PSYCHOMETRIKA—VOL. 49, NO. 3, 289–310
SEPTEMBER 1984

THE ESTIMATION OF ULTRAMETRIC AND PATH LENGTH TREES FROM RECTANGULAR PROXIMITY DATA

GEERT DE SOETE

UNIVERSITY OF GHENT

WAYNE S. DESARBO

AT&T BELL LABORATORIES

GEORGE W. FURNAS

BELL COMMUNICATIONS RESEARCH

J. DOUGLAS CARROLL

AT&T BELL LABORATORIES

A least-squares algorithm for fitting ultrametric and path length or additive trees to two-way, two-mode proximity data is presented. The algorithm utilizes a penalty function to enforce the ultrametric inequality generalized for asymmetric, and generally rectangular (rather than square) proximity matrices in estimating an ultrametric tree. This stage is used in an alternating least-squares fashion with closed-form formulas for estimating path length constants for deriving path length trees. The algorithm is evaluated via two Monte Carlo studies. Examples of fitting ultrametric and path length trees are presented.

Key words: Cluster Analysis, Trees

Introduction

Two-way, two-mode $n \times m$ asymmetric rectangular matrices ($\Delta = \| \delta_{ij} \|$) of proximity data are quite frequently collected in the behavioral (and other) sciences. They indicate the relationships between two different classes of entities (e.g., objects and variables, subjects and stimuli, stimuli and responses). The analysis of such data often seeks understanding of the structure of these sets of objects and their inter-relationships. For example, profile data of a stimulus by scale type are often formed when one evaluates or rates a number of different stimulus objects on various attribute scales, e.g., a number of different soft drinks are rated on various prespecified scales relating to taste, level of carbonation, etc. If these ratings are interpreted as judgments of closeness to an ideal exemplar of the attribute being rated, the resulting matrix can be viewed as a rectangular proximity matrix.

Another typical example of such rectangular data occurs where one collects subject × stimulus data. For instance, various subjects render preference judgments over a number of different stimuli (e.g., subject preference ratings for various brands of telephones).

G. De Soete is "Aspirant" of the Belgian "Nationaal Fonds voor Wetenschappelijk Onderzoek" at the University of Ghent, Belgium. W. S. DeSarbo and J. D. Carroll are Members of Technical Staff at AT&T Bell Laboratories, Murray Hill, N.J. G. W. Furnas is Member of Technical Staff at Bell Communications Research, Murray Hill, N.J.

Requests for reprints should be sent either to G. De Soete, Department of Psychology, University of Ghent, Henri Dunantlaan 2, B-9000 Ghent, Belgium; or to W. S. DeSarbo, AT&T Bell Laboratories, Room 2C-256, 600 Mountain Avenue, Murray Hill, N.J. 07974.

0033-3123/84/0900-6027$00.75/0

Still another typical type of nonsymmetric data encountered in the behavioral sciences are confusions data where the rows and columns correspond to the same objects (thus $n = m$), yet, δ_{ij} is not necessarily equal to δ_{ji}. For example, the row elements might be various messages sent and the column elements the various messages received. The main diagonal elements would record the accuracy of the communication process, while the off-diagonal elements would reflect the error of the process. The famous Miller-Nicely (1955) data on confusions of phonemes in a variety of noise conditions provide an excellent example of such proximities. Also, one could consider brand-switching matrices such as those presented in DeSarbo (1982) where probabilities of switching from brand/product i in period t to j in period $t + 1$ represent the δ_{ij} entries in $\mathbf{\Delta}$. Here the δ_{ii} entries represent brand loyalty and the δ_{ij} ($i \neq j$) entries reflect the degree of brand switching. Similar types of matrices are found in the sociological literature in the form of social interaction and mobility matrices.

In summary, there are a number of different types of rectangular data found across many diverse disciplines. We propose here a new methodology for fitting ultrametric and additive tree structures to such data. We begin with a review of the related literature in two-way, two-mode clustering. The new proposed model and algorithm are then described in detail. The results of two detailed Monte Carlo studies are then presented. Two applications, one fitting an ultrametric tree to confusions data, and the other fitting an additive tree to word associations data, are discussed. Finally, future research opportunities in this area are described.

Previous Work in Two-Mode Clustering

There exist a number of quite different approaches to clustering both row and column elements in a two-way, two mode rectangular data matrix. Tryon and Bailey (1970) describe a heuristic for examining both variables and individuals in a clustering setting. Here, a "V-analysis", or "clustering" of the variables (columns), is performed initially via a factor analysis on the variable by variable correlation matrix or on a subset of the most collinear variables. Then, an "O-analysis" of the individuals is performed where subjects are initially scored on the several clusters/factors obtained in the "V-analysis" and a clustering of the subjects is then performed. One finally obtains a reduced "space" for variables and one for subjects based on their scores from the "V-analysis". The user can then examine profile scores of the various clusters. Thus, the Tryon and Bailey (1970) procedure does not really provide a *joint* clustering of subjects and variables, but rather a clustering of subjects whose resultant clusters vary continuously on a set of factor scores.

Hartigan (1975, 1976) develops a block clustering method for categorical rectangular data. Each block is defined by a cluster of cases and a cluster of variables such that each variable in the block is constant over the cases in the block, except for cases that also belong to other blocks (blocks may overlap). The constant value taken by a variable in a block is called the "modal value" for that block. The goal of the analysis is to provide a succinct representation of the data by a few large blocks with corresponding block modal value, together with residual single blocks consisting of single values deviating from the appropriate block modal value. An iterative algorithm in Hartigan (1975, 1976) is utilized to cluster both objects and variables simultaneously.

McCormick, Schweitzer, and White (1972) propose their "bond energy algorithm" to provide a joint clustering of row and column elements in rectangular data. This is accomplished by permuting the rows and columns of an input data array in such a way as to push the numerically larger array elements together (or as the authors mention, to maximize the summed bond energy over all row and column permutations in the input array—the "beta measure of effectiveness".) An iterative sequential-selection heuristic is utilized to provide an at least locally optimum solution.

DeSarbo (1982) has developed the GENNCLUS methodology for the simultaneous clustering of both row and column elements. He generalizes the ADCLUS (Shepard and Arabie (1979)) model, representing interstimulus proximities as combinations of discrete and possibly overlapping properties, to the case of asymmetric proximities. The GENNCLUS procedure allows for the estimation of either overlapping or nonoverlapping clusters. It utilizes a series of gradient-based procedures and combinatorial optimization methods in an alternating least-squares framework. Sarle (personal communication, 1982) has recently developed a similar procedure for the GENNCLUS "dual-domain" case where separate clusters would be derived for both row and column objects, and be subsequently related to each other via an estimated weights matrix.

None of the above mentioned methods for jointly clustering row and column elements in rectangular data have dealt with fitting tree structures (ultrametric and/or additive trees) to such data. While such methods (Carroll, 1976; Carroll and Chang, 1973; Carroll and Pruzansky, 1975; Carroll and Pruzansky, 1980; Cunningham, 1974; De Soete, 1983; Hartigan, 1967; Sattath and Tversky, 1977) have been developed for fitting such structures to the one-mode symmetric proximities (see Carroll, Clark, and DeSarbo, 1984, for the three-way case), only Furnas (1980) has investigated the "tree-unfolding" problem of fitting a tree to general rectangular proximities, providing a joint representation of both row and column objects. (Cunningham's (1978) bidirectional trees can only accomodate asymmetric single mode data (row and column elements are the same) allowing for different path lengths for row and column objects.) He develops an "ultrametric inequality" condition for such rectangular data (he also considers additive trees) and proposes an agglomerative clustering method to fit an ultrametric tree.

We shall extend the work of Furnas (1980) to accommodate both ultrametric and additive trees by utilization of a mathematical programming approach based on a penalty function algorithm. The approach to be described here is aimed at explicitly optimizing a least-squares loss function, whereas Furnas' (1980) earlier approach was heuristic in nature and only approximately least-squares.

The Penalty Function Approach

Theory Underlying the "Tree Unfolding" Model

The underlying model and theory for unfolding via ultrametric and path length or additive trees was worked out by Furnas (1980). He derived necessary and sufficient conditions, and uniqueness properties for such representations and devised some data analytic methods. Those methods were only heuristics (agglomerative algorithms for Ultrametric trees; ad hoc methods for error-free Additive tree data). The purpose of the current work is to devise and evaluate explicit least squares methods for fitting these models, and give examples of their usefulness.

In its simplest form, the usual proximity analysis begins with a single set of objects and a square symmetric matrix of all the pairwise measurements between them. The goal is to find a single global structure (e.g., a spatial configuration or tree structure) representing all the objects and, as faithfully as possible, the proximities between them.

Unfolding analysis is a slight variant. It begins with *two* separate classes of objects (e.g., people and cars) and with the rectangular matrix of measurements only between pairs of opposite classes (e.g., how much each person likes each car). Its goal is still the same—to find a single global representation, placing in it both sorts of objects such that the distances between them represent, as faithfully as possible the original between-class data. As in the usual one-class analysis, the resulting structure represents the data with fewer, hopefully interpretable parameters. An useful byproduct of an unfolding analysis is that the global structure provides a representation for the implicit within-class proximities, in addition to the between-class relationships it explicitly tried to fit. Thus for example it becomes evident which cars are similar, by virtue of being liked by the same people, and which people are similar by virtue of liking the same cars.

Unfolding theory has been well elaborated for multidimensional models, but there is good evidence that the similarity structure of many familiar domains is better fit by tree structures (Sattah & Tversky, 1977; Pruzansky, Tversky, & Carroll, 1982). One would expect unfolding analyses in such domains to require tree representation as well (see for example Furnas, 1980).

As background to the work presented here, we review the necessary and sufficient conditions for, and the resulting uniqueness of, the two-mode tree representations. A brief summary is presented here (for the errorless case); further details may be found in Furnas (1980).

Ultrametric distances (i.e., distances associated with ultrametric trees) must obey the Ultrametric Inequality:

$$d_{ij} \leq \max\ (d_{ik}, d_{jk}). \tag{1}$$

In the case of rectangular distance matrices, however, it is not possible to test the ultrametric inequality since one of the three distances will be missing for every triple. Furnas showed that, for rectangular matrices with a distance measure defined only between items of different classes (A, with elements represented by letters early in the alphabet and Z represented by letters late in the alphabet) the following Two-Class Ultrametric condition is necessary and sufficient for representation as an ultrametric tree:

$$t_{az} \leq \max\ (t_{ax}, t_{bx}, t_{bz}). \tag{2}$$

When this inequality is satisfied, the representation is unique up to the internal structure of purely one-class subtrees. For example the three trees in Figure 1 are equivalent in their two-class structure. For example, the (vd) and (ze) subtrees are well determined, but the (abc) and (xy) subtrees are not. By default, such two-class trees will be shown in their least structured form—that is with no structure for the pure subtrees and the fewest possible number of extra nodes, as in the middle example above. Note that this indeterminacy is usually small, in that most subtrees have mixed membership, particularly once they get to be of reasonable size. In the example above, only four of the 45 distances in the final tree were not completely determined, and even those four were strictly bounded.

Necessary and sufficient conditions for unfolding path length or additive trees are much more complicated (Furnas (1980) presented a bounded deterministic algorithm for this) and will not be given here. It suffices instead to note that overall distances in a path length or additive tree can be decomposed into the sum of an ultrametric part and an additively decomposable part (see Carroll and Pruzansky, 1980) which can be represented by a "star" or "bush" tree (an additive tree having only one interior node). The same is true for rectangular submatrices of distances from a path length or additive tree, and the analytic techniques proposed here make use of this decomposition.

In path length or additive trees, the unfolded representations have two sorts of nonuniqueness. One is exactly equivalent to the ultrametric case—purely one-class subtrees

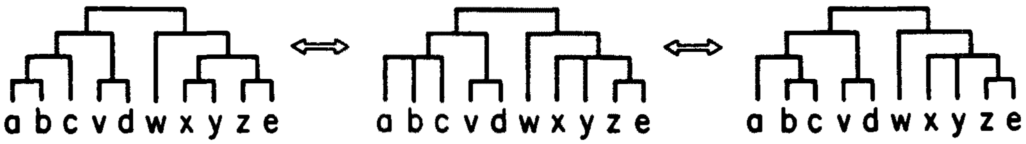

FIGURE 1
Indeterminacies in Two-Class Trees: Internal Structure of One-Class Clusters.

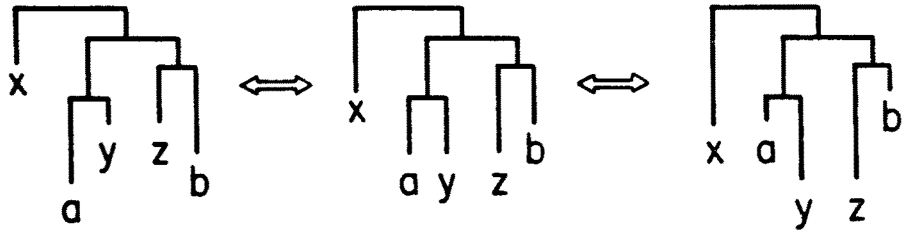

FIGURE 2
Indeterminacy in Two-Class Additive Trees: Adding a Constant to Terminal Branches of one class and subtracting it from the other.

have indeterminable internal structure. The second is a single continuous free parameter, shifting relative lengths of the terminal branches of the two classes. By adding a constant to all terminal branches of one class and subtracting it from the terminal branches of the other class, all between-class distances have a net change of zero. Thus, for example, the three trees displayed in Figure 2 are compatible with the same two-class matrix. A convention is used in this paper that attempts to balance the length of the terminal branches of the two classes in a manner to be described later.

These indeterminacies require a degree of caution so as not to "over-interpret" the tree representations resulting from an unfolding analysis, just as the orientation of axes cannot be determined in a two-way multidimensional scaling analysis.

One important point to note is that the approach discussed in this paper makes a very strong assumption about the data (Δ)—namely that of comparability of the data from row to row (and/or column to column) of the data matrix. In the terminology coined by Coombs (1964), the matrix is regarded as an *unconditional* (rather than a row or column conditional) proximity matrix.

The Algorithm

For Estimating an Ultrametric Tree. The algorithm consists of the following phases:

a. Transform the data matrix **Δ** into a matrix **T** best approximating **Δ** in a least-squares sense where **T** satisfies the two class ultrametric inequality:

$$t_{i\ell} \leq \max\,(t_{ik}, t_{jk}, t_{j\ell}), \tag{3}$$

for $i, j = 1 \ldots n \; (i \neq j)$ and $k, \ell = 1 \ldots m \; (k \neq \ell)$. An alternative statement of the two-class ultrametric inequality which can be shown to be equivalent to expression (3) is that for every quadruple of points comprised of two from each class, the two largest of the four defined distances must be equal, i.e., given $t_{i\ell}$, t_{ik}, $t_{j\ell}$, and t_{jk} (the only four distances among i and j in class one and k and ℓ in class two defined by the rectangular proximity data), the two largest of those four distances must be equal. The problem can be reformulated as that of solving the optimization problem:

$$\min \left\{ L(\mathbf{T}) = \sum_{i=1}^{n} \sum_{k=1}^{m} (\delta_{ik} - t_{ik})^2 \right\}, \tag{4}$$

subject to the condition that **T** satisfies the two-class ultrametric inequality. To do this, an exterior penalty function approach (Rao, 1979) is utilized to convert the constrained problem into a series of unconstrained ones. The augmented function:

$$\Phi(\mathbf{T}, \rho) = L(\mathbf{T}) + \rho P(\mathbf{T}), \tag{5}$$

with $\rho > 0$, is minimized for an increasing sequence of values of ρ, where the penalty part of expression (5) $P(\mathbf{T})$, is defined as

$$P(\mathbf{T}) = \sum_{i=2}^{n} \sum_{j=1}^{i-1} \sum_{k=2}^{m} \sum_{\ell=1}^{k-1} (u_{ijk\ell} - v_{ijk\ell})^2, \tag{6}$$

where:

$$u_{ijk\ell} = \max\,(t_{i\ell}, t_{ik}, t_{jk}, t_{j\ell}), \tag{7}$$

and

$$v_{ijk\ell} = \begin{cases} \max\,(t_{i\ell}, t_{jk}, t_{j\ell}) & \text{if } u_{ijk\ell} = t_{ik} \\ \max\,(t_{ik}, t_{jk}, t_{j\ell}) & \text{if } u_{ijk\ell} = t_{i\ell} \\ \max\,(t_{i\ell}, t_{ik}, t_{j\ell}) & \text{if } u_{ijk\ell} = t_{jk} \\ \max\,(t_{i\ell}, t_{ik}, t_{jk}) & \text{if } u_{ijk\ell} = t_{j\ell} \end{cases} \tag{8}$$

The specific steps of the penalty function algorithm are summarized in Appendix I.

b. Construct a square $(n + m)$ by $(n + m)$ matrix $\mathbf{D}$ which satisfies the ordinary one-class ultrametric inequality. $\mathbf{D} = ((d_{ab}))$, for a, $b = 1 \ldots n + m$ is symmetric $(d_{ab} = d_{ba})$ and is defined for $a \neq b$. Because of symmetry, we need only define d_{ab} for $a > b$. $\mathbf{D}$ can be thought of as the matrix having $\mathbf{T}$ as the $n \times m$ submatrix consisting of the last n rows and the first m columns. The problem is to fill in the (lower half of the symmetric) $m \times m$ and $n \times n$ submatrices comprising the first m rows and columns and the last n rows and columns respectively. This is accomplished by use of the following equations (Furnas, 1980):

$$d_{ab} = \begin{cases} t_{(a-m)b} & \text{if } m + 1 \leq a \leq m + n \\ & \text{and } 1 \leq b \leq m \\ \min\limits_{i=1 \ldots n}\,[\max\,(t_{ia}, t_{ib})] & \text{if } 1 \leq a \leq m \\ & \text{and } 1 \leq b \leq m \\ \min\limits_{k=1 \ldots m}\,[\max\,(t_{(a-m)k}, t_{(b-m)k})] & \text{if } m + 1 \leq a \leq m + n \text{ and} \\ & m + 1 \leq b \leq m + n. \end{cases} \tag{9}$$

If necessary, a positive constant is added to the d_{ab} so that they satisfy the triangle inequality.

c. Using standard hierarchical clustering methods (see Johnson, 1967), the ultrametric tree representation of both row and column elements is obtained from $\mathbf{D}$.

For Estimating a Path Length or Additive Tree. As discussed in Carroll (1976), Carroll and Pruzansky (1980), Carroll, Clark, and DeSarbo (1984), based on the work of Farris (1972) and Hartigan (1975), given an ultrametric tree, it is possible to convert it into an additive tree by adding a trivial "star" or "bush" tree (i.e., an additive tree having only one interior node) to it. The algorithm here is thus based on the fact that any set of path length tree distances can be decomposed into a set of ultrametric distances plus a set of additive constants for each of the row and column elements. The numerical problem can be stated as:

$$\text{Min}\left\{L(\mathbf{T}, r_i, c_k) = \sum_{i=1}^{n} \sum_{k=1}^{m} (\delta_{ik} - t_{ik} - r_i - c_k)^2\right\}, \tag{10}$$

subject to the condition that $\mathbf{T}$ satisfies the two-class ultrametric inequality. Once $\mathbf{T}$ is estimated via the algorithm in Appendix I, the additive constants are estimated in closed form via:

$$\begin{aligned}\hat{r}_i &= \frac{\sum_{k=1}^{m} (\delta_{ik} - t_{ik})}{m} - \frac{1}{2}\,\frac{\sum_{j=1}^{n} \sum_{k=1}^{m} (\delta_{jk} - t_{jk})}{nm} \\ \hat{c}_k &= \frac{\sum_{i=1}^{n} (\delta_{ik} - t_{ik})}{n} - \frac{1}{2}\,\frac{\sum_{i=1}^{n} \sum_{\ell=1}^{m} (\delta_{i\ell} - t_{i\ell})}{nm},\end{aligned} \tag{11}$$

as generalized from Carroll and Pruzansky (1980). Note that this computation of the $\hat{r}_i$ and $\hat{c}_k$ gives them equal means. This is the exact convention used for fixing the indeterminate constant in the relative lengths of the terminal branches of the row and column objects. Once the $\hat{r}_i$ and $\hat{c}_k$ constants are estimated, the algorithm cycles back to the ultrametric tree estimation phase in estimating **T** given $\hat{r}_i$ and $\hat{c}_k$. This alternating least-squares procedure (Wold, 1966) continues cycling back and forth over these two major phases until convergence in the loss function and/or t_{ik} values is reached. Once the final estimates of **T** and the r_i and c_k are obtained, **D** is reconstructed from **T** using the method in expression (9), and the appropriate additive constants are added. The additive tree is recovered from this tree (see Dobson, 1974). This can be done by simply converting the ultrametric tree into an additive tree, by defining the length of every branch to be the difference in height values of the two nodes connected by that branch (thus defining the heights of terminal nodes to be zero), and then adding the constants $\hat{r}_i$ and $\hat{c}_k$ to the lengths of the "leaves" of the tree (the branches connecting the terminal nodes to the first nonterminal or internal node). Because of the indeterminacy mentioned earlier, however, an additive constant can be added to the leaves corresponding to the elements of one class and that same constant subtracted from those of the other set. Finally, if desired, the root node of the ultrametric tree can be removed since that node is redundant for a path length or additive tree, where the two branches issuing from that root node (assuming the tree starts at that node), being replaced by a single branch whose length is the sum of the lengths of the two. Recall that a path length or additive tree is, in a fundamental sense, unrooted; also that the ultrametric tree corresponding to it is not unique—so that the root is highly arbitrary, and could in fact be placed between any two nodes or even at any node of the additive tree. However, for most purposes, it is convenient to represent the path length tree as a hierarchical tree, implying a root. While the root is not unique, the present fitting procedure would tend to place it at the root of the "dominant" or best fitting ultrametric tree, which seems intuitively a reasonable choice. Therefore, in many cases, it may be desirable to retain the root node and the lengths of its two branches, despite the fact that it is not needed.

Monte Carlo Analysis

In order to examine the performance of the algorithm in a systematic manner, a Monte Carlo analysis was designed for each type of tree fitting method: ultrametric and additive trees.

Ultrametric Tree Estimation

Table 1 presents five factors and their various levels that were experimentally combined to examine the performance of the penalty function procedure. We wished to examine what impact starting values, number of row elements, number of column elements, error levels introduced in the data, and shape of the underlying tree structure had in

TABLE 1

Factor Definition for Ultrametric Tree Monte Carlo

	Factor	Levels	
A.	Starting Values	Random Start	(0)
		Δ	(1)
		Δ + error	(2)
B.	No. Row Elements (n)	$n=10$	(0)
		$n=15$	(1)
		$n=20$	(2)
C.	No. Column Elements (m)	$m=.6n$	(0)
		$m=.8n$	(1)
		$m=n$	(2)
D.	Error in $\underline{\Delta}$	$\sigma^2=.25$	(0)
		$\sigma^2=0$	(1)
		$\sigma^2=.50$	(2)
E.	Shape of Tree*	$p=.1$	(0)
		$p=.3$	(1)
		$p=.5$	(2)

* see Appendix II

estimating ultrametric tree structures. A fractional factorial experimental design (see Addelman, 1962) was utilized for main effects estimation which appears in Table 2. Four dependent measures were defined as indicators of the algorithm performance: (1) metric recovery—the variance accounted for by **T** in the true underlying distances; (2) variance accounted for by **T** in the actual (error perturbed) data; (3) the C.P.U. time measured in seconds on a CRAY-I computer; and (4) the number of total function and gradient evaluations. The actual measurements for the 25 trials for these four variables appear in Table 3. Each experimental trial (row in the design) defined a specific combination of levels of factors to be combined to generate data for our methodology to fit. The first experimental factor defined how $\mathbf{T}^{(0)}$, the initial starting values for the parameters to be estimated via the penalty function, was specified. The second and third factors set n and m respectively. The fourth factor defined the amount of error added to the true distances which were initially normalized to unit variance, and the error levels define the variance (σ^2) in the $N(0, \sigma^2)$ error that was added. This affected the actual input data (Δ) being fit in the loss function (this may or may not affect the respective starting values). The last factor defined the shape of the true underlying ultrametric tree. As p approaches .5, the more symmetric in shape the resulting tree becomes. The splitting algorithm employed in generating the various shapes is described in Appendix II. Figure 3 depicts some shapes of random trees for the two most extreme levels specified. Note, these are the random shapes only.

The results of the experiment were analyzed via multiple regression methods where the experimental design was converted to dummy independent variables (k levels coded

TABLE 2

3^5 Fractional Factorial Design for Ultrametric Tree Monte Carlo Analysis

	FACTOR				
TRIAL	A	B	C	D	E
1	0	0	0	0	0
2	0	1	1	2	2
3	0	2	2	0	1
4	0	2	2	1	0
5	0	0	0	2	2
6	1	0	1	1	1
7	1	1	2	2	0
8	1	2	2	0	2
9	1	2	0	2	0
10	1	0	0	0	2
11	2	0	2	2	2
12	2	1	2	0	0
13	2	2	0	1	2
14	2	2	0	2	1
15	2	0	1	0	0
16	2	0	2	2	2
17	2	1	0	0	1
18	2	2	0	2	0
19	2	2	1	0	2
20	2	0	2	1	0
21	0	0	0	0	0
22	0	1	0	1	2
23	0	2	1	2	0
24	0	2	2	0	2
25	0	0	2	2	1

into $k - 1$ dummy variables). Since the first two dependent measures are basically proportions of variance accounted for ranging from 0 to 1, logit and arc sine transformations (see Snedecor and Cochran, 1981; DeSarbo, 1982 for a similar example) were also performed and rendered similar results to ordinary regression (and will thus not be reported). Table 4 presents the results for each of the four dependent measures. The coefficients displayed next to each of the coded factor levels (coded 1) represents the regression coefficient for that respective level. The intercept term contains the aggregated effects of all factors coded "0". All regressions were significant at $\alpha \leq 0.01$. The error conditions were the only significant variables concerning metric recovery. Specifically, the expected result observed here was that better recovery was obtained with no error than with the highest error level ($\sigma^2 = .5$).

This same result was demonstrated with the second dependent variable (as to be expected again), but here the middle level ($n = 15$) of row elements is significant, indicating that larger numbers of row elements appear to distract from this variance-accounted-for measure. While not significant, the $n = 20$ and higher column elements also have

TABLE 3

Dependent Variable Measurements for Ultrametric Tree Monte Carlo

Trial	Metric Recovery	V.A.F.	C.P.U. Time	No. of Function Evaluation
1	.905	.864	1.09	138
2	.816	.707	30.91	412
3	.976	.832	153.45	400
4	1.000	1.000	49.43	127
5	.816	.798	1.12	141
6	1.000	1.000	0.72	46
7	.842	.712	68.20	581
8	.975	.829	143.87	374
9	.876	.771	67.33	503
10	.935	.891	1.04	130
11	.892	.774	5.03	225
12	.972	.804	27.55	233
13	1.000	1.000	.76	1
14	.893	.733	50.04	373
15	.961	.857	2.30	162
16	.890	.686	6.68	298
17	.937	.883	12.75	248
18	.851	.737	65.97	436
19	.961	.825	97.42	400
20	1.000	1.000	.16	1
21	.893	.875	1.02	128
22	1.000	1.000	4.09	76
23	.914	.740	176.16	730
24	.959	.838	189.99	495
25	.788	.789	5.00	224

negative regression coefficients suggesting that larger rectangular arrays tend to decrease variance accounted for between T and the data.

Concerning C.P.U. time, the Δ+error start appeared to give starting estimates leading to quicker convergence. As expected, larger rectangular arrays (large n and m) lead to larger computation time as indicated by the significant coefficient for the medium and high levels of n and m. In addition, convergence was quicker when no error was added to the true distances ($\sigma^2 = 0$), i.e., when the data perfectly satisfied the two class ultrametric inequality.

A similar pattern of results was found for the number of function and gradient evaluations. While no starting method was significantly superior to others, larger n and m levels did significantly increase such evaluations. Similarly, specifying $\sigma^2 = 0$ significantly decreased function and gradient evaluations, while $\sigma^2 = .5$ increased them.

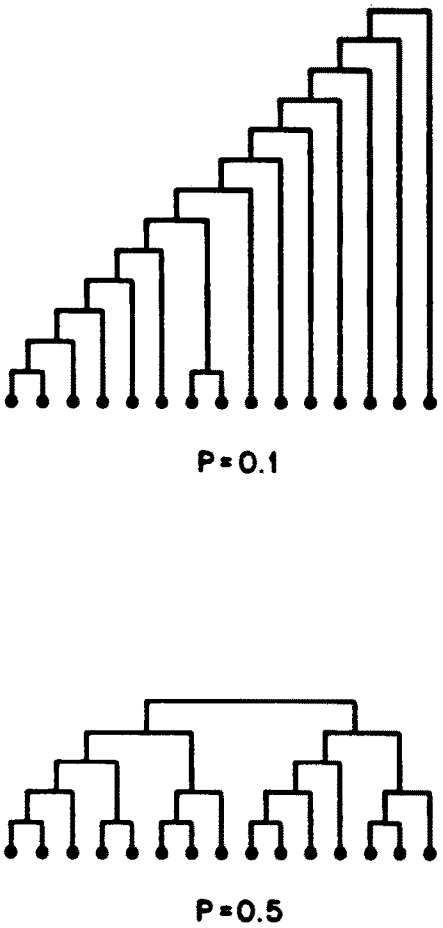

FIGURE 3
Examples of Trees Generated with Different Shape Parameters.

We noticed that in all four regression analyses, there was *no* effect due to the shape of the true underlying tree generating Δ!

Additive Tree Estimation

Based on the experience obtained with the previous Monte Carlo analysis, an abbreviated Monte Carlo analysis was designed for additive tree estimation. Table 5 presents six factors and their corresponding levels that were experimentally combined in this testing phase. The first five factors (A–E) are basically the same as those described in the previous Monte Carlo analysis for ultrametric trees, although some factors possess fewer levels. The sixth factor examined was designed to measure the impact of estimating the ultrametric tree portion first and the constants last or vice versa.

Table 6 presents the fractional factorial design specifying sixteen trials of various different factor level combinations in order to estimate main effects.

Table 7 presents the measurements for five dependent variables for each of the sixteen trials. Note that the first four are defined the same as the four utilized in the ultrametric tree Monte Carlo analysis. The fifth dependent measure records the total number of giant iterations or alternating least-square cycles required for convergence.

As before, multiple regression techniques were used to analyze the results of the

TABLE 4

Multiple Regression Results for Ultrametric Tree Monte Carlo Analysis

Factor/Level	Metric Recovery	V.A.F.	C.P.U. Time	No. of Evaluations
A. Data	.019	-.004	-4.994	39.700
A. Data+error	.029	-.019	-34.360*	-49.400
B. n=15	.005	-.042*	26.284	160.700
B. n=20	.033	-.023	97.026*	234.600*
C. m=.8n	.020	-.024	40.981*	132.600*
C. m=n	.019	-.024	44.415*	78.400
D. σ^2=0	.053*	.155*	-52.016*	-220.600*
D. σ^2=.5	-.090*	-.100*	-15.404	121.500*
E. p=.3	-.003	.001	-1.529	-45.700
E. p=.5	.003	-.001	2.170	-48.700
intercept	.906	.886	7.199	127.380
S.E.	.028	.022	28.563	75.344
R^2	.892	.971	.869	.908
adj R^2	.815	.951	.776	.842
F	11.553*	47.340*	9.304*	13.763*

* significant at $\alpha < .01$

Monte Carlo analysis where the experimental design in Table 6 was converted to dummy variables and each of the five dependent measures were regressed on this resulting design matrix. Again, since the first two dependent measures are basically proportions of variance accounted for, logit and arc sine transformations of these variables were also enacted and regressed on the design matrix. As before, the results were basically the same as obtained with ordinary multiple regression.

Table 8 presents the multiple regression results for each of the five dependent measures. Concerning the first dependent variable, metric recovery, we see that the two error levels significantly detract from the ability to recover the true underlying tree—similar to what was observed in the previous Monte Carlo analysis and to be expected here. The same pattern is observed for the second dependent variable where the two error levels also significantly detract from the fit between the solution and the data.

None of the other regressions are significant. For the C.P.U. dependent measure, large n and m tend to increase computing time required for convergence, a resulted noted before in the previous Monte Carlo analysis and one to be expected. However, for the number of function and gradient evaluations and number of giant iterations, *no* variable was significant at $\alpha \leq .05$. Note again, that the shape of the tree did not significantly affect *any* dependent measure as found before in the previous analysis.

Caveats

The two Monte Carlo analyses presented represent a *preliminary* examination of the performance of our methodology. A more thorough investigation might include ad-

TABLE 5

Factor Definition for Monte Carlo Analysis for Additive Tree

	Factor	Levels	
A.	Starting Values	Random	(0)
		Data	(1)
B.	# Row Objects (n)	8	(0)
		16	(1)
C.	# Col. Objects (m)	$.5n$	(0)
		$.75n$	(1)
D.	Error in Data	None ($\sigma^2=0$)	(0)
		$\sigma^2=.25$	(1)
		$\sigma^2=.50$	(2)
E.	Shape of Tree	$p=.1$	(0)
		$p=.5$	(1)
F.	Order of Estimation	Ultrametric 1st	(0)
		Constants 1st	(1)

TABLE 6

$2^5 3^1$ Fractional Factorial Design for Additive Tree Monte Carlo Analysis

	FACTOR					
TRIAL	A	B	C	D	E	F
1	0	0	0	0	0	0
2	1	0	1	0	1	1
3	1	1	0	0	1	1
4	0	1	1	0	0	0
5	0	0	1	1	1	0
6	1	0	0	1	0	1
7	1	1	1	1	0	1
8	0	1	0	1	1	0
9	0	1	0	2	1	1
10	1	1	1	2	0	0
11	1	0	0	2	0	0
12	0	0	1	2	1	1
13	0	1	1	1	0	1
14	1	1	0	1	1	0
15	1	0	1	1	1	0
16	0	0	0	1	0	1

TABLE 7

Dependent Measures for Additive Tree Monte Carlo Analysis

Trial	Metric Recovery	V.A.F.	C.P.U. Time	No. of Function Evaluations	No. of Giant Iterations
1	.999	.999	3.11	1368	45
2	1.000	1.000	2.10	275	10
3	1.000	1.000	13.02	248	6
4	1.000	1.000	80.05	840	9
5	.905	.907	8.32	1695	17
6	.844	.949	1.42	630	9
7	.947	.885	213.65	2427	9
8	.930	.881	122.32	3680	18
9	.797	.810	69.43	1804	7
10	.876	.774	205.91	2318	9
11	.717	.941	1.66	751	12
12	.758	.802	7.63	1545	11
13	.911	.872	190.06	2141	7
14	.862	.823	269.25	7287	35
15	.824	.906	7.33	1488	12
16	.827	.961	1.67	760	8

TABLE 8

Multiple Regression Results for the Additive Tree Monte Carlo Analysis

Factor/Level	Metric Recovery	V.A.F.	C.P.U. Time	No. of Function Evaluations	No. of Giant Iterations
A. Data Start	-.007	.006	28.969	198.875	2.500
B. n=16	.056	-.053	141.306*	1529.125	-3.000
C. m=.75n	.030	-.027	29.146	-474.875	-7.000
D. σ^2=.25	-.119*	-.102*	77.183	1830.750	-3.125
D. σ^2=.50	-.213*	-.168*	46.588	921.750	-7.750
E. p=.5	-.005	-.032	-24.766	848.375	1.000
F. constants 1st	-.003	.006	-24.871	-1199.625	-11.250
intercept	.965	1.049	-50.322	231.813	28.875
S.E.	.040	.039	57.203	1401.747	10.419
R^2	.893	.865	.804	.643	.506
adj R^2	.799	.747	.633	.330	.074
F	9.516*	7.317*	4.694	2.054	1.171

* significant at $\alpha \leqslant .05$

ditional factors (e.g., varying various penalty function parameters such as ρ), additional levels for various factors (especially concerning error levels), allowing for the estimation of various interaction effects (e.g., through use of a full factorial design), replicating the design, etc. While the results of the two Monte Carlo analyses appear to be quite favorable, definitive conclusions concerning the robustness of the methodology cannot be drawn until more complete investigations are completed.

Applications

Ultrametric Tree—Miller and Nicely Data

Miller and Nicely (1955) collected data on confusions among 16 English consonants under each of several conditions of noise (varying the signal-to-noise ratio), low-pass filtering (filtering out acoustical energy in the higher frequencies), and high-pass filtering (filtering out acoustical energy in the lower frequencies). The subjects listened to speakers read c-v syllables (each syllable consisted of one of the 16 constants followed by the vowel a as in father), and tried to identify the consonant they heard after each syllable was spoken. In each of the 17 experimental conditions the speech was acoustically degraded in a different manner. A matrix of frequencies of stimulus-response confusions was derived for the data in each experimental condition. All 17 nonsymmetric matrices are presented in the original Miller and Nicely paper.

We chose to sum the confusions data for the six noise conditions and fit an ultrametric tree to the resulting 16 × 16 aggregate asymmetric data (this matrix was converted to dissimilarities by subtracting it from a large constant). Figure 4 presents the ultrametric tree derived from the algorithm. It accounts for 99.19% of the variance in the aggregated data and took 35.9 seconds of C.P.U. time on the CRAY-I machine (it converged in 6 major iterations taking 230 function and gradient evaluations).

Starting from the top of the tree, the first split divides the consonants into voiceless vs. voiced ones. Within the voiceless consonants on the left side, the next split separates long fricatives, short fricatives, and stops. Within the voiced consonants, on the right side, the next split separates consonants pronounced in the back of the mouth, those pronounced in the front of the mouth, and nasals. Within those pronounced in the back of the mouth, the next split separates stops from fricatives as with the split within those pronounced in the front of the mouth.

Note that the respective row and column elements corresponding to the same consonant are grouped together at the first interior nodes of the tree indicating a strong main diagonal component in the confusions data (indicating a strong tendency to identify correctly those consonants given as stimuli in the identification task leading to large main diagonal entries as compared to the off-diagonal elements) in the data and not too much asymmetry. The relative height of the level at which each of the two corresponding row and column elements join gives an indication of the confusability of that consonant with itself over the 6 noise conditions, or an inverse indication of the discriminability of that consonant. (Note that while small main diagonal elements (for the case of dissimilarities) tend to suggest that the same row and column objects be joined together at relatively short height levels of the corresponding tree, the loss function is defined over the *entire* matrix weighing each entry equally.) It might be noted that if we identify the row and column elements corresponding to the same consonant completely (or, take the "penultimate" tree whose terminal nodes are the internal nodes to which the leaves or terminal branches attach), the tree is topologically equivalent to the one Shepard (1972) obtained by applying a hierarchical clustering procedure to the Miller-Nicely data averaged over all 17 conditions, and symmetrized by averaging the (i, j) with the (j, i) data values. It

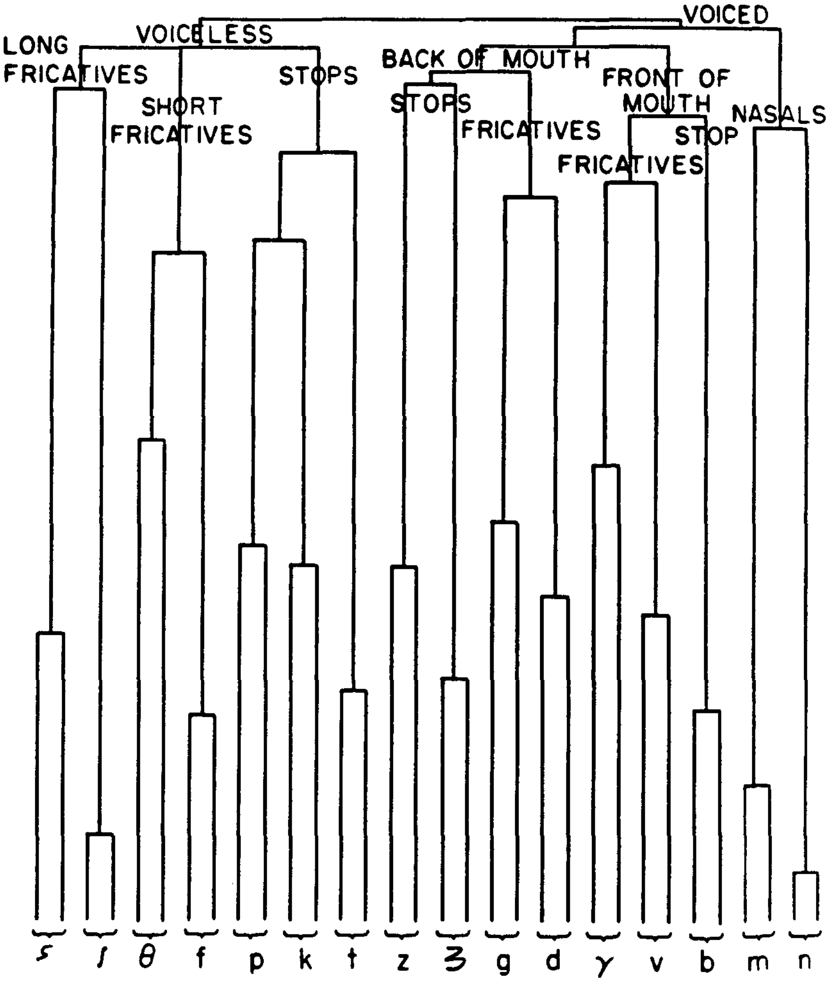

FIGURE 4
Ultrametric Tree Representation for the Miller-Nicely Data.

might prove interesting also to analyze this symmetric data via our tree-unfolding methodology and compare the resulting solution.

Additive Tree—Green and Tull Shampoo Benefit Data

Green and Tull (1978) present a rectangular array of free-association responses from a marketing study completed on shampoo benefits. The purpose of the free-association task was to examine what semantic associations were conjured up by various words or phrases that were initially thought to be related to the central benefit of "body". Table 9 presents the nonsymmetric subset of the data (frequencies) for the eight stimulus phrases utilized and the eleven associated words from $N = 84$ subjects. (Note that the original data contained these eleven associated words plus the eight stimulus phrases as column elements. Because main diagonal elements for the eight by eight (stimulus phrases) submatrix were not collected, we chose to simply analyze the eight by eleven matrix presented in Table 9.)

Figure 5 presents the additive tree fit to this rectangular array which accounts for

TABLE 9

Word Association Frequencies Involving Eight Stimulus Phrases Regarding Shampoos (sample size = 84)*

Stimulus Phrase	Clean	Sheen	Curly	Long	Grooming Aid	Soft	Nice	Combs Easily	Healthy	Alive	Pretty
1. Body	6	9	8	3	4	6	1	1	3	4	2
2. Fullness	7	7	7	10	1	1	5	1	5	2	4
3. Holds set	4	2	17	2	14	33	5	2	0	1	6
4. Bouncy	12	14	22	9	4	3	4	2	1	9	5
5. Not limp	15	5	12	3	5	2	4	1	8	1	1
6. Manageable	15	7	3	0	8	8	5	18	0	0	1
7. Zesty	16	26	11	8	6	1	2	2	7	10	1
8. Natural	26	27	5	11	2	8	5	3	4	1	7

*Taken from Green and Tull (1978)

84.2% of the variance taking 50.6 seconds of C.P.U. time on the CRAY-I (13 giant iterations and 2979 function and gradient evaluations).

Beginning at the top of the tree, the response phrases "nice" and "pretty" are grouped away from any stimulus. These are general descriptions of the cosmetic appearance of hair which are not tied to any *specific* row benefits/attribute (which are underlined in Figure 5). The next group of "manageable", "combs easily", "holds set", "soft", "grooming aid", and "body" refers mostly to hair control. The following group of phrases: "bouncy", "curly", and "alive" are related to the liveliness or activity of the hair. "Fullness" and "long" are grouped together reflecting hair density. "Not limp" and "healthy" are clustered together describing general hair quality or condition. Finally, "natural", "sheen", "zesty", and "clean" group together to describe adjectives related to overall appearance.

The tree also provides insight as to which specific adjectives are associated with the various stimulus phrases. For example, when "manageable" is stated, "combs easily" is evoked. "Natural" and "zesty" are most often associated with "sheen" and "clean". The analysis provides insight into appropriate advertising copy for the general marketing of shampoos in identifying various inter-related dimensions of hair shampoo attributes.

Discussion

We have presented a methodological description of the model and algorithm which provides an ultrametric or additive tree representation for row and column objects in a two-way, two mode rectangular proximities matrix. Monte Carlo testing for each type of tree provided positive evidence concerning the performance of the algorithms employed. Finally, two examples were provided where the procedure rendered insight into the structure of associated rectangular data matrices.

There are a number of possible extensions for this procedure. One obvious extension is to redesign the algorithm(s) in order to estimate multiple tree structures as done in Carroll and Pruzansky (1980) and Carroll, Clark, and DeSarbo (1984). This proposed

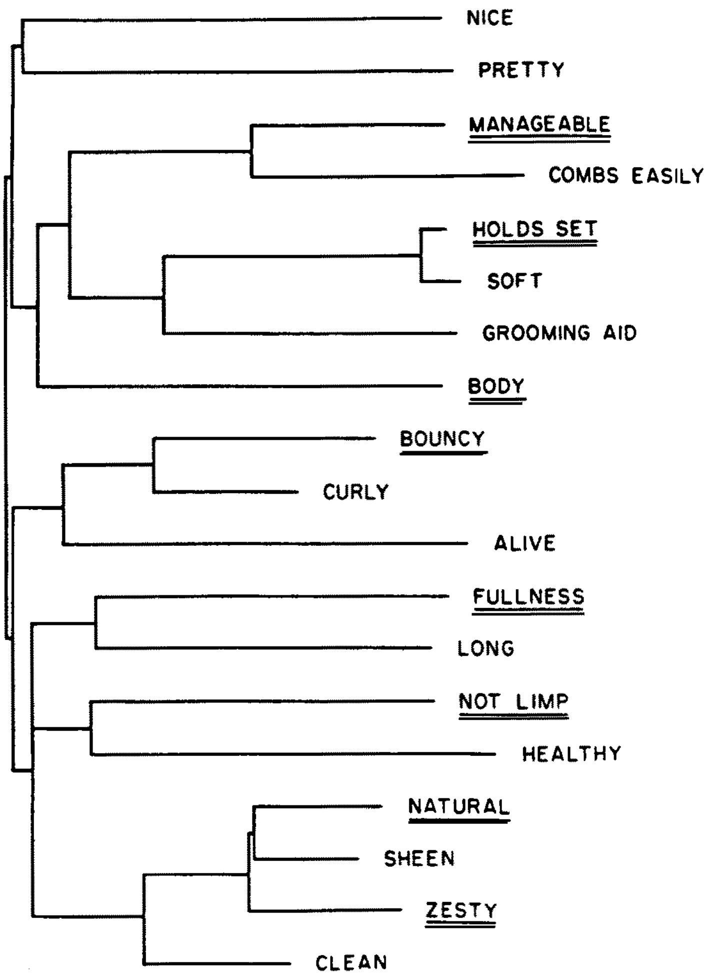

FIGURE 5
Additive Tree Representation for the Shampoo Word Association Data.

model can be formally expressed in the ultrametric case as:

$$\Delta \cong \mathbf{T}_1 + \mathbf{T}_2 + \cdots + \mathbf{T}_R, \tag{12}$$

where $\mathbf{T}_1$ through $\mathbf{T}_R$ arise from R different ultrametric tree structures. In the case of additive trees, expression (12) becomes:

$$\Delta \cong \mathbf{T}_1 + \mathbf{T}_2 + \cdots + \mathbf{T}_R + \mathbf{A}, \tag{13}$$

where $\mathbf{A} = ((a_{ij}))$ and $a_{ij} = r_i + c_j$ from expression (11).

Similarly, a procedure can be generalized to accommodate "hybrid models" where both discrete tree structures and continuous spatial dimensions are estimated. Here, for the ultrametric case, one can express this as:

$$\Delta \cong \mathbf{T}_1 + \mathbf{T}_2 + \cdots + \mathbf{T}_R + \mathbf{H}, \tag{14}$$

where $\mathbf{H}$ is a proximity matrix generated from, say, an S-dimensional nonasymmetric multidimensional scaling procedure (e.g., Harshman, 1978). One merely has to add $\mathbf{A}$ from expression (13) to expression (14) to express the hybrid model for the additive tree case.

Another logical extension, and one that is currently being pursued by the authors, is to generalize this procedure to handle three-way rectangular proximities arrays where, as in INDTREES (Carroll, Clark, and DeSarbo (1984), a common tree topology is estimated across slices of the array, allowing for differential branch lengths or node heights.

Finally, the procedure must be tested more thoroughly on other types of *real* data sets. Currently, work is progressing on fitting tree structures to brand switching matrices to examine the nature of respective competitive market structures. Preliminary results indicate great success in fitting such tree structures and obtaining valuable insight into brand switching behavior within a particular (soft-drinks and automobiles) competitive market. It is also interesting to point out that our "tree-unfolding" methodology appears *not* to be susceptible to degeneracies often encountered with traditional multidimensional scaling unfolding. On the basis of analyses on several real data sets and the two Monte Carlo analyses reported, we have not encountered degenerate solutions (e.g., all row elements in one cluster and all column elements in another). Further testing is required, however, before definitive conclusions can be drawn.

Appendix I
Penalty Function Algorithm to Estimate T

1. Normalize $\mathbf{\Delta}$ such that

$$\sum_{i=1}^{n} \sum_{k=1}^{m} (\delta_{ik} - \bar{\delta})^2 = 1, \tag{A-1}$$

where $\bar{\delta}$ is the grand mean of the δ_{ik}'s.

2. Define $\mathbf{T}^{(0)}$ using one of the following three methods:

—$t_{ik}^{(0)}$ are uniformly distributed random numbers;

—$t_{ik}^{(0)} = \delta_{ik} + \varepsilon_{ik}$, where $\varepsilon_{ik} \sim N\left(0, \dfrac{1}{3mn}\right)$;

—$\mathbf{T}^{(0)} \equiv \mathbf{\Delta}$.

Initialize the iteration index $q = 1$ and let:

$$\rho^{(1)} = \begin{cases} L(\mathbf{T}^{(0)})/P(\mathbf{T}^{(0)}) & \text{if } L(\mathbf{T}^{(0)}) > 0 \text{ and if } P(\mathbf{T}^{(0)} > 0 \\ .0001 & \text{otherwise.} \end{cases} \tag{A-2}$$

3. Minimize $\Phi(\mathbf{T}^{(q)}, \rho^{(q)})$ starting from $\mathbf{T}^{(q-1)}$ to obtain $\mathbf{T}^{(q)}$. This is done using Powell's (1977) conjugate gradient method with automatic restarts where the partial derivatives of $\Phi(\mathbf{T}, \rho)$ are

$$\frac{\partial \Phi(\mathbf{T}, \rho)}{\partial t_{ik}} = \frac{\partial L(\mathbf{T})}{\partial t_{ik}} + \rho \frac{\partial P(\mathbf{T})}{\partial t_{ik}}, \tag{A-3}$$

where:

$$\frac{\partial L(\mathbf{T})}{\partial t_{ik}} = 2(t_{ik} - \delta_{ik})$$

$$\frac{\partial P(\mathbf{T})}{\partial t_{ab}} = 2\sum_{i=1}^{n} \sum_{j=1}^{i-1} \sum_{k=1}^{m} \sum_{\ell=1}^{k-1} (u_{ijk\ell} - v_{ijk\ell})(e_{ijk\ell}^{ab} - f_{ijk\ell}^{ab})$$

$$e_{ijk\ell}^{ab} = \begin{cases} 1 & \text{if } u_{ijk\ell} = t_{ab} \text{ and } a = i \text{ or } j, \text{ while } b = k \text{ or } \ell \\ 0 & \text{otherwise} \end{cases}$$

$$f_{ijk\ell}^{ab} = \begin{cases} 1 & \text{if } v_{ijk\ell} = t_{ab} \text{ and } a = i \text{ or } j, \text{ while } b = k \text{ or } \ell \\ 0 & \text{otherwise.} \end{cases}$$

4. Test for convergence:

   if $\left[\sum_{i=1}^{n}\sum_{k=1}^{m}(t_{ik}^{(q)} - t_{ik}^{(q-1)})^2\right]^{1/2}$ is less than some small constant, stop; otherwise, go to step 5.

5. Update ρ: $\rho^{(q+1)} = 10 \times \rho^{(q)}$. Let $q = q + 1$ and go to step 3.

Note, because of the indeterminacy of such trees with respect to an overall additive constant, a variance-accounted-for statistic is utilized as the appropriate overall goodness of fit measure between Δ and $\mathbf{T}$ (or $\mathbf{T}$ plus the path length constants). Note that since the variance-accounted-for measure has an upper bound of 1.0, and since each stage of estimation can be shown conditionally to improve this measure, one can use a "limiting sums" argument (Courant, 1965) to prove that the entire algorithm converges to at least a locally optimum solution.

Appendix II
Generation of Random Trees

Trees were created in two steps: the generation of random topological shapes and the assignment of branch lengths. We wanted to generate tree forms with a specifiable parameter of shape, so we used a method derived from successive hierarchical partitioning with a symmetry-of-partition parameter. Below is the algorithm used to generate a tree on n terminal nodes:

Let $N(s)$ be the number of objects in any set s. Let S be the set of all n objects. Let p be some probability $0 < p < 1$, which serves as the shape parameter.

The random tree is generated by performing SPLIT(S, p). Internal nodes are associated with sets and terminal nodes with the objects, as in hierarchical clusterings.

PROCEDURE SPLIT(set s, probability p).

1. Partition s into two subsets, s_0 and s_1; use probability p to assign each of the $N(s)$ objects independently and randomly to the two subsets. If one set is of size zero, redo this step.
2. Make a link from the internal node s to s_0 and to s_1.
3. If $N(s_0) > 1$, perform SPLIT(s_0).
   If $N(s_1) > 1$, perform SPLIT(s_1).
4. Return

Such a method is equivalent to sampling subtrees from an infinite binary tree. (See Furnas (1984) for the case of $p = .5$. A simple extension encorporates other values of p.)

The effect of the shape parameter is easily understood. If p is near .5 then the decedents of any given node will tend to be of nearly equal size, as dictated by the binomial distribution. If p is far from .5 (the situation is symmetric, so $p = .9$ is the same as $p = .1$), the descendants tend to be of different size. Several examples, two generated with $p = .5$ and two with $p = .1$, are presented in Figure 3.

These random tree shapes were turned into random Ultrametric and Additive trees by the assignment of appropriate lengths to the trees' branches. Additive trees were simplest, obtained by assigning uniform (0, 1) random branch lengths independently to all branches, with the exception that one of the two branches from the root node was set to zero length. This is because the root has no real significance in an additive tree and assigning a zero branch effectively removes it, by preventing it from influencing distances in the tree.

Ultrametric branch lengths must be assigned so that all terminal nodes are equidistant from the root. This constraint allows one to assign random heights to internal nodes of the tree and get branch lengths later by subtraction. Certain simple methods for assigning such node heights suffer from the problem that the average height of nodes in the left and right descendent subtrees of any given node will depend on the number of nodes in the subtree. In order to prevent this, node heights were assigned as follows. The set of node heights to be used was sampled ahead of time from the uniform distribution. At each branching, the set of heights available to descendents of a node was randomly but proportionately partitioned between the two descendants. The highest value in each of the subsets was assigned to the highest node in the respective subtrees. The assigned value was removed from the pool of heights available to its descendents, and the procedure was repeated recursively on each of the two subtrees.

REFERENCES

Addelman, S. (1962). Orthogonal main-effect plans for asymmetrical factorial experiments. *Technometrics, 4,* 21–46.

Carroll, J. D. (1976). Spatial, non-spatial and hybrid models for scaling. *Psychometrika, 41,* 439–463.

Carroll, J. D. and Chang, J. J. (1973). A method for fitting a class of hierarchical tree structure models to dissimilarities data, and its application to some body parts data of Miller's. In *Proceedings of the 81st Annual Convention of the American Psychological Association,* Vol. 8, 1097–1098.

Carroll, J. D., Clark, L. A., and DeSarbo, W. S. (1984). The representation of three-way proximities data by single and multiple tree structure models. *Journal of Classification,* (in press).

Carroll, J. D. and Pruzansky, S. (1975). Fitting of hierarchical tree structure (HTS) models, mixtures of HTS models, and hybrid models, via mathematical programming and alternating least squares. *Paper presented at the U.S.-Japan Seminar on Multidimensional Scaling,* University of California at San Diego, La Jolla, California.

Carroll, J. D. and Pruzansky, S. (1980). Discrete and hybrid scaling models. In E. D. Lantermann and H. Feger (Eds.), *Similarity and Choice.* Bern: Hans Huber.

Coombs, C. H. (1964). *A theory of data.* New York: Wiley.

Courant, R. (1965). *Differential and integral calculus* (2nd edition), Vol. 1. New York: Wiley.

Cunningham, J. P. (1974). Finding the optimal tree realization of a proximity matrix. *Paper presented at the Mathematical Psychology Meetings,* Ann Arbor Michigan.

Cunningham, J. P. (1978). Free trees and bidirectional trees as a representations of psychological distance. *Journal of Mathematical Psychology, 17,* 165–188.

DeSarbo, W. S. (1982). GENNCLUS: New models for general nonhierarchical clustering analysis. *Psychometrika, 47,* 449–475.

DeSoete, G. (1983). A least squares algorithm for fitting additive trees to proximity data. *Psychometrika, 48,* 621–626.

Dobson, A. G. (1974). Unrooted trees for numerical taxonomy. *Journal of Applied Probability, 11,* 32–42.

Farris, J. S. (1972). Estimating phylogenetic trees from distance matrices. *American Naturalist, 106,* 645–668.

Furnas, G. W. (1980). Objects and their features: The metric representation of two class data. *Unpublished Doctoral Dissertation,* Stanford University.

Furnas, G. W. (1984). The construction of random, terminally labeled, binary trees. *Journal of Classification,* (in press).

Green, P. E. and Tull, D. S. (1978). *Research for marketing decisions,* 4th ed., Englewood Cliffs, N.J.: Prentice-Hall.

Harshman, R. (1978). Models for analysis of asymmetrical relationships among *N* objects or stimuli. *Unpublished Paper,* University of Western Ontario, Canada.

Hartigan, J. A. (1975). *Clustering algorithms,* New York: Wiley.

Hartigan, J. A. (1976). Modal blocks in definition of west coast mammals. *Systematic Zoology, 25,* 149–160.

Hartigan, J. A. (1967). Representation of similarity matrices by trees. *Journal of the American Statistical Association, 62,* 1140–1158.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika, 32,* 241–254.

McCormick, W. T., Schweitzer, P. J., and White, T. W. (1972). Problem decomposition and data reorganization by a clustering technique, *Operations Research, 20,* 993–1009.

Miller, G. A., and Nicely, P. E. (1955). An analysis of perceptual confusions among some English consonants. *Journal of the Acoustical Society of America, 27,* 338–352.

Powell, M. J. D. (1977). Restart procedures for the conjugate gradient method. *Mathematical Programming, 12,* 241–254.

Pruzansky, S., Tversky, A., and Carroll, J. D. (1982). Spatial versus tree representations of proximity data. *Psychometrika, 47,* 3–24.

Rao, S. S. (1979). *Optimization theory and applications.* New York: Wiley.

Sattath, S. and Tversky, A. (1977). Additive similarity trees. *Psychometrika, 42,* 319–345.

Shepard, R. N. (1972). Psychological representation of speech sounds. In E. E. David and P. B. Denes (eds.), *Human communication: A unified view.* New York: McGraw Hill.

Shepard, R. N. and Arabie, P. (1979). Additive clustering: Representation of similarities as combination of discrete overlapping properties. *Psychological Review, 86,* 87–123.

Snedecor, G. W. and Cochran, W. G. (1981). *Statistical methods,* 7th edition, Ames, Iowa: Iowa State University Press.

Sonquist, J. A. (1971). *Multivariate model building: the validation of a search strategy.* Institute for Social Research, University of Michigan, Ann Arbor, Michigan.

Tryon, R. C. and Bailey, D. E. (1970). *Cluster analysis.* New York: McGraw Hill.

Wold, H., (1966). Estimation of principal components and related models by iterative least squares. In P. R. Krishnaiah (Ed.), *Multivariate analysis.* New York: Academic Press.

Manuscript received 10/18/83
Final version received 5/1/84