PSYCHOMETRIKA—VOL. 48, NO. 4.
DECEMBER, 1983
NOTES AND COMMENTS

A LEAST SQUARES ALGORITHM FOR FITTING ADDITIVE TREES TO PROXIMITY DATA

GEERT DE SOETE

UNIVERSITY OF GHENT, BELGIUM

A least squares algorithm for fitting additive trees to proximity data is described. The algorithm uses a penalty function to enforce the four point condition on the estimated path length distances. The algorithm is evaluated in a small Monte Carlo study. Finally, an illustrative application is presented.

Key words: tree structures, clustering, proximity data.

Additive trees have proved to be a valuable alternative to multidimensional scaling for representing proximity data [Sattath & Tversky, 1977]. Especially when the stimuli are conceptual rather than perceptual, an additive tree representation can be very useful [cf. Pruzansky, Tversky, & Carroll, 1982].

A tree is a connected graph where every pair of nodes is connected by a unique path. In an additive tree a nonnegative weight is attached to each link such that the distance between any pair of nodes is the sum of the weights associated with the links that connect the two nodes. Alternative names for additive trees are path length trees [Carroll, 1976] and weighted free trees [Cunningham, 1978].

Let δ be a nonnegative symmetric dissimilarity measure defined on a finite set of n objects. Then the n objects can be represented by the n terminal nodes of an additive tree whenever $\Delta = \{\delta_{ij} | i < j\}$ satisfies

$$\delta_{ij} + \delta_{kl} \leq \max (\delta_{ik} + \delta_{jl}, \delta_{il} + \delta_{jk}) \tag{1}$$

for all i, j, k and l. Or equivalently, whenever the two largest of $(\delta_{ij} + \delta_{kl})$, $(\delta_{ik} + \delta_{jl})$, and $(\delta_{il} + \delta_{jk})$ are equal for all i, j, k, and l. This condition, which is often referred to as the additive inequality or the four point condition, is both necessary and sufficient for the existence of a unique additive tree representation of Δ [cf. e.g., Dobson, 1974]. When Δ satisfies (1) perfectly, the additive tree representation can be constructed quite easily. As remarked by Sattath and Tversky [1977] every proof of the sufficiency of (1) provides a method for constructing the tree. With fallible data, however, an algorithm is needed that finds an additive tree with path length distances $D = \{d_{ij} | i < j\}$ that optimally approximate Δ. Cunningham [1978] suggested to use a least squares criterion. This means that path length distances are sought which minimize

$$L = \sum_{i<j} \sum (\delta_{ij} - d_{ij})^2. \tag{2}$$

Cunningham [1978] presented an algorithm that is meant to minimize (2). His procedure is based on the assumption that for all i, j, k, and l the optimal path length

The author is "Aspirant" of the Belgian "Nationaal Fonds voor Wetenschappelijk Onderzoek". The author is indebted to Professor J. Hoste for providing computer facilities at the Institute of Nuclear Sciences at Ghent.

Requests for reprints should be sent to Geert De Soete, Department of Psychology, University of Ghent, Henri Dunantlaan 2, B-9000 Ghent, Belgium.

0033-3123/83/1200-4063$0050/0

distances satisfy

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

whenever

$$\delta_{ij} + \delta_{kl} \leq \delta_{ik} + \delta_{jl} \leq \delta_{il} + \delta_{jk}.$$

This clearly need not always be the case, especially when the data contain a lot of noise. Therefore, the path length distances estimated with Cunningham's procedure are not necessarily optimal in a least squares sense.

Carroll and Pruzansky [Note 1, 1980] proposed an alternating least squares algorithm that is based on the fact that any set of path length distances can be decomposed into a set of distances that satisfy the ultrametric inequality and n additive constants. In the next section we propose a more straightforward algorithm for obtaining path length distances that minimize (2).

A New Algorithm

In order to simplify the notation, it is assumed throughout this section that the data are normalized such that

$$\sum_{i<j}\sum (\delta_{ij} - \bar{\delta})^2 = 1,$$

where

$$\bar{\delta} = \frac{2}{n(n-1)} \sum_{i<j}\sum \delta_{ij}.$$

We want to find a set of distances D which are closest to the data in least squares sense and which satisfy the four point condition perfectly. This constrained optimization problem is solved by sequentially minimizing the unconstrained function

$$F(D, r) = L(D) + rP(D) \tag{3}$$

for an increasing sequence of values of r. The first component of $F(D, r)$ is given in (2) and measures the departure from the data while the second part, $P(D)$, is a penalty function that expresses how strongly D violates the four point condition. $P(D)$ is defined as

$$P(D) = \sum_{\Omega} (d_{ik} + d_{jl} - d_{il} - d_{jk})^2$$

where

$$\Omega = \{(i, j, k, l) \mid i, j, k, l \text{ distinct and } d_{ij} + d_{kl} \leq \min(d_{ik} + d_{jl}, d_{il} + d_{jk})\}.$$

Using q as the iteration index, the basic algorithm for obtaining least squares estimates of the path length distances is as follows:

(i) Initialize: $q = 1$,

$$D^{(0)} = \{\delta_{ij} + \varepsilon_{ij} \mid i < j\} \qquad \text{where} \qquad \varepsilon_{ij} \sim N(0, \sigma_\varepsilon^2), \qquad r^{(1)} = \frac{L(D^{(0)})}{P(D^{(0)})}.$$

(ii) Minimize $F(D, r^{(q)})$ starting from $D^{(q-1)}$ to obtain $D^{(q)}$.

(iii) Test for convergence: If

$$\left[\sum_{i<j}\sum (d_{ij}^{(q)} - d_{ij}^{(q-1)})^2\right]^{1/2}$$

is less than some small constant stop, otherwise continue.

(iv) Update r: $r^{(q+1)} = 10 \times r^{(q)}$.
Increment q and go back to step (ii).

This procedure converges usually within five to six major iterations. Since the number of unknowns can be quite large, we decided to use Powell's [1977] conjugate gradient procedure with automatic restarts to minimize $F(D, r^{(q)})$ in phase (ii). This method requires only the first-order partial derivatives of $F(D, r)$ with respect to the elements in D. These are given below

$$\frac{\partial F}{\partial d_{st}} = -2(\delta_{st} - d_{st}) + 2r \sum_{\Omega} \{(e_{st,ik} + e_{st,jl} - e_{st,il} - e_{st,jk})(d_{ik} + d_{jl} - d_{il} - d_{jk})\}.$$

where the indicator variable $e_{st,ij}$ is defined by

$$e_{st,ij} = \begin{cases} 1 & \text{when } s = i \text{ and } t = j. \\ 0 & \text{otherwise.} \end{cases}$$

Once least squares estimates of the path length distances are obtained by the procedure outlined above, the additive tree can be constructed in a straightforward way [cf. e.g., Cunningham, 1978; Dobson, 1974].

Monte Carlo Evaluation

In order to evaluate the performance of the algorithm described in the previous section, a Monte Carlo study was undertaken using error-perturbed path length distances as input data. Two factors were factorially combined: the set size and the amount of error added to the true path length distances. Following Pruzansky et al. [1982] three different set sizes (n) were used: 12, 24, and 36.

Error-perturbed path length distances were generated by adding a random central normal deviate with variance σ_e^2 to the unit variance normalized path length distances between the n terminal nodes of a random additive tree. The same levels of σ_e^2 as in Pruzansky et al. [1982] were used: 0.00, 0.25, and 0.50. In order to construct a random additive tree with n terminal nodes, a weight randomly sampled from a uniform distribution on the unit interval was added to each link of a random binary tree with n terminal nodes which was constructed by uniform subsampling from the infinite binary tree (for more details on constructing random binary trees, see Furnas, Note 2).

There were 11 replications per condition and for each data set a new random additive tree was generated. Each of the $3 \times 3 \times 11 = 99$ data sets was analyzed using 10 different sets of initial parameter estimates, so that a total of 990 least squares additive tree analyses were performed. The initial parameter estimates were constructed by adding a random normal deviate with zero mean and variance $s_\delta^2/3$ to the input data, where s_δ^2 denotes the variance of the input data.

It was found that for each data set, ten exactly identical solutions were obtained. This suggests that the algorithm is relatively stable and does not suffer from serious local minima problems. Note, however, that the starting configurations were not completely random, but that they were obtained by randomly "shaking" the data.

For each data set the *squared* product-moment correlation between the true and the derived path length distances was calculated as an index of metric recovery. Table 1 lists per condition the median metric recovery as well as the interquartile distance. As can be inferred from the table, the metric recovery increases as the number of stimuli increases, while it decreases when σ_e^2 increases.

TABLE 1

Summary of Monte Carlo Results on Metric Recovery

	$\sigma_e^2 = 0.00$	$\sigma_e^2 = 0.25$	$\sigma_e^2 = 0.50$
n = 12	1.000[a]	0.953	0.906
	(0.000)[b]	(0.010)	(0.083)
n = 24	1.000	0.979	0.955
	(0.000)	(0.009)	(0.017)
n = 36	1.000	0.984	0.961
	(0.000)	(0.003)	(0.011)

[a]Median metric recovery

[b]Interquartile distance of metric recovery

Although Pruzansky et al. [1982] utilized a slightly different method for generating error-perturbed path length distances, it is instructive to compare Table 1 with the first table in Appendix A of Pruzansky et al. [1982] which presents results about the metric recovery of ADDTREE [Sattath & Tversky, 1977] for the same combinations of n and σ_e^2. Comparing the two tables reveals that the present algorithm outperforms ADDTREE in terms of metric recovery whenever $\sigma_e^2 > 0$. Moreover, the larger the set size, the bigger the difference.

Illustrative Application

Kuennapas and Janson [1969] obtained from 57 subjects similarity judgments between all lower-case Swedish letters. A least squares additive tree analysis was performed on the symmetrized average ratings. In order to be able to compare our results with the ADDTREE solution reported by Sattath and Tversky [1977], the three special letters å, ä and ö were omitted. The least squares additive tree accounts for 77% of the variance of the data. The tree is displayed in parallel form in Figure 1. In this form, the distance between any pair of nodes is the sum of the horizontal links that connect them. Just like in the ADDTREE analysis, five major clusters emerge: the curved and circular letters (o, c, e, a), the curved and tailed letters (g, p, q, d, b), the arched letters (m, n, h, u), the angular letters (s, z, y, v, x), and the vertical letters (r, l, t, f, j, i, k). Although the ADDTREE solution presented in Sattath and Tversky [1977] gives approximately an equally good account of the data (76% variance accounted for), there are some interesting qualitative differences between the two trees. The most remarkable one is that in Figure 1 k is grouped together with the vertical letters, whereas in the ADDTREE solution k is classified in the cluster of the angular letters.

Discussion

In this paper an iterative algorithm for constructing a least squares additive tree representation of a set of dissimilarity data was presented. The algorithm uses a penalty function to enforce the four point condition on the estimated path length distances. Con-

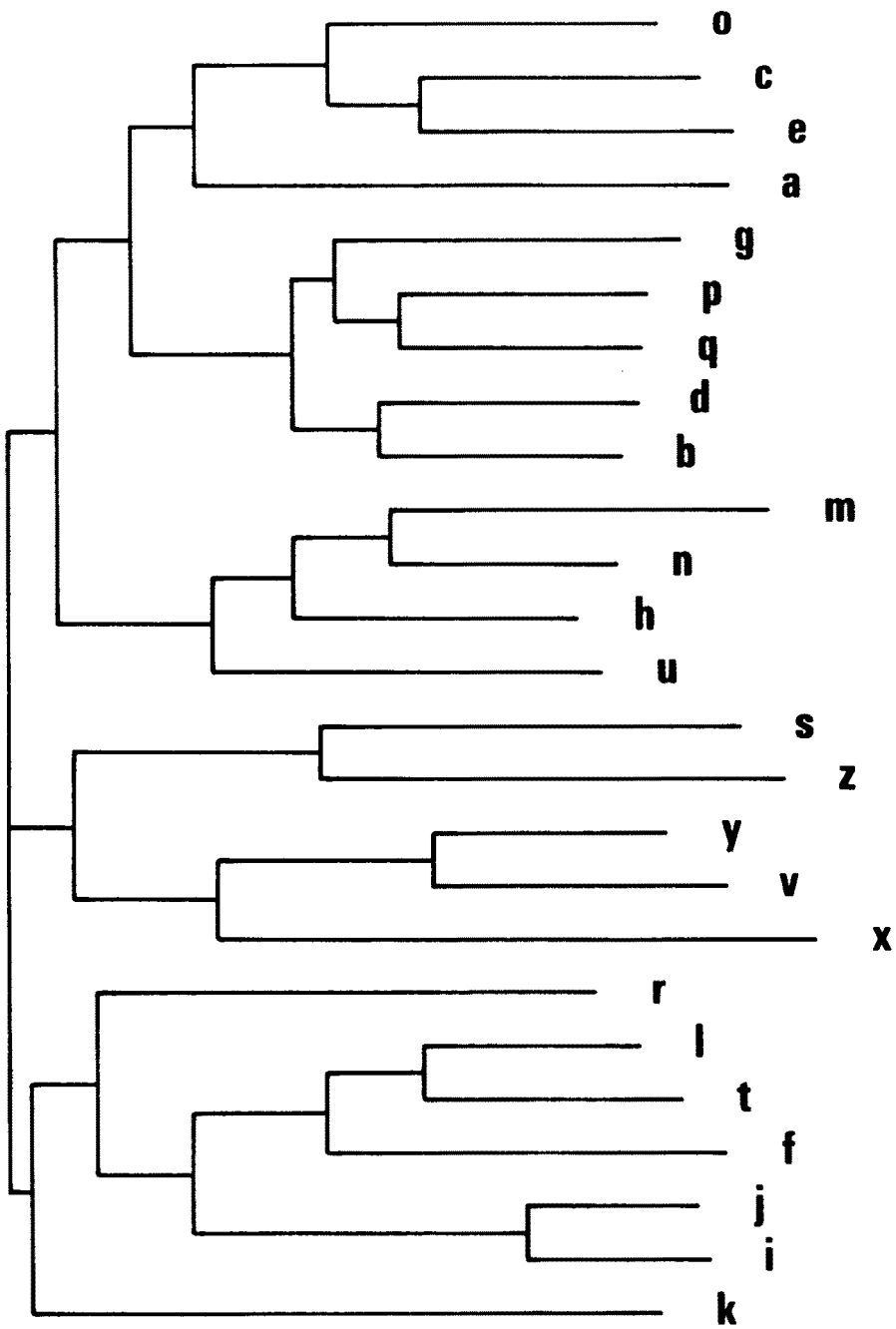

FIGURE 1.
Least squares additive tree representation of the Kuennapas and Janson [1969] data.

ceptually, this approach is closely related to Carroll and Pruzansky's [Note 1, 1980] method for fitting ultrametric trees to proximity data. Numerically, the present method differs from the Carroll and Pruzansky algorithm in that a more stable nonlinear minimization procedure has been used. As evidenced by the Monte Carlo study, the present algorithm is quite insensitive to the choice of the initial parameter estimates.

Contrary to Cunningham's [1978] procedure, the present algorithm is iterative. However, as has been argued in the first section, Cunningham's method does not always yield a least squares solution. Moreover, his method requires the inversion of a symmetric matrix of order $\binom{n}{4}$. This limits its applicability to fairly small data sets. The present algorithm, on the other hand, can analyze dissimilarity data between and up to fifty stimuli without much problem.

REFERENCES NOTES

1. Carroll, J. D., & Pruzansky, S. *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 Theory, Methods, and Applications of Multidimensional Scaling and Related Techniques, San Diego, August 1975.
2. Furnas, G. W. *The construction of random, terminally labeled, binary trees.* Unpublished paper, Bell Laboratories, Murray Hill, New Jersey, 1981.

REFERENCES

Carroll, J. D. Spatial, nonspatial and hybrid models for scaling. *Psychometrika*, 1976, *41*, 439–463.

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

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

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

Kuennapas, T., & Janson, A. J. Multidimensional similarity of letters. *Perceptual &Motor Skills*, 1969, *28*, 3–12.

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

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

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

Manuscript received 12/18/81
First revision received 11/29/82
Final version received 5/13/83