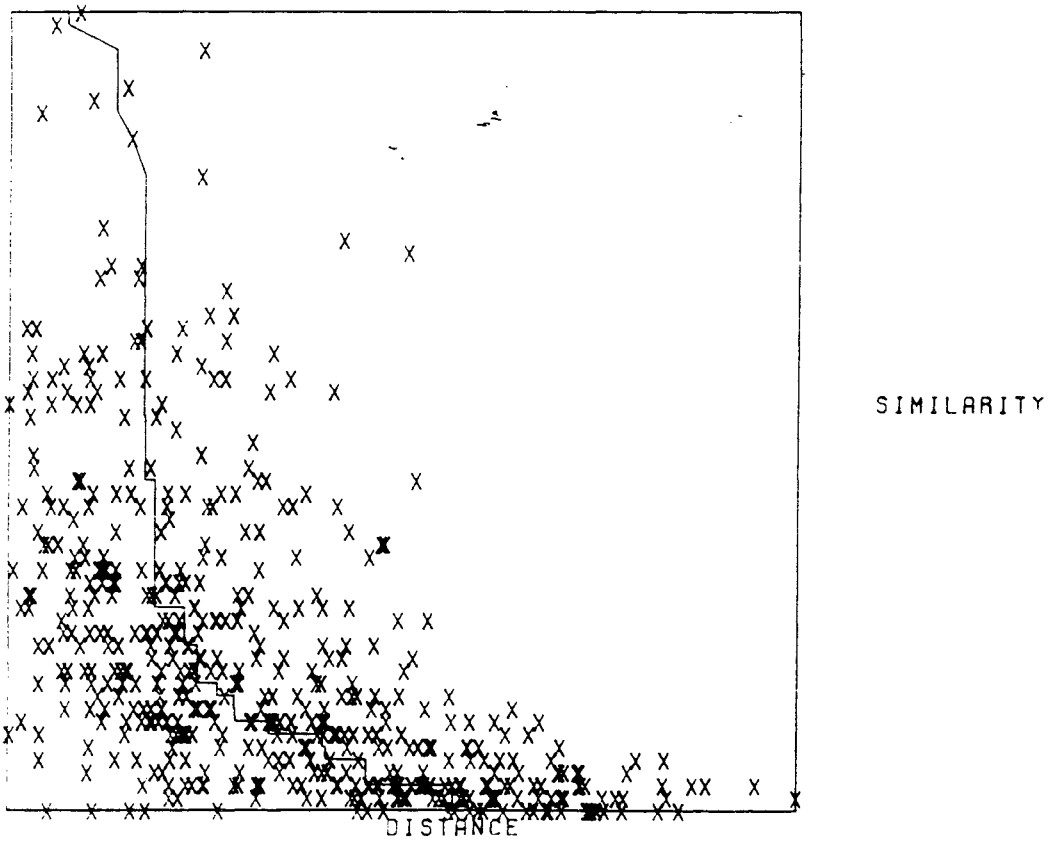
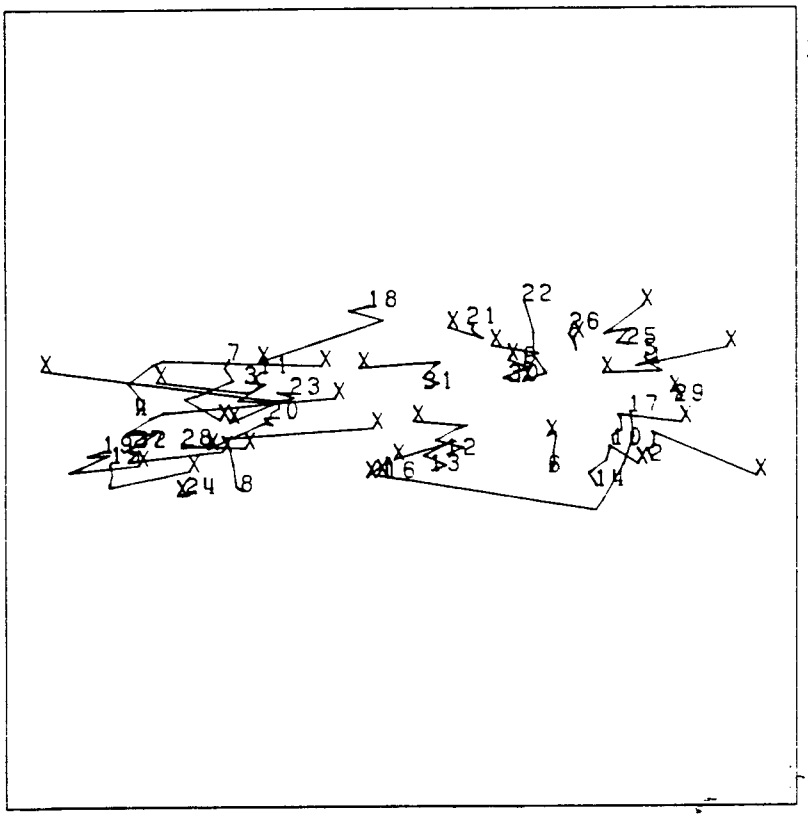


Figure 3.12b *Shepard diagram at iteration 5*



ITER	STRESS	NEXTSTEP
5	.676890	.516218
6	.356057	.414221
7	.338799	.102525
8	.324861	.090250
9	.314566	.066588
10	.301137	.064490

Figure 3.13a *Iterations 5-10: moves in positioning of points*

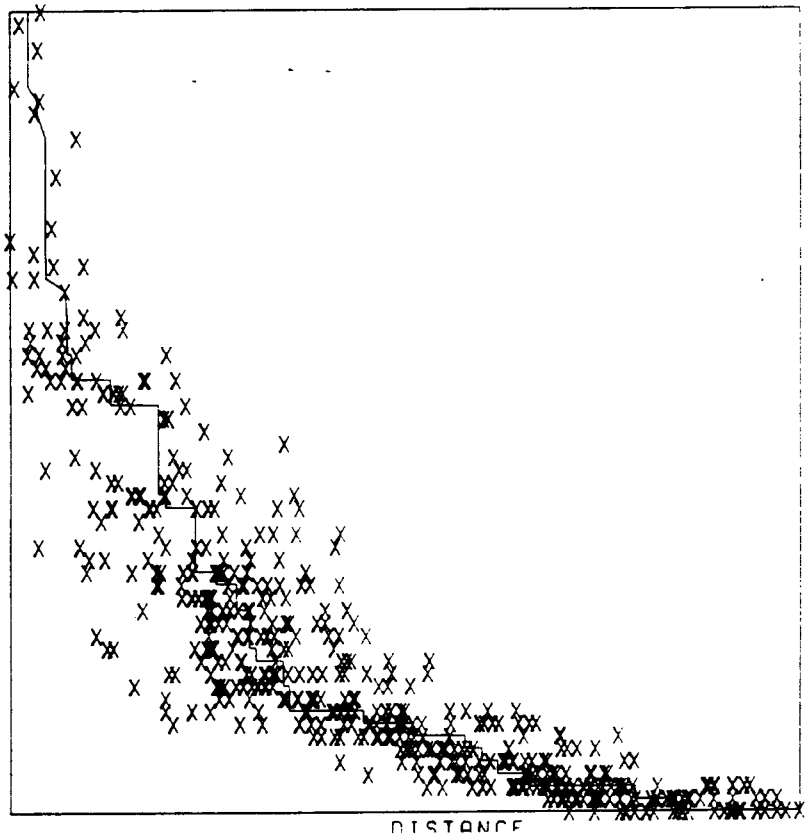
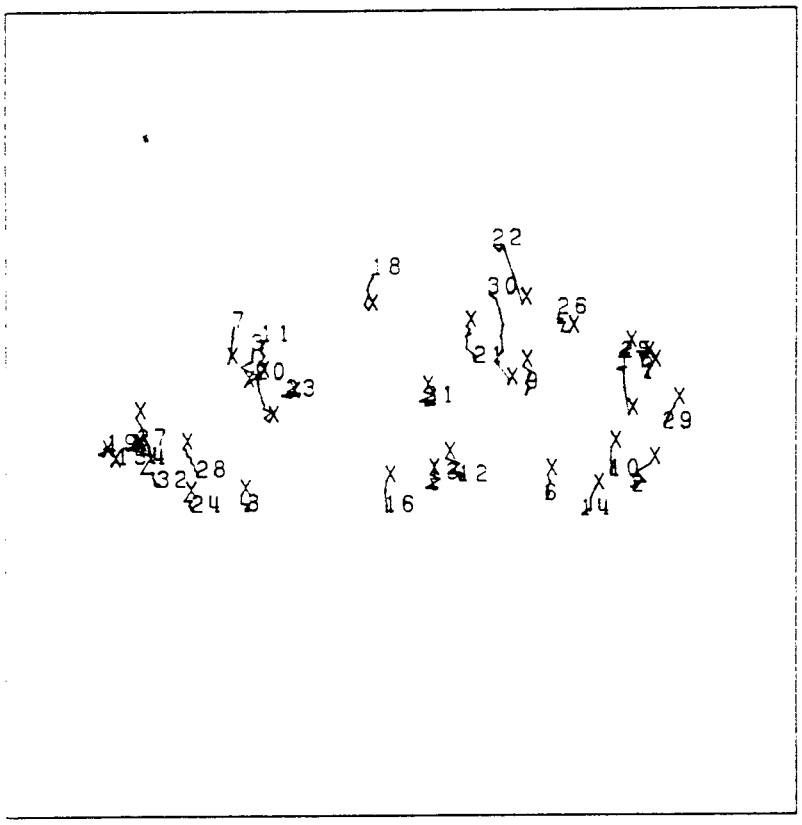


Figure 3.13b *Shepard diagram at iteration 10*



ITER	STRESS	NEXTSTEP
10	.301137	.059137
11	.292243	.067665
12	.289220	.038631
13	.284668	.022784
14	.283156	.017844
15	.281975	.020362
16	.280862	.023589
17	.280504	.012010
18	.279853	.008215
19	.279694	.007531
20	.279665	.003592
21	.279580	.003236
22	.279548	.005209
23	.279535	.003290

MINIMUM ACHIEVED

Figure 3.14a *Iterations 10–23: moves in positioning of points*

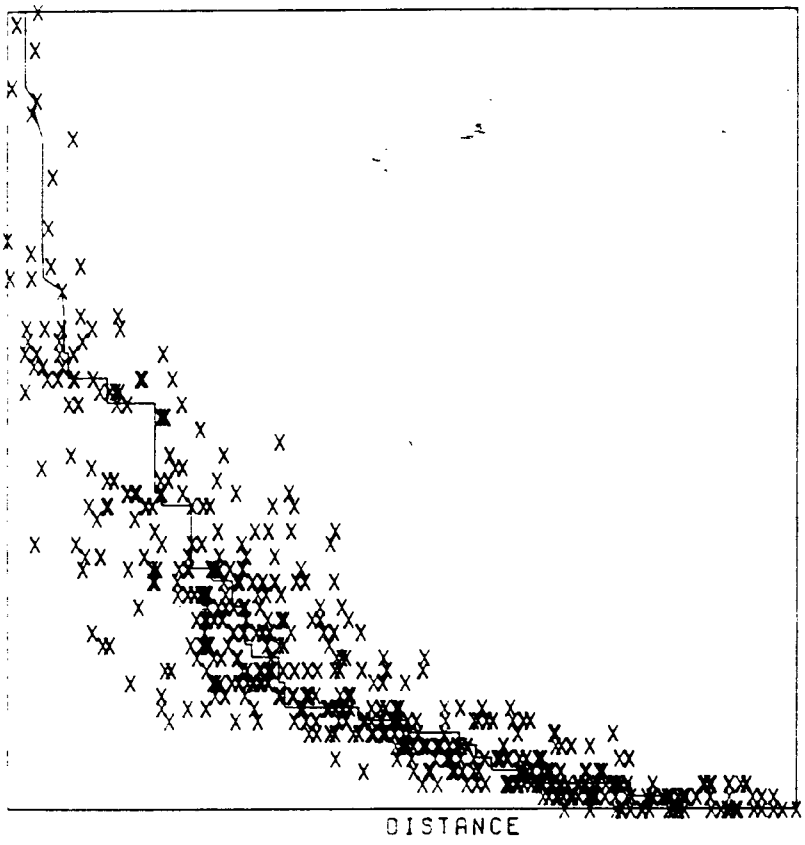


Figure 3.14b *Shepard diagram at iteration 23*

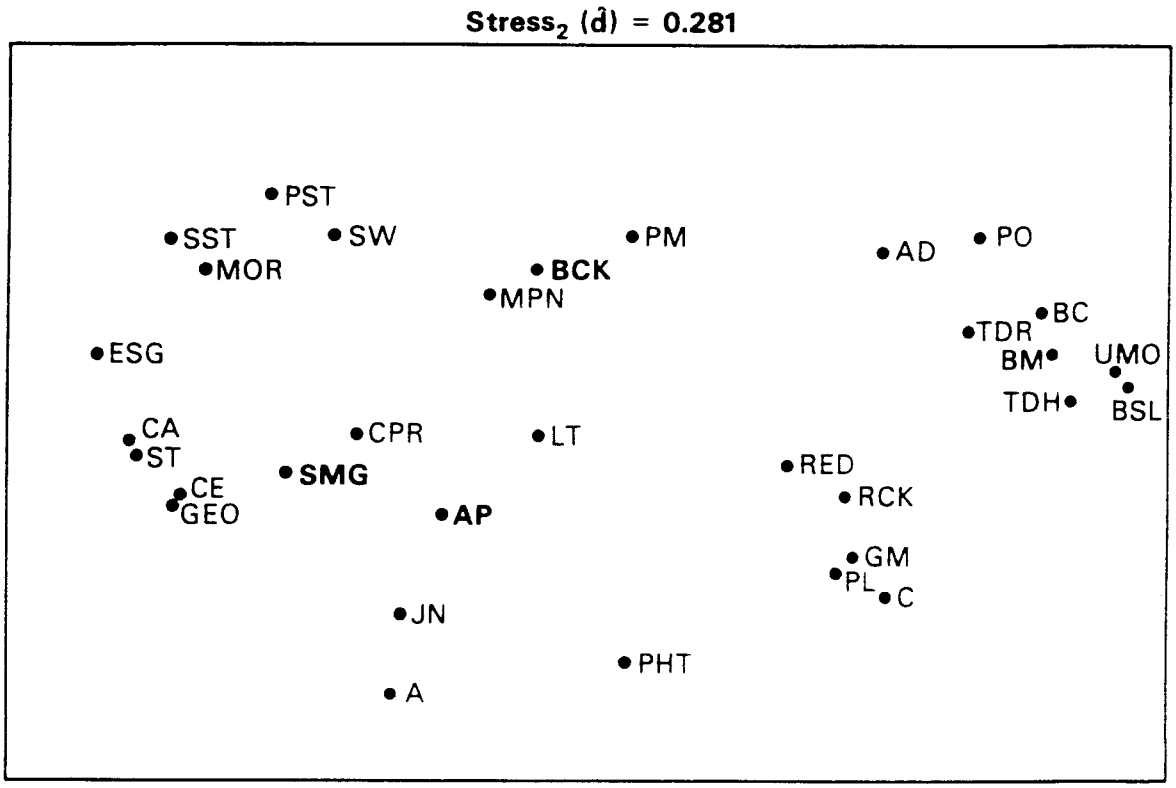


Figure 3.15 *2-D MDS final configuration (rotated to principal axes): co-occurrence of 32 occupational titles*

After the final, 23rd iteration, the configuration is rotated to principal axes and output, together with information on fit (stress values) and, if desired, also on the distances, the disparities and the residuals ($d_{jk} - d^0_{jk}$). The final configuration (centred and normalised) is given in Table 3.5, and is plotted in Figure 3.15. (Since the axes may be rotated at will, they are not drawn in; this allows attention to be concentrated upon characteristics of the configuration other than the arbitrary positioning of the axes.)

3.6.2 Diagnostics

Before beginning to interpret the final configuration, it is a good practice to assess first the adequacy and stability of the solution. The main questions are:

(a) Is a configuration with this stress value an acceptable approximation to the data?

(b) Is the dimensionality we have chosen likely to be the correct one?

(The answers to these questions are deferred until the next section.) Of more immediate relevance is the following set of questions:

(c) How well is each datum fit by the solution?

Are the residual values evenly distributed, or concentrated in a few rather badly fit instances?

Is one particular stimulus causing most of the trouble?

3.6.2.1 Analysis of residuals

The basic information necessary to answer all these questions is contained in the final Shepard diagram (see Figure 3.14b), which should be read like any other regression plot. It consists of a scatter plot of the data points with the best-fitting transformation function (in this case, the weak monotonic or ordinal function) drawn through it. The horizontal spread from this function represents the overall

Occupational Titles Abbrev.	No.	Title	*Final Configuration Dimensions* I	II
CA	1	Chartered accountant	-1.246	-0.082
SST	2	Secondary school teacher	-1.152	0.472
GM	3	Garage mechanic	0.733	-0.392
BM	4	Barman	1.262	0.156
ST	5	Statistician	-1.241	-0.117
SW	6	Social worker	-0.703	0.488
C	7	Carpenter	0.814	-0.496
AD	8	Ambulance driver	0.804	0.416
CPR	9	Computer programmer	-0.638	-0.061
MOR	10	Minister of religion	-1.045	0.391
PL	11	Plumber	0.678	-0.431
MPN	12	Male psychiatric nurse	-0.267	0.321
BCK	13	Bank clerk	-0.144	0.391
PST	14	Primary school teacher	-0.886	0.574
UMO	15	Unskilled machine operator on a factory assembly line	1.445	0.110
PM	16	Policeman	0.108	0.473
CE	17	Civil engineer	-1.126	-0.229
PHT	18	Photographer	0.094	-0.678
BSL	19	Building-site labourer	1.485	0.049
RCK	20	Restaurant cook	-0.704	-0.227
AP	21	Airline pilot	-0.399	-0.285
A	22	Actor	-0.556	-0.785
RED	23	Railway engine driver	0.555	-0.149
PO	24	Postman	1.078	0.417
GEO	25	Geologist	-1.143	-0.254
SMG	26	Sales manager	-0.827	-0.154
TDH	27	Trawler deckhand	1.335	0.033
TDR	28	Taxi driver	1.049	0.221
ESG	29	Eye surgeon	-1.353	0.156
JN	30	Journalist	-0.528	-0.543
LT	31	Laboratory technician	-0.138	-0.059
BCR	32	Bus conductor	1.249	0.275
		Mean	0.000	0.000
		S.D.	27.762	4.238

$\text{Stress}_2(\hat{d}) = 0.2808$ (primary approaches to ties)

Table 3.5 *Final 2-dimensional configuration of the data from Table 3.4*

degree of fit (recall that stress is a normalised measure of the dispersion of the distances from this monotonic regression function). The further a data point is from its corresponding disparity value (measured on the distance axis) the worse fit it is, and the larger the associated residual value ($d_{jk} - \hat{d}_{jk}$) will be. The matrix of residual values, rounded and multiplied by 10 (so that 0 represents a residual between 0 and 0.499, 1 a residual between 0.5 and 1.499 etc.) is presented in Table 3.6, accompanied by a frequency diagram. As is often the case, the distribution of residuals is strongly skewed toward the small values, indicating overall goodness of fit. But it is worthwhile giving special attention to the high residual values, and especially to the eighteen with values over 0.45 (i.e. values of 5 and 6 in Table 3.6). A

CA	1	–																																	
SST	2	0	–																																
GM	3	0	2																																
BM	4	3	0	3																															
ST	5	1	0	2	1																														
SW	6	1	2	0	0	0																													
C	7	1	2	0	2	0	1																												
AD	8	1	1	1	2	1	5	1																											
CPR	9	0	1	0	2	0	2	1	2																										
MOR	10	0	0	0	0	2	0	0	2	4																									
PL	11	1	2	0	2	0	0	0	0	1	0																								
MPN	12	0	2	2	1	0	2	4	3	2	1	2																							
BCK	13	5	1	1	0	5	0	0	0	1	1	1	6																						
PST	14	1	1	0	1	0	0	2	2	0	0	0	1	2																					
UMO	15	0	0	0	1	3	0	0	0	0	3	1	2	0	0																				
PM	16	2	4	0	2	3	2	3	0	1	3	2	2	3	3	1																			
CE	17	0	1	4	0	3	0	3	4	0	0	5	0	1	0	4	0																		
PHT	18	2	0	2	4	0	0	1	0	1	1	2	2	1	0	1	2	2																	
BSL	19	0	0	0	0	3	2	1	0	1	0	1	1	0	0	0	2	4	1																
RCK	20	1	3	1	1	3	3	0	1	4	1	0	3	2	0	0	1	0	0	1															
AP	21	3	1	1	2	0	2	1	1	4	0	0	4	6	3	0	1	2	4	0	4														
A	22	0	0	1	4	0	3	1	1	1	4	2	3	0	1	0	1	2	2	0	4	5													
RED	23	1	1	0	2	1	2	0	0	2	1	1	0	1	1	1	0	0	1	0	4	2	2												
PO	24	0	5	0	1	0	1	0	0	1	3	1	0	2	3	0	3	1	0	1	0	1	0	1											
GEO	25	1	2	1	0	1	1	3	2	2	1	4	1	3	0	0	0	2	0	4	1	0	1	2	1										
SMG	26	1	1	3	5	2	2	1	1	4	3	3	3	3	1	0	2	6	1	0	1	5	0	2	3	5									
TDH	27	2	0	1	1	3	2	1	1	1	0	0	3	0	1	0	0	1	0	0	0	3	0	0	0	3	1								
TDR	28	0	2	1	1	1	0	1	1	0	3	1	1	0	0	2	1	2	3	1	2	5	4	0	1	0	2	2							
ESG	29	2	2	0	0	4	0	0	3	2	0	2	4	5	1	0	0	0	2	0	0	5	2	2	1	0	4	3	0						
JN	30	1	3	0	0	0	2	0	1	3	2	0	1	1	4	0	3	2	4	0	2	6	0	2	0	1	2	1	1	1					
LT	31	1	1	2	0	1	1	3	2	1	0	2	2	3	0	1	1	2	2	0	0	4	5	1	4	2	3	1	1	2	4	–			
BCR	32	1	0	0	1	3	0	1	0	3	0	1	2	1	0	0	3	0	1	1	1	5	2	1	0	1	4	0	0	0	0	3			
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)	(21)	(22)	(23)	(24)	(25)	(26)	(27)	(28)	(29)	(30)	(31)	(32)			

Table 3.6 *Absolute values of residuals* ($d_{jk} - \hat{d}_{jk}$), ×10, rounded

look along row. and down column, 21 of Table 3.6 (corresponding to the Airline Pilot) shows that seven of the worst fit values occur in association with this one occupational title. A similar inspection will show that 26 (Sales Manager) and 13 (Bank Clerk) are also over-represented among the highest residuals.

A more systematic way of examining where badness of fit is concentrated is to look at the contribution which each point makes to the overall stress value, and this is done in Figure 3.16.† Note that we are taking into account all the residuals in which each point is involved. and not just the extreme ones. Nonetheless, the same conclusion is evident: points 21 and 26 contribute significantly more than the others to the badness of fit. We do not know why these particular points should be so troublesome. but analysis presented elsewhere (Coxon and Jones 1979, p. 42 et seq.) suggests that the worst fit occupations have characteristics which are not common to the remaining ones. In any event, it is worth considering simplifying the analysis either by removing the worst fit point(s) by deleting the relevant rows and columns from the data matrix and re-running the program, or by re-running the program in a higher dimensionality to see whether the additional dimensions allow the fit to be significantly improved.

It could also be that we have encountered a local minimum, in the sense that some other configuration may exist with lower stress, which would locate these worst-offending points in another position but would be substantially similar in other respects. The only way to check this,‡ is to re-run the program with a different initial configuration—say, the current final configuration with the points relocated to where the user thinks they ought to be. In the present example, different starting configurations and methods of minimisation produce virtually identical configurations (compare Figure 3.15 with Figure 2.8 in Coxon and Jones. 1978a. p. 43) so we can conclude with a fair degree of certainty that the point locations are as accurate as they can be. Nonetheless, the actual location of the worst-fit points must be treated with considerable caution when interpreting the configuration. and to emphasise this, these points are in bold print in Figure 3.15. (We shall return to the interpretation of Figure 3.15 in the next chapter.)

3.6.3 Degenerate and trivial solutions

Occasionally. final configurations can be produced which have very low stress values but are substantively meaningless. This arises when the low stress value has been obtained by the program capitalising on some technical feature of the minimisation process. such as weak monotonicity, or upon some unanticipated features of the data. Such configurations are often termed 'degenerate' or 'trivial' solutions (see Shepard 1974, pp. 391–9).

A good example of this occurs in very highly clustered data. If the data are such that the stimuli fall into a small number of clusters where the dissimilarities within each cluster are uniformly smaller than those between the clusters, the effect will be to produce a very low stress solution. where points within a cluster will condense or

†The individual point contributions do not add up to the stress value, because we are actually examining all the $(p - 1)$ *pairwise* contributions to stress involving a particular point. The point contributions do not therefore contribute additively to stress.

‡Interactive MDS programs such as SPACES (Schneider and Weisberg, 1974) allow the user to re-position or delete points to see how stress values change.

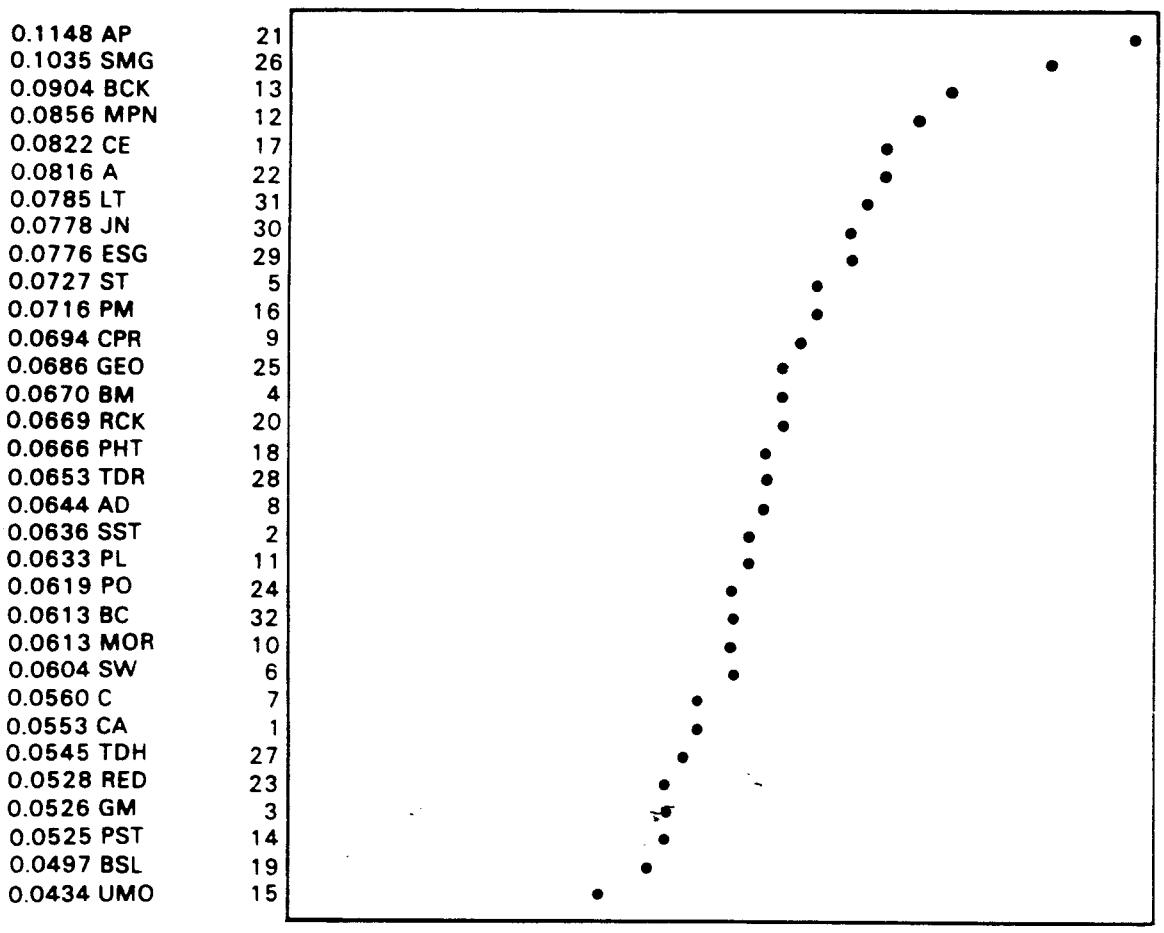


Figure 3.16 *Individual point contributions to total STRESS, of 0.2808 in two dimensions*

even collapse upon a single position and the program will then be able to minimise stress simply by maximising the distance between the positions of the clusters. This is not to say that if there are recognisable clusters of points in a final configuration the solution is therefore degenerate. For instance, it is not difficult in Figure 3.15 to spot at least two fairly coherent, genuinely distinct, clusters at the right hand side of the space, the groupings (TDR, BC, BM, UMO, TDH, BSL) and (RED, RCK, GM, PL, C), which comprise the 'unskilled' and 'trades' categories used by the subjects who made the judgments on which the data are based (Coxon and Jones 1979, pp. 39–41). However, if a degenerate clustering does occur, it is worthwhile making a separate scaling analysis of the stimuli involved in each cluster.

Two other examples of possible degeneracy have already been mentioned earlier when discussing types of monotonicity and different approaches to ties in the data. First, *weak* monotonicity allows distinct data values to be fit by the same disparity value: indeed, the block-averaging procedure used in monotone regression does just this. Usually there should not be a markedly smaller number of blocks (disparity values) than there are distinct data (dissimilarity values). However, if there are very few blocks and they contain a large number of entries, then the solution may well be degenerate. This situation shows up on the Shepard diagram in the form of long vertical segments on the monotone function, each with a large number of associated data points, because a large number of data are being fitted by a single disparity value.

Secondly, if the primary approach to ties is chosen, the program is given the freedom to fit different disparity values to data which have the same value, without this counting towards badness of fit. Sometimes this freedom is grossly exploited by the program, especially when the data contain only a few distinct values (for example, when a 5-point rating scale is used on pairwise judgments of a large number of stimuli). Where this occurs it is indicated on the Shepard diagram by the appearance of long horizontal segments on the monotone function, highly populated by data points.

It is very difficult to determine decisively whether a final configuration is 'really' degenerate or trivial, but this is not the point. Rather, the user should be alerted to the danger signs of artificially low values of stress which can often indicate serious loss of information, and to the reasons for their occurrence. At the very least, it is good practice always to inspect the Shepard diagram and the set of disparity values relating to the final configuration. If any of the tell-tale signs of a trivial or degenerate solution appear in a given solution, then a re-analysis should be made using options which counteract the weakness concerned.

1 If the researcher has objective information, or even a strong hunch, about what the configuration *should* look like, then Confirmatory MDS should be used, or alternatively the points whose positioning is known should be fixed (by the FIX POINTS command in SSAM or by using PREFMAP), and the analysis run to determine the position of the other points.

2 If the monotone function is very 'steppy', containing many long vertical and horizontal segments, then a more restrictive function which excludes such steps may be used in preference to the monotone function. A fairly common option is to choose to fit a *linear* (or even a power) function between the data and the distances, and this can be done by using the MRSCAL (*m*etric *scal*ing) program (see 6.1.4).

3 If the difficulties arise because the primary approach to ties has been used, the data can be re-analysed with MINISSA, using the *secondary* ties option. If they arise due to weak monotonicity, the data can be re-analysed using SSAM, applying the strong monotonicity option.

Having discussed the relatively rare and unusual problems of degeneracy, let us turn to the more important general issue of assessing the stress value of a configuration.

3.7 Stress, Dimensionality and Recovery of Metric Information

Three important and related issues arise in using MDS programs. These are:

(i) How is the stress value of a configuration to be interpreted?
(ii) What is the 'real' or 'best' dimensionality for a solution?
(iii) How well can non-metric MDS recover information if the data are 'noisy' or error-prone?

3.7.1 Evaluation of stress

A number of factors affect the value of stress. The most important are:

(i) *The number of points.* In general, the larger the number of points the more the information to be fitted and the higher the stress.

(ii) *The dimensionality of the solution.* The higher the dimensionality of the solution, the easier it is to fit the information, and therefore the lower the stress value. In general, it is possible to fit *any* data relating to p points perfectly in $p - 2$ dimensions. Such a solution is termed a 'trivial solution'.

As we have noted before, a further set of factors holds as a necessary result of options (or default values) chosen by the user in obtaining a solution, and discussed above. These are:

(iii) *The type of stress.* Raw stress/phi is necessarily larger than normalised stress and, because of the different normalising factors, stress_2 is normally about twice as large as stress_1.

(iv) *The type of monotonicity criterion.* Guttman's strong monotonicity criterion is more stringent than Kruskal's weak monotonicity criterion and therefore stress based on strong monotonicity will necessarily be larger than stress based on weak monotonicity.

(v) *Tying approach.* The secondary approach to ties, treating tied values as equivalences, produces higher stress values than the primary approach.

Because stress is affected by all these factors, it is meaningless to talk about what an 'acceptable value of stress is' without further specification. To simplify matters, we shall therefore restrict attention to the paradigm case of stress:

stressform 1;
based upon Kruskal's $\hat{d}$ (weak monotone) fitting quantities;
using the primary approach towards ties.

In asking what is an acceptable level of stress, we are asking a variant of the common statistical question, 'Does the non-metric MDS model fit the data well enough that the stress value could not have arisen by chance?' (Cf. Kruskal 1972b.) Put slightly differently, we advance the null hypothesis that some chance mechanism could have generated the data, and we use this as a bench-mark to assess how far the actual configuration departs from a random distribution. Very little progress has been made in analytically deriving a statistical distribution of stress, and recourse has been had instead to so-called Monte Carlo simulation methods (Young 1970; Wagenaar and Padmos 1971; Isaac and Poor 1974; Spence 1972; Spence and Graef 1974).

3.7.2 Simulation of stress values

The basic process in simulating the distribution of stress consists of producing a configuration of p points in t dimensions and calculating the distances between the points in the configuration. This is the 'true' or generating configuration. The distances (or in some cases the coordinates) are then distorted by adding to them 'noise', i.e. error in differing but specified amounts. Sometimes in addition the distances are transformed monotonically. The resulting set of error-perturbed distances are now treated as if they were dissimilarities data, and are scaled by a non-metric MDS distance model program in a number of different dimensions, including the 'true' dimensionality t (for we wish to assess the effect of scaling data in the wrong dimensionality). A large number of such dissimilarity matrices are

generated which vary in the number of points in the configuration, the 'true' dimensionality and the amount of error added. All of these matrices are scaled and the resulting $stress_1$ values are noted.

Spence and Graef's M-SPACE *procedure*

Although there are several variants of the procedure, the most well known is the Spence-Graef M-SPACE procedure (Spence and Graef 1974), a variant of which is incorporated in the MINISSA program in the MDS(X) series as a way of helping users decide upon the 'true' dimensionality and likely error present in their data.

Spence and Graef constructed random configurations containing a given number of points ($p = 12, 18, 26, 36$) in a given number of dimensions ($t = 1, 2, 3, 4, 5$), and then added error at five levels (from a unit normal distribution with standard deviations σ, of 0, 0.0625, 0.1225, 0.2550 and infinity) to each co-ordinate in this case. For each combination of points, true dimensionality and error level, a number of dissimilarity matrices was produced and these were scaled in five through one dimensions. The resulting $stress_1$ values were averaged and put together to produce a set of diagrams (nomographs) such as those reproduced in Figure 3.17.

This set of four diagrams refers to configurations of 36 points with *true* (generating) dimensionality of 1 (top left hand diagram), 2 (top right), 3 (bottom left) and 4 (bottom right). (Similar diagrams exist for from 12 to 36 points). Within each diagram there are five 'curves'—one for each dimensionality m in which the configurations were actually scaled. Each curve joins the average $stress_1$ values of the configurations as the level of added error is increased. For example, (see the top right hand diagram) in a true dimensionality of 2 and with no added error, the configurations of 36 points scaled in two, three, four and five dimensions yield a

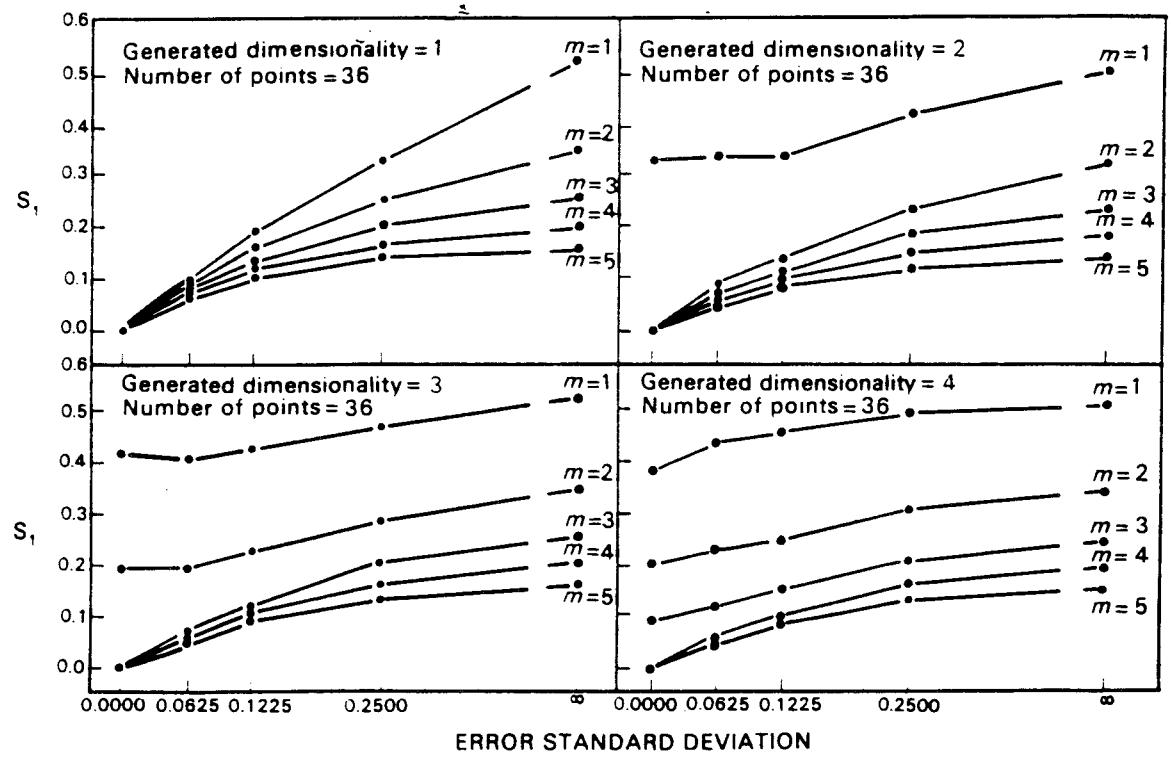


Figure 3.17 *Stress of a recovered configuration in spaces of varying dimensionality as a function of the error level in a known undelying configuration*

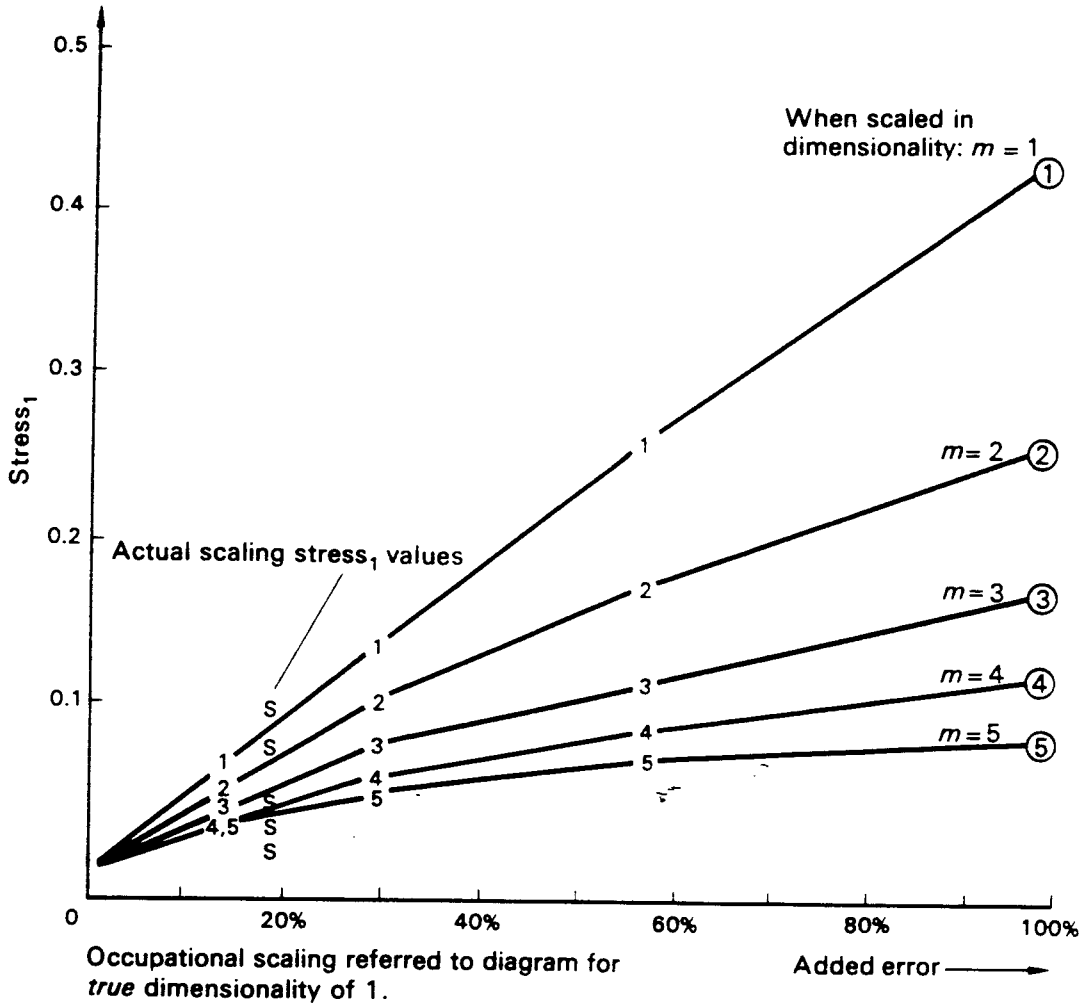


Figure 3.18 *Spence-Graef M-SPACE analysis*

zero-stress$_1$ solution, whilst those scaled in *one* dimension have a stress$_1$ value of about 0.32 on average. At each level of error, the stress$_1$ of the solutions in two through five-dimensions become more separated—and it is the difference between these values which will be important when the diagrams are used to interpret actual scaling results.

The idea of M-SPACE is simply to compare the actual set of stress$_1$ values which the user has obtained with the results of the Spence-Graef simulations, in order to assess the likely 'true' dimensionality, and the probable amount of error. An example will help illustrate the process. Coxon and Jones (1979, pp. 73–4) scaled a set of averaged pairwise similarity ratings referring to 16 occupations, obtaining stress$_1$ values of: 0.024, 0.030, 0.043, 0.060 and 0.100 for solutions in 5, 4, 3, 2 and 1 dimensions. This 'actual set' of stress values was then compared to Spence and Graef's results obtained from scaling random configurations of 16 points. The M-SPACE procedure compares the actual set of stress$_1$ values with each 'true dimensionality' diagram in turn, finding the point at which the actual set fits the simulated results most closely by locating the point along the error axis at which the actual stress$_1$ values conform most closely to the simulated ones. As a result, the user is given the error level to which the actual set of stress$_1$ values best corresponds by reference to each 'true' dimensionality in turn. Usually the fit will be best in one particular 'true' dimensionality, and this information is taken to

mean that this is the most likely underlying dimensionality of the user's data. To our surprise, the use of M-SPACE on these data strongly suggested a 'true' one-dimensional (or possibly a two-dimensional) solution and the relevant M-SPACE nomograph for true dimensionality of 1 is presented in Figure 3.18: the 'actual set' fits best the true dimensionality of 1 for an error level of 18 per cent, (on any account a low level of error. and by far the best fitting in any of the true dimensionalities). M-SPACE should not be used uncritically as an automatic detector of 'real' dimensionality: if anything, it tends to underestimate dimensionality, and the authors mention a number of other cautions (Spence and Graef 1974, p. 3).

The results of other simulation studies point in the same direction and confirm Shepard's (1966) intuitions about the number of points and dimensions needed to produce a stable solution. Klahr (1969) shows, for example, that stress values of sets of randomly generated dissimilarities for between 6 and 8 points in three-dimensions yield 'good' solutions according to Kruskal's original rules! Fortunately, a small increase in points rapidly diminishes the likelihood of such a mistaken inference. Stenson and Knoll's (1969) study is interesting mostly for the evidence it provides of the effect on $stress_1$ of the choice of primary or secondary approach to ties. To estimate this, he ties dissimilarity values for 30 points 'coarsely' (into 10 approximately equal sets of tied values) and 'finely' (into 50 such sets). Surprisingly the fineness of grouping has little apparent effect on $stress_1$ values. Wagenaar and Padmos (1971) were the first to provide a realistic and systematic investigation of the effect of adding error into the process of generating the dissimilarity matrices, and their results here have been generalised by Spence and Graef.

3.7.3 Other approaches to assessing dimensionality, stability and metric recovery

From the practice and lore of factor analysis come other approaches to determine the likely dimensionality of the data. The most famous is the 'scree test', or 'elbow test'. The recommended strategy here is to perform the scalings in a high number of dimensionalities, stepping down to a uni-dimensional scaling. The stress values are then charted against the dimensionality, and joined together to form a polygon. If a noticeable bend or 'elbow' occurs, indicating that the improvement in fit is not significantly altered by the addition of a further dimension, then the lower-dimensional solution is to be preferred. A common variant of this rule is ascribed (probably unfairly) to Shepard and often referred to as 'Shepard's Law': if a solution exists, it probably exists in two dimensions. Extended a little, such a rule contains a good deal of sense. Uni-dimensional solutions are quite often degenerate and unable to portray a situation that sometimes occurs—that the points in fact form a non-linear, but uni-dimensional sequence such as a 'horseshoe' (see Kendall 1971a, pp. 224–7 and 4.6 below). So even if a uni-dimensional solution is likely, it is prudent to scale in two dimensions. A two-dimensional solution is the easiest to comprehend because it can be readily assimilated, and whereas a third dimension can be relatively easily visualised, higher dimensionalities, can not. 'If a solution exists, it probably exists in two dimensions; if it doesn't then it certainly exists in three', might be a reasonable extension to Shepard's law.

As we have seen, Shepard (1966, p. 288) shows that non-metric constraints, if

imposed in sufficient number, begin to act like metric constraints. That is, information on the pairwise ordering of stimuli is normally sufficient to constrain the location of points in a configuration to such an extent that the distances between them are virtually fixed. The key word is 'normally'. In fact, the number of data need to exceed the number of points times the number of dimensions to a considerable degree before the configuration is really stable. Again, Monte Carlo studies have been used extensively to see how well non-metric scaling can recover *known* configurations (see especially Young 1970 and Shepard 1966). The results are conclusive: so long as the number of order relations in the data exceed the number of co-ordinates of the solution by at least a factor of 2 (and so long, obviously, as there is not an overwhelming amount of error), non-metric analysis can recover the correct underlying configuration (or, rather, the distances) extremely accurately. For 'true' dimensionality of two, the extent of metric recovery has been studied extensively, and the results of Shepard (1966, p. 299) still hold:

> While the reconstruction of the configuration can occasionally be quite good for a small number of points, it is apt to be rather poor (for p less than eight, say). As p increases, however, the accuracy of the reconstruction systematically improves until even the worst of ten solutions becomes quite satisfactory with 10 points, and, to all practical purposes, essentially perfect with 15 or more points.

But, as he later points out (Shepard 1974, p. 395), the lesson has not always been well heeded by users:

> A distressing number of two- and even three-dimensional solutions have been published in which, despite the inclusion of only six to eight objects, no evidence is provided that the configuration has a reasonable degree of metric determinacy and is not a prematurely arrested case of convergence toward a degeneracy.

A footnote should be added to these studies of recoverability and stability of MDS solutions which is of particular importance to users who wish to have subjects judge a large number of stimuli, and is also of general importance in interpreting a configuration:

Information about *larger* distances/dissimilarities is far more crucial in ensuring satisfactory recovery of metric information than is information about medium or smaller dissimilarities; and

small distances in the solution are far less stable than large ones, implying that 'global' information (between distant points or clusters of points) is a more reliable basis for interpreting a solution than 'local' information between highly proximate points.

These issues are given extended treatment in an important study by Graef and Spence (1979).

APPENDIX A3.1 COMPARISON OF MEASURES OF FIT BETWEEN DATA AND SOLUTION USED IN NON-METRIC MDS

1 All measures of fit used in the basic model compare a set of disparities (ratio-level quantities which are a function of the data, $d^0_{jk} = f(\delta_{jk})$, which are *monotonic* for ordinal data and *linear* for metric or interval data with the ratio-level distances, d_{jk}. Two forms of comparison are used:

(i) *the difference* $(d_{jk} - d^0_{jk})$, which forms the basis of badness-of-fit measures, since the greater the discrepancy between a solution and the data, the greater will be the differences; and

(ii) *the scalar product* $(d_{jk}d^0_{jk})$, which forms the basis of goodness-of-fit measures, since the greater the covariance (or the less the angular separation) between data and the solution, the greater will be the scalar products.

2 The basic measure of goodness-of-fit used in non-metric programs emanating from the Bell Laboratories is *stress*, and in particular (normalised) stress$_1$, based upon Kruskal's BFMF disparities, $\hat{d}_{jk}$. These and alternative measures are dealt with in sections 3.3 and 3.5.2 and are extensively discussed in Kruskal and Carroll, 1969.

3 The measures used by Lingoes (Michigan), Guttman (Israel) and Roskam (Nijmegen) include stress measures (often based upon Guttman's rank-image disparities, d^*_{jk}), but also include a number of less familiar measures. In particular:

(a) Mu $(\mu) = \sum d_{jk}d^0_{jk}$

$$\sqrt{\left\{\sum d^2_{jk} \sum \left(d^0_{jk}\right)^2\right\}}$$

(Goodness of fit, varies between -1 and $+1$)

This measure is akin to the Pearsonian correlation coefficient. It is independent of the scale of both the distances *and* the disparities if the data are scaled by a ratio transformation, as in the metric MDS model (with the logarithmic option) implemented as MRSCAL in the MDS(X) series.

If the data are scaled by a linear transformation, as in MRSCAL (under the linear option), then mu is formally identical to the Pearsonian correlation coefficient r. It may also be used for monotone transformations, but it is not entirely clear whether it is dependent in this case on the scale of the disparities.

(b) Alienation (K) $= \sqrt{(1 - \mu^2)}$ (Badness of fit, varies between 0 and 1)

This measure is akin to stress$_1$ and in some cases is identical to it. In any event, K is strictly monotonic with stress$_1$. The coefficient measures the extent of residual variance from the fitted regression.

(c) (Normalised) Phi (ϕ) = (Raw Stress/(2 × NF 1))
$= \sum (d_{jk} - d^0_{jk})/2 \sum d^2_{jk}$
(Badness of fit, varies between 0 and 1)

This measure is also akin to stress_1, but differs in the scaling factor—twice that of stress_1—and in the fact that the index is not reduced by its square root. It differs from the coefficient of alienation K, in being based upon the difference, rather than the scalar product, of the distances and the disparities.

Strictly speaking, any of these three measures may be used either with Kruskal's monotone regression disparities $\hat{d}_{jk}$, or Guttman's rank images d^*_{jk}, although by convention they are normally used with the latter.

4 *Relation between fit measures*

Relationships between the fit measures depend most importantly on whether Kruskal's $\hat{d}$ or Guttman's d^* quantities are being used. (In reporting measures of fit, users should always indicate which fitting quantities are being referred to and MDS(X) programs indicate the referent quantities as d-hat and d-star respectively.) In general, for any of these badness-of-fit measures, a measure based on d^* will be higher (indicating worse fit) than the same measure based on $\hat{d}$, since the former attempts to preserve strong monotonicity and the latter preserves only weak monotonicity with the data.

This can best be exemplified by relating various measures to μ, which represents the cosine of the angle separating the distances and the fitting quantities, d^0, in the measurement space (see Roskam 1969, p. 13):

	Fitting quantities (disparities)	
Measure	Kruskal's $\hat{d}$ (weak)	Guttman's d^* (strong)
μ	$\cos(d, \hat{d})$	$\cos(d, d^*)$
Stress_1 (S_1)	$\sqrt{(1 - \mu^2)}$	$\sqrt{2(1 - \mu)}$
Phi	$\frac{1}{2}(1 - \mu^2)$	$(1 - \mu)$
K (Alienation)	$(= S_1)$	$\sqrt{(1 - \mu^2)}$

Other useful relationships are as follows:

Alienation and phi (i) $K = \sqrt{1 - (1 - \phi)^2} = \sqrt{(1 - \mu^2)}$

Alienation and stress_1 (ii) If $\hat{d}$ is used, $K = S_1$, and if d^* is used, $K = S_1\sqrt{1 - (\frac{1}{2}S_1)^2}$

Phi and stress_1 (iii) $\phi = \frac{1}{2}S^2$

APPENDIX A3.2 CREATION OF THE INITIAL CONFIGURATION

A3.2.1 User-provided configuration

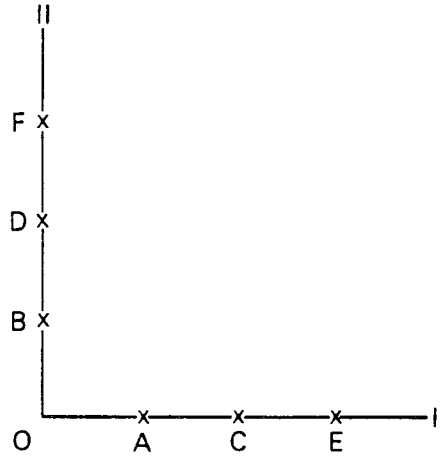
Most MDS programs give the user the option of providing an initial configuration. Usually, this will be a configuration thought to be close to the final configuration, either on a priori grounds or it will be a configuration from a similar study.

A3.2.2 Random or arbitrary start

The initial configuration may be a *random* start, formed simply by allocating random numbers to the $p \times r$ co-ordinates, or it may be an *arbitrary* start, positioning the points regularly at unit intervals along the dimensions of the initial configuration, as in the following 2-dimensional case, where they form a regular L-shaped configuration:

Stimulus	Dimension I	Dimension II
A	1	0
B	0	1
C	2	0
D	0	2
E	3	0
F	0	3

(a) *Co-ordinates*



(b) *Corresponding Configuration*

In general, such a configuration can be produced from the series:

Stimulus	Dimension I	II	III	...	r
A	1	0	0	...	0
B	0	1	0	...	0
C	0	0	1	...	0
⋮		⋮			
(r th)	0	0	0	...	1
($r + 1$ th)	2	0	0	...	0
($r + 2$)	0	2	0	...	0
⋮		⋮			
($2 \times r$ th)	0	0	0	...	2
	3	0	0	...	0
	0	3	0	...	0

... and so forth.

Such a configuration ensures that the co-ordinates are orthogonal, no matter what the dimensionality. Usually, the configuration is also centred and normed. It is the most common method used by Kruskal to create an initial configuration. It has the advantage that it in no way prejudices or influences the shape of the final configuration, but it is known to make the iterative procedure especially prone to sub-optimal (local minimum) solutions (Lingoes and Roskam 1973, p. 69) and should generally be avoided.

A3.2.3 Metric initial configuration

The data are treated as estimates of Euclidean distances, converted into scalar products, and the eigenvectors corresponding to the first r largest eigenvalues are used as the best (least squares) estimate of an r-dimensional initial configuration. This option is identical to Torgerson's (1958, pp. 254–9) classic method of metric MDS (see Appendix A5.2) and is closely allied to principal components analysis and Eckart-Young singular value decomposition. It generally produces a fairly good initial estimate of the solution, unless the configuration of points forms some highly non-linear shape (cf. Arabie and Boorman 1973). It is the strategy adopted by the programs TORSCA (Young 1968) and KYST (Kruskal et al. 1973) to form the starting configuration.

A3.2.4 Quasi non-metric initial configuration

Here the data are first reduced to rank order, thereby jettisoning all non-ordinal information. From these data a ranks matrix, **C**, is formed:

$$c_{jk} = \begin{cases} 1 - \rho_{jk}/r & (j = k\text{: off-diagonal elements}) \\ 1 + \sum_{l} \rho_{il}/r & (j = k\text{: diagonal elements}) \end{cases}$$

where ρ_{jk} is the rank number of dissimilarity δ_{jk} and r is the maximum rank number.

The entries of **C** are similar to scalar products, and are a strict monotone function of data.

A principal components analysis is performed on **C**, dropping the first (constant) eigenvector, $\mathbf{C} = \mathbf{FF}'$, and **F** is then the Guttman-Lingoes-Roskam initial configuration (see Roskam 1975, A7–8; Lingoes and Roskam 1973, pp. 17–19). Like the metric initial configuration, it will be quite close to the final configuration, and so will greatly reduce the number of iterations. Unlike the metric start, the quasi non-metric configuration has the advantage of using only ordinal information, and cannot therefore capitalise on possibly irrelevant quantitative (interval level) properties of the data.

In the MDS(X) series, MINISSA (and all programs in the Guttman-Lingoes-Roskam tradition) uses the quasi non-metric method of producing an initial configuration, and also allows users to input an initial configuration of their choice, if preferred. Although the quasi non-metric initial configuration certainly seems to guard best against suboptimal solutions, users are strongly advised to check solutions by using several different initial configurations.