LOCAL MINIMUM SOLUTIONS IN NONMETRIC MULTIDIMENSIONAL SCALING

Ian Spence, University of Toronto

Although it has been widely recognized that nonmetric scaling algorithms are potentially susceptible to local minimum problems, there is little systematic data on the relative frequency with which these programs will become trapped in nonoptimal stationary positions. This study provides detailed information on this question.

An empirically obtained measure of the dissimilarity of any two objects in a set P of n objects gives rise to an order relation on P x P. Nonmetric multidimensional scaling seeks to represent this dissimilarity ordering as a geometrical model by mapping the objects of P into a metric space, where the ordering of the distances corresponds to the observed ordering of the dissimilarities. Thus, if s_{ij} is the dissimilarity measure obtained for objects i and j,

$$(1) \qquad d_{ij} \leqq d_{kl} \text{ iff } s_{ij} \leqq s_{kl} \quad ,$$

where d_{ij} is the metric distance between points i and j.

Practical computer algorithms (e.g. Kruskal [2], McGee[4], Guttman[1], Young[7]) for obtaining a configuration of n points in m-dimensional space have been based on an iterative procedure designed to minimize the residual sum of squares (called "stress")

$$(2) \qquad S = \sum_{i,j}^{n} (d_{ij} - \delta_{ij})^2 \quad ,$$

or some other quantity which is monotonically related to S (see Spence[6]). The $\{d_{ij}\}$ are <u>metric distances</u>, and the $\{\delta_{ij}\}$ are <u>pseudo distances</u>. The pseudo distances have the property of being order isomorphic to the dissimilarities $\{s_{ij}\}$, and have the same scale as the $\{d_{ij}\}$. Further, if the $\{\delta_{ij}\}$ are chosen to minimize S, given a particular set of $\{d_{ij}\}$, the resulting pseudo distances are known as the $\{\hat{d}_{ij}\}$ (Kruskal [2]). The minimization of S is not trivial since an optimal set of $\{\delta_{ij}\}$ is not known. Consequently, an alternating algorithm has been proposed (Kruskal[2], Guttman[1]): this procedure switches between satisfying the metric distance requirement and the order isomorphy requirement, with the hope that satisfying both in turn will eventually result in the algorithm arriving at a stationary position where both requirements will be optimally satisfied. In the Euclidean case, the following algorithm may be used:

1. Choose an initial $X_{n \times m}$: a configuration of points in m-space.
2. Compute $D_{n \times n}$ from $X_{n \times m}$, where
$$d_{ij} = \left[\sum_{a=1}^{m} (x_{ia} - x_{ja})^2 \right]^{1/2} \quad .$$
3. Choose $\Delta_{n \times n}$ with general entry δ_{ij} .
4. Move $D_{n \times n}$ towards $\Delta_{n \times n}$ by use of a gradient algorithm to minimize S. Typically only one step is taken, viz.,
$$(3) \quad x'_{ia} = x_{ia} - \frac{1}{n} \sum_{j=1}^{n} \left(1 - \frac{\delta_{ij}}{d_{ij}} \right) (x_{ia} - x_{ja}) \quad ;$$
however, if desired, more than one step may be taken--with a fixed set of $\{\delta_{ij}\}$--as in Guttman[1].
5. Test for termination: if further improvement is desired, go to 2 with the new X. Else,
6. End.

TABLE 1

Summary of the Essential differences among the Algorithms

	Algorithms		
Options	M-D-SCAL	SSA-1	TORSCA
Choice of initial configuration.	does not make full use of information in the input data.	attempts to make "optimal" use of the input data.	attempts to make "optimal" use of the input data.
Choice of pseudo distances, $\{\delta_{ij}\}$.	to minimize S, hence these are the $\{\hat{d}_{ij}\}$.	ascending permutation of the $\{d_{ij}\}$ called $\{d^*_{ij}\}$. Does not minimize S.	to minimize S, hence these are the $\{\hat{d}_{ij}\}$.

There are two arbitrary points in the specification of the above algorithm: both the initial configuration and the pseudo distances may be chosen in a variety of ways. These choices may be expected to have some effect on the overall performance of the algorithm, especially with respect to the avoidance of nonoptimal stationary solutions.

In a study which is reported in greater detail elsewhere (Spence[6]), three widely used algorithms were compared using Monte Carlo techniques. The programs were Kruskal's M-D-SCAL (Kruskal[3]), Guttman-Lingoes's SSA-I (Guttman [1]), and Young-Torgerson's TORSCA-9 (Young[7]). The essential differences among these procedures are summarized in Table 1. Known configurations were used as the basis for computing sets of dissimilarities which were then scaled by each of the algorithms. The recovered configurations were then compared with the known generated configurations. Specifically, the following method was used:

1. Choose n and m and generate the n x m coordinates of X by sampling from the rectangular interval [-1, +1] with the additional constraint that all points lie within a hypersphere of unit radius.

2. Compute $d^e_{ij} = [\sum_{a=1}^{m} (x^e_{ia} - x^e_{ja})^2]^{1/2}$,

where $x^e_{ia} = x_{ia} + N(0, \sigma^2_k)$, and

$\sigma_k = 0.00; 0.15; 0.25; 0.35;$ is the set of error values used. This method of injecting error corresponds to a multidimensional analogue of Thurstone's Case V (see Ramsay[5]). The dissimilarity matrices were computed with entry

$$(4) \quad s_{ij} = \begin{cases} 1.8\,(d^e_{ij})^2 + 5.5 & \text{if } \sigma_k = 0.00, \\ d^e_{ij} & \text{otherwise.} \end{cases}$$

3. Obtain scaling solutions in m = 1, ... , 5 recovered dimensions using each of the algorithms. Compute $\hat{S}_1 = [\sum (d - \hat{d})^2 / \sum d^2]^{1/2}$ and $r(d,d_t)$ --the product moment correlation coefficient between the recovered and true distances--as measures of goodness of fit and metric recovery respectively.

In a single computer run n was varied from 6 to 36 and m from 1 to 4; in total, each of the three algorithms processed:

DISTINCT CONFIGURATIONS(18) x ERROR LEVELS(4)

x RECOVERED DIMENSIONS(5) = 360 SOLUTIONS .

Two replications were obtained; hence, 2160 solutions were computed.

Results and Discussion

The initial configurations generated by the TORSCA program were invariably much better, in terms of goodness of fit and metric recovery, than the approximations generated by the other two algorithms. SSA-I and M-D-SCAL did not differ significantly in their abilities to produce a starting configuration. The final solutions produced by the three procedures were, in the vast majority of cases, virtually identical; indeed, analysis of variance showed that the hypothesis of no differences among the programs could not be rejected (see Spence[6]). However, in some of the solutions, one or more of the algorithms obtained a solution which deviated considerably from the best attempt.

A simple method was used to investigate this local minimum problem: it was assumed that at least one of the three algorithms would, in all cases, get very close to the global minimum. This seems to be a plausible assumption, since these programs use rather different methods to produce a solution. Furthermore, this assumption is reinforced by the fact that inspection of the data showed that in almost all cases at least two of the algorithms finished in virtually identical positions. Using the criterion of goodness of fit defined above, $\hat{S}_1$, and considering each of the 720 solutions attempted per algorithm separately, the deviations of the values obtained by the other two programs from the lowest stress value were computed. If any deviation exceeded a preset threshold criterion, then the deviating algorithm was considered to be in a local minimum position.

The results of this analysis are shown in Table 2 for different values of the threshold (varying from 0.005 to 0.050). The 0.005 threshold is probably too stringent a criterion since a deviation of this magnitude may simply indicate that the offending algorithm was moving slowly in the region of the minimum, and had not quite converged. However, as the threshold size is increased to 0.010, and above, an interesting pattern emerges. It seems to be clear that TORSCA is least troubled by local minimum problems (in terms of the above operational definition), and SSA-I is only a little worse; although, it does appear from the data that SSA-I may not have fully converged, since the vast majority of SSA-I deviations are of the order of 10^{-3}. M-D-SCAL appears to be much more sensitive to local minimum problems than the other two algorithms, and, more importantly, 27 of its solutions deviated by more than 0.050 from the presumed minimum stress value. By contrast, only one of the 1440 SSA-I and TORSCA solutions deviated by more than this amount from the minimum value, and, in fact, only five of the SSA-I and TORSCA solutions deviated by more than 0.030. In one dimension the problem seems to be especially severe, and consequently, in practical situations, it would be prudent to regard any one dimensional solution with some suspicion, irrespective of the method used to derive the initial configuration.

Conclusions

It is reasonable to attribute the excellent performance of TORSCA to its very good initial configuration; as has already been noted, the TORSCA starting position was easily the best. Likewise, the poorest performance (by M-D-SCAL) may be partially attributed to its often unsatisfactory initial configuration. The fairly good performance of SSA-I is rather more difficult to explain on the basis of the quality of the initial configuration; the SSA-I initial configuration was not much better, in general, than the M-D-SCAL starting configuration. Indeed, it is

TABLE 2

Frequency of Deviation from the Presumed Minimum Stress Value

Threshold	Program	Dimensionality					Totals
		1	2	3	4	5	
0.005	TORSCA	30	11	8	8	21	78
	SSA-I	25	32	36	44	38	175
	M-D-SCAL	64	29	10	10	10	123
0.010	TORSCA	18	0	4	1	2	25
	SSA-I	14	12	12	16	8	62
	M-D-SCAL	55	22	5	3	5	90
0.020	TORSCA	5	0	0	0	0	5
	SSA-I	3	2	4	5	0	14
	M-D-SCAL	43	9	4	1	4	46
0.030	TORSCA	2	0	0	0	0	2
	SSA-I	2	0	0	1	0	3
	M-D-SCAL	32	5	4	1	4	46
0.050	TORSCA	0	0	0	0	0	0
	SSA-I	1	0	0	0	0	1
	M-D-SCAL	19	0	3	1	4	27

Note.--The total number of solutions per algorithm was 720, hence there were 144 solutions per algorithm per dimension.

probable that the use of the $\{d^*_{ij}\}$ produces a more "jumpy" algorithm which tends to step over small local depressions on its way to the minimum.

Acknowledgement

This research was partially supported by a predoctoral fellowship from the National Research Council of Canada, and also by an NRC grant (APA-176) to John C. Ogilvie.

REFERENCES

[1] Guttman, L. A general nonmetric technique for finding the smallest coordinate space for a configuration of points. Psychometrika, 33 (1968), 469-506.

[2] Kruskal, J.B. Nonmetric multidimensional scaling: A numerical method. Psychometrika, 29 (1964), 115-129.

[3] Kruskal, J.B. How to use M-D-SCAL, a program to do multidimensional scaling and multidimensional unfolding. Unpublished MS (1968). Bell Laboratories, Murray Hill, New Jersey.

[4] McGee, V.E. The multidimensional analysis of "elastic" distances. British Journal of Mathematical and Statistical Psychology, 19 (1966), 181-196.

[5] Ramsay, J.O. Some statistical considerations in multidimensional scaling. Psychometrika, 34 (1969), 167-182.

[6] Spence, I. Multidimensional scaling: An empirical and theoretical investigation. Unpublished Ph. D. Thesis. University of Toronto (1970).

[7] Young, F.W. A FORTRAN IV program for nonmetric multidimensional scaling. L.L. Thurstone Psychometric Laboratory Report No. 56 (March, 1968). University of North Carolina, Chapel Hill, North Carolina.