

The Representation of Three-Way Proximity Data by Single and Multiple Tree Structure Models

J. Douglas Carroll

Linda A. Clark

Wayne S. DeSarbo

AT&T Bell Laboratories

Abstract: Models for the representation of proximity data (similarities/dissimilarities) can be categorized into one of three groups of models: continuous spatial models, discrete nonspatial models, and hybrid models (which combine aspects of both spatial and discrete models). Multidimensional scaling models and associated methods, used for the *spatial* representation of such proximity data, have been devised to accommodate two, three, and higher-way arrays. At least one model/method for overlapping (but generally non-hierarchical) clustering called INDCLUS (Carroll and Arabie 1983) has been devised for the case of three-way arrays of proximity data. Tree-fitting methods, used for the *discrete* network representation of such proximity data, have only thus far been devised to handle two-way arrays. This paper develops a new methodology called INDTREES (for Individual Differences in TREE Structures) for fitting various (*discrete*) tree structures to three-way proximity data. This individual differences generalization is one in which different individuals, for example, are assumed to base their judgments on the same family of trees, but are allowed to have different node heights and/or branch lengths.

We initially present an introductory overview focussing on existing two-way models. The INDTREES model and algorithm are then described in detail. Monte Carlo results for the INDTREES fitting of four different three-way data sets are presented. In the application, a single ultrametric tree is fitted to three-way proximity data derived from intention-to-buy-data for various brands of over-the-counter pain relievers for relieving three common types of maladies. Finally, we briefly describe how the INDTREES procedure can be extended to accommodate hybrid modelling, as well as to handle other types of applications.

Keywords: Clustering; Alternating least squares; Discrete optimization

1. Introduction

Psychometricians have developed three major classes of models for the analysis of *two-way* proximity data (i.e., a single, usually square [one mode] and symmetric matrix of proximities between every pair of some set of objects or stimuli):

1.1 Spatial Models

Continuous spatial models of the kind usually associated with multidimensional scaling typically embed the objects or stimuli of interest (e.g., colors, words, products, people, etc.) in some coordinate space so that a specified distance measure (usually, but not always, Euclidean) between points represents the observed proximities among the respective objects. Thus, a metric space is assumed to underlie the proximity data. Distances in this space are postulated to be related in some simple way — e.g., linearly or monotonically — to the corresponding proximities. The problem here is, given the two-way proximity data, and specifying the general form of the metric, to construct an estimate of the underlying space.

1.2 Nonspatial Models

The most straightforward nonspatial model or method is simple (nonhierarchical and nonoverlapping) clustering, in which each object belongs to one and only one cluster or class. Such clusters, which comprise a partition of the object set, can be thought of as corresponding to a number of discrete attributes which are mutually exclusive (i.e., each object possesses one and only one of these attributes).

A widely used class of methods for the discrete representation of two-way similarity or dissimilarity data is hierarchical clustering (e.g., Johnson 1967; Hartigan 1967). A hierarchical clustering method yields a family of nested clusters (i.e., any two clusters are disjoint or one includes the other). In the usual representation, the objects being clustered appear as terminal nodes of a tree, while a metric is defined in which the distances between objects are the heights of their “lowest common ancestor” node (the first internal node at which they meet as one proceeds “up” the hierarchy).

The underlying model implies that, given two disjoint clusters, all distances between objects in the same cluster are smaller than distances between objects in two different clusters, and that these between-cluster distances are equal. This property is called the ultrametric inequality and the associated tree representation is called an ultrametric tree. More formally, the ultrametric inequality states that

$$d_{ik} \leq \max(d_{ij}, d_{jk}) \quad \text{for all } i, j, k . \quad (1)$$

This, of course, is a much stronger condition than the triangle inequality, which states that

$$d_{ik} \leq d_{ij} + d_{jk} \quad \text{for all } i, j, k . \quad (2)$$

Satisfaction or nonsatisfaction of the ultrametric inequality is invariant under strictly increasing monotonic functions of the distances, whereas the triangle inequality can always be satisfied, for example, by applying the trivial monotonic function that adds a “sufficiently large” constant to all distances (or violated by subtracting a sufficiently large constant). Given a set of distances satisfying the ultrametric inequality, the associated tree can easily be constructed (and height values defined). Given a hierarchical tree, an infinite family of ultrametrics is defined; if the height values of nonterminal or internal nodes are specified, the particular ultrametric is uniquely defined.

Hartigan (1967) was the first to suggest treating the problem of hierarchical clustering as one of fitting a certain geometric model — namely a rooted tree structure on which an ultrametric is defined — using combinatorial optimization techniques. Hartigan proposed an explicit algorithm aimed at optimizing a least-squares criterion of fit between input data “distances” and distances calculated from an ultrametric tree structure.

A few years later Carroll and Chang (1973) devised a procedure that generalized Hartigan’s approach in two different ways. While Hartigan’s procedure (and, indeed, essentially all other hierarchical clustering procedures) restricted the objects (or stimuli, to use psychological terms) to *terminal* nodes of the tree, the Carroll and Chang (1973) procedure allowed some or all of the interior, or nonterminal, nodes of the tree to correspond to objects or stimuli. Secondly, these authors allowed for a more flexible definition of the metric defined on the tree. In addition to the ultrametric, two other kinds of metrics were allowed. The first of these is a path length metric, in which lengths are associated with branches, or links in the tree, and distance is simply the length of the (unique) path joining those two nodes of the tree. The second is a mixed case, in which “heights” are associated with nonterminal nodes *and* lengths with branches, while distance is defined as a sum of the path length and height of the lowest common ancestor node. This last “mixed” metric can be meaningfully distinguished from the simpler path length metric *only* in the case in which some of the objects are at nonterminal nodes. In the more usual case in which all objects are at terminal nodes, the ultrametric and this “mixed” metric are special cases of the path length metric. When some objects are at interior nodes, the three models are all meaningfully different.

A tree with path length metric, or simply a path length tree is synonymous with the term "free tree." Farris (1972) calls this a tree with "four point metric," while Sattath and Tversky (1977) call it an "additive similarity tree." Unlike an ultrametric tree which has a natural "root" node (the bottommost or topmost node, depending on which way the tree is drawn), a path length tree has no root. It is not necessary to think of it as being vertically organized into a hierarchy.

Both Cunningham (1974) and Sattath and Tversky (1977) have devised algorithms for fitting free trees to data. Sattath and Tversky's method is a kind of natural generalization of the "pair group" method (e.g., single, average and complete linkage) often used to generate hierarchical clustering solutions. Cunningham's (1974) method attempts to approximate a least-squares solution.

Cunningham's solution rests on a certain four point condition (Bunemann 1974) that must be satisfied by path length distances. One way of stating this condition is:

if

$$d_{ij} + d_{kl} \geq d_{ik} + d_{jl} \geq d_{jk} + d_{il}$$

then

$$d_{ij} + d_{kl} = d_{ik} + d_{jl} . \quad (3)$$

That is, the two largest sums of pairs of distances involving the subscripts i, j, k, l must be equal. Cunningham's approach, in effect, assumes that the order of these sums of distances for the data will (at least weakly) match the order of the same sums in the optimal solution (a rather dubious assumption, unless the error variances are very small indeed).

Thus if δ_{ij} represents the data values (dissimilarities), Cunningham imposes the linear constraint on the distances (the d 's) that

$$d_{ij} + d_{kl} = d_{ik} + d_{jl}$$

if

$$\delta_{ij} + \delta_{kl} \geq \delta_{ik} + \delta_{jl} \geq \delta_{jk} + \delta_{il} . \quad (4)$$

One such linear constraint on the distances is defined for each tetrad of points. He then seeks the d 's yielding a best-least squares fit to the δ 's, but

subject to these linear constraints. This is a problem that has a straightforward analytic solution, (although it is possible for such anomalies as negative distances to occur).

Cunningham's procedure works very nicely for small data sets with relatively low rates of error. For larger data sets it quickly becomes unwieldy, since it is necessary to invert an $\binom{N}{2} \times \binom{N}{2}$ matrix. More critically, it begins to break down once the ratio of error variance to "true" variance exceeds about 10%. This problem occurs apparently because the constraints seemingly implied by the data are then not necessarily the constraints that hold for the optimal solution. In such cases, the solution Cunningham's method obtains is frequently a rather uninteresting tree of the form with all branch lengths equal (so that all distances are equal to the same constant). A tree of this form is sometimes called a "star" by graph theorists, or a "bush" by numerical taxonomists.

Because of these problems, Cunningham (1978) has devised an improved, more robust approach that avoids this "degeneracy" problem to some degree. This approach is somewhat ad hoc however, and is prone to various numerical problems. In any case, it does not provide an overall least-squares solution to the problem of fitting trees of this nature.

Carroll (1976) and Carroll and Pruzansky (1980) have recently taken a mathematical programming approach to fitting ultrametric trees or additive trees, to two-way proximity data. In fitting an ultrametric tree, their approach attempts to find a least-squares fit of a distance matrix constrained to satisfy the ultrametric inequality, by use of a "penalty function" which measures the degree of violation of that inequality, as defined in expression (1) above, to a given matrix of dissimilarities. The approach is also easily extended to fitting of path length trees satisfying the four point condition. A more direct procedure entailing a direct generalization of the Carroll and Pruzansky penalty approach, using a penalty function to enforce the four point condition, has recently been proposed by DeSoete (1983 a,b).

In overlapping nonhierarchical clustering, such as the ADCLUS model proposed by Shepard and Arabie (1975), each stimulus object either has or does not have each of a number of attributes, but these attributes are not restricted to be either exhaustive or mutually exclusive. A practical procedure for fitting the ADCLUS model is available in the MAPCLUS method devised by Arabie and Carroll (1980).

1.3 Multiple Trees and Hybrid Models

Carroll (1976) and Carroll and Pruzansky (1980) have recently introduced two new classes of models used to fit two-way proximity data —

multiple trees and hybrid models. Multiple tree models assume that the proximities data are simply related to a sum of distances coming from two or more different tree structures. Hybrid models combine the fitting of both tree structure(s) and continuous spatial components so that for a given two-way proximity data set, say, one might fit two tree structures and a two dimensional Euclidean space. This hybrid model was demonstrated on the Rosenberg and Kim (1975) data in Carroll (1976) and Carroll and Pruzansky (1980).

1.4 Research Objectives

Three-way spatial models and associated methods such as INDSCAL (Carroll and Chang 1970), Tucker's (1964,1972) three-mode scaling, Harshman's (1972) PARAFAC-2, and others have been devised to fit three-way proximity data in which a matrix of object \times object proximity matrices is given for each of a number of individual subjects or other data sources (Carroll and Wish 1974). Similarly a three-way nonhierarchical overlapping clustering method called INDCLUS (Carroll and Arabie 1979,1983) has been developed also to fit such three-way proximity data. However, prior to the present work, there were no known techniques for extending the fitting of hierarchical or other tree structure models to such three-way data. Nor has there been any effort as yet in extending hybrid models to the three-way case. Our goal is to develop such tree structure models designed to fit three-way proximity data. First, we examine existing two-way discrete tree-fitting methods.

2. INDTREES

2.1 Objectives

We propose an individual differences generalization of the Carroll (1976) and Carroll and Pruzansky (1980) two-way hierarchical tree-fitting models and methods to three-way proximities in which, say, different individuals are assumed to base their judgments on the same family of trees, but are allowed to have different node heights and/or branch lengths. This new methodology, entitled INDTREES (for INdividual Differences in TREE Structures), can fit single or multiple, ultrametric or additive trees to a given three-way proximity (dissimilarity) array. In the hybrid model case, individual subject weights, similar to INDSCAL, can also be introduced.

2.2 The Model

Let:

δ_{ijl} = the input dissimilarity between objects i and j in the l -th array;

d_{ijl} = the reconstructed dissimilarity (distance) between objects i and j in the l -th array; generally d_{ijl} can be thought of as a distance or, in some cases, a squared distance, plus an additive constant where the distances are defined in certain well specified ways in terms of one or more tree structures;

e_{ijl} = the error ($\delta_{ijl} - d_{ijl}$) term for i and j in the l -th slice or array;

$i, j = 1 \dots N$ objects;

$l = 1 \dots L$ arrays.

Then, the INDTREES model can be simply stated as:

$$\delta_{ijl} = d_{ijl} + e_{ijl} , \quad (5)$$

where one wishes to estimate the d_{ijl} which satisfy certain specific constraints depending upon the type and number of trees to be fitted. We shall discuss the major options separately.

2.2.1 Fitting a Single Ultrametric Tree

We first discuss the fitting of a single ultrametric tree to three-way proximity data because this phase is essential to all other types of trees that can be fit in INDTREES. We wish to estimate d_{ijl} to “best fit” the δ_{ijl} , and yet satisfy the three-way ultrametric inequality:

$$d_{ijl} \leq \max (d_{ikl}, d_{jkl}) \quad \forall i, j, k, l . \quad (6)$$

That is, we wish to estimate one general tree topology for L slices of the array, but allow different node heights and/or branch lengths for each slice. This can be translated into a constrained optimization problem where one attempts to estimate d_{ijl} in order to minimize:

$$T = \sum_{l=1}^L \sum_{i < j}^N [\delta_{ijl} - d_{ijl}]^2, \quad (7)$$

subject to satisfying the ultrametric inequality constraints in expression (6). Note that an equivalent statement of the ultrametric inequality in expression (6) is that all triangles formed by connecting all possible triples of points (d_{ijl}) are acute isosceles. This property can be restated as:

$$d_{ikl} \geq d_{jkl} \geq d_{ijl} \text{ iff } d_{ikl} = d_{jkl} \quad (8)$$

$$\forall i, j, k, \text{ and } l.$$

In addition, we want the associated ultrametric trees all to have the *same* topology. This is equivalent to requiring the same two pairs of distances of each triple (i, j, k) to be the longest two (and therefore to be equal) for all values of l . One can re-express the ultrametric inequality constraints in expression (8), as well as the requirement of a common topology, in a more convenient mathematical form via the following equality constraints:

$$P_1 = \sum_{l=1}^L \left[\sum_{i=1}^N \sum_{\substack{j=1 \\ i < k \\ j \neq i, k}}^N \sum_{k=1}^N w_{ik}^j (d_{ijl} - d_{jkl})^2 \right] = 0, \quad (9)$$

and

$$P_2 = \sum_{l=1}^L \left[\sum_{i=1}^N \sum_{\substack{j=1 \\ i < k \\ j \neq i, k}}^N \sum_{k=1}^N w_{ik}^j \left((d_{ikl} - d_{ijl})_+^2 + (d_{ikl} - d_{jkl})_+^2 \right) \right] = 0, \quad (10)$$

where:

$$w_{ik}^j = \begin{cases} 1 & \text{if } d_{ik.} \leq \min \{ d_{ij.}, d_{jk.} \} \\ 0 & \text{otherwise} \end{cases}$$

$$d_{ij.} = \frac{1}{L} \sum_{l=1}^L d_{ijl} ,$$

$$(x)_+ = \max \{0, x\} .$$

The equality constraints in expression (9) force the two largest distances (on average) to be equal while the equality constraints in expression (10) force the two largest (and therefore equal) distances to be the same for all values of l . Together, *both* expressions (9) and (10) enforce the ultrametric inequality based on a single tree structure. Note that the indicator function w_{jk}^l is defined in terms of an average two-way squared distance matrix $((d_{ij.}))$. This is just one way of defining the indicator function so as to move toward a “consensus” or common tree structure, yet it seems to be quite effective as demonstrated in the synthetic data analyses that will be discussed later. Both expressions (9) and (10) also restrict the tree topology of each slice to be the same (since the ultrametric inequality constraints are defined for all i, j, k , and l) outside of differential branch lengths and node heights for different values of l . Appendix I presents a proof that the conditions enforced by P_1 and P_2 are necessary and sufficient conditions for ultrametricity and topological equivalence of the L trees (slices).

A problem in the definition of these penalty functions, as pointed out by two referees, is that because of the definition of w_{jk}^l , neither P_1 nor P_2 is everywhere continuous (nor are their derivatives). The penalty function used by Carroll and Pruzansky (1980), and later by De Soete (1983 a,b) for the two way case of fitting ultrametric trees was continuous and had continuous first derivatives (although the higher order derivatives were not continuous). It would be possible to define penalty functions for the present case that are continuous and have continuous first derivatives, but these would involve excessive computation both to compute the functions themselves and their derivatives. Despite the fact that there are theoretical problems entailed in using discontinuous penalty functions, the approach presented here seems, nonetheless, to work well in practice and it was decided that the additional computational complexity, which would slow the convergence of the iterative procedure, would not be pursued. A methodology involving sub-gradients is an alternative approach.

Thus, the three-way ultrametric tree fitting problem can be expressed as a nonlinear equality constrained optimization problem in attempting to minimize expression (7) subject to the nonlinear equality constraints in (9) and (10). INDTREES utilizes an exterior penalty function approach (Rao 1979) in order to produce an at-least local optimum (minimum) and feasible estimate of d_{ijl} . The optimization problem can thus be restated as:

$$\text{Min}_{d_{ijl}} Z = (A + B), \quad (11)$$

where:

$$A = \frac{T}{a}, \quad (12)$$

$$a = \sum_{l=1}^L \sum_{i<j}^N \sum_{i<j}^N [\delta_{ijl} - \delta_{..l}]^2, \quad (13)$$

$$\delta_{..l} = \frac{2}{N(N-1)} \sum_{i<j}^N \sum_{i<j}^N \delta_{ijl} \quad (14)$$

$$B = \lambda^{(IT)} \left(\frac{P_1 + P_2}{b} \right) \quad (15)$$

$\lambda^{(IT)}$ = a positive penalty parameter at iteration IT,

$$b = \sum_{l=1}^L \sum_{i<j}^N \sum_{i<j}^N (d_{ijl} - d_{..l})^2 \quad (16)$$

$$d_{..l} = \frac{2}{N(N-1)} \sum_{i<j}^N \sum_{i<j}^N d_{ijl} . \quad (17)$$

The A term in expression (11) represents a normalized sums of squares which, when minimized, attempts to provide estimates of d_{ijl} which are as "close" as possible to the given data δ_{ijl} . Clearly, without constraints placed on d_{ijl} , one could drive this A -term to zero by making $d_{ijl} = \delta_{ijl}$, $\forall i, j, l$. The B term in expression (11) designates the magnitude of the violation of the ultrametric inequality constraints incorporated in the analysis. The net effect of the B -term in expression (11) is to increase Z in proportion to the amount by which the constraints in expressions (9) and (10) are violated. There will be a penalty for violating the constraints, and the amount of the penalty will increase at a rate based on the amount of violation of the constraint. The function Z , or $Z(d_{ijl}, \lambda^{(IT)})$, usually possesses a

minimum as a function of d_{ijl} in the infeasible region. The unconstrained minima, $d_{ijl}^{*(IT)}$, converge to at least a locally optimal solution of the original problem as $IT \rightarrow \infty$ and $\lambda^{(IT)} \rightarrow \infty$. Thus, since the unconstrained minima approach the feasible domain gradually as $IT \rightarrow \infty$, the $d_{ijl}^{*(IT)}$ eventually lie in (or arbitrarily close to) the feasible region. This process is why the exterior penalty function name applies — a “penalty” is applied to violations of the constraints and the process is usually initiated with a set of infeasible (exterior) points. The procedure converts a difficult constrained nonlinear programming problem into a sequence of unconstrained problems.

Note that both sets of constraints involving P_1 and P_2 are normalized by a sum of squares term b . This scaling is done for two reasons. One is that the division of P_1 and P_2 by b in the constraints will avoid the degenerate, but feasible, solution $d_{ijl} = 0, \forall i, j, l$. The other reason is one of numerical convenience. The A part of expression (11) typically ranges between 0 and 1. However, the B part, ignoring $\lambda^{(IT)}$ for now, can be unbounded. In addition, different types of constraints in B can have quite different ranges or variances in typical applications. This variation implies that, without normalizing the sets of constraints, the algorithm may inherently “pay more attention” to the constraints, or to those constraints with larger ranges since they can contribute more to Z . Thus, it is desirable to normalize each set of constraints so that they have roughly equivalent ranges.

The various phases of INDTREES which pertain to the fitting of a single ultrametric tree are described as follows:

2.2.1.1 Phase I: Input The user must specify: if input data are distances or profile data, preprocessing options, starting values for d_{ijl} , gradient convergence tolerance (TOL), maximum number of minor iterations (MIT*), maximum number of major iterations (IT*), initial penalty multiplier ($\lambda^{(1)}$), penalty multiplier factor (t), number of trees to be fitted, type of trees to be estimated, maximum number of giant iterations (GIT*), output and diagnostic options. Each of these will be defined in the various phases of the INDTREES algorithm to which it relates.

2.2.1.2 Phase II: Preprocessing If the user specifies three-way profile data (Shepard 1972) $X_{ik}^{(l)}$ as input, INDTREES will calculate three-way squared Euclidean distances δ_{ijl} via:

$$\delta_{ijl} = (\mathbf{X}_i^{(l)} - \mathbf{X}_j^{(l)})(\mathbf{X}_i^{(l)} - \mathbf{X}_j^{(l)})', \quad (18)$$

where $\mathbf{X}_i^{(l)}$ is the $1 \times K$ ($K =$ the number of variables or columns) vector of observed variable values for object i in the l -th array. The user can also provide his/her own dissimilarities as input.

The user has his/her choice of one of three preprocessing options involving δ_{ijl} :

1. using the “raw” δ_{ijl} , i.e., no preprocessing;
2. normalizing each $N \times N$ “slice” of the array (actually, just the upper/lower triangular half matrix because of symmetry) to unit variance (matrix conditional);
3. normalizing the entire three-way array (again, only the upper/lower triangular halves) to unit variance.

The type of normalization utilized will often be a function of the conditionality (Takane, et al. 1977) assumptions made about the data, as well as of numerical convenience. If the data are not “comparable” across slices (matrix conditional), then the user should select either no preprocessing or normalizing the data *within* each slice. Note that this latter option could have profound consequences in altering a pre-normalized tree structure. If the data can be assumed to be comparable between slices (matrix unconditional), then the user can either use the raw data or normalize the entire data array *across* slices. This latter option would not alter the topology of a pre-normalized tree structure, but would only change node heights or branch lengths. Depending upon assumptions concerning the conditionality of the data, normalization options could also be utilized in order to obtain data in some given specified range. Since trees are defined up to an additive constant, there is really no need for adding/subtracting scale factors to/from the data. Because of this property, the use of variances, as opposed to sums-of-squares, is permitted in both the normalization procedures and goodness-of-fit measures, as will later be described.

2.2.1.3 Phase III: Starting Values There are five different options for obtaining starting values for the d_{ijl} :

1. A random start where positive numbers are selected randomly from a uniform distribution $U[0, r]$, where the range r is user specified;
2. A given start where the user provides given starting values of d_{ijl} ;
3. Using the original δ_{ijl} ($d_{ijl} = \delta_{ijl}$) as starting values;
4. Performing a standard hierarchical clustering analysis (e.g., Johnson 1967) on each slice of \mathbf{A} using either the complete, single, or average linkage method and generating $\mathbf{D} = ((d_{ijl}))$ as predicted values (node heights) from each of the L trees;
5. Compute

$$\delta_{ij.} = \frac{1}{L} \sum_{l=1}^L \delta_{ijl}, \quad (19)$$

an average data matrix (averaged over slices), perform one of the various hierarchical clustering procedures mentioned above on

$\Delta. = (\delta_{ij.})$ and, assign the predicted values of the one “average” tree to all L slices.

Of the five options presented, only the last option provides an initial starting set of values that are feasible in the sense of generating a common tree structure topology for each slice of the data δ_{ijl} . In this simple case, however, all the node heights across slices are identical. Option four provides d_{ijl} which satisfy the ultrametric inequality *separately* within each slice — but there is no guarantee that the topology of the tree structures generated for each slice will be the same. Option three renders starting values that clearly minimize the A -term of expression (11), the sums of squares between actual and predicted data values. However, these starting values will rarely be feasible in satisfying constraint sets (9) and (10). Option one will generally neither render “close” nor feasible starting values. The second option might be useful for informally testing an *á priori* hypothesis concerning the underlying tree structure of δ_{ijl} based on either theory or previous analyses, and will also provide efficient starting values in cases in which “good” initial estimates are available on theoretical or other grounds.

2.2.1.4 Phase IV: Set Penalty Parameter $\lambda^{(IT)}$ Since the unconstrained minimization of Z is to be performed for an increasing sequence of $\lambda^{(IT)}$, it might initially appear that by choosing a very large initial value of $\lambda^{(1)}$, one could avoid an excessive number of penalty major iterations to minimize Z . However, what happens in practice computationally is that the penalty function will move to a solution which will satisfy all the constraints in the B -term of expression (11), but will not be the optimal constrained solution since the A -term will *not* be jointly minimized. In other words, one would be likely to derive a feasible solution whose A -term would be quite high (a poor local optimum).

Assuming $0 < \lambda^{(IT)} < \lambda^{(IT+1)}$, the exterior penalty function has been shown under fairly general conditions to converge to at least a locally optimum solution. On the basis of his empirical work, Rao (1979) recommends setting:

$$\lambda^{(IT+1)} = t\lambda^{(IT)}, \quad (20)$$

where t is the user specified penalty multiplier > 1 . Initially, $\lambda^{(1)} = .001$ and $t = 10$ appears to work quite well in INDTREES practice to date.

2.2.1.5 Phase V: Calculate Conjugate Gradients As mentioned, the exterior penalty function method transforms a constrained optimization problem into a series or sequence of unconstrained problems. This implies that one needs a numerical method to solve the various unconstrained problems. The method of conjugate gradients (Fletcher and Reeves 1964) is utilized to

solve the unconstrained problem suggested in equation (11). The complete conjugate gradient procedure encompassing Phases V, VI, VII can be summarized as follows:

- (i) Start with initial d_{ijl} 's : $\mathbf{D}^{(1)} = ((d_{ijl}^{(1)}))$; set MIT = 1.
- (ii) Set the first search direction $\mathbf{S}^{(1)} = -\nabla \mathbf{Z}^{(1)}$, where $\nabla \mathbf{Z}^{(\text{MIT})}$ is the gradient of $\mathbf{Z}^{(\text{MIT})}$. (The interested reader may write the authors for an unpublished early draft of this paper in which the gradient is derived in detail.)
- (iii) Find $\mathbf{D}^{(2)} = (d_{ijl}^{(2)})$ according to the relation:

$$\mathbf{D}^{(2)} = \mathbf{D}^{(1)} + u^{(1)}\mathbf{S}^{(1)}, \quad (21)$$

where $u^{(1)}$ is the optimal step length in the direction $\mathbf{S}^{(1)}$ (calculated in Phase VI). Set MIT = 2.

- (iv) Calculate $\nabla \mathbf{Z}^{(\text{MIT})}$ and set

$$\mathbf{S}^{(\text{MIT})} = -\nabla \mathbf{Z}^{(\text{MIT})} + \frac{(\nabla \mathbf{Z}^{(\text{MIT})})'(\nabla \mathbf{Z}^{(\text{MIT})})}{(\nabla \mathbf{Z}^{(\text{MIT}-1)})'(\nabla \mathbf{Z}^{(\text{MIT}-1)})} \mathbf{S}^{(\text{MIT}-1)}. \quad (22)$$

- (v) Compute the optimal step length $u^{(\text{MIT})}$ (Phase VI) in the direction $\mathbf{S}^{(\text{MIT})}$, and find

$$\mathbf{D}^{(\text{MIT}+1)} = \mathbf{D}^{(\text{MIT})} + u^{(\text{MIT})}\mathbf{S}^{(\text{MIT})}. \quad (23)$$

- (vi) If the convergence criteria (computed in Phase VII) are satisfied, stop (Phase VII). Otherwise set MIT = MIT + 1 and go to step (iv) above (i.e., undertake another minor iteration).

It has been demonstrated empirically that conjugate gradient procedures can avoid the typical "cycling" often encountered with steepest descent algorithms.

2.2.1.6 Phase VI: Step Size Determination Two step size options have been programmed and tested in INDTREES — a Fibonacci line search (Cooper and Steinberg 1972) and a quadratic interpolation procedure (see Carroll and Pruzansky 1980). Both line search procedures assume a unimodal objective function in a specified range. The Fibonacci procedure calculates $u^{(\text{MIT})}$ via an adaptive sequence of objective value evaluations while the quadratic line search approximates the objective function locally in a specified interval with a quadratic function and then takes the minimum of this quadratic approximation. Care must be exercised in either option since a bimodal objective function, in a specified interval, may be encountered

near a constraint boundary. Either procedure may render a $u^{(\text{MIT})}$ that would provide an optimal \mathbf{D} *outside* the feasible region. For this reason, the Fibonacci procedure is preferred, where the interval of search is shortened adaptively after each major iteration.

2.2.1.7 Phase VII: Minor Iteration Tests Three convergence tests are performed in the minor estimation cycle to test whether additional iterations of the conjugate gradient procedure are necessary or beneficial:

1. If $\text{MIT} \geq \text{MIT}^*$, stop;
2. If $\|\nabla Z^{(\text{MIT})}\| < \text{TOL}$, stop;
3. If $|Z^{(\text{MIT}-1)} - Z^{(\text{MIT})}| < \text{TOL}$, stop.

If none of the three are satisfied, MIT is incremented and control returns to step (iv) of Phase V. If one or more are satisfied the algorithm proceeds to Phase VIII.

2.2.1.8 Phase VIII: Major Iteration Tests A number of convergence tests are performed after a complete major iteration cycle (completing phases III-VII) to test if additional computation is necessary:

1. If $\text{IT} \geq \text{IT}^*$, stop;
If B in equation (11) $< \text{TOL}$ and $\|\nabla Z^{(\text{MIT})}\| < \text{TOL}$, stop.

Otherwise, set $\text{IT} = \text{IT} + 1$ and return to Phase IV above. Note that once this penalty procedure converges or terminates, the INDTREES algorithm checks to see that:

$$(Z^{(\text{IT}-1)} - Z^{(\text{IT})}) \geq 0 \quad (24)$$

so that each penalty phase reduces the augmented loss function in expression (11). If this reduction is not the case, then the results of this most recent penalty phase are ignored, and $\mathbf{D}^{(\text{IT}-1)}$ is retained for the next phase.

Gill, Murray, and Wright (1981) raise two important concerns regarding the practical use of penalty functions. One is that the effect of a penalty-type transformation is, *at best*, to create a *local minimum* of Z which, for sufficiently large $\lambda^{(\text{IT})}$, is “near” the true solution. However, the subproblem (expression 11) may be defective, and there are simple examples where a penalty function is unbounded below for any value of the penalty parameter. When an objective function is unbounded below, an unconstrained algorithm may experience difficulty in converging to the desired local minimum; for example, the iterates in the unconstrained subproblem may move outside the region of the local minimum introduced by the transformation. This property has important implications if a standard unconstrained method is to be used to solve the unconstrained subproblem. In our use of the penalty function however, Z has a lower bound of zero and such a potential problem is really not relevant.

The other concern is related to the fact that the Hessian matrix of second derivatives often becomes ill-conditioned near the constraint boundaries or if $\lambda^{(IT)}$ is excessively large. The conditioning of the Hessian is significant because the minima of successive penalty functions become increasingly poorly determined in the null space of the active constraint gradients. If the initial value of $\lambda^{(IT)}$ is "too large," even a robust unconstrained algorithm will typically experience great difficulty in the attempt to compute an optimum solution. Therefore, in order to solve (11) by a penalty function method, a sequence of unconstrained problems is solved, with increasing values of the penalty parameter. Each successive $D^{(IT)}$ solution is used as the starting point for minimization with the next value of the penalty parameter, until acceptable convergence criteria for the solution of the original problem are satisfied.

2.2.2 Fitting Multiple Ultrametric Trees

There are many sets of proximities data that are not well represented by either hierarchical or partition-type clusterings. One alternative model is the ADCLUS model of Shepard and Arabie (1979) as implemented in the MAPCLUS procedure (Arabie and Carroll 1980) or its generalization to the three-way case, called INDCLUS (Carroll and Arabie 1983) (see also the GENCLUS model and method by DeSarbo (1982)), in which proximities data are assumed to arise from discrete attributes that define overlapping, but nonhierarchically organized clusters. It may be, in such a case, however, that the clusters can be organized into two or more separate hierarchal clusterings. Each of these separate hierarchies could represent, for example, an organized family of subordinate and superordinate concepts. For instance, in the case of animal names one might imagine one hierarchical conceptual scheme based on the phylogenetic scale, and another based on function or relationship to man (e.g., tame vs. wild, with tame animals further broken down into pets, work animals, and animals raised for food, pets further broken down into house pets versus outdoor pets, and so on). Two such conceptual hierarchical structures would obviously be far from independent of one another — whether or not an animal is a pet, for example, is hardly independent of the phylogenetic classification of the animal, but they could be sufficiently distinct such that an appropriate technique could separate them. Such multiple hierarchies in data may often be obscured in standard clustering analyses, simply because of this problem of a possibly high degree of correlation among separate structures. Apparently what is needed (in this case) is a method that would allow fitting a model entailing *multiple* tree structures to data — a multidimensional generalization of the single tree structure, as it were. Such a generalization has been pursued in the two-way case by Carroll and Pruzansky (1975, 1980).

In this section we consider fitting Δ with a mixture of M different ultrametric trees (each satisfying the ultrametric inequality). In particular, we want to approximate the three-way array $\Delta = ((\delta_{ijl}))$ as a sum

$$\Delta \approx \mathbf{D}_1 + \mathbf{D}_2 + \cdots + \mathbf{D}_M, \quad (25)$$

where each matrix \mathbf{D}_m ($m = 1 \dots M$) comprises a three-way array of distance matrices each satisfying the ultrametric inequality and consistent with the same (the m^{th}) hierarchical tree.

After a single ultrametric tree has been estimated to minimize expression (11), one can estimate further trees if desired. If not, INDTREES progresses to Phase X. We examine, in detail, the numerical procedure utilized for calculating $M > 1$ ultrametric trees. This is based on what has come to be called an alternating least-squares (or ALS) approach (Wold 1966).

2.2.2.1 Phase IX: Computing Residuals and Reestimation of Component Trees Denote the values of d_{ijl} for the m -th ultrametric tree as $\mathbf{D}_m = ((d_{ijl}^{(m)}))$. In order for INDTREES to reestimate the m' -th ultrametric tree, residuals are first computed via the equation:

$$r_{ijl}^{(m')} = \delta_{ijl} - \sum_{\substack{m=1 \\ m \neq m'}}^M d_{ijl}^{(m)}, \quad (26)$$

and the INDTREES algorithm returns to Phase III, initiating another penalty estimation cycle, using $r_{ijl}^{(m')}$ as starting estimates for $d_{ijl}^{(m')}$ on the first estimation of those parameters (in subsequent iterations we may use $r_{ijl}^{(m')}$ or the previous estimates of $d_{ijl}^{(m')}$ as starting values.). We thus fit additional ultrametric trees in a stage-wise fashion after removing the ‘‘information’’ or structure contained in previously estimated ultrametric trees. This process begins by estimating \mathbf{D}_1 , with initial values of $\mathbf{D}_2 = \mathbf{D}_3 = \dots = \mathbf{D}_m = \mathbf{O}$. We continue this procedure in an iterative fashion until the process as a whole converges. In case one is fitting additive trees, an additional phase is added after each cycle of (re)estimating these ultrametric tree components.

2.2.3 Fitting Path Length or Additive Trees

After one cycle of (re)estimating the complete number of ultrametric trees desired, one can estimate constants to convert such ultrametric trees into path length or additive trees, if desired. If this is not desired, the INDTREES algorithm proceeds directly to phase XI.

As discussed in Carroll (1976) and Carroll and Pruzansky (1980), from the work of Farris (personal communication) and Hartigan (1975), given a path length tree, it is always possible to decompose that tree into an ultrametric tree plus an additive residual that can be associated with a special additive tree called a "star" or "bush". Appendix II provides a proof.

In the three-way case, this decomposition can be stated as:

$$d_{ijl} = d_{ijl}^* + c_{il} + c_{jl} , \quad (27)$$

which states that each slice of the path length array, \mathbf{D}_l , is decomposable into a \mathbf{D}_l^* that satisfies the ultrametric inequality plus an additive residual, which we shall simply call $\mathbf{C}_l \equiv (c_{il} + c_{jl})$, with the understanding that diagonals of each \mathbf{C}_l are undefined, or zero if defined. \mathbf{C}_l is, in fact, just the distance matrix for a special path length or additive tree with only one non-terminal or internal node, i.e., a "star."

Now, let us consider the case in which we want to decompose our data array \mathbf{A} into a mixture of path length \mathbf{D} 's. We then have

$$\mathbf{A} \approx \mathbf{D}_1 + \mathbf{D}_2 + \cdots + \mathbf{D}_M , \quad (28)$$

where each \mathbf{D} is a path length distance array. We thus have

$$\mathbf{D}_m = \mathbf{D}_m^* + \mathbf{C}_m , \quad (29)$$

where \mathbf{D}_m^* satisfies the ultrametric inequality and $\mathbf{C}_m \equiv (c_{ilm} + c_{jlm})$. Thus

$$\mathbf{A} \approx \mathbf{D}_1^* + \mathbf{D}_2^* + \cdots + \mathbf{D}_M^* + \sum_{m=1}^M \mathbf{C}_m = \mathbf{D}_1^* + \mathbf{D}_2^* + \cdots + \mathbf{D}_M^* + \mathbf{C} , \quad (30)$$

where $\mathbf{C} \equiv \sum \mathbf{C}_m$, and \mathbf{C} is of the same form as each individual \mathbf{C}_m . That is, $\mathbf{C} \equiv (c_{il} + c_{jl})$ where $c_{il} = \sum_m c_{ilm}$ and $c_{jl} = \sum_m c_{jlm}$.

It follows that, in a mixture of path length trees, we cannot "partial out" the additive components associated with different trees, but may combine them into a single additive component. Thus, a sum of M path length (or additive) trees can be decomposed into a sum of M ultrametric trees plus a *single* "star" or "bush." (Note that the analogous argument was made for the two-way case by Carroll (1976) and Carroll and Pruzansky (1980).)

2.2.3.1 Phase X: Compute Path Constants Since each D_m^* is an ultrametric distance matrix, it follows that we may fit mixtures of path length trees by simply adding an additional step to the alternating least-squares strategy in which the constants c_{ij} are estimated by least squares procedures. That is, given estimates of D_1^*, \dots, D_M^* , we define

$$\Delta^{**} = \Delta - \sum_{m=1}^M D_m^*, \quad (31)$$

and fit Δ^{**} by $\hat{C} \equiv (\hat{c}_{ij} + \hat{c}_{ji})$ such that $|\Delta^{**} - \hat{C}|^2$ is minimized.

It remains only to define the least squares estimates \hat{c}_{ij} of the c_{ij} 's. For this purpose, we shall drop the $**$'s from Δ , and consider the problem of finding \hat{c}_{ij} 's minimizing:

$$|\Delta - \hat{C}|^2 \equiv \sum_l^L \sum_{i < j}^N (\delta_{ijl} - c_{ij} - c_{ji})^2 \quad (32)$$

The solution, obtained by straightforward means, is:

$$\hat{c}_{ij} = \frac{1}{n-2} \sum_{j \neq 1}^N \delta_{ijl} - \frac{1}{(n-1)(n-2)} \sum_{j < k}^N \sum \delta_{jkl}. \quad (33)$$

[Note: The constants c_{ij} are almost the same as the constants that would be added to rows and columns to double center each Δ , and differ primarily because only the off diagonals of Δ are fit, the diagonals of Δ being treated, in effect, as missing data.]

2.2.3.2 Phase XI: GIANT Iteration Tests Let us now assume that INDTREES has estimated D_1, D_2, \dots, D_M for each of M ultrametric trees desired and (if path trees are being fit) optional path length constants C . Unless one of the following criteria are met, the procedure cycles back to Phase III setting $GIT = GIT + 1$ and computing residuals via:

$$R_1 = \Delta - \left[\sum_{m \neq 1}^M D_m + C \right] \quad (34)$$

in order to improve the estimates of D_1 , and then similarly reestimate $D_2, D_3 \dots D_m$. Note that $C \equiv O$ if phase X is omitted. The final criteria for INDTREES termination are:

- (i) If $GIT = GIT^*$, (the maximum number of GIANT iterations specified by the user) stop; (Note that, in the case where only *one ultrametric* tree is to be fit, with no path constants to convert it to a path length tree, $GIT^* \equiv 1$ by default).
- (ii) If $\left[\left(\sum_m^M |D_m^{(GIT)} - D_m^{(GIT-1)}|^2 \right) + (|C^{GIT} - C^{(GIT-1)}|^2) \right] < TOL$, stop;
- (iii) If $(V^{(GIT)} - V^{(GIT-1)}) < TOL$, stop, where V is the variance accounted-for measure (R^2) between $\mathbf{\Delta}$ and $\hat{\mathbf{\Delta}}$, where $\hat{\mathbf{\Delta}}$ is defined as:

$$\hat{\mathbf{\Delta}} = \mathbf{D}_1 + \mathbf{D}_2 + \dots + \mathbf{D}_M + \mathbf{C}, \quad (35)$$

for all off-diagonal elements of each slice.

3. Some Analyses of Synthetic Data

Four different synthetic data sets were generated for INDTREES testing. Each data set was generated from one or more known tree structure(s). The test was to examine how well the INDTREES method could recover the tree structure(s) from a random start (the first option listed above for starting values).

3.1 Ultrametric Tree

Table 1 presents the three-way array $\mathbf{\Delta}$ generated for $N = 8$ objects from a single ultrametric tree and associated node heights (with no error).

Basically, $\mathbf{\Delta}$ is errorless data which should be perfectly fit by INDTREES. Using a random start, the INDTREES methodology converged in nine major iterations providing a variance accounted for value of .99997. Table 2 presents the feasible predicted values d_{ijl} . As one can see, the differences between δ_{ijl} and d_{ijl} are in the third decimal place.

Figures 1a-c present the associated tree structures for the predicted d_{ijl} values. Notice that the topology for each tree (slice) is the same except for node heights. The average-link method was utilized for the tree construction, although either single-, complete-, or average-link methods would render identical structures, given that the data satisfy the ultrametric inequality.

As a secondary analysis it was desirable to examine the impact of introducing error into the δ_{ijl} so that the input data would not satisfy the ultrametric inequality. Table 3 presents the revised three-way array. The data here are similar to those provided in Table 1 except that one entry (and

TABLE 1
Synthetic Three-Way Array from a
Single Ultrametric Tree

		Δ							
		0							
		1	0						
		3	3	0					
Slice: 1		3	3	1	0				
		3	3	1	1	0			
		1	1	3	3	3	0		
		3	3	1	1	1	3	0	
		1	1	3	3	3	1	3	0
		0							
		2	0						
		2	2	0					
Slice: 2		2	2	1	0				
		2	2	2	2	0			
		1	2	2	2	2	0		
		2	2	2	2	1	2	0	
		2	1	2	2	2	2	2	0
		0							
		2	0						
		3	3	0					
Slice: 3		3	3	2	0				
		3	3	3	3	0			
		2	2	3	3	3	0		
		3	3	3	3	2	3	0	
		2	2	3	3	3	2	3	0

its symmetric counterpart) in each slice was altered so that the new data would *not* satisfy the ultrametric inequality. The bold, italicized entries in Table 3 designate the numbers changed from the data in Table 1. The variance accounted for statistic between the data in Table 1 and those in Table 3 was .883. Thus, one feasible solution — the data in Table 1 — accounted for over 88% of the variance of the data in Table 3. Three analyses were performed. In the first analysis, a random start was utilized. In the second, Δ from Table 1 was utilized as the starting solution. In the third, Δ from Table 3 was utilized as the starting solution. The three solutions were identical up to the third decimal place. Table 4 presents one of the solutions (random start) which accounted for 88.7% of the variance of the synthetic

TABLE 2

The Recovered Solution for the Single Ultrametric
Tree Fitted to Synthetic Data

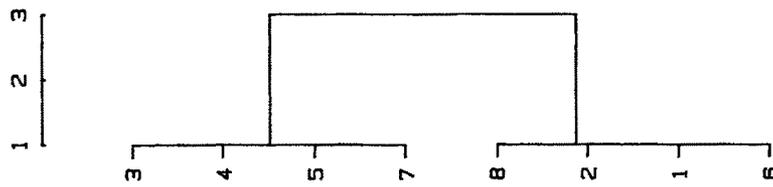
D								
	.000							
	1.003	.000						
	3.003	3.003	.000					
Slice: 1	3.002	3.003	.998	.000				
	3.003	3.003	1.002	1.000	.000			
	.999	1.003	3.002	3.003	3.002	.000		
	3.003	3.003	1.001	.999	.998	3.003	.000	
	1.005	1.003	3.003	3.003	3.003	1.004	3.003	.000
	.000							
	1.995	.000						
	2.003	2.004	.000					
Slice: 2	2.003	2.003	1.003	.000				
	2.005	2.004	1.991	1.993	.000			
	.996	1.996	2.004	2.003	2.004	.000		
	2.004	2.004	1.994	1.996	1.006	2.004	.000	
	1.996	1.005	2.004	2.005	2.004	1.997	2.004	.000
	.000							
	1.999	.000						
	3.002	3.002	.000					
Slice: 3	3.002	3.001	2.001	.000				
	3.001	3.002	2.997	2.997	.000			
	1.994	1.999	3.001	3.002	3.002	.000		
	3.001	3.002	2.997	2.996	2.004	3.001	.000	
	2.002	1.998	3.001	3.001	3.002	2.001	3.002	.000

data with error in Table 3. Figures 2a-c depict the fitted ultrametric trees derived from an average linkage analysis of the data in Table 4. Note that the trees here are nearly (except for node heights and branch lengths) equal to their counterparts in Figures 1a-c.

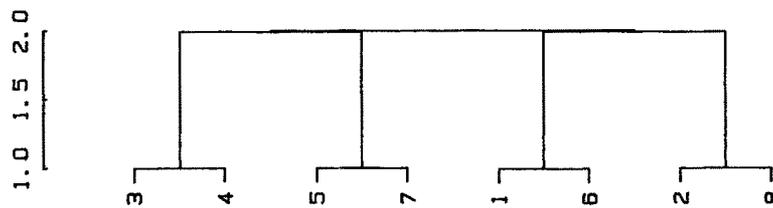
3.2 Path Length or Additive Tree

As mentioned earlier, a path length or additive tree can be decomposed into an ultrametric tree plus a bush or star graph. Given the synthetic data (without error) presented in Table 1, row and column constants (c_{il}, c_{jl}) were generated to produce the data in Table 5. Also presented in Table 5

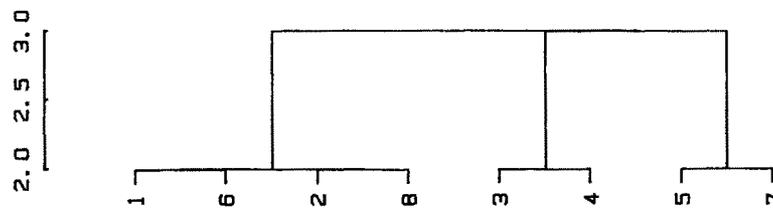
(a)
Single Ultrametric Tree - Slice One



(b)
Single Ultrametric Tree - Slice Two



(c)
Single Ultrametric Tree - Slice Three



Figures 1a-c. Fitted Tree Structure for the Single Ultrametric Tree Synthetic Data Example in Table 2.

TABLE 3
Synthetic Data with Error for a Single
Ultrametric Tree

		Δ							
		0							
	<i>1</i>	0							
	1	3	0						
Slice: 1	3	3	1	0					
	3	3	1	1	0				
	1	1	3	3	3	0			
	3	3	1	1	1	3	0		
	1	1	3	3	3	1	3	0	
		0							
	2	0							
	2	2	0						
Slice: 2	2	2	1	0					
	2	2	2	2	0				
	1	2	2	2	2	0			
	2	<i>1</i>	2	2	1	2	0		
	2	1	2	2	2	2	2	0	
		0							
	2	0							
	3	3	0						
Slice: 3	3	3	2	0					
	3	3	3	2	0				
	2	2	3	3	3	0			
	3	3	3	3	2	3	0		
	2	2	3	3	3	2	3	0	

Bold, italicized entries denote changes made to data in Table 1.

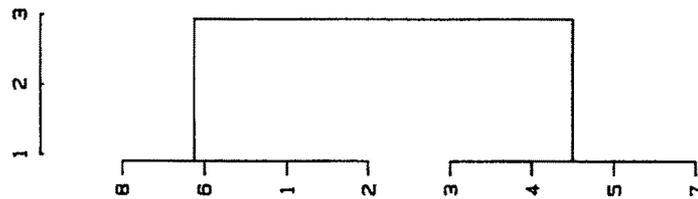
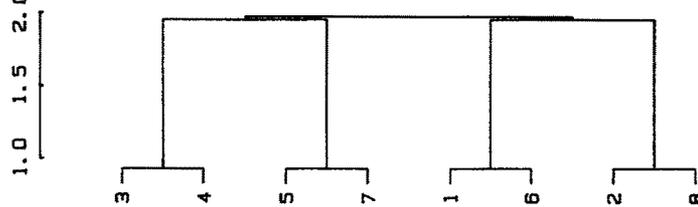
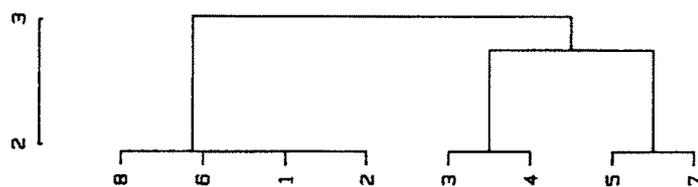
are the constants. Note as Cunningham (1978) demonstrates, there are some indeterminacies involved in estimating an additive tree. One indeterminacy involves where one decides to place the root. The root can be placed between any two nodes in the tree. There is also an additive constant indeterminacy. Thus while we should be able to recover the basic topology of this additive tree, there are problems inherent in such a tree as to where to place the root and associated additive constants.

TABLE 4

The Recovered Solution for the Single Ultrametric Tree Fitted to Synthetic Data with Error

D									
	.000								
	.905	.000							
Slice: 1	2.945	2.946	.000						
	2.946	2.946	.905	.000					
	2.946	2.946	.905	.905	.000				
	.905	.905	2.946	2.946	2.946	.000			
	2.946	2.946	.905	.905	.905	2.946	.000		
	.905	.905	2.946	2.946	2.946	.905	2.946	.000	
	.000								
	1.955	.000							
Slice: 2	1.978	1.978	.000						
	1.978	1.978	.927	.000					
	1.978	1.978	1.955	1.955	.000				
	.927	1.955	1.978	1.978	1.978	.000			
	1.978	1.977	1.955	1.955	.927	1.978	.000		
	1.955	.927	1.978	1.978	1.978	1.955	1.978	.000	
	.000								
	1.940	.000							
Slice: 3	3.028	3.028	.000						
	3.028	3.028	1.940	.000					
	3.028	3.028	2.756	2.756	.000				
	1.940	1.940	3.028	3.028	3.028	.000			
	3.028	3.028	2.756	2.756	1.940	3.028	.000		
	1.940	1.940	3.028	3.028	3.028	1.940	3.028	.000	

Table 6 presents the recovered solutions for both the ultrametric tree part and constant part. The procedure converged in five giant iterations producing a variance accounted for of .9998. Note that while neither Δ in Table 1 nor the C in Table 5 were recovered due to the above mentioned indeterminacies, the overall additive tree was recovered quite well. Figures 3a-c present the ultrametric tree topology from D in Table 6. Note the obvious similarity in the topology of these trees with their counterparts in Figure 1a-c indicating that, except for node heights and branch lengths, the overall tree topology has been recovered quite well.

(a)
Single Ultrametric Tree With Error: Slice One(b)
Single Ultrametric Tree With Error: Slice Two(c)
Single Ultrametric Tree With Error: Slice Three

Figures 2a-c. Fitted Tree Structures for the Single Ultrametric Tree Synthetic Data with Error Example in Table 4.

TABLE 5
Synthetic Data for a Single Additive Tree

C

ADDITIVE TREE CONSTANTS:

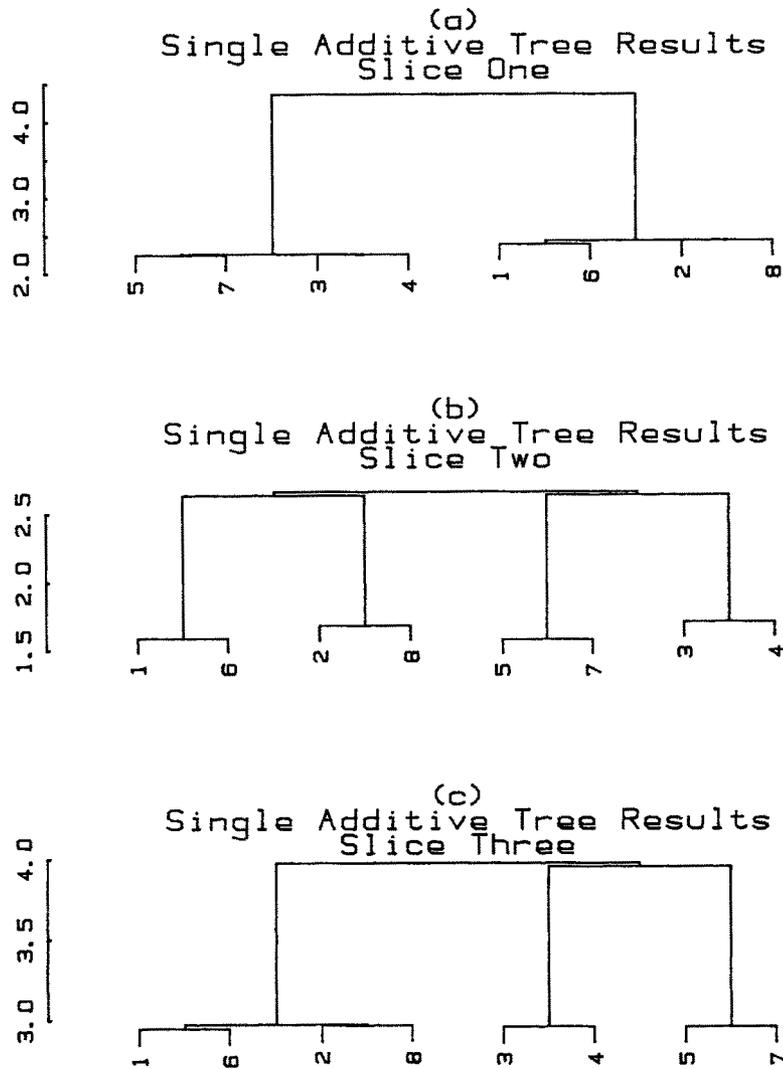
	1	.850	.940	.840	.520	.600	.120	.550	.990
Slice =	2	.330	.630	1.000	.170	.100	.080	.090	.200
	3	.700	.360	.800	.160	.390	.110	.570	.790

Δ

	.000								
	2.790	.000							
Slice: 1	4.690	4.780	.000						
	4.370	4.460	2.360	.000					
	4.450	4.540	2.440	2.120	.000				
	1.970	2.060	3.960	3.640	3.720	.000			
	4.400	4.490	2.390	2.070	2.150	3.670	.000		
	2.840	2.930	4.830	4.510	4.950	2.110	4.540	.000	
	.000								
	2.960	.000							
Slice: 2	3.330	3.360	.000						
	2.500	2.800	2.170	.000					
	2.430	2.730	3.100	2.270	.000				
	1.410	2.710	3.080	2.250	2.180	.000			
	2.420	2.720	3.090	2.260	1.190	2.170	.000		
	2.530	1.830	3.200	2.370	2.300	2.280	2.290	.000	
	.000								
	3.060	.000							
Slice: 3	4.500	4.160	.000						
	3.860	3.520	2.960	.000					
	4.090	3.750	4.190	3.550	.000				
	2.810	2.470	3.910	3.270	3.500	.000			
	4.270	3.930	4.370	3.730	2.960	3.680	.000		
	3.490	3.150	4.590	3.950	4.180	2.900	4.860	.000	

3.3 Multiple Ultrametric Trees

Table 7 presents the synthetic data utilized to test INDTREES in its ability to recover two ultrametric trees. Here, both Δ_1 and Δ_2 each satisfy the ultrametric inequality and represent two different ultrametric trees ($\Delta_1 = \Delta$ in Table 1). The array Δ in Table 7 is the sum of Δ_1 and Δ_2 .



Figures 3a-c. Fitted Ultrametric Tree Structures for the Single Additive Tree Data Example in Table 6.

TABLE 7
Synthetic Data for Two Ultrametric Trees

		Δ_1								
	0									
	1	0								
	3	3	0							
Slice: 1	3	3	1	0						
	3	3	1	1	0					
	1	1	3	3	3	0				
	3	3	1	1	1	3	0			
	1	1	3	3	3	1	3	0		
	0									
	2	0								
	2	2	0							
Slice: 2	2	2	1	0						
	2	2	2	2	0					
	1	2	2	2	2	0				
	2	2	2	2	1	2	0			
	2	1	2	2	2	2	2	0		
	0									
	2	0								
	3	3	0							
Slice: 3	3	3	2	0						
	3	3	3	3	0					
	2	2	3	3	3	0				
	3	3	3	3	2	3	0			
	2	2	3	3	3	2	3	0		
		Δ_2								
	0									
	1	0								
	2	2	0							
Slice:1	2	2	1	0						
	3	3	3	3	0					
	3	3	3	3	1	0				
	3	3	3	3	2	2	0			
	3	3	3	3	2	2	1	0		
	0									
	2	0								
	3	3	0							
Slice: 2	3	3	2	0						
	4	4	4	4	0					
	4	4	4	4	2	0				
	4	4	4	4	3	3	0			
	4	4	4	4	3	3	2	0		
	0									
	3	0								
	4	4	0							
Slice 3	4	4	3	0						
	5	5	5	5	0					
	5	5	5	5	3	0				
	5	5	5	5	4	4	0			
	5	5	5	5	4	4	3	0		

TABLE 7 (Continued)

		$\Delta = \Delta_1 + \Delta_2$							
		0							
		2	0						
		5	5	0					
Slice: 1		5	5	2	0				
		6	6	4	4	0			
		4	4	6	6	4	0		
		6	6	4	4	3	5	0	
		4	4	6	6	5	3	4	0
		0							
		4	0						
		5	5	0					
Slice: 2		5	5	3	0				
		6	6	6	6	0			
		5	6	6	6	4	0		
		6	6	6	6	4	5	0	
		6	5	6	6	5	5	4	0
		0							
		5	0						
		7	7	0					
Slice: 3		7	7	5	0				
		8	8	8	8	0			
		7	7	8	8	6	0		
		8	8	8	8	6	7	0	
		7	7	8	8	7	6	6	0

The IND TREES procedure converged in four giant iterations providing a .99998 variance accounted-for measure.

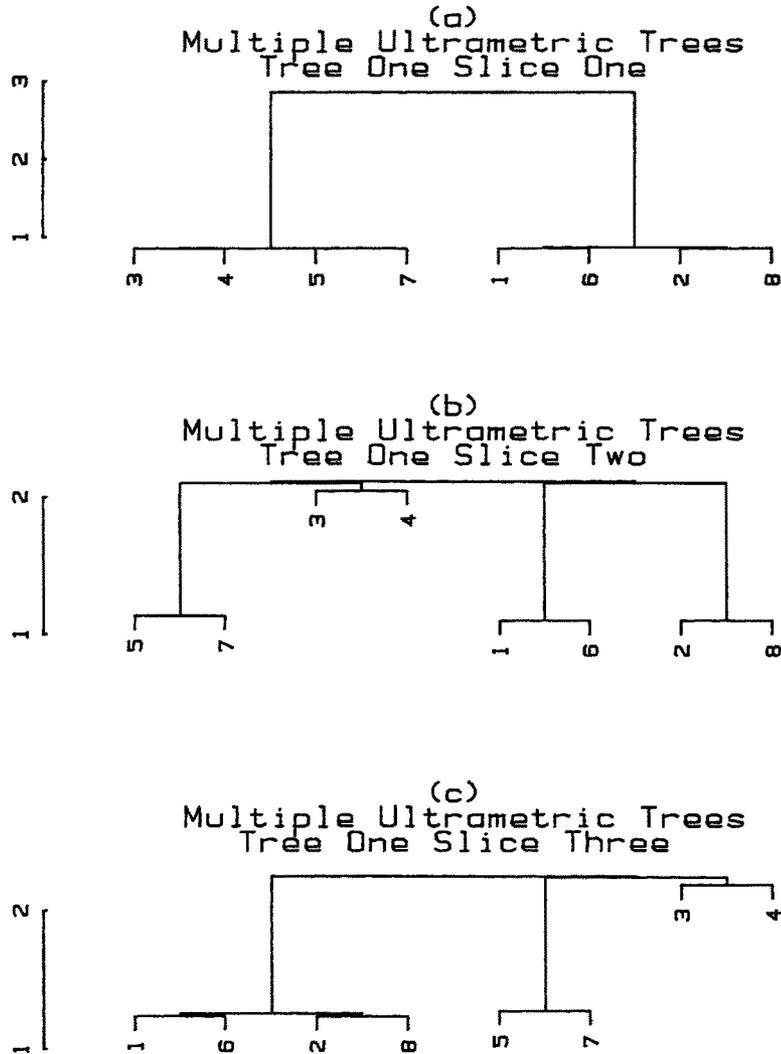
Table 8 presents the predicted D_1 and D_2 arrays. These differ from Δ_1 and Δ_2 in Table 7 by different additive constants. In fact we can arbitrarily add a positive constant to the off-diagonal elements of D_2 to get rid of the negative entries and still not affect the tree topology. Figures 4a-c present the first set of recovered ultrametric trees. Note that these are nearly identical to their counterparts in Figure 1a-c except for node heights and branch lengths because of the additive constant indeterminacy just described. Figures 5a-c present the second set of recovered ultrametric trees after adjusting D_2 for negative entries. These figures also recover the true known topology of Δ_2 (except for an additive constant affecting node heights and branch lengths).

3.4 Multiple Additive Trees

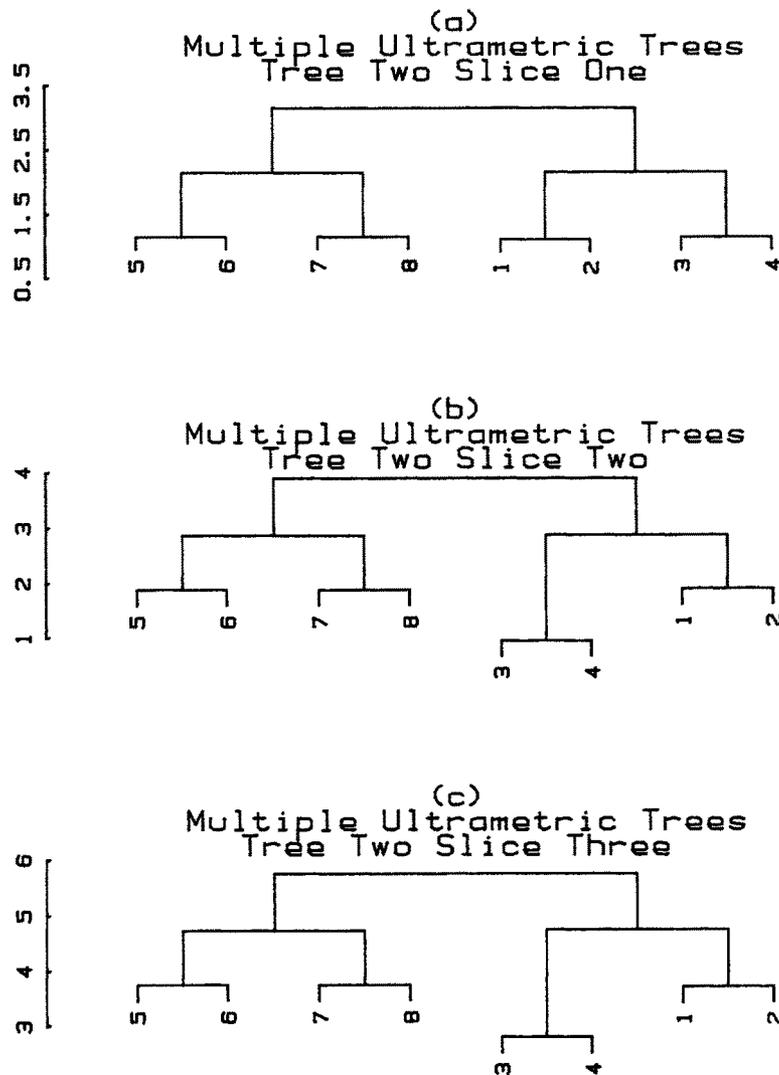
In order to test the recovery power of the IND TREES procedure for the case of multiple additive trees, c_{ij} constants were added to the two ultrametric tree proximities (Δ) in Table 7. Table 9 presents these additive tree constants and the new data Δ . One can define two additive trees as two ultrametric trees plus a set of constants (c_{ij}). Because of root and additive constant indeterminacies alluded to earlier, one should be able to recover the basic topology of both trees except for node heights and branch lengths.

TABLE 8
Recovered Solution for the Two
Ultrametric Trees Example

		D_1									
		.000									
		3.112	.000								
		4.149	4.149	.000							
Slice: 1		4.149	4.149	3.144	.000						
		5.143	5.143	5.142	5.142	.000					
		5.142	5.140	5.143	5.143	3.152	.000				
		5.143	5.143	5.142	5.142	4.144	4.143	.000			
		5.140	5.142	5.143	5.153	4.143	4.141	3.152	.000		
		.000									
		3.903	.000								
		4.880	4.880	.000							
Slice: 2		4.880	4.880	2.956	.000						
		5.897	5.897	5.900	5.900	.000					
		5.901	5.899	5.897	5.897	3.884	.000				
		5.897	5.897	5.900	5.900	4.864	4.870	.000			
		5.899	5.901	5.897	5.897	4.870	4.874	3.884	.000		
		.000									
		5.722	.000								
		6.746	6.746	.000							
Slice: 3		6.746	6.746	4.812	.000						
		7.754	7.754	7.755	7.755	.000					
		7.759	7.751	7.754	7.754	5.755	.000				
		7.754	7.754	7.755	7.755	6.720	6.725	.000			
		7.751	7.759	7.754	7.754	6.725	6.727	5.755	.000		
		.000									
		-1.130	.000								
		.854	.854	.000							
Slice: 1		.854	.854	-1.444	.000						
		.855	.855	-1.142	-1.142	.000					
		-1.143	-1.133	.855	.855	.854	.000				
		.855	.855	-1.142	-1.142	-1.144	.855	.000			
		-1.133	-1.143	.855	.855	.855	-1.137	.854	.000		
		.000									
		.101	.000								
		.114	.114	.000							
Slice: 2		.114	.114	.044	.000						
		.112	.112	.100	.100	.000					
		-.901	.104	.112	.112	.113	.000				
		.112	.112	.100	.100	-.865	.114	.000			
		.014	-.901	.112	.112	.114	.107	.113	.000		
		.000									
		-.737	.000								
		.252	.252	.000							
Slice: 3		.252	.252	.188	.000						
		.251	.251	.245	.245	.000					
		-.759	-.738	.251	.251	.250	.000				
		.251	.251	.245	.245	-.750	.254	.000			
		-.738	-.759	.251	.251	.254	-.738	.250	.000		



Figures 4a-c. First Ultrametric Tree Solution for the Synthetic Multiple Tree Data Example in Table 8.



Figures 5a-c. Second Ultrametric Tree Solution for the Synthetic Multiple Tree Data Example in Table 8.

TABLE 9
 Synthetic Data for Two Additive Trees

ADDITIVE TREE CONSTANTS:

	1	.420	.610	.190	.470	.060	.560	.400	.240
Slice =	2	.110	.990	.730	.230	.250	.920	.800	.880
	3	.870	.540	.750	.680	.970	.720	.260	.340

Δ

	0								
	3.03	0							
	5.61	5.80	0						
Slice: 1	5.89	6.08	2.66	0					
	6.48	6.67	4.25	4.53	0				
	4.98	5.17	6.75	7.03	4.62	0			
	6.82	7.01	4.59	4.87	3.46	5.96	0		
	4.66	4.85	6.43	6.71	5.30	3.80	4.64	0	
	0								
	5.10	0							
	5.84	6.72	0						
Slice: 2	5.34	6.22	3.96	0					
	6.36	7.24	6.98	6.48	0				
	6.03	7.91	7.65	7.15	5.17	0			
	6.91	7.79	7.53	7.03	5.05	6.72	0		
	6.99	6.87	7.61	7.11	6.13	6.80	5.68	0	
	0								
	6.41	0							
	8.62	8.29	0						
Slice: 3	8.55	8.22	6.43	0					
	9.84	9.51	9.72	9.65	0				
	8.59	8.26	9.47	9.40	7.69	0			
	9.13	8.80	9.01	8.94	7.23	7.98	0		
	8.21	7.88	9.09	9.02	8.31	7.06	6.60	0	

The INDTREES procedure converged in five giant iterations, providing a variance accounted-for measure of .9997.

Table 10 presents the recovered solution in terms of D_1 , D_2 , and C . Note that these were not adjusted for negative values. Figures 6a-c present the first set of ultrametric trees recovered. Except for branch lengths and node heights, these are quite similar to their counterparts in Figures 4a-c. Figures 7a-c present the second set of ultrametric trees recovered from D_2 adjusting for negative values. These figures also resemble their counterparts in Figures 5a-c as expected.

TABLE 10
Recovered Solution for the Two Additive
Multiple Tree Example

		D_1							
		.000							
		3.976	.000						
		4.987	4.987	.000					
Slice: 1		4.990	4.989	3.932	.000				
		5.883	5.885	5.885	5.885	.000			
		5.884	5.882	5.886	5.880	3.751	.000		
		5.881	5.882	5.882	5.882	4.777	4.787	.000	
		5.879	5.881	5.884	5.878	4.787	4.781	3.787	.000
		.000							
		5.199	.000						
		6.066	6.061	.000					
Slice: 2		6.066	6.065	4.088	.000				
		7.123	7.130	7.137	7.137	.000			
		7.134	7.140	7.137	7.132	5.048	.000		
		7.117	7.125	7.131	7.129	6.125	6.138	.000	
		7.133	7.129	7.138	7.132	6.135	6.117	5.158	.000
		.000							
		7.373	.000						
		8.294	8.295	.000					
Slice: 3		8.295	8.295	6.432	.000				
		9.050	9.048	9.049	9.049	.000			
		9.064	9.061	9.051	9.051	6.694	.000		
		9.051	9.051	9.050	9.051	7.681	7.677	.000	
		9.062	9.066	9.051	9.050	7.678	7.672	6.668	.000
		.000							
		D_2							
		.000							
		-.991	.000						
		.857	.858	.000					
Slice: 1		.855	.856	-1.260	.000				
		.855	.855	-1.267	-1.312	.000			
		-1.014	-.944	.856	.855	.859	.000		
		.857	.857	-1.265	-1.310	-1.308	.854	.000	
		-.992	-.994	.858	.857	.855	-.995	.856	.000
		.000							
		.120	.000						
		.138	.138	.000					
Slice: 2		.136	.139	.006	.000				
		.126	.126	.005	-.027	.000			
		-.862	.117	.126	.125	.146	.000		
		.133	.132	.009	-.015	-.963	.132	.000	
		.116	-.748	.133	.132	.134	.111	.135	.000
		.000							
		-.892	.000						
		.295	.294	.000					
Slice: 3		.295	.294	.270	.000				
		.280	.280	.271	.266	.000			
		-.896	-.882	.280	.280	.291	.000		
		.278	.278	.271	.266	-.584	.294	.000	
		-.883	-.921	.278	.278	.293	-.868	.294	.000

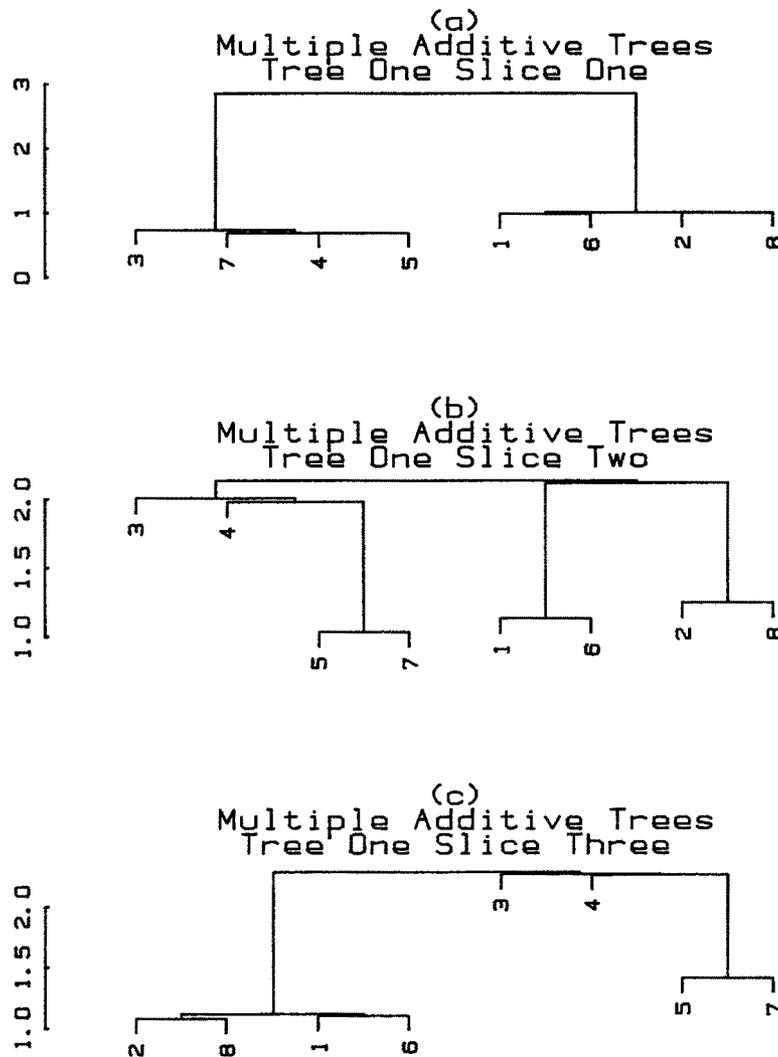
TABLE 10 (Continued)

		C									
		.000									
		.041	.000								
Slice: 1		-.238	-.051	.000							
		.059	.246	-.033	.000						
		-.253	-.067	-.345	-.048	.000					
		.100	.287	.008	.305	-.007	.000				
		.082	.268	-.010	.287	-.025	.328	.000			
		-.228	-.042	-.320	-.023	-.336	.018	.001	.000		
		.000									
		-.200	.000								
		-.366	.493	.000							
Slice: 2		-.855	.003	-.163	.000						
		-.858	.001	-.166	-.655	.000					
		-.248	.611	.444	-.045	-.047	.000				
		-.325	.533	.367	-.122	-.125	.485	.000			
		-.323	.535	.369	-.120	-.123	.487	.410	.000		
			.000								
		-.098	.000								
		.022	-.304	.000							
Slice: 3		-.046	-.372	-.252	.000						
		.548	.222	.342	.274	.000					
		.394	.068	.188	.120	.714	.000				
		-.159	-.484	-.365	-.433	.161	.007	.000			
		.023	-.303	-.184	-.252	.343	.189	-.364	.000		

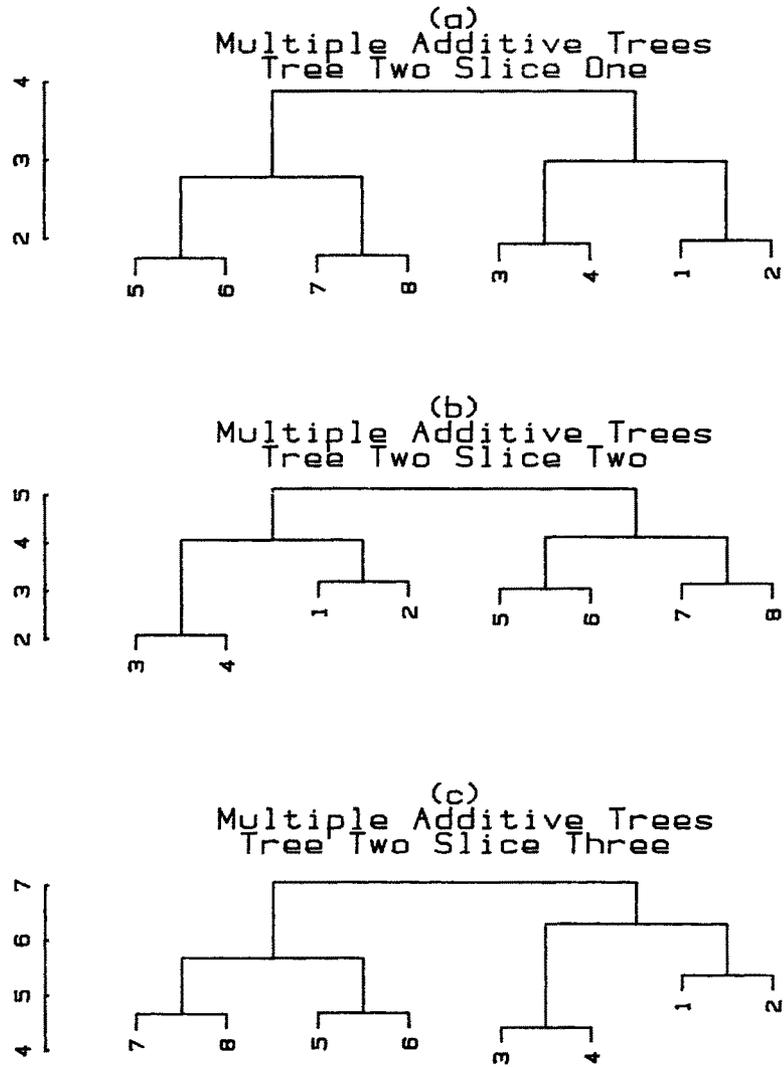
4. Application: Pain Reliever Data

A study was conducted among a sample of 61 MBA students at Fairleigh Dickenson University (DeSarbo and Rao (1983)) to investigate over-the-counter pain reliever usage in remedying three common maladies. The brands tested were: Anacin, Ascriptin, Bayer Aspirin, Bufferin, Cope, Datriil, Excedrin, Hudson Aspirin, Tylenol, and Vanquish. The subjects were given three different maladies: headache, fever, and muscle aches, and were asked to evaluate their intention to use each of the ten brands on a ten-point category scale (1 = least likely to use, 10 = most likely to use) for relieving each of the three maladies. We wished to investigate how such intended usage for the different brands would vary according to how the product would be used. A color photograph with information concerning the manufacturer, cost per 100 tablets, ingredients, and package description/claims were initially shown to the subjects prior and during the evaluation of each drug usage for each situation. (It should be noted that these data were collected prior to the so-called "Tylenol scare," so were not affected by that unfortunate incident.)

A three-way array of dominance data, having dimensions 3 (maladies) \times 61 (subjects) \times 10 (brands), was thus collected and coded. Euclidean distances between brands within each slice (malady) were then constructed



Figures 6a-c. First Ultrametric Tree Solution for the Synthetic Multiple Additive Tree Data Example in Table 10.



Figures 7a-c. Second Ultrametric Tree Solution for the Synthetic Multiple Additive Tree Data Example in Table 10.

TABLE 11
Derived Euclidean Distances for the
Pain Reliever Data

Δ

	0																				
	36.11	0																			
	22.78	39.94	0																		
	22.49	35.67	21.14	0																	
	36.97	21.79	43.41	37.99	0																
Slice: 1	33.91	33.56	36.59	30.13	34.31	0															
	25.01	40.38	30.08	20.29	39.23	34.37	0														
	37.05	25.10	34.31	33.87	30.90	29.55	40.11	0													
	33.68	43.20	31.86	27.73	44.69	26.61	28.25	39.22	0												
	35.49	23.73	37.67	32.69	20.93	33.97	38.00	28.92	40.35	0											
	0																				
	34.49	0																			
	23.39	41.01	0																		
	26.16	38.57	20.24	0																	
	31.61	27.89	38.70	33.14	0																
Slice: 2	35.13	34.69	39.30	34.17	27.11	0															
	28.87	35.37	32.52	27.19	29.21	27.51	0														
	36.13	28.01	33.43	34.35	31.77	29.85	31.81	0													
	34.87	44.67	31.84	26.38	35.72	27.49	28.74	37.1	0												
	34.70	23.15	39.13	34.18	21.93	29.77	29.84	28.98	37.05	0											
	0																				
	35.25	0																			
	24.82	33.59	0																		
	23.56	33.65	23.13	0																	
	33.32	25.17	37.54	35.69	0																
Slice: 3	34.06	32.81	34.04	30.00	30.91	0															
	28.27	37.15	31.38	23.57	31.05	32.53	0														
	34.33	23.45	27.99	30.10	30.00	25.20	35.54	0													
	35.79	40.15	34.50	28.84	40.03	27.76	30.48	35.36	0												
	31.40	24.03	32.95	30.42	20.03	30.57	32.12	29.10	36.01	0											

(δ_{ijl}) producing a 3 (maladies) \times 10 (brands) \times 10 (brands) array of dissimilarities for INDTREES analysis. Table 11 presents this three-way array of dissimilarities.

A single set of ultrametric trees with the same topology was fit to this three-way array. The INDTREES procedure converged in eleven major iterations producing a solution with a variance accounted-for of 60.11%. Figures 8a-c present the derived ultrametric trees fitted to these data for each malady.

Three distinct clusters appear to emerge from the analysis. Cluster one contains Datriil and Tylenol. These two brands are both aspirin substitutes containing 325 mg of acetaminophen and nothing else. Cluster two contains Excedrin, Anacin, Bayer and Bufferin. These four brands are the market share leaders for brands of aspirins. They are perhaps the most heavily advertised of all aspirins. Table 12 presents a market share comparison (Crain 1982) of some major brands which substantiates this.

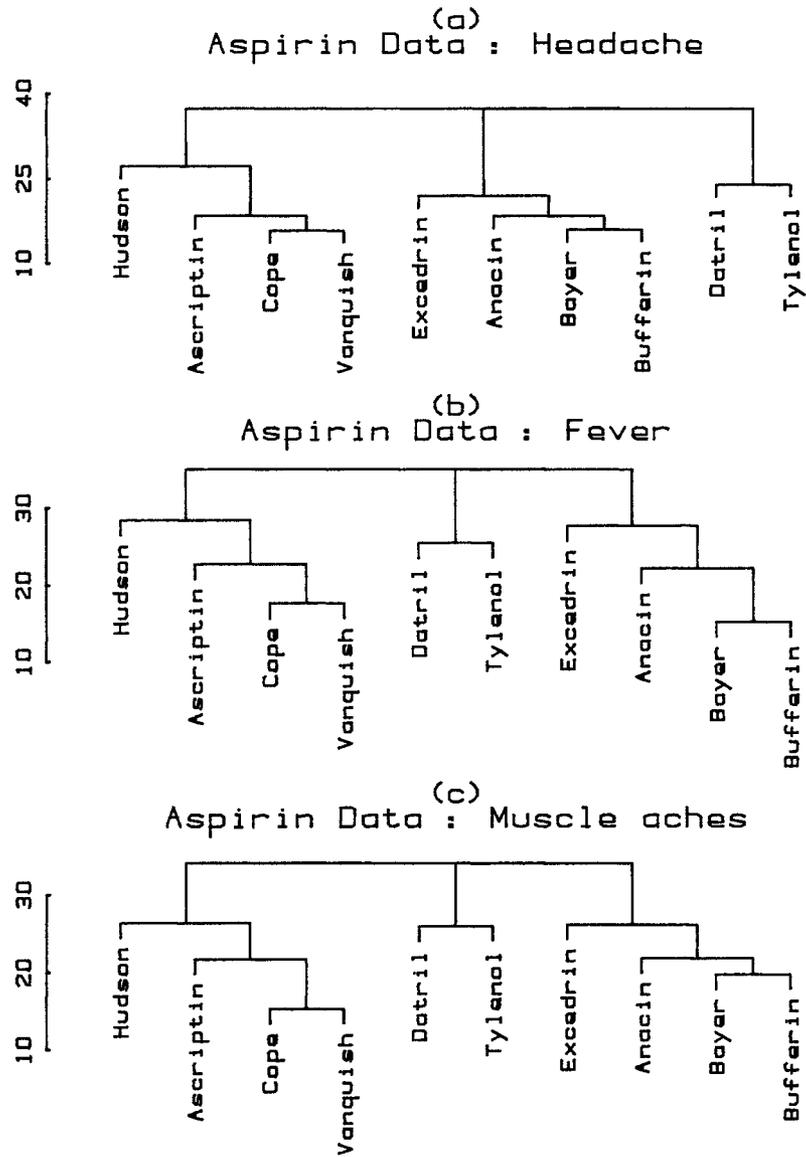
Cluster three contains Hudson, Cope, Ascriptin and Vanquish. These four brands are not as popular as those in clusters one and two. The manufacturers do not spend as much money on advertising on these as others. With the exception of Hudson aspirin, the remaining brands in cluster three contain such antacid ingredients as Aluminum Hydroxide and Magnesium Hydroxide which are intended to avoid the occasional stomach-upsetting side effects of aspirin. (Bufferin describes itself as containing "small amounts of Aluminum Glycinate and Magnesium Carbonate" on its package).

Table 13 displays the means and standard deviations of intended drug use across the three usage occasions. There is higher intended usage for the brands in clusters one and two across all occasions reflecting the higher sales penetration levels of these major brands.

From an examination of the three trees, respondents' intended usage in relieving headache pain is most dissimilar among the three clusters of brands, indicating that subjects distinguish product differentiation claims most in situations involving headaches. Cluster two brands appear to be tightly clustered together. Ascriptin, Cope, and Vanquish are tightly clustered together in cluster three, but Hudson joins the cluster at a much higher distance (α) level.

For relieving fever discomfort, intended usage patterns are not as dissimilar among the three clusters as in the case of headache. Here clusters two and three are not nearly so compact as in the case of headache remedy. In cluster two, Bayer and Bufferin usage is most similar, but Anacin and Excedrin join the cluster at larger α levels. This result is also displayed in Table 13 where intended usage for Bayer and Bufferin is highest for this cluster in relieving fever discomfort. Similarly, in cluster three, Cope and Vanquish usage is most similar, but Ascriptin and Hudson join the cluster at large α levels.

For relieving muscle aches, as in the case of fever remedy, intended usage patterns are not so dissimilar as in the case of headache remedy. While cluster two brands remain reasonably compact (as in the case of headaches), cluster three brands display a tree structure more like that seen in fever remedy, i.e., where Hudson and Ascriptin are not so tightly clustered with Cope and Vanquish. As Table 13 demonstrates, Hudson



Figures 8a-c. Single Set of Ultrametric Trees Fit to the Pain Reliever Dissimilarities.

TABLE 12

Market Share Comparisons for Major Brands
of Over-the-Counter Pain Relievers

Brand*	Estimated Market Share
Tylenol	46.7%
Anacin	15.2
Excedrin	10.1
Bufferin	9.6
Bayer	8.6
Nonaspirin	50.5%
Aspirin	49.5

* Taken from Crain (1982, p. 1).

TABLE 13

Means and Standard Deviations of Intended
Drug Usage Across the Three Maladies

Brand	Headache		Fever		Muscle Ache	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
Anacin	6.77	2.79	6.56	2.75	6.52	2.88
Ascriptin	3.69	2.60	3.80	2.76	3.85	2.67
Bayer	7.54	2.61	7.59	2.73	6.62	2.80
Bufferin	6.71	2.23	6.83	2.59	6.48	2.72
Cope	3.44	2.74	4.21	2.74	3.71	2.67
Datril	5.72	2.85	5.15	2.95	5.20	2.89
Excedrin	6.90	2.84	5.52	2.68	6.14	3.14
Hudson	4.83	3.10	5.20	3.27	5.07	2.98
Tylenol	7.53	2.73	7.00	2.82	6.62	3.01
Vanquish	3.87	2.54	3.54	2.14	3.92	2.44

intended usage is highest across usage occasions than that of any of the other brands in cluster three — perhaps because of the fact that Hudson is 100% aspirin, without other additives. This fact may account for its lack of overall similarity with the other three brands in cluster three. It is also interesting to note how Ascriptin intended usage patterns vary when compared to Cope and Vanquish across usage occasions. For headache, the three brands are joined at a low level in the tree, while for fever and muscle aches, Ascriptin joins at a higher level.

5. Discussion

We have presented a methodological description of the INDTREES model and algorithm. Artificial data testing on a number of synthetic data sets provided some evidence concerning the ability of the INDTREES method to recover known structure. However, more rigorous testing is required. Finally, an example was presented where INDTREES rendered some insight into the structure of real three-way proximity data, derived from judgments on likelihood of usage of various analgesics in situations defined in terms of different types of maladies.

There are a number of research areas that can be pursued with this INDTREES procedure. One important area concerns the generalization of INDTREES to accommodate hybrid models where *both* tree structures and spatial components are estimated. The proposed model can be formally expressed as:

$$\Delta \cong \mathbf{D}_1 + \mathbf{D}_2 + \dots + \mathbf{D}_M + \mathbf{D}_{E_r}^2, \quad (36)$$

where \mathbf{D}_1 through \mathbf{D}_M are distance matrices arising from tree structures (either ultrametric or additive trees) and $\mathbf{D}_{E_r}^2$ is a three-way array of squared distances arising from an r -dimensional three-way spatial representation (say, an INDSCAL type model entailing different weighted Euclidean metrics defined on a common r dimensional vector space). Carroll and DeSarbo are currently working on such a generalization.

Another area of future research is the generalization of INDTREES to accommodate asymmetric three-way proximity data where $\Delta_{ij'j}$ may not be equal to $\Delta_{ijj'}$, as in the case of either confusions data or data where the row and column objects are different. This would amount to a three-way generalization of the work done by DeSoete, DeSarbo, Furnas, and Carroll (1983).

Finally, research should attempt to generalize INDTREES to accommodate merely ordinal proximities. Carroll has demonstrated typical degeneracies involved in attempting to fit such discrete models to such data via standard nonmetric procedures entailing totally general monotone transformations and a stress measure as the objective function. Conceivably, an

alternative formulation could be developed where such degeneracies would not occur. One such approach under consideration is that of using the monotone splines successfully applied elsewhere by Winsberg and Ramsay (1983) as a compromise between the one extreme of often excessively "strong" metric assumptions (e.g., of linear functions relating dissimilarities and distances) and the other of the very general, but sometimes excessively weak, assumption of otherwise unrestricted monotone functions typically assumed in nonmetric approaches.

Appendix I

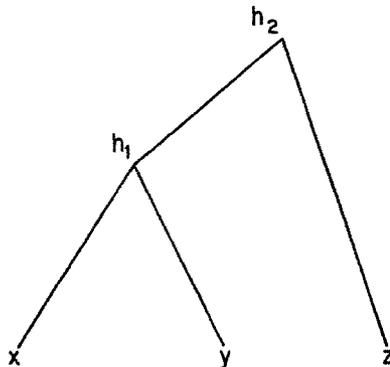
Proof that:

$P_1 = P_2 = 0$ is a necessary and sufficient condition for distance matrices to be consistent with ultrametric trees having the same topology.

Since one of the reviewers felt it was not obvious that the condition that the two penalty functions equaling zero is equivalent to the distance matrices all being consistent with the same hierarchical tree, but with ultrametric defined via different height values, we now prove this fact.

Necessity:

If all distance matrices are ultrametrics on the same tree, the ultrametric tree induced for every triple of points is, upon appropriate labeling of the points, isomorphic to that shown in Figure AI-1



while ultrametric distances are defined by assigning values to h_1 and h_2 such that $h_2 \geq h_1$, then defining

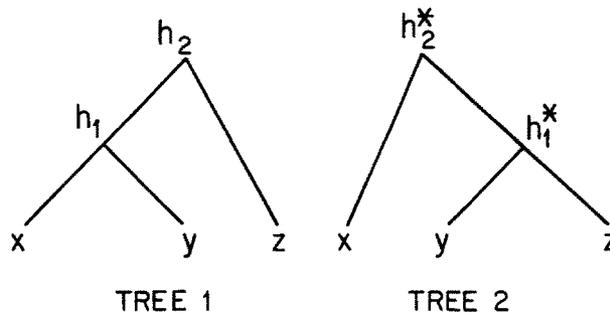
$$\begin{aligned}d_{xy} &= h_1 \\d_{xz} &= d_{yz} = h_2 .\end{aligned}$$

Since this definition of ultrametric distance must hold for all the distance matrices, only h_1 and h_2 (but with $h_2 \geq h_1$) varying, it follows that $d_{xy} \leq d_{xz} = d_{yz}$ for all distance matrices. Thus the pair of points having the smallest distance for each triple of points must be the same for all the distance matrices. (Thus implying $P_2 = 0$). Conversely, the two largest distances of each triple must also be the same for all matrices. This, combined with the fact that each distance matrix must satisfy the ultrametric conditions, implies $P_1 = 0$.

Sufficiency:

It is straightforward that $P_1 = 0$ and $P_2 = 0$ together guarantee that each distance matrix satisfies the ultrametric conditions, so each is consistent with *some* ultrametric tree. We need only now prove that the topologies of these trees all be mutually compatible.

To do so let us assume the contrary, i.e. that the topologies are not mutually compatible. Then at least two of the distance matrices must be generated from different trees. Call these matrices D_1 and D_2 . The two trees must differ in the topology induced on at least one triple of points. Let that triple be x , y and z . Then, under appropriate labeling, the two trees induced on x , y and z are isomorphic to those shown in Figure AI-2 with $h_2 > h_1$ and $h_2^* > h_1^*$.



It follows that, in D_1 ,

$$d_{xy} < d_{xz} = d_{yz} ,$$

while in D_2

$$d_{yz} < d_{xy} = d_{xz} ,$$

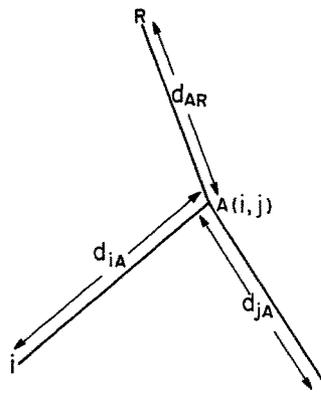
which implies $P_2 > 0$. .

Appendix II

Proof:

A path length or additive tree can always be decomposed into an ultrametric tree plus a "star" or "bush".

Given a path length tree, let us arbitrarily "root" it at one of its internal nodes, R . Then we may define the distance, d_{iR} , from any terminal node i to R as the sum of lengths of the branches in the (unique) path from i to R . The distance between any two terminal nodes i and j is the sum of the lengths of the branches in the (unique) path from i to j . If we focus on the tree induced on just the four nodes i , j , $A(i, j)$ and R , where $A(i, j)$ is the "least common ancestor" node in the tree (now rooted at R , which induces a partial order on the nodes, and thus serves to define such least common ancestor nodes for each pair of terminal nodes), this tree is isomorphic to the tree shown in Figure AII-1



From this tree, it is clear that $d_{ij} = d_{iA} + d_{jA}$, while $d_{iR} = d_{iA} + d_{AR}$ and $d_{jR} = d_{jA} + d_{AR}$. Therefore $d_{ij} - d_{iR} - d_{jR} = -2d_{AR}$. Define c as: $c = \max (d_{AR})$ where $A \in \{\text{internal nodes}\}$ and $c_i = d_{iR} - c$. Then $d_{ij} - c_i - c_j = 2(c - d_{AR}) > 0$, so that $d_{ij} = u_{ij} + c_i + c_j$, where $u_{ij} = c - d_{A(i,j)R}$ is a nonnegative number that is a function of only the least common ancestor node of i and j , and which satisfies the partial order necessary for heights defining an ultrametric. Therefore u_{ij} is an ultrametric.

Consider now the term:

$$c_{ij} = c_i + c_j .$$

This is precisely the metric of a “star” or “bush” tree with a single internal node, where c_i is the length of the branch from terminal node i to that single internal node.

This, of course, does not guarantee that $c_i \geq 0$ for all i . However, one can add a sufficiently large constant to the c 's to guarantee this, and subtract this constant from the u_{ij} 's. This may lead to u 's that satisfy the ultrametric inequality for all distinct triples of points, but not positivity. In a very central sense, however, the ultrametric condition is defined only up to an additive constant, (indeed, only up to the much broader class of ordinal transformations of scale) so that negativity need not be taken as contradicting the ultrametric condition, broadly defined. It may be the case, however, that for some path length or additive trees there is no placement of the root that will result in a decomposition such that both the ultrametric and “bush” component are nonnegative. (In such cases there is a fundamental “tradeoff” — one may choose to make the ultrametric component positive but allow some negative c_i 's, or make the c_i 's positive but allow some negative values in the ultrametric component.)

One fact that should be obvious from this proof is that, in any case, this decomposition is *not* unique. The root can be placed at any node (including a terminal node) or even between any two nodes (by inserting an additional node and defining branch lengths from that new node to the two nodes between which it is placed so that the sum of those branch lengths equals the length of the single branch connecting those two nodes in the original tree). Thus there are in fact an infinite number of decompositions of an additive tree into a sum of an ultrametric tree and a “bush”. Viewed from the point of view of the topology of the ultrametric tree component (treating trees with different roots as topologically distinct for this purpose), there are $3n - 5$ different ways of generating topologically different ultrametric trees corresponding to a single additive or path length tree, as discussed in the text.

References

- ARABIE, P., and CARROLL, J.D. (1980), "MAPCLUS: A Mathematical Programming Approach to Fitting the ADCLUS Model," *Psychometrika*, 45, 211-235.
- BUNEMANN, P.A. (1974), "A Note on the Metric Properties of Trees," *Journal of Combinatorial Theory*, 17, 48-50.
- CARROLL, J.D. (1975), "Handout for Models for Individual Differences in Similarities Judgments." (Paper presented at Mathematical Psychology Meetings, 25-27 August Purdue University, West Lafayette, Indiana.)
- CARROLL, J.D. (1976), "Spatial, Non-Spatial and Hybrid Models for Scaling," *Psychometrika*, 41, 439-463.
- CARROLL, J.D., and ARABIE, P. (1979), "INDCLUS: A Three-Way Approach to Clustering." (Paper presented at meeting of Psychometric Society, Monterey, California.)
- CARROLL, J.D., and ARABIE, P. (1983), "INDCLUS: An Individual Differences Generalization of the ADCLUS Model and the MAPCLUS Algorithm," *Psychometrika*, 48, 157-169.
- CARROLL, J.D., and CHANG, J.J. (1970), "Analysis of Individual Differences in Multidimensional Scaling via an N-Way Generalization of Eckart-Young Decomposition," *Psychometrika*, 35, 283-319.
- 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," *Proceedings of the 81st Annual Convention of the American Psychological Association*, 8, 1097-1098.
- 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 U.S.-Japan Seminar of Multidimensional Scaling, 20-24 August, University of California at San Diego, La Jolla, California.)
- CARROLL, J.D., and PRUZANSKY, S. (1980), "Discrete and Hybrid Scaling Models," in *Similarity and Choice*, eds. E.D. Lantermann and H. Feger, Bern: Hans Huber.
- CARROLL, J.D., and WISH, M. (1974), "Models and Methods for Three-Way Multidimensional Scaling," in *Contemporary Developments in Mathematical Psychology (Vol. II)*, eds. D.H. Krantz, R.C. Atkinson, R.D. Luce, and P. Suppes, San Francisco: Freeman.
- COOPER, L., and STEINBERG, D. (1970), *Introduction to Methods of Optimization*, Philadelphia: W. B. Saunders Co.
- CRAIN, R., (ed.) (1982), *Advertising Age*, Chicago: Crain Communications.
- CUNNINGHAM, J.P. (1974), "Finding the Optimal Tree Realization of a Proximity Matrix." (Paper presented at the Mathematical Psychology Meetings, August, Ann Arbor, Michigan.)
- CUNNINGHAM, J.P. (1978), "Free Trees and Bidirectional Trees as a Representation 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.
- DeSARBO, W.S., and RAO, V.R. (1983), "A Constrained Unfolding Model for Product Positioning and Market Segmentation," unpublished paper, Bell Laboratories, Murray Hill, NJ.
- DeSOETE, G. (1983a), "Algorithms for Constructing Least-Squares Ultrametric and Additive Tree Representations of Dissimilarities Data," unpublished dissertation (in Dutch), University of Ghent, Belgium.
- DeSOETE, G. (1983b), "A Least-Squares Algorithm for Fitting Trees to Proximity Data," *Psychometrika*, 48, 621-626.
- DeSOETE, G., DeSARBO, W.S., FURNAS, G.W., and CARROLL, J.D. (1983), "The Estimation of Ultrametric and Path Length Trees from Rectangular Proximity Data," unpublished paper, Bell Laboratories, Murray Hill, NJ.

- FARRIS, J.S. (1972), "Estimating Phylogenetic Trees from Distance Matrices," *American Naturalist*, 106, 645-668.
- FLETCHER, R., and REEVES, C.M. (1964), "Function Minimization by Conjugate Gradients," *Computer Journal*, 7, 149-154.
- GILL, P.E., MURRAY, W., and WRIGHT, M.H. (1981), *Practical Optimization*, New York: Academic Press.
- HARSHMAN, R.A. (1972), "PARAFAC2: Mathematical and Technical Notes," *Working Papers in Phonetics*, UCLA, 22, 30-47.
- HARTIGAN, J.A. (1967), "Representation of Similarity Matrices by Trees," *Journal of the American Statistical Association*, 62, 1140-1158.
- HARTIGAN, J.A. (1975), *Clustering Algorithms*, New York: Wiley.
- HIMMELBLAU, D.M. (1972), *Applied Nonlinear Programming*, New York: McGraw-Hill.
- JOHNSON, S.C. (1967), "Hierarchical Clustering Schemes," *Psychometrika*, 32, 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), "A Taxonomy of Some Principal Types of Data and of Multidimensional Methods for Their Analysis," in *Multidimensional Scaling: Theory and Applications in the Behavioral Sciences*, Vol. 1: *Theory*, eds. R.N. Shepard, A.K. Romney and S. Nerlove, New York: Seminar Press.
- SHEPARD, R.N., and ARABIE, P. (1979), "Additive Clustering: Representation of Similarities as Combination of Discrete Overlapping Properties," *Psychological Review*, 86, 87-123.
- TAKANE, Y., YOUNG, F.W., and DeLEEUW, J. (1977), "Nonmetric Individual Differences Multidimensional Scaling: An Alternating Least-Squares Method with Optimal Scaling Features," *Psychometrika*, 42, 7-67.
- TUCKER, L.R. (1964), "The Extension of Factor Analysis to Three-Dimensional Matrices," in *Contributions to Mathematical Psychology*, eds. N. Frederiksen and H. Gulliksen, New York: Holt, Rinehart and Winston.
- TUCKER, L.R. (1972), "Relations Between Multidimensional Scaling and Three-Mode Factor Analysis," *Psychometrika*, 32, 3-27.
- WINSBERG, S., and RAMSAY, J.O. (1983), "Monotone Spline Transformations for Dimension Reduction," *Psychometrika*, 48, 575-596.
- WOLD, H. (1966), "Estimation of Principal Components and Related Models by Iterative Least Squares," in *Multivariate Analysis*, ed. P.R. Krishnaiah, New York: Academic Press.