PSYCHOMETRIKA—VOL. 45, NO. 2.
JUNE, 1980

MAPCLUS: A MATHEMATICAL PROGRAMMING APPROACH TO FITTING THE ADCLUS MODEL

PHIPPS ARABIE

UNIVERSITY OF MINNESOTA AND BELL LABORATORIES

J. DOUGLAS CARROLL

BELL LABORATORIES

We present a new algorithm, MAPCLUS (*MA*thematical *P*rogramming *CLUS*tering), for fitting the Shepard-Arabie ADCLUS (for *AD*ditive *CLUS*tering) model. MAPCLUS utilizes an alternating least squares method combined with a mathematical programming optimization procedure based on a penalty function approach, to impose discrete (0,1) constraints on parameters defining cluster membership. This procedure is supplemented by several other numerical techniques (notably a heuristically based combinatorial optimization procedure) to provide an efficient general-purpose computer implemented algorithm for obtaining ADCLUS representations. MAPCLUS is illustrated with an application to one of the examples given by Shepard and Arabie using the older ADCLUS procedure. The MAPCLUS solution uses half as many clusters to achieve nearly the same level of goodness-of-fit. Finally, we consider an extension of the present approach to fitting a three-way generalization of the ADCLUS model, called INDCLUS (*IN*dividual *D*ifferences *CLUS*tering).

Key words: additive clustering, nonhierarchical clustering, alternating least squares.

Shepard and Arabie [1979] have recently given a detailed exposition of their ADCLUS (for "additive clustering") model, which represents interstimulus proximities as combinations of discrete overlapping properties. More concretely, an ADCLUS representation consists of a set of m (possibly overlapping) subsets or clusters, each having an associated numerical weight, w_k (where $k = 1, \cdots, m$). For any pair of stimuli, the predicted similarity is simply the sum of the weights of those subsets containing the given pair of stimuli. Shepard and Arabie described the ADCLUS *model* at considerable length and also provided illustrative applications to several data sets. In the present paper, we wish only to provide a brief recapitulation of this model, since *our primary emphasis is upon an alternative algorithm, MAPCLUS (and associated computer program), for fitting that same model.* (MAPCLUS is an acronym for *MA*thematical *P*rogramming *CLUS*tering.) Moreover, we shall argue that the numerical approach described here offers several advantages over the algorithm described by Shepard and Arabie [1979] for fitting the ADCLUS model.

We are indebted to Scott A. Boorman, W. K. Estes, J. A. Hartigan, Lawrence J. Hubert, Carol L. Krumhansl, Joseph B. Kruskal, Sandra Pruzansky, Roger N. Shepard, Edward J. Shoben, Sigfrid D. Soli, and Amos Tversky for helpful discussions of this work, as well as the anonymous referees for their suggestions and corrections on an earlier version of this paper. We are also grateful to Pamela Baker and Dan C. Knutson for technical assistance. The research reported here was supported in part by LEAA Grant 78-NI-AX-0142 and NSF Grants SOC76-24512 and SOC76-24394.

Requests for reprints should be sent to either Phipps Arabie, Department of Psychology, Elliott Hall, University of Minnesota, Minneapolis, Minnesota 55455, or to J. Douglas Carroll, Room 2C-553, Bell Laboratories, 600 Mountain Avenue, Murray Hill, New Jersey 07974.

0033-3123/80/0600-2753$00.75/0

In matrix notation, the ADCLUS model is written

$$\hat{\mathbf{S}} = \mathbf{PWP'} \tag{1}$$

where $\hat{\mathbf{S}}$ is an $n \times n$ symmetric matrix of reconstructed similarities $\hat{s}_{ij}$ (with ones in the principal diagonal), $\mathbf{W}$ is an $m \times m$ diagonal matrix with the weights w_k ($k = 1, \cdots, m$) in the principal diagonal (and zeroes elsewhere), and $\mathbf{P}$ is the $n \times m$ rectangular matrix of *binary* values p_{ik}. Here $\mathbf{P}'$ is the $m \times n$ matrix transpose of the matrix $\mathbf{P}$.

The following qualifications to the preceding variables should be noted. The input data to which the model is fitted are assumed to be the $M = n(n - 1)/2$ entries constituting a two-way symmetric (or symmetrized) proximity matrix having no missing entries. Although the raw data may be in the form of either similarities or dissimilarities, we first transform them linearly to be similarities on the interval [0,1]. (Since the data are assumed to be on an interval scale, this transformation in no way effects the goodness-of-fit, but does allow for the standardization of various parameters in the program described below.) $\mathbf{S} \equiv ||s_{ij}||$ will always refer to these transformed proximities, to which the fitted $\hat{\mathbf{S}}$ matrix is being compared. Turning to the $\mathbf{P}$ matrix, note that each column represents one of the m subsets (or clusters—we use the terms interchangeably), with the ones of that column defining constituency of stimuli within the respective subset. Shepard and Arabie imposed the constraint that the m^{th} subset (and only that subset) was a column of all ones, whose weight was in effect an additive constant for (1), as required for the use of variance accounted for to gauge the goodness-of-fit. In the present usage, we prefer to express the model as

$$\hat{\mathbf{S}} = \mathbf{PWP'} + \mathbf{C} \tag{2}$$

where $\mathbf{C}$ is an $n \times n$ matrix having zeroes in the principal diagonal, and the (fitted) additive constant c in all the remaining entries. [That constant is simply the weight w_m fitted for the complete subset in (1).] Strictly speaking, m in (1) and in the Shepard and Arabie [1979] description corresponds to $m - 1$ in (2). However, we feel that this inconsistency is sufficiently transparent as to allow uniform references below to m as the number of subsets, plus an $(m + 1)^{\text{st}}$ weight as the additive constant.

Overview

The ADCLUS program described by Shepard and Arabie [1979] is briefly summarized as follows. In the first phase, the ordinal information in $\mathbf{S}$ is used to extract the complete set of all m' "elevated" subsets (i.e., maximal complete subgraphs) for each distinct proximity value and thus define the $\mathbf{P}$ matrix. In practice m' turns out to be too large vis-à-vis either substantive interpretation, or parsimonious, nonredundant representation and reduction of data, not to mention the amount of core required to handle even medium-sized (e.g., $n \approx 30$) data sets. (The upper bound for m' can be obtained from results of Moon and Moser, 1965.) A crude initial estimate of each weight, w_k, is also obtained, using interval scale assumptions. The second phase of the ADCLUS program uses an iterative procedure with a modified gradient approach to reduce the large number, m', of subsets, while approximately maximizing the variance accounted for (VAF). The final decision as to what constitutes an acceptable tradeoff between m, the final number of retained subsets to be substantively interpreted, and VAF remains the responsibility of the data analyst.

The most obvious difference between the ADCLUS and MAPCLUS programs is that, for the latter, the (fairly small) number of subsets, m, is specified by the user at the beginning of an analysis and never changes throughout the computation. Thus the difference between the two programs, with respect to the number of clusters, parallels the different approaches to dimensionality taken by Shepard [1962a,b] and Kruskal [1964a,b] in the earliest developments of nonmetric multidimensional scaling. In practice, MAPCLUS has been able to obtain solutions that were acceptable in terms of both interpretability and goodness-of-fit, using considerably fewer clusters for various data sets, than was ever possible with the ADCLUS program. Moreover, the subsets in MAPCLUS are not constrained to be maximal complete subgraphs either at the beginning or conclusion of an analysis. As with all the numerical computation after the ADCLUS program defines its initial configuration, MAPCLUS assumes the data are on an interval scale.

Since the detailed description of the MAPCLUS algorithm in the next section of this paper may strike some readers as being difficult to follow, we offer the following cursory overview of MAPCLUS. The **P** matrix is *initially* considered to have *continuously* varying p_{ik} in spite of the ultimate binary nature of **P**. The *initial* values of **P** can be taken from any of several sources (detailed below), and **W** is initially all zeroes. We use a gradient approach to minimizing a loss function which is the weighted sum of an A- and a B-part. The former is simply a normalized measure of sum of squared error. The more novel B-part consists of a "penalty function" in the form of a polynomial designed to move all pairwise products $p_{ik}p_{jk}$ toward 0,1. Thus the overall algorithm comprises a "mathematical programming" approach to solving a discrete problem by treating it as a continuous problem with constraints allowing only a particular set of discrete values of parameters. Another way of looking at this specific "penalty function" approach is that we attempt to approach the discrete solution by a sequence of increasingly close continuous approximations.

The subsets and weights are computed as follows. Given whatever estimates we have for the first subset $[p(1,1), \cdots, p(n,1)]$, univariate regression is used to estimate w_1, and afterward the p_{i1} values are improved iteratively. Then, following an alternating least squares approach, we take residuals and fit them with a second subset, and so on, until the fit of the m^{th} subset has been iteratively improved and its weight estimated. We also apply multiple linear regression to improve our estimates of all the w_k weights simultaneously. The whole procedure is then repeated with increases in the weight for the B-part relative to the A-part of the loss function, until asymptotically, the (0,1) constraint holds essentially perfectly. When no further improvement in goodness-of-fit is forthcoming, we apply three additional techniques ("polishing", *de novo* iterations, and combinatorial optimization) to refine the fit still further.

We now turn to a detailed description of MAPCLUS.

The Algorithm

The Penalty Function Approach

The iterative maximization of objective functions in clustering and scaling methods [e.g., Hubert, 1972; Katz, 1947; Kruskal, 1964a,b] has heralded much of the current progress in these areas of research. A further step, taken in more recent years, is the combination within a single loss function of several objective functions suitable to different goals, with weights for the different components reflecting importance of the associated "goal". In particular, if one "goal" (e.g., imposition of certain constraints) is to be met precisely, the weight for that goal is (asymptotically) infinitely large relative to other weights. Recent examples of this practice are Cunningham and Shepard [1974], Shepard and Crawford [1975, Note 1], and Carroll and Pruzansky [1975, Note 2; 1980]. In the latter application, objective functions suitable to maximizing variance accounted for *and* satisfying the ultrametric inequality were imposed while fitting multiple tree structures to data. In the present instance, we seek to maximize the variance accounted for, subject to the constraint that **P** is asymptotically binary. As explained in detail below, our loss function takes the form

$$L_k(\alpha_k,\beta_k,\Delta,\mathbf{P}) = \alpha_k A_k + \beta_k B_k. \tag{3}$$

Considering first the left side of (3), note that the loss function is computed only for subset k. Moreover, we do not sum the penalty function over k. The reason is that we are using an alternating least squares approach [Wold, 1966] which underlies the iterative fitting in turn of each subset $p_{ik}(i = 1, \cdots, n)$ and its associated weight w_k. Wold has shown that for problems posed in a continuous form, the alternating least squares approach will asymptotically lead to at least a local optimum (minimum) for the *overall* least squares problem for all m subsets of parameters in the model. (In cases where there is only a single optimum this solution will, under very general conditions, be that global optimum, but this situation will generally not hold for the kind of highly nonlinear model we are fitting in the present case.) Since we are only fitting the k^{th} subset at any instant, Δ in (3) refers to the (centered) residuals computed for the remaining $m - 1$ subsets and the reader may wish to associate an implicit subscript "k" with Δ. For the present, the reader is asked only to note this statement of the procedure, since we feel that the explanation is most easily presented in the next two subsections, which explain in detail the alternating least squares implementation in MAPCLUS.

In the right side of (3), the term $\alpha_k A_k$ refers to the weight α_k applied to the normalized sum of squared error, A_k. Specifically,

$$A_k = \frac{a_k}{d_k}, \tag{4}$$

where

$$a_k = \sum_{i}^{n} \; \sum_{>\; j}^{n-1} (\delta_{ij} - w_k p_{ik} p_{jk})^2 , \tag{5}$$

and

$$d_k = \frac{4 \sum_{i}^{n} \; \sum_{>\; j}^{n-1} \delta_{ij}^2}{M} . \tag{6}$$

In (5), a_k is simply the sum of squared error, the minimization of which is equivalent to maximizing VAF. The denominator of A, d_k, is a normalization factor interpreted as the variance of the residuals, δ_{ij}, computed over the remaining subsets. The factor of 4 represents the maximum variance of $\frac{1}{4}$ that could be obtained from the input data s_{ij}, which, as noted earlier, are in the range [0,1].

If our loss function consisted only of the A term (i.e., the B weight $\beta = 0$), then our problem would reduce to performing principal components analysis, for which well-known continuous methods are available. However, our model demands that $p_{ik} = 0, 1$ and the B-part of the loss function is designed, by successively closer continuous approximations, to enforce this discrete constraint. We refer to this as a "mathematical programming" approach, since it entails optimizing a nonlinear function (A_k) with constraints on parameters imposed by use of the "penalty function" method.

Elaborating on the right side of (3), we have

$$B_k = \frac{u_k}{v_k}, \tag{7}$$

where

$$(8) \qquad u_k = \frac{1}{2} \sum_{i}^{n} \sum_{j}^{n} [(p_{ik}p_{jk} - 1)p_{ik}p_{jk}]^2 ,$$

and

$$(9) \qquad v_k = \sum_{i \; >}^{n} \sum_{j}^{n-1} (p_{ik}p_{jk} - T_k)^2 ,$$

where T_k is simply the mean of the pairwise products of $p_{ik}p_{jk}$, namely

$$(10) \qquad T_k = \frac{1}{M} \sum_{i \; >}^{n} \sum_{j}^{n-1} p_{ik}p_{jk} .$$

The numerator of B_k, u_k, is designed to force the pairwise products $p_{ik}p_{jk}$ to be 0,1. B is deliberately nonhomogeneous, since otherwise, the p_{ik} could approach *any* relatively distinct pair of values, instead of only 0,1. That is, B could be made arbitrarily small by making one of the p_{ik} very large and all the others very small, so that the product of the large p_{ik} with the small ones is equal either to 1.0 or near 0.0. Such a pattern is not consistent with the ADCLUS model. Products of *pairs* of the form $p_{ik}p_{jk}$ are emphasized in order to reduce the likelihood of singleton subsets (i.e., all but one of the $p_{ik}(i = 1, \cdots, n)$ being zero) occurring. We deliberately include the diagonal terms in the numerator of B_k, but exclude them in the denominator. If we included the diagonal terms in the variance-like normalizing factor that forms the denominator of B_k, a singleton subset would satisfy the constraint, since the denominator would be nonzero (owing to the one diagonal term differing from all the other diagonal and off-diagonal terms). Singletons are unacceptable in the present context since they cannot account for any of the variance in the $n(n - 1)/2$ interstimulus proximities. Recent theoretical emphases upon self-similarities [Krumhansl, 1978; Podgorny & Garner, 1979] suggest that permitting input of the diagonal entries (when available) of a proximity matrix and allowing their participation in determining the solution may be a promising future development for both clustering and scaling techniques.

We feel that the penalty function defined by the B-part of the loss function is potentially applicable to a fairly large range of models where coefficients are constrained to assume a small number of discrete values. However, selection of the "right" form for B has proved to be crucial in the development of MAPCLUS. The present formula is based on intuition and our previous experiences with three alternatives for B which were successively discarded, viz.,

$$\frac{\sum_{i}^{n} [p_{ik}(1 - p_{ik})]^2}{\left(\sum_{i}^{n} p_{ik}^2\right)^2} , \qquad \frac{\sum_{i}^{n} [p_{ik}(1 - p_{ik})]^2}{\sum_{i}^{n} p_{ik}^2} ,$$

and

$$(11) \qquad \frac{\sum_{i}^{n} [p_{ik}(1 - p_{ik})]^2}{\left[\sum_{i}^{n} p_{ik}^2 - \frac{1}{n}\left(\sum_{i=1}^{n} p_{ik}\right)^2\right]} .$$

Since MAPCLUS relies on a minimization procedure, we require the gradient of the loss function L_k, ∇L_k, with respect to p_{ik}. It is straightforward that the i^{th} component of ∇L_k ($\nabla_i L_k$) is:

$$\nabla_i L_k = \frac{\partial L_k}{\partial p_{ik}} = \frac{\alpha}{d_k}\left[\frac{\partial a_k}{\partial p_{ik}}\right] + \beta\left[\frac{v_k \dfrac{\partial u_k}{\partial p_{ik}} - u_k \dfrac{\partial v_k}{\partial p_{ik}}}{v_k^2}\right] \tag{12}$$

where

$$\frac{\partial a_k}{\partial p_{ik}} = -2w_k \sum_{j\neq i} p_{jk}(\delta_{ij} - w_k p_{ik} p_{jk}), \tag{13}$$

$$\frac{\partial u_k}{\partial p_{ik}} = 2p_{ik} \sum_{j} [(p_{ik}p_{jk} - 1)\,(p_{jk}^2)\,(2p_{ik}p_{jk} - 1)], \tag{14}$$

and

$$\frac{\partial v_k}{\partial p_{ik}} = 2\sum_{j\neq i}\left(p_{jk} - \frac{1}{M}\sum_{h\neq i} p_{hk}\right)(p_{ik}p_{jk} - T_k). \tag{15}$$

Finally, returning to the weights, α_k and β_k in the loss function of (3) and (12), we use the constraint that $\alpha_k + \beta_k = 1$. We typically begin with $\alpha_k = \alpha_0 = .50$, and as computation proceeds, we increase the value of β_k relative to α_k (details given below), to ensure that the final values of the $p_{ik} = 0,1$. This adjustment of α_k and β_k is done according to

$$\alpha'_k = \frac{\alpha_k}{\alpha_k + K_1\beta_k} \quad \text{and} \quad \beta'_k = \frac{K_1\beta_k}{\alpha_k + K_1\beta_k}, \tag{16}$$

where K_1 is currently defined as 2.0. Also, α_k is not permitted to be less than 10^{-6}, so as to avoid annoying underflow messages from various FORTRAN compilers.

The Alternating Least Squares Structure of MAPCLUS

Thus far, we have presented the loss function and the associated gradient required for steepest descent, as well as the strategy for changing the weights (α_k and β_k) to assure satisfaction of the (0,1) constraints. In this and the two following subsections, we cover the essential details of the numerical methods used in MAPCLUS.

Characteristic of alternating least squares approaches, MAPCLUS has iterative computing nested to several levels of depth. Figure 1 gives an overview of this nesting, as well as an indication of the terminology employed here. Unfortunately, it is necessary for the reader to master our usage of "major," "outer," and "inner" iterations, as we will be making extensive use of these terms throughout the rest of the paper. An *inner iteration* is used for obtaining an estimate via univariate linear regression of the weight w_k and the additive constant c_k specific to subset k, as well as moving each of the p_{ik} ($i = 1, \cdots, n$) elements by one step of the computed gradient. Details of these procedures are given in the following two subsections of this paper, but for the present it is sufficient to note that an inner iteration consists of the description in the preceding sentence plus subsequent checking for convergence.

The basic principle on which the alternating least squares (ALS) approach is based in the present instance is quite simple. In seeking a least squares solution for a model of the form

$$\underline{s_{ij} \cong} \sum_{k=1}^{m} w_k p_{ik} p_{jk} + c, \tag{17}$$

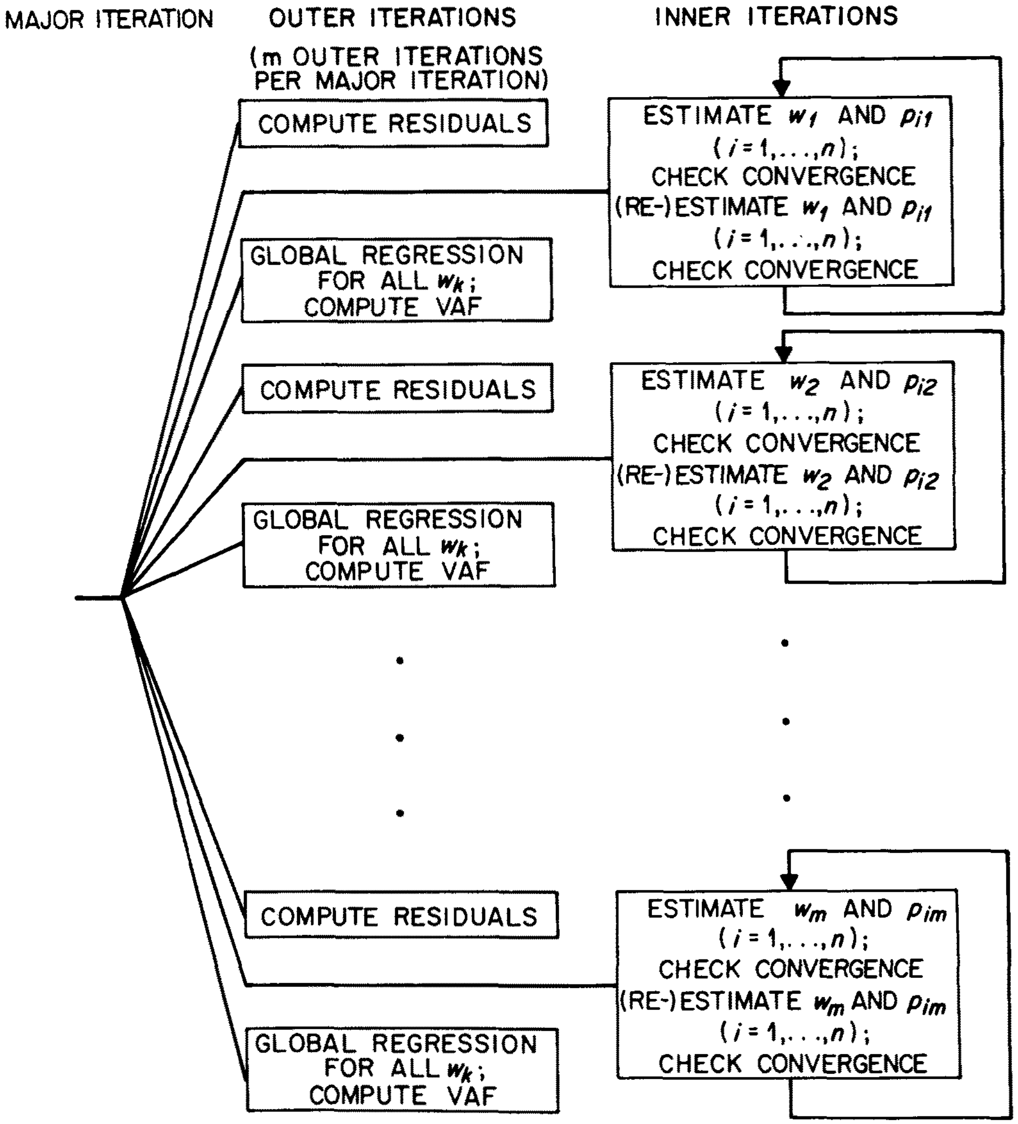

MAJOR ITERATION
OUTER ITERATIONS
(m OUTER ITERATIONS PER MAJOR ITERATION)
INNER ITERATIONS
COMPUTE RESIDUALS
ESTIMATE w_1 AND p_{i1} ($i = 1, \ldots, n$); CHECK CONVERGENCE
(RE-)ESTIMATE w_1 AND p_{i1} ($i = 1, \ldots, n$); CHECK CONVERGENCE
GLOBAL REGRESSION FOR ALL w_k; COMPUTE VAF
COMPUTE RESIDUALS
ESTIMATE w_2 AND p_{i2} ($i = 1, \ldots, n$); CHECK CONVERGENCE
(RE-)ESTIMATE w_2 AND p_{i2} ($i = 1, \ldots, n$); CHECK CONVERGENCE
GLOBAL REGRESSION FOR ALL w_k; COMPUTE VAF
COMPUTE RESIDUALS
ESTIMATE w_m AND p_{im} ($i = 1, \ldots, n$); CHECK CONVERGENCE
(RE-)ESTIMATE w_m AND p_{im} ($i = 1, \ldots, n$); CHECK CONVERGENCE
GLOBAL REGRESSION FOR ALL w_k; COMPUTE VAF

FIGURE 1
Structure of major, outer, and inner iterations in the alternating least squares approach used in MAPCLUS.

we are using a least squares (overall) loss function; that is, we solve for the p_{ik} and w_k by minimizing

$$(18) \qquad E^2 = \sum_{i}^{n} \, {}_{>} \sum_{j}^{n-1} \left(s_{ij} - \sum_{k}^{m} w_k p_{ik} p_{jk} - c\right)^2 \cdot$$

In an ALS approach, we divide our parameters into subsets, and solve the *conditional* least squares problem for one subset holding all others fixed. In the present case, the subsets correspond to the $\mathbf{p}_k$ vectors (excluding the $m + 1^{\text{st}}$ "constant" vector) and associated w_k ($k = 1, 2, \cdots, m$). It is easy to see, then, that the conditional least squares problem for parameter subset k defines

$$(19) \qquad \delta_{ij}^{(k)} \equiv s_{ij} - \sum_{l \neq k} w_l p_{il} p_{jl} - c$$

and fits $\mathbf{p}_k$ and w_k via least squares to $\delta^{(k)}$; thus the conditional least squares problem is to minimize

$$(20) \qquad E^2_{(k)} \equiv \sum_{i}^{n} \sum_{> \ j}^{n-1} (\delta_{ij}^{(k)} - w_k p_{ik} p_{jk})^2 .$$

This is exactly the least squares problem whose solution is sought via the "mathematical programming" algorithm described earlier.

We can, without loss of generality, center the residuals (the $\delta_{ij}^{(k)}$) after computing them, because we calculate, for each inner iteration, a constant ($\hat{c}_k$) as well as an estimate of the weight ($\hat{w}_k$) for the k^{th} subset, using least squares univariate linear regression. The constant $\hat{c}_k$ is, in effect, absorbed into the weight for the $(m + 1)^{\text{st}}$ implicit subset (corresponding to the complete set); i.e., into the estimate of the constant c. In effect, the current estimate of c is thus $\hat{c} = \hat{c}_k + b_k$, where b_k is the negative of the constant added to the residuals in centering them. Therefore, we are in fact reestimating c each time we update the estimate of each $\mathbf{p}_k$ vector.

A series of these inner iterations, for successively better estimates of the parameters of the k^{th} subset, are the core of an *outer iteration*. To elaborate, an *outer iteration* consists of (a) taking the residuals (i.e., the difference between $\mathbf{S}$ and the values of $\hat{\mathbf{S}}$ computed over all the other subsets and additive constant, but excluding subset k) and centering them, (b) a series of inner iterations, each refining the estimated parameters for subset k, and (c) "global" multiple linear regression for reestimating all m weights (plus c) simultaneously, and subsequent computation of VAF.

The number of inner iterations in an outer iteration will vary, depending on convergence criteria and arbitrary limits. However, there will always be m outer iterations (one for each subset) within a single *major iteration*. In the computation of a major iteration, the subsets are considered in the order $1, 2, \cdots, m$ on odd-numbered major iterations, and in reverse (descending) order on even iterations. Without this alternating reversal, we have found that the comparative fit of the m different subsets is often subject to sequential problems (e.g., the lion's share goes to subset 1, and few morsels remain for subset m), and that goodness-of-fit is unduly sensitive to the number of inner iterations allowed in the outer iterations. Finally, a distinction will later be made between "prepolishing" major iterations versus "polishing" major iterations, where the latter refer to the process of forcing $\beta_k \rightarrow 1$, so that the p_{ik} are "polished off" at 0,1.

Description of Computation in Inner, Outer, and Major Iterations

An outer iteration is essentially a shell for the inner iterations, in which most of the real computational work in MAPCLUS is done. Accordingly, the present subsection of the paper is largely concerned with the proceedings of an inner iteration.

In the flow chart given in Figure 2, the inner iteration consists of the procedures between junctures 1 and 6. The computation preceding juncture 1 and following 6 is the enveloping outer iteration. The latter begins by taking the default value of λ (currently 1) as the initial step-size, but retaining the values of α_k and β_k from any previous computation on that subset. It is then necessary to compute the residuals Δ to which subset k is being fitted. We have

$$(21) \qquad \delta'^{(k)}_{ij} = s_{ij} - \sum_{l \neq k} w_l p_{il} p_{jl} .$$

The residuals $\delta'^{(k)}_{ij}$ are then centered on their mean to form $\Delta = ||\delta_{ij}||$, which will be used as starting values throughout the outer iteration. Henceforth, we have dropped from δ the superscript k, understanding that the latter is implicitly present. [If "global" regression of all w_k (explained in the next section) were applied at the end of each inner iteration, then the residuals would have to be computed anew for each inner iteration; however, we have chosen to apply global regression only at the end of an outer iteration.]

Given the residuals to which the loss function L_k is to be applied, we are now ready to begin an inner iteration (juncture 1 of Figure 2). If this is the very first time the p_{ik} ($i = 1, \cdots, n$) are being considered (i.e., the first inner iteration of outer iteration k in the first major iteration), then initial values for the p_{ik} must come from one of the three methods given below. Given those values, we use univariate linear regression to estimate the weight,

$$\hat{w}_k = \frac{\sum_{i>j}^{n} \sum^{n-1} \delta_{ij} p_{ik} p_{jk}}{\left[\sum_{i>j}^{n} \sum^{n-1} p_{ik}^2 p_{jk}^2 - \frac{1}{M} \left(\sum_{i>j}^{n} \sum^{n-1} p_{ik} p_{jk} \right)^2 \right]} \tag{22}$$

and the regression constant c_k for that subset's weight,

$$\hat{c}_k = -\hat{w}_k T_k + \frac{1}{M} \sum_{i>j}^{n} \sum^{n-1} \delta_{ij} . \tag{23}$$

We then translate the residuals by that regression constant, so that

$$\delta_{ij}^{\text{new}} = \delta_{ij}^{\text{old}} - \hat{c}_k . \tag{24}$$

The translated residuals are used for computation during the remainder of the given inner iteration.

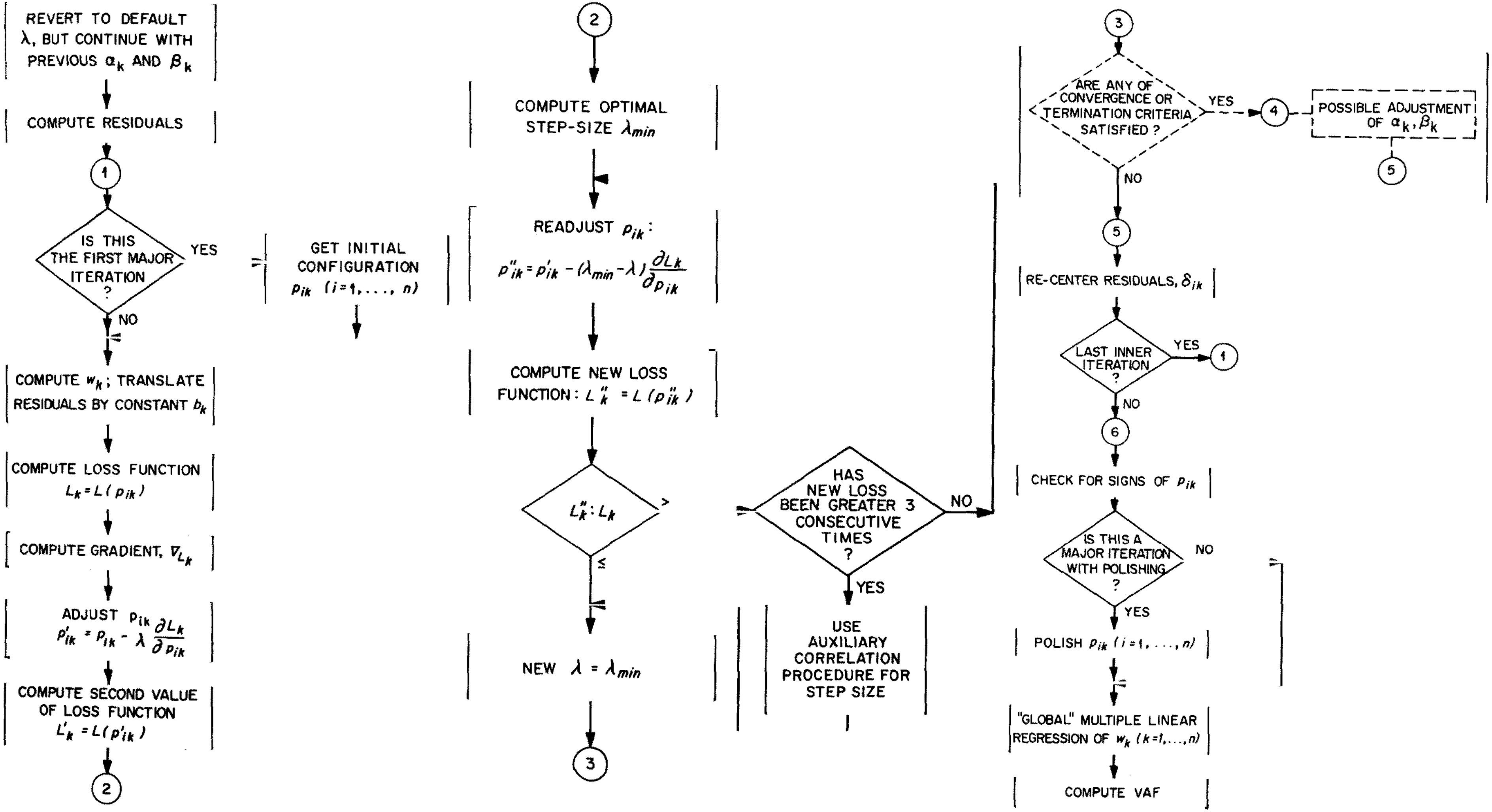
The next step is to compute the gradient, ∇L_k, according to (12) through (15). Given this quantity, we are now faced with the perennial problem of determining the best step-size. That is, we are adjusting the old p_{ik} ($i = 1, \cdots, n$) values according to

$$p'_{ik} = p_{ik} - \lambda \frac{\partial L_k}{\partial p_{ik}} , \tag{25}$$

where λ is a "trial" value of the step size. Having taken that step for each of the p_{ik} ($i = 1, \cdots, n$), we then evaluate the loss function L_k, using (3) through (10). We now have all the information needed to estimate what the value of λ *should* have been, $\lambda_{\min}$, to minimize the loss function. The procedure we use for estimating $\lambda_{\min}$ is described, among other places, in Adby and Dempster [1974, pp. 62–64], and has also been used successfully by Carroll and Pruzansky [1980]. Although this procedure has worked well in practice, such quadratic approximation techniques perform poorly in various circumstances [Robinson, 1979]. Therefore, when this procedure fails on a given inner iteration to improve the loss function, we use an auxiliary method based on the product-moment correlation between the present and past gradient (cf. reliance upon the cosine of the angle between successive gradients, as used in Kruskal, 1964b, 1977; and Kruskal & Carroll, 1969). Both the quadratic interpolation and the auxiliary correlation procedures could be replaced by other numerical techniques, but the latter appear to us to be considerably more expensive computationally.

The next step is to check for satisfaction of various convergence criteria; see junctures 3 and 5 in Figure 2. A more detailed flow chart for that segment of an inner iteration is given in Figure 3. We will depend largely on that graphic summary to convey the details of the decision structure since (a) the comparisons and outcomes are rather complicated, (b) it is obviously ad hoc, and (c) with further development of MAPCLUS, modifications and revisions are likely (cf. Ramsay, 1977, p. 249 on nonmetric multidimensional scaling algorithms).

FIGURE 2
Flow chart for an inner iteration.


REVERT TO DEFAULT λ, BUT CONTINUE WITH PREVIOUS α_k AND β_k
COMPUTE RESIDUALS
1
IS THIS THE FIRST MAJOR ITERATION ?
YES
NO
COMPUTE w_k; TRANSLATE RESIDUALS BY CONSTANT b_k
COMPUTE LOSS FUNCTION $L_k = L(p_{ik})$
COMPUTE GRADIENT, ∇_{L_k}
ADJUST p_{ik} $p'_{ik} = p_{ik} - \lambda \frac{\partial L_k}{\partial p_{ik}}$
COMPUTE SECOND VALUE OF LOSS FUNCTION $L'_k = L(p'_{ik})$
2
GET INITIAL CONFIGURATION p_{ik} $(i=1,\dots,n)$
2
COMPUTE OPTIMAL STEP-SIZE λ_{min}
READJUST p_{ik}:
$p''_{ik} = p'_{ik} - (\lambda_{min} - \lambda)\frac{\partial L_k}{\partial p_{ik}}$
COMPUTE NEW LOSS FUNCTION: $L''_k = L(p''_{ik})$
$L''_k : L_k$
>
≤
NEW $\lambda = \lambda_{min}$
3
HAS NEW LOSS BEEN GREATER 3 CONSECUTIVE TIMES ?
NO
YES
USE AUXILIARY CORRELATION PROCEDURE FOR STEP SIZE
3
ARE ANY OF CONVERGENCE OR TERMINATION CRITERIA SATISFIED ?
YES
4
POSSIBLE ADJUSTMENT OF α_k, β_k
5
NO
5
RE-CENTER RESIDUALS, δ_{ik}
LAST INNER ITERATION ?
YES
1
NO
6
CHECK FOR SIGNS OF p_{ik}
IS THIS A MAJOR ITERATION WITH POLISHING ?
NO
YES
POLISH p_{ik} $(i=1,\dots,n)$
"GLOBAL" MULTIPLE LINEAR REGRESSION OF w_k $(k=1,\dots,n)$
COMPUTE VAF

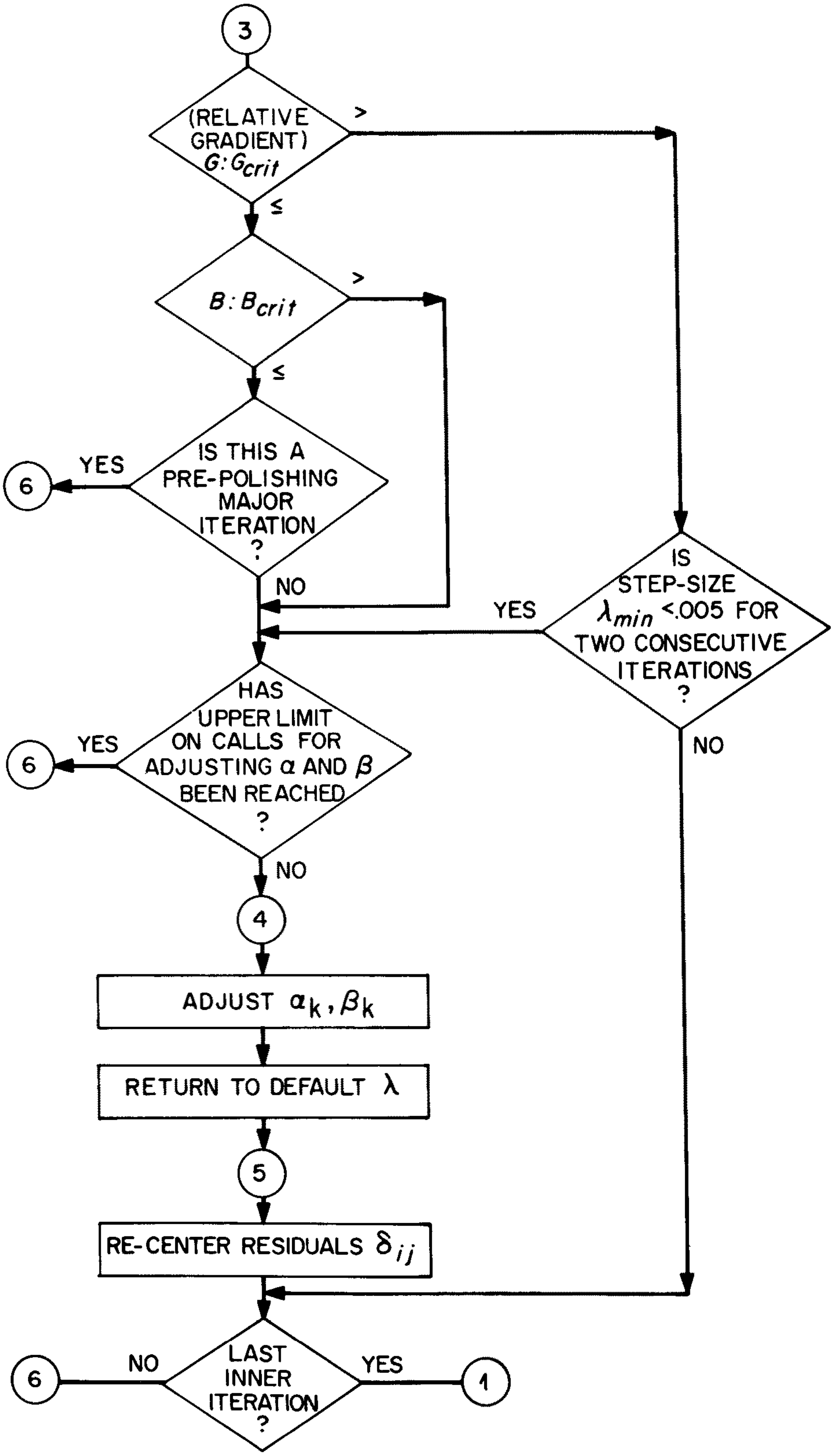

3
(RELATIVE GRADIENT) $G : G_{crit}$
>
≤
$B : B_{crit}$
>
≤
IS THIS A PRE-POLISHING MAJOR ITERATION ?
YES
6
NO
IS STEP-SIZE λ_{min} <.005 FOR TWO CONSECUTIVE ITERATIONS ?
YES
NO
HAS UPPER LIMIT ON CALLS FOR ADJUSTING α AND β BEEN REACHED ?
YES
6
NO
4
ADJUST α_k, β_k
RETURN TO DEFAULT λ
5
RE-CENTER RESIDUALS δ_{ij}
LAST INNER ITERATION ?
NO
6
YES
1

FIGURE 3
Elaboration of boxed area in last panel of Figure 2, concerning tests for convergence.

A summary of the computation is as follows. Three quantities are compared with arbitrary critical values or thresholds. The three quantities are G, the "relative gradient" (defined as the product of the length of the gradient times the length of the $\mathbf{p}_k$ vector), the B-part of the loss function, and the estimated step-size $\lambda_{\min}$. Currently, the default values for the criteria are: $G_{\text{crit}} = .005$, $B_{\text{crit}} = .05$, $\lambda_{\text{crit}} = .005$. When various combinations of these criteria are met, either the outer iteration terminates, or β_k is increased relative to α_k, as in (16), up to a previously specified number of times.

Which of the two preceding alternatives is taken depends largely on whether the major iteration is designated as "prepolishing," or "polishing." To elaborate, on the first several outer iterations, we have found that it is useful to pay closer attention to the A-part of the penalty function. After a specified number of "prepolished" major iterations, we are ready to begin the first "polishing" major iteration. When that happens, we revert to the default values for α_k and β_k (i.e., reduce β_k back to its initial value) and include 30 additional inner iterations in each outer iteration. During those additional inner iterations, we repeatedly adjust β_k so that $\beta_k \to 1$. (Later on in these "polishing" major iterations, at the end of the outer iteration we set $\beta_k = 1$ to "polish" the p_{ik} to 0,1, as explained below.)

At the end of an inner iteration, we recenter the residuals δ_{ij}. If this is not the last inner iteration of the given outer iteration, we return to juncture 1 of Figure 2; otherwise we have reached juncture 6 and the final stages of an outer iteration. Here, the first check is to see that the signs of the p_{ik} are "correct." That is, since the B-part of the penalty function [see equations (8) and (9)] uses the pairwise products $(p_{ik}p_{jk})$, the signs of all the p_{ik} could be reversed without affecting the value of B_k. The simplest remedy for this possibility is to set p_{hk} (where that entry corresponds to $\max_i|p_{ik}|$) positive and to multiply the rest of the p_{ik} by that same sign.

If the major iteration is designated as "prepolishing," we are now ready for multiple linear regression of all the weights $w_k (k = 1, \cdots, m)$, as explained below. For a "polishing" major iteration, however, we first consider each of the p_{ik} $(i = 1, \cdots, n)$ and redefine them as

$$p_{ik} = \begin{cases} 1 & \text{if} \quad p_{ik} \geq 2^{-1/2} \\ 0 & \text{if} \quad p_{ik} < 2^{-1/2} . \end{cases} \tag{26}$$

This redefinition is equivalent to setting $\beta_k = 1$ and forces the condition $p_{ik} = 0,1$. If the cutoff chosen here creates a singleton subset (i.e., only one of the n values p_{ik} is unity, and the rest are zeroes), then the second largest of the original p_{ik} values is redefined as 1. Conversely, if all the original p_{ik} values are greater than the cutoff, then the smallest of the original values is redefined as 0. The rationale for the cutoff used in (26) is as follows. The B-part of the loss function emphasizes pairwise products, $p_{ik}p_{jk}$, since these are the values being pushed toward 0,1. If we consider the *pairwise* threshold to be 0.5, then a reasonable value for an individual p_{ik} would seem to be $0.5^{1/2}$. Empirically, this slight advantage for zero entries of p_{ik} seems reasonable, since one of the symptoms of a bad solution (e.g., one in which the data analyst requested too many clusters) is too many ponderous subsets.

We then apply multiple linear regression to the model of (2) to obtain best-fitting estimates of all the weights $w_k (k = 1, \cdots, m)$ as explained in the next subsection. Finally the VAF over all m subsets is computed. At this point, we could apply the Miles-Kruskal algorithm for monotone regression [Kruskal, 1964b] and thus render MAPCLUS fully nonmetric, but we have not yet done so.

Multiple Linear Regression Estimates of the Weights

The technique to be discussed in this section is the use of multiple linear regression simultaneously to fit the weights w_k $(k = 1, \cdots, m)$ and the additive constant. While there is nothing original in our present use of regression, we think that as an application of the ADCLUS model to an hypothesized set of (binary) distinctive features, this "stand-alone" regression for fitting a constrained ADCLUS model has wide applicability to feature-oriented research [cf., Carroll & Arabie, 1980; Hubert & Baker, 1977; Tversky, 1977] and thus merits explicit discussion. Shepard and Arabie [1979], for example, used the regression approach in evaluating the distinctive features proposed by Gibson, Osser, Schiff, and Smith [1963, Note 3] to account for confusions between upper-case Roman letters.

In MAPCLUS, regression can be invoked either (a) to fit weights w_k to a user-supplied set of binary features (represented as subsets; see Shepard and Arabie, 1979, and the subsection below on initial configurations) in a "stand-alone" mode, without recourse to any of the iterative computation described above, or (b) at the end of each outer iteration (see Figure 2), using the iteratively fitted values of p_{ik}.

For either usage, we begin with (2), considering **S** and **P** to be fixed, while solving for the diagonal matrix **W** and the off-diagonal constant matrix **C**. Equation (2) can be recast in summation notation as

$$\hat{s}_{ij} = \sum_{k=1}^{m} w_k p_{ik} p_{jk} + c. \tag{27}$$

Now let

$$q_{ijk} = p_{ik} p_{jk}. \tag{28}$$

At this point, it is easier to view **S** as a (singly subscripted) one-dimensional array, and **Q** as a two-dimensional array. To achieve this reduction in dimensionality, we concatenate subscripts i, j as (ij) where $(ij) = 1, \cdots, M$ (where $M = n(n-1)/2$). We may now rewrite (27) as

$$\hat{s}_{(ij)} = \sum_{k=1}^{m+1} w_k q_{(ij)k} \tag{29}$$

where $w_{m+1} = c$ and $q_{(ij)m+1} = 1$. In vector-matrix notation, we have

$$\mathbf{s} = \mathbf{Qw} \tag{30}$$

where **s** is the column vector of M elements with $s_{(ij)}$ as the general entry, **Q** is the $M \times (m + 1)$ matrix with $q_{(ij)k}$ as the general entry, and **w** is the $(m + 1)$-dimensional column vector with general entry w_k (with $w_{m+1} = c$).

As a standard least squares result, we estimate the weights

$$\hat{\mathbf{w}} = \mathbf{Q}^+\mathbf{s} \tag{31}$$

where $\mathbf{Q}^+$ is the Moore-Penrose generalized inverse [see Kruskal, 1975; Rao & Mitra, 1971] of **Q**. Under most conditions, in particular if **Q** is of full column rank (f.c.r.),

$$\mathbf{Q}^+ = (\mathbf{Q}'\mathbf{Q})^{-1}\mathbf{Q}'. \tag{32}$$

If **Q** is not of f.c.r., $\mathbf{Q}'\mathbf{Q}$ will be singular and $(\mathbf{Q}'\mathbf{Q})^{-1}$ nonexistent, so that a more complex definition of $\mathbf{Q}^+$ is required. In practice, however, we have been able to use the algorithm of Bunch, Kaufman, and Partlett [1976] to solve for $(\mathbf{Q}'\mathbf{Q})^{-1}$. In our experience the only difficulties that have arisen have resulted from inadvertent duplication of a subset (i.e., two columns of **P** are identical) in the initial configuration. When a feature is thus present in duplicate, the simplest remedy is to remove the duplicate(s) and reduce m accordingly. Although other difficulties are conceivable (e.g., when several disjoint subsets and their union as an additional subset are all in **P**), they have yet to occur.

In passing, we should note that **P** can be of less than f.c.r., while **Q** *is* nevertheless of f.c.r. For example if **P** corresponds to a partition of the n stimuli, or to a hierarchical clustering (i.e., nested partitions) of the stimuli, **P** will necessarily be of less than f.c.r., but **Q** will frequently be of f.c.r., so that $(\mathbf{Q}'\mathbf{Q})$ will be invertible and $\mathbf{Q}^+$ will thus exist.

Sharpening the Output

Taking stock of the computational description thus far, we have covered inner and outer iterations, as well as the distinction between prepolishing and polishing major iterations. In one sense, MAPCLUS is now complete, since the elements of **P** are 0,1, and the variance accounted for has been approximately maximized. However, varying certain of the parameters (e.g., the initial values of α and β, the weights in the loss function) has sometimes resulted in improved solutions, and negative weights ($w_k < 0$) have occasionally been encountered. We have therefore devised two additional procedures which are appended to the algorithm.

Shepard and Arabie [1979] noted that negative weights are uninterpretable for the ADCLUS model, just as they are in INDSCAL [Carroll & Chang, 1970]. Although in our experience, negative weights have been infrequent, they are sufficiently annoying that we have developed a procedure that has generally been successful in eliminating this problem. Our approach differs markedly in rationale from various well-known techniques for nonnegative least squares regression. Such methods, if employed in MAPCLUS, would upon finding a negative weight reduce the effective number of clusters, so that the number originally specified by the data analyst would just be an upper bound. Moreover, we have observed that weights which become negative often do so only temporarily during the iterative process of fitting the ADCLUS model via MAPCLUS. Thus, we prefer to retain subsets whose weights become negative, but use a method (described below) which encourages those weights to become positive.

It will be recalled from the earlier discussion that the first polishing major iteration for subset k involves redefining $p_{ik} \geqslant 2^{-1/2}$ (where $i = 1, \cdots, n$) as unity, or otherwise as zero. This redefinition satisfies the requirement that elements of **P** be 0,1 and is equivalent to setting $\beta_k = 1$. There is little advantage in conducting a second major iteration with polishing. If we were to do so and retain the $\beta_k = 1$ or even $\beta_k \approx 1$, then the p_{ik} would hardly budge; reverting to the default initial value of α and β would, computationally speaking, be a return to whence we came.

An alternative that has worked well, which we have called a *"de novo"* major iteration, works as follows: For the second (and succeeding) major iteration(s) with polishing, we begin each outer iteration for subset k by "zeroing out" both the weight, w_k, and the fitted subset p_{ik} ($i = 1, \cdots, n$). We then reestimate the subset (i.e., the p_{ik}) using (42) through (48) given below for obtaining a rational initial configuration. Then, starting with α_k and β_k at their default values, we gradually increase β_k (relative to α_k) over successive inner iterations. The *de novo* approach in effect lets the remaining $m - 1$ subsets completely determine the k^{th} during the latter subset's outer iteration. This technique appears to eliminate negative weights. The VAF sometimes drops slightly on the first *de novo* major iteration (i.e., the second polishing major iteration), but generally tends to recover in succeeding *de novo* iterations, often to a higher value than before the start of the *de novo* iterations. When improvement in VAF becomes negligible, the iterative computation is terminated.

We now turn to combinatorial optimization, the final procedure in the MAPCLUS program. Shepard and Arabie [1979] noted that for n stimuli, there are $2^{2^n-1} - 1$ distinct ADCLUS clusterings. Any attempt at exhaustive enumeration (to get the maximum VAF) would be infeasible; for as few as $n = 5$ stimuli, there are more than 10^9 possible solutions. However, one strategy, often employed in fitting discrete models, is the exhaustive exploration of a subset of all possible solutions, typically those related to a very good starting solution. (In the present application, the last item is furnished by the output from the *de novo* major iterations.)

In their discussion of the ADCLUS representation of some of the Roethlisberger and Dickson [1939] data, Shepard and Arabie [1979] noted that several of the subsets with relatively small weights were "off" by one stimulus. That is, if one member were added to or dropped from these subsets, then interpretability would have been enhanced. This observation suggests one avenue of combinatorial optimization, namely reversing each of the individual p_{ik} values in turn (i.e., $p'_{ik} = 1 - p_{ik}$) to see if VAF increases. To develop this idea formally, we begin with V_k, the variance accounted for by the k^{th} subset (and only that subset, as distinct from VAF computed over all m subsets) and by the associated regression constant c_k. That is,

$$V_k = \frac{\left[\sum_{i\,>\,j}^{n}\sum^{n-1} \delta_{ij}p_{ik}p_{jk}\right]^2}{H\left(\sum_{i\,>\,j}^{n}\sum^{n-1} p_{ik}^2p_{jk}^2\right)} = \frac{E_k^2}{F_k}, \tag{33}$$

where the function H in the denominator is simply

$$H(x) = x - \frac{x^2}{M}, \tag{34}$$

and the residuals δ_{ij} have been centered. It follows that if we reverse an entry p_{ik}, the resulting variance accounted for by subset k becomes

$$V_{k\bar{i}} = \frac{[E_k - (2p_{ik} - 1)g_{ik}]^2}{H(F_k - (2p_{ik} - 1)h_{ik})}, \tag{35}$$

where

$$g_{ik} = \sum_{j \neq i} \delta_{ij}p_{jk} \tag{36}$$

and

$$h_{ik} = \sum_{j \neq i} p_{jk}^2 . \tag{37}$$

If we consider all "doubleton" reversals (i.e., $p'_{ik} = 1 - p_{ik}$ and $p'_{jk} = 1 - p_{jk}$, $i \neq j$), the appropriate formula is

$$V_{k(\overline{ij})} = \frac{[E_k - (2p_{ik} - 1)g_{ik} - (2p_{jk} - 1)g_{jk} + u_{ijk}]^2}{H(F_k - (2p_{ik} - 1)h_{ik} - (2p_{jk} - 1)h_{jk} + v_{ijk})}, \tag{38}$$

where

$$v_{ijk} = (2p_{ik} - 1)(2p_{jk} - 1), \tag{39}$$

and

$$u_{ijk} = v_{ijk}\delta_{ij}. \tag{40}$$

More generally, for an "*l*-tuple" reversal, we have

$$V_{k(i_1 i_2 \ldots i_l)} = \frac{\left[E_k - \sum_{i_s \in I}(2p_{i_sk} - 1)g_{i_sk} + \sum_{\substack{i_s\,<\,i_{s'}\\(i_s, i_{s'} \in I)}}\sum u_{i_si_{s'}k}\right]^2}{H\left(F_k - \sum_{i_s \in I}(2p_{i_sk} - 1)h_{i_sk} + \sum_{\substack{i_s\,<\,i_{s'}\\(i_s, i_{s'} \in I)}}\sum v_{i_si_{s'}k}\right)} \tag{41}$$

where $\bar{I} = (i_1, i_2, \cdots, i_l)$ is the subset of subscripts denoting the elements being reversed in the l-tuple change.

Two points should be noted. First, for all the types of reversals of the p_{ik} considered thus far, the only permissible ones are those leaving subset k with a cardinality of more than 1 and less than n stimuli. Second, it is possible for a reversal to result in improved VAF while causing previously positive weight(s) of any of the subsets to become negative. Our approach begins by evaluating (38) for each of the p_{i1} and p_{j1} values in subset 1. If any of the permissible trial values of the $n(n - 1)/2$ different predicted $V_{1(\bar{ij})}$ exceeds V_1 as computed in (33), then we execute the reversals corresponding to the largest such value (if any), arbitrarily breaking a tie if necessary, and apply global regression to see if the improved VAF entails any newly negative weights. If so, the reversal is not implemented, and the procedure goes to the next subset. Otherwise, the reversal is retained, and (38) is once again used to look for further advantageous changes within the subset. If any are found, they are tested just as the first such change was. The search for further reversals within a subset ends when (38) fails to yield a value superior to that from (33), which will have been updated whenever an advantageous reversal has been made. We proceed to do the same in turn for subsets $2, \cdots, m$. We are then ready for the second part of our combinatorial optimization approach, which considers singletons. Specifically, we use (35) to compute the admissible values of the n different predicted $V_{k(\bar{i})}$, and look at the largest one, again arbitrarily breaking ties if necessary. The rest of the singleton strategy follows along in the same manner as was described for doubletons.

The doubleton and singleton strategies are integrated in a "round robin" arrangement as follows. We make repeated passes through the two consecutive procedures until no advantageous reversals are found in a complete loop. When there are no changes made in either of the two procedures on consecutive passes, the optimization is terminated. If the alternating least squares procedure has worked well, then our combinatorial optimization typically finds no advantageous changes and therefore terminates after one "round robin" loop. On the other hand, when the iterative solution is not a good one, as many as five or six loops of combinatorial optimization can occur before no more improvement in the VAF is forthcoming.* By way of evaluation, we note that our approach to combinatorial optimization typically uses a nonnegligible amount of computer time, and sometimes produces a worthwhile increase in VAF, but not always. This somewhat disappointing state of affairs cannot, as far as we can determine, be ameliorated by known methods. We note that, in particular, an ostensibly promising method of Ivanescu and Rudeanu [1966] would require an explicit representation of w_k as a function of the remaining subsets and their weights (p_{il}, w_l, $l \neq k$) and is therefore of no use in the present situation. A more recent proposal by Banfield and Bassill [1977] is extremely limited in its scope of application. For instance, if the starting point were subsets constituting a partition of the stimuli (i.e., no overlap), the Banfield and Bassill approach would only test other partitions, without ever considering overlapping subsets.

In developing MAPCLUS and varying initial values of α, β, and λ on the same set(s) of data, we have occasionally found different solutions, each with over 90% VAF, using the same number m of subsets. The solutions occasionally differed by as little as .4% VAF (absolute) but had few subsets in common. Subjectively, these solutions were discrepant in that one or two stimuli seemed to "float" around to different subsets across the two solutions. This observation was in fact part of the motivation for developing the combinatorial optimization procedure just described. Unfortunately, that procedure has not eliminated the problem of nonunique (approximately) best solutions, and we realize that

* The variable run times that ensue from the appended combinatorial optimization preclude our giving any guidelines as to compute time required for specific combinations of n, m.

some data analysts will view this situation with discomfort. However, we have come to suspect that this nonuniqueness is inherent in a fairly wide class of discrete models such as ADCLUS. Consider, for example the competing proposals for sets of distinctive features for the perception of consonant phonemes [see Arabie & Soli, 1980; Soli & Arabie, 1979]. Formally, that research may be viewed as the comparison of different sets of discrete and (typically) correlated predictor variables applied to data from a specific psychological context. The fact that no consensus as to a preferred set of such features has yet to emerge (cf., Wang & Bilger, 1973) closely parallels the nonuniqueness of different MAPCLUS solutions, each with a highly acceptable VAF. Moreover, the nonuniqueness we believe to be inherent in ADCLUS and similar models is also present in the method of hierarchical clustering most frequently used by psychologists, namely the complete-link approach. Although many data analysts appear unaware of this fact, the latter method yields not a single tree or dendrogram, but rather a *class* of such structures, when certain patterns of tied proximity values are encountered [Hubert, 1973; Peay, 1975]. Although the complete-link procedure can be modified (see Hubert, 1973, or the first article describing complete-link clustering, Sørenson, 1948) to avoid this lack of uniqueness, most data analysts seem uninterested in doing so. Finally, we note that various results from the theory of hypergraphs can be used in principle to suggest that discrete representations such as those of the ADCLUS model may not be unique [Lawrence J. Hubert, Note 7].

Initial Configurations

The initial configuration for MAPCLUS simply refers to the starting values for the entries of matrix **P** in (2). There are three possible sources for this matrix: user-supplied entries, the output from a random number generator, and a rational strategy. We now consider each of these procedures in turn.

A user-supplied list of subsets ($p_{ik}(i = 1, \cdots, n), k = 1, \cdots, m$) may be read as input by the MAPCLUS program. Those values of **P**, plus the input proximities (transformed to be similarities, **S**) are sufficient for calling the multiple linear regression procedure (described above) and obtaining least squares estimates of the weights, w_k, and the additive constant.

In Shepard and Arabie [1978], the distinctive features proposed by E. J. Gibson [Gibson, Osser, Schiff, & Smith, 1963, Note 3] for confusions between upper-case Roman letters were used as an initial configuration. Those binary features defined **P**, and regression was then used to estimate the weights and evaluate the goodness-of-fit of those features vis-à-vis the model in (2). In "stand-alone" regression applications such as this one (i.e., no iterative computation is involved), it is necessary for the user-supplied subsets to be binary if the additive model is to be valid. In addition, each subset should be unique, and with cardinality greater than 1 and less than n.

There are of course many situations in which a user might wish to supply an initial configuration at the start of the iterative computation. In this type of usage, multiple linear regression is called after the initial configuration is read in, to obtain initial estimates of the weights and constant, prior to the first (prepolishing) major iteration. In such an application, there is no requirement that the user-supplied matrix **P** be binary.

For the second source of starting values of **P**, a random initial configuration is supplied in MAPCLUS by using Kruskal's [1969] random number generator that was designed for extreme portability across computers with different word lengths. The implementation of that random number generator in MAPCLUS is similar to the usage in the multidimensional scaling program KYST [Kruskal, Young, & Seery, 1973, Note 4].

Finally, our strategy for a rational initial configuration uses a simple linear approximation to fit a vector (which is a column of **P**) of bimodally distributed entries to the residuals for each subset.

Formally, we begin with

$$\mathbf{z} = \mathbf{1}\boldsymbol{\Delta}, \tag{42}$$

where there is a column vector **z** with an entry z_i for each of the n stimuli, **1** is a $1 \times n$ row vector of unities, and $\boldsymbol{\Delta}$ is the $n \times n$ symmetric and centered matrix of residuals with zeroes in the principal diagonal (i.e., $\delta_{ii} = 0$). Next, we center the entries of the **z** vector so that

$$\sum_{i=1}^{n} z_i = 0. \tag{43}$$

Then, assuming $\boldsymbol{\Delta}$ was not a double-centered matrix, we tally the number of positive elements in **z** as n_+ and negative ones as n_-. We then compute the mean, g_+, of the positive centered z_j values and, g_-, of the negative values. That is

$$g_+ = \frac{1}{n_+} \sum_{i=1}^{n} (z_i)^+ \quad \text{and} \quad g_- = \frac{1}{n_-} \sum_{i=1}^{n} (z_i)^- \tag{44}$$

where

$$(z_i)^+ = \begin{cases} z_i & \text{if } z_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad (z_i)^- = \begin{cases} z_i & \text{if } z_i < 0 \\ 0 & \text{otherwise.} \end{cases} \tag{45}$$

Our goal is to define

$$p_{ik} = a + bz_i, \tag{46}$$

where $p_{ik}(i = 1, \cdots, n)$ is each element in turn of subset k for which the rational initial configuration is being obtained. The coefficients a and b (unrelated to the same variable names used earlier) in this linear approximation are defined so that

$$a + bg_+ = 1 \quad \text{and} \quad a + bg_- = 0. \tag{47}$$

Thus

$$b = \frac{1}{g_+ - g_-} \quad \text{and} \quad a = \frac{-g_-}{g_+ - g_-}\cdot \tag{48}$$

After the values of a and b have been determined, (46) is used routinely to supply the initial values for subset k. Note in Figure 2 that the rational initial configuration is computed for one subset at a time, just before the very first outer iteration dealing with the k^{th} subset. Thus, when the initial configuration for the $(k + 1)^{\text{st}}$ subset is computed, the matrix of residuals $\boldsymbol{\Delta}$ has changed since the computation of the **z** vector for subset k. In practice, our strategy for a rational initial configuration has worked fairly well, and we also routinely use several different random initial configurations when analyzing any given set of data.

An anonymous referee has suggested an alternative means of obtaining a rational initial configuration, beginning with the two objects having the greatest similarity. To this dyad single stimuli are accrued in a stepwise manner until VAF fails to increase. The procedure is to be applied iteratively to the residuals.

Application to Confusions Between 16 Consonant Phonemes

Shepard and Arabie [1979] presented an ADCLUS representation of the data from the Miller-Nicely [1955] experimental investigation of subjects' errors of identification of 16 English consonant phonemes under different conditions of filtering and added noise. The (two-way) data from the so-called "flat noise" masking conditions have been studied by (two-way) multidimensional scaling methods [Arabie & Soli, 1980; Shepard, 1972] as well as by complete-link hierarchical clustering [Shepard, 1972].

Since psychologists and phoneticians have for many years sought a preferred set of discrete underlying perceptual features of consonant phonemes [Jakobson, Fant, & Halle, 1963; Klatt, 1968; Liberman, Cooper, Shankweiler, & Studdert-Kennedy, 1967; Wickelgren, 1966], ADCLUS seemed like a natural vehicle for pursuing such discrete features. Accordingly, Shepard and Arabie [1979] obtained the solution reproduced here as Table 1. The 16 subsets plus the additive constant accounted for 94.5% of the variance in the original confusions data. An extensive substantive discussion of this solution is given in

TABLE 1

Shepard-Arabie ADCLUS Solution for Confusions Between 16 Consonant Phonemes

Rank	Weight	Elements of Subset	Interpretation[a]
1	.730	f θ	front unvoiced fricatives
2	.575	d g	back voiced stops
3	.479	p t k	unvoiced stops
4	.464	p k	unvoiced stops, omitting t
5	.340	v ð	front voiced fricatives
6	.296	s θ	middle unvoiced fricatives
7	.281	m n	nasals
8	.267	b v ð	(front voiced consonants)
9	.197	s ʃ	back unvoiced fricatives
10	.191	p f θ	(front unvoiced consonants)
11	.190	z ð	middle voiced fricatives
12	.156	d g z ʒ	back voiced consonants
13	.153	b v	(front voiced consonants)
14	.114	v z ð	front and middle voiced fricatives
15	.081	g z ʒ	back voiced consonants
16	.009	z ʒ	back voiced fricatives

Note: The data are from Miller and Nicely [1955]. Variance accounted for = 94.5% with 16 subsets and additive constant (corresponding to the complete set of 16 consonants) = 0.057.

[a] Subsets of questionable interpretation are in parentheses.

Shepard and Arabie (see also below). For the moment, however, it should be noted that the representation in Table 1 came from the ADCLUS computer program, for which development was subsequently terminated. By way of review, that program generated the entire list of all maximal complete subgraphs from the input proximity matrix, and then used an iterative procedure for simultaneously estimating weights of the subsets while reducing the overall number of such subsets. Subsequently, multiple linear regression (details given above) was used to sharpen the estimates of the weights and additive constant.

It is naturally of interest to compare the performance of MAPCLUS and ADCLUS in analyzing the Miller-Nicely data. We therefore used MAPCLUS on these data, stipulating 16 subsets (plus an additive constant) and a rational initial configuration. The resulting MAPCLUS solution accounted for 98.1% of the variance. Although we will not be presenting the 16-cluster MAPCLUS solution, we note by way of comparison that the most heavily weighted clusters (and the pattern of their weights) compared favorably with the ADCLUS solution (Table 1), whereas for the least weighted subsets, the ADCLUS solution was substantively preferable. It has been our experience that for those data sets which were tractable to the ADCLUS program, MAPCLUS has typically given better fits using considerably fewer subsets. For example, the ADCLUS solution in Table 1 for the Miller-Nicely [1955] data represents the minimum number of clusters for which the ADCLUS program gave a convergent solution. (Owing to problems with the iterative adjustment of the weights and the lack of close control over the number of subsets present at any given stage of the iterative procedure, ADCLUS often failed to yield a usable solution for many sets of data.) To demonstrate the facility with which MAPCLUS usually requires fewer subsets, we present in Table 2 an 8-cluster solution accounting for 89.6% of the variance. These subsets are portrayed graphically in Figure 4, where they are embedded in a two-dimensional scaling configuration obtained by metric procedures developed by Shepard and Chang [Chang & Shepard, 1966, Note 5; Shepard, 1972]. In passing, we note that other MAPCLUS solutions with 8 clusters (plus the additive constant) sometimes gave still better fit (90.7% VAF) but were less interpretable.

In Table 2, the subsets have been rank ordered according to their fitted weights. To facilitate comparison with Table 1, the leftmost column of Table 2 shows the rank from the ADCLUS solution for the six subsets common to both the MAPCLUS and ADCLUS solutions.

Although brief interpretations of the MAPCLUS subsets are given in Table 2, further comments and comparisons with Table 1 (ADCLUS solution) are in order. Considering the most heavily weighted subsets, we note that the top two consist of parallel clusters of front unvoiced fricatives /*f θ*/ and front voiced fricatives /*v* ð/. A priori, these subsets are plausible contenders for the largest weights, since the stimuli within each pair are quite easily confused in the presence of masking noise [Wang & Bilger, 1973, p. 1252]. A similar pairing of clusters by weights holds for the third and fourth subsets, corresponding to back voiced stops /*d g*/ and unvoiced stops /*p t k*/. However, these two clusters are not completely parallel, since all the unvoiced stops /*p t k*/ are included in Cluster 4, whereas /*b*/ is excluded from the remaining back voiced stops /*d g*/ in Cluster 3, consistent with Shepard's [1972] results based on complete-link hierarchical clustering analyses of the same data. The voiced stop /*b*/, although similar phonetically to /*d g*/, is presumably segregated because, of those three voiced stops, /*b*/ alone has a rising second formant. It has been noted elsewhere that the shape of that formant's transition is a sufficient cue for distinguishing the voiced stops [Arabie & Soli, 1980; Soli & Arabie, 1979].

The fifth subset, /*b v*/, was noted by Shepard and Arabie [1979] to be of dubious interpretation. The fact that MAPCLUS gives that pair of front voiced consonants an even higher ranked (5) weight than did ADCLUS (13) is therefore all the more perplexing.

TABLE 2
MAPCLUS Solution for Confusions

Between 16 Consonant Phonemes

ADCLUS Rank[a]	MAPCLUS Rank	MAPCLUS Weight	Elements of Subset	Interpretation[b]
1	1	0.814	f θ	front unvoiced fricatives
5	2	0.729	v ð	front voiced fricatives
2	3	0.577	d g	back voiced stops
3	4	0.487	p t k	unvoiced stops
13	5	0.428	b v	(front voiced consonants)
4	6	0.348	p k	unvoiced stops, omitting t
-	7	0.162	b d gðz ʒ	voiced consonants, omitting v
-	8	0.116	p k f θ s ʃ	(unvoiced consonants, omitting t)

Note: As in Table 1, the data are from Miller and Nicely [1955]. Variance accounted for = 89.6% with 8 subsets (additive constant = 0.049).

[a] Ranks are given only for the six subsets common to both the ADCLUS and the MAPCLUS solutions.

[b] Subsets of questionable interpretation are in parentheses.

However, in general, it is fair to say that the weights of the top five MAPCLUS subsets are intuitively more appealing than those of the Shepard-Arabie solution (Table 1).

The sixth subset simply drops /*t*/ from the fourth. This grouping of /*k p*/ to form Subset 6 is not surprising, since both phonemes have relatively low frequency noise spectra at the time of burst release, unlike the corresponding high frequency for /*t*/ in the context of the vowel /*a*/ [Liberman, Delattre, & Cooper, 1952, p. 504]. (Speech spectrograms for the 16 phonemes are given in Carroll & Wish, 1974, and are also reprinted in Arabie & Soli, 1980, and Soli & Arabie, 1979.)

We now turn to the two least weighted subsets. Cluster 7 consists of the voiced consonants, omitting /*v*/. The exclusion of that phoneme is explained by the fact that, of the voiced consonant phonemes, /*v*/ alone has the weakest voiced formant transitions; thus, /*v*/ may be the most vulnerable to the masking noise [cf. Soli & Arabie, 1979]. The un-

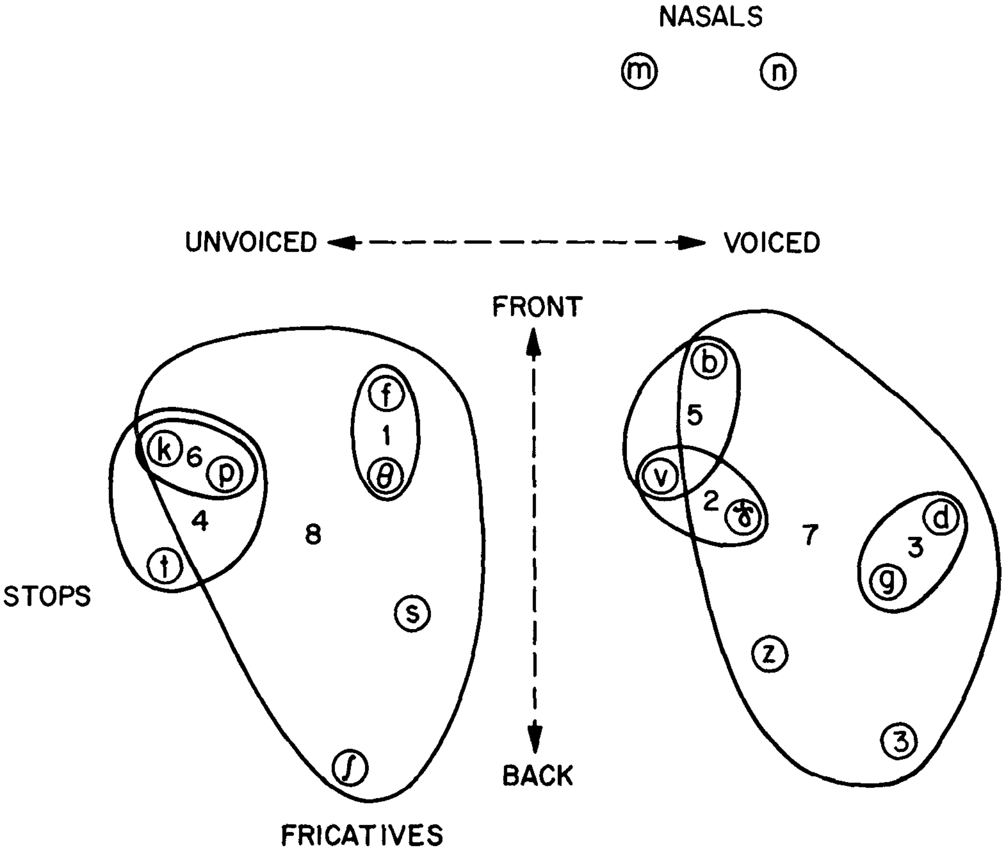


FIGURE 4
MAPCLUS solution for the 16 consonant phonemes used in the Miller-Nicely [1955] study. Variance accounted for = 89.6% with 8 subsets.

voiced consonant phonemes, excluding /*t*/, comprise the eighth subset. The same reasons given for the exclusion of /*t*/ from Subset 6 are again relevant to the composition of this cluster.

We suspect that some readers will be substantively perturbed, as we were, by the lack of a cluster for the nasals, /*m n*/, which were Subset 7 in the ADCLUS solution (Table 1). First, we note that each of the five clusters in Table 2 which were dyad subsets correspond to entries having greater pairwise confusability in the Miller-Nicely data [reprinted in Shepard, 1972, p. 75] than for /*m n*/. Thus, if the nasals were included instead of some of the substantively and traditionally less expected dyad subsets, the result would have been a decrease in VAF. However, the absence of the nasals subset in all of our 8-cluster solutions raises the question of how many clusters are required before this highly familiar pair is included. (Note that they are the seventh cluster in the ADCLUS solution of Table 1.) We therefore obtained a series of 10-cluster MAPCLUS solutions, accounting for 91.5 to 93.7% of the variance, and none of them included the nasals. We found that by specifying 12 clusters (and getting 94.1 to 95.6% VAF), /*m n*/ were sometimes present in the solutions. Thus, we suspect that the nasals are not as clearly present as a feature underlying the two-way Miller-Nicely data ("flat noise" conditions) as traditional phonology might insist. (In the 10- and 12-cluster solutions, the most heavily weighted subsets generally included the first six of those listed in the 8-cluster solution (Table 2), but the ordering of the weights was variable. The lesser weighted subsets tended to vary greatly in composition, although the cardinality was typically in the range of four to six.)

We would like to emphasize that with only 8 clusters MAPCLUS accounted for 89.6% of the variance, whereas the ADCLUS program required 16 to account for 94.5% and was unable to obtain a solution with fewer subsets. Computationally, MAPCLUS has a considerable advantage over the ADCLUS program of Shepard and Arabie [1979] with respect both to core requirements and computation involved in obtaining a rational initial configuration. These two considerations are closely related since the ADCLUS strategy of obtaining an exhaustive list of maximal complete subgraphs demands much computation and often results in matrices of large dimensions. Moreover, the computational advantage of the MAPCLUS algorithm mounts dramatically with increasing *n*.

Future Prospects: INDCLUS as the Three-Way Generalization

We are currently extending this general approach to the three-way case. The three-way model, called INDCLUS (for *IN*dividual *D*ifferences *CLUS*tering) is of the form:

$$s_{ij}^{h} \cong \sum_{k=1}^{m} w_{hk} p_{ik} p_{jk} + c_h, \tag{49}$$

where s_{ij}^{h} is the similarity between stimuli (or other objects) i and j for subject (or other data source) h, w_{hk} is a weight for subject h on subset k (closely analogous to INDSCAL dimension weights), while c_h is a constant (or weight for the complete subset) for subject h. A fairly straightforward extension of the MAPCLUS procedure allows fitting of this INDCLUS model [Carroll & Arabie, Note 6, 1979]. We hope that this model and method will have many of the same advantages that the INDSCAL model and method have for three-way multidimensional scaling. One particular advantage we anticipate is a somewhat greater stability of the solution (i.e., of subset definition). The reasoning on which this is based is that the (two-way) ADCLUS model may suffer from a quasi-indeterminacy analogous in some ways to a rotational problem in two-way MDS. This quasi-indeterminacy may in fact underlie the general tendency of this model (and other discrete models such as the Carroll-Pruzansky multiple tree structure model) to exhibit multiple local optima having about the same value of the least squares (or other) loss function measuring goodness-of-fit. The three-way INDCLUS model, it is hoped, may have a property analogous to the "dimensional uniqueness" property of INDSCAL that may tend to diminish or eliminate this tendency to multiple solutions. In addition, many of the other advantages of INDSCAL should accrue to INDCLUS—in particular, the differential subject weights may prove to be important individual differences parameters.

REFERENCE NOTES

1. Shepard, R. N., & Crawford, G. *Multidimensional scaling based on the fitting of constrained difference functions.* Paper presented at the U.S.-Japan Seminar on Multidimensional Scaling, University of California at San Diego, La Jolla, California, August 20–24, 1975.
2. 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 Multidimensional Scaling, University of California at San Diego, La Jolla, California, August 20–24, 1975.
3. Gibson, E. J., Osser, H., Schiff, W., & Smith, J. An analysis of critical features of letters, tested by a confusion matrix. In, *A basic research program on reading.* (Cornell University and United States Office of Education Cooperative Research Project No. 639.), Ithaca, N. Y.: Cornell University, 1963.
4. Kruskal, J. B., Young, F. W., & Seery, J. B. *How to use KYST, a very flexible program to do multidimensional scaling and unfolding.* Murray Hill, N. J.: Bell Telephone Laboratories, 1973.
5. Chang, J. J., & Shepard, R. N. *Exponential fitting in the proximity analysis of confusion matrices.* Presented at the annual meeting of the Eastern Psychological Association, New York, April 14, 1966.

6. Carroll, J. D., & Arabie, P. *INDCLUS: A three-way approach to clustering.* Paper presented at meeting of Psychometric Society, Monterey, CA, 1979.
7. Hubert, Lawrence J. Personal communication, 1978.

REFERENCES

Adby, P. R., & Dempster, M. A. H. *Introduction to optimization methods.* New York: Wiley, 1974.

Arabie, P., & Soli, S. D. The interface between the type of regression and methods of collecting proximities data. In R. Golledge & J. N. Rayner (Eds.), *Multidimensional analysis of large data sets.* Minneapolis: University of Minnesota Press, 1980.

Banfield, C. F., & Bassill, L. C. A transfer algorithm for non-hierarchical classification. *Journal of the Royal Statistical Society (Series C), Applied Statistics,* 1977, *26,* 206–210.

Bunch, J. R., Kaufman, L., & Parlett, B. N. Decomposition of a symmetric matrix. *Numerische Mathematik,* 1976, *27,* 95–109.

Carroll, J. D., & Arabie, P. Multidimensional scaling. In M. R. Rosenzweig & L. W. Porter (Eds.), *Annual review of psychology.* Palo Alto, CA: Annual Reviews, 1980.

Carroll, J. D., & Chang, J.-J. Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition. *Psychometrika,* 1970, *35,* 283–319.

Carroll, J. D., & Pruzansky, S. Discrete and hybrid scaling models. In E. D. Lantermann & H. Feger (Eds.), *Proceedings of Aachen symposia on decision making and multidimensional scaling.* Berlin: Springer-Verlag, 1980.

Carroll, J. D., & Wish, M. Models and methods for three-way multidimensional scaling. In D. H. Krantz, R. C. Atkinson, R. D. Luce, & P. Suppes (Eds.), *Contemporary developments in mathematical psychology* (Vol. II). San Francisco: Freeman, 1974.

Cunningham, J. P., & Shepard, R. N. Monotone mapping of similarities into a general metric space. *Journal of Mathematical Psychology,* 1974, *11,* 335–363.

Hubert, L. Some extensions of Johnson's hierarchical clustering algorithms. *Psychometrika,* 1972, *37,* 261–274.

Hubert, L. J. Monotone invariant clustering procedures. *Psychometrika,* 1973, *38,* 47–62.

Hubert, L. J., & Baker, F. B. Analyzing distinctive features. *Journal of Educational Statistics,* 1977, *2,* 79–98.

Ivanescu, P. L., & Rudeanu, S. *Pseudo-Boolean methods for bivalent programming. Lecture notes in mathematics,* (Vol. 23). New York: Springer-Verlag, 1966.

Jakobson, R., Fant, G. M., & Halle, M. Preliminaries to speech analysis: The distinctive features and their correlates. Cambridge, Mass.: MIT Press, 1963.

Katz, L. On the matric analysis of sociometric data. *Sociometry,* 1947, *10,* 233–241.

Klatt, D. H. Structure of confusions in short-term memory between English consonants. *Journal of the Acoustical Society of America,* 1968, *44,* 401–407.

Krumhansl, C. L. Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. *Psychological Review,* 1978, *85,* 445–463.

Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika,* 1964, *29,* 1–27. (a)

Kruskal, J. B. Nonmetric multidimensional scaling: A numerical method. *Psychometrika,* 1964, *29,* 115–129. (b)

Kruskal, J. B. An extremely portable random number generator. *Communications of the ACM,* 1969, *12,* 93–94.

Kruskal, J. B. Multidimensional scaling and other methods for discovering structure. In K. Enslein, A. Ralston, & H. S. Wilf (Eds.), *Statistical methods for digital computers* (Vol. 3). New York: Wiley-Interscience, 1977.

Kruskal, J. B., & Carroll, J. D. Geometrical models and badness-of-fit functions. In P. R. Krishnaiah (Ed.), *Multivariate analysis II.* New York: Academic Press, 1969.

Kruskal, W. H. The geometry of generalized inverses. *Journal of the Royal Statistical Society (Series B),* 1975, *37,* 272–283.

Liberman, A. M., Delattre, P., & Cooper, F. S. The role of selected stimulus-variables in the perception of the unvoiced stop consonants. *American Journal of Psychology,* 1952, *65,* 497–516.

Liberman, A. M., Cooper, F. S., Shankweiler, D. P., & Studdert-Kennedy, M. Perception of the speech code. *Psychological Review,* 1967, *74,* 431–461.

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

Moon, J. W., & Moser, L. On cliques in graphs. *Israel Journal of Mathematics,* 1965, *3,* 23–28.

Peay, E. R. Nonmetric grouping: Clusters and cliques. *Psychometrika,* 1975, *40,* 297–313.

Podgorny, P., & Garner, W. R. Reaction time as a measure of inter- and intraobject visual similarity: Letters of the alphabet. *Perception & Psychophysics,* 1979, *26,* 37–52.

Ramsay, J. O. Maximum likelihood estimation in multidimensional scaling. *Psychometrika,* 1977, *42,* 241–266.

Rao, C. R., & Mitra, S. K. *Generalized inverse of matrices and its applications.* New York: Wiley, 1971.

Robinson, S. M. Quadratic interpolation is risky. *SIAM Journal of Numerical Analysis,* 1979, *16,* 377–379.

Roethlisberger, F. J., & Dickson, W. J. *Management and the worker.* Cambridge, Massachusetts: Harvard University Press, 1939.

Shepard, R. N. Analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika,* 1962, *27,* 125–140. (a)

Shepard, R. N. Analysis of proximities: Multidimensional scaling with an unknown distance function. II. *Psychometrika,* 1962, *27,* 219–246. (b)

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

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

Soli, S. D., & Arabie, P. Auditory versus phonetic accounts of observed confusions between consonant phonemes. *Journal of the Acoustical Society of America,* 1979, *66,* 46–59.

Sørenson, T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter,* 1948, *5,* 1–34.

Tversky, A. Features of similarity. *Psychological Review,* 1977, *84,* 327–352.

Wang, M. D., & Bilger, R. C. Consonant confusions in noise: A study of perceptual features. *Journal of the Acoustical Society of America,* 1973, *54,* 1248–1266.

Wickelgren, W A. Distinctive features and errors in short-term memory for English consonants. *Journal of the Acoustical Society of America,* 1966, *39,* 388–398.

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

Manuscript received 7/19/78
First revision received 9/11/79
Final version received 1/3/80