XIV. THE DETERMINATION OF CLUSTERS BY MATRIX ANALYSIS

A. Richard LeSchack

ABSTRACT

The use of clustering techniques related to factor analysis is discussed in this section. The application of matrix spectral analysis to the detection of clusters is based on a matrix model for strong clustering. The methods used and the programs developed to implement them on the IBM 7090 computer are described; detailed test results are given; and attempts at mechanizing the decision procedure are treated.

1. Introduction

Taxonomy, the science of classification, is an activity fundamental not only to descriptive biology but to many other areas of scientific and intellectual endeavor as well. Categorizing individuals - assigning them to one of a tractably small number of subpopulations, the members of which are sufficiently alike to justify ignoring individual differences - suppresses the inessential detail which obscures underlying relationships. The result is a more economical description of those features of the population which are truly significant.

A particular case of the classification problem is considered here. The basic raw data are estimates of the pairwise similarity between members of the population. It is desired to identify subsets in which the members are so similar to one another, and so dissimilar to nonmembers, that it is most useful to consider them all as a unit. This process has been termed "clumping" or "clustering." It is clear that an exact formulation of the clustering problem must be in terms of optimization; for the goal is the best balance between loss of precision, which is the inevitable consequence of representing each individual by the typical or average characteristics of his cluster, and gain in economy, which is the result of replacing many individuals by a few clusters. This general formulation is not pursued further; we limit ourselves to those situations of strong clustering in which the adherence of individuals to subsets is relatively unambiguous. It is shown that under certain assumptions regarding the nature of the strong clustering, techniques of matrix spectral analysis will identify the members of the various clusters. These techniques are extended — with empirical success, although without rigorous justification — to the general case in which the assumption of strong clustering is not so clearly tenable.

A number of techniques for determining clusters have been disussed in the literature (see Bonner[3] for a review of many of them): these include clump theory (Parker-Rhodes and Needham),[11] factor analysis (Bonner;[3] Borko and Bernick)[4] and latent class analysis (Baker).[1] The present algorithm bears a close resemblance to factor analysis, but differs from this well-known statistical technique in that it does not require that the similarity data be given in terms of the classical product-moment correlation coefficient. It is based on a different mathematical model (the theory of reducible matrices of nonnegative elements) and is directed toward a different goal.

A few of the useful shorthand notations of the Iverson programming language are used in formulas and flowcharts throughout this section. The reader is referred to Brooks and Iverson[5] for the general features of the notation, and to Iverson[8] for a detailed presentation.

2. A Model for Strong Clustering

The intuitive notion of perfect clustering is best explained by considering a population of elements which can be partitioned into non-overlapping subsets, such that within each subset there is a nonzero linkage between each pair of elements, but no linkage between a member of the subset and an element which is not a member. To generalize to the idea of strong clustering, assume that we are able to distinguish between strong and weak links; the definition is then modified to read:

> ... within each subset there is a strong linkage between each pair of elements, but at most a weak linkage between a member of the subset and an element which is not a member.

This part presents the theory of reducible matrices as a model for the case of perfect clustering, and generalizes the results to the case of strong clustering. The properties of the eigenvectors of such matrices justify the use of spectral analysis to determine strong clusters. The exact boundaries between "strong" and "weak," and the rigorous justification of these techniques in the case of clustering which is not strong, remain open questions.

A square n × n matrix $\underline{A}$ is *reducible* (decomposable, uncoupled) if its rows and columns can be permuted to yield a form partitionable as follows:

$$\left[\begin{array}{c|c} \underline{X} & \underline{Y} \\ \hline \underline{0} & \underline{Z} \end{array}\right]$$

where $\underline{0}$ is a matrix of all zeros, and $\underline{X}$ and $\underline{Z}$ are square matrices. More formally, we may state the definition in any of these equivalent forms:

(1) $\underline{A}$ is *reducible* $\Longleftrightarrow$ there exists a permutation matrix $\underline{P}$ such that

$$\underline{PAP}^T = \left[\begin{array}{c|c} \underline{X} & \underline{Y} \\ \hline \underline{0} & \underline{Z} \end{array}\right]$$

(2) $\underline{A}$ is *reducible* $\Longleftrightarrow$ there exists a pair of integers (i,j) such that there exists no chain

$$a_{ip_1} a_{p_1 p_2} \cdots a_{p_r j}$$

with all terms nonzero.

$$(i, j, p_k \leq n)$$

(3) A is reducible $\Longleftrightarrow$ the set $\{1,2,\ldots,n\}$ can be partitioned into mutually exclusive, collectively exhaustive subsets S and T, such that whenever $i \in S$ and $j \in T$, then $a_{ij} = 0$.

The usual correspondence between a matrix and a directed graph† leads to the conclusion that $\underline{A}$ is reducible if and only if the corresponding directed graph is *not* strongly connected, that is, if there are two nodes p and q for which no path leads from p to q.

If the matrix $\underline{A}$ is symmetric, then the definitions take on this stronger form: The symmetric matrix $\underline{A}$ is *reducible* if and only if there exists a permutation matrix $\underline{P}$ such that

$$\underline{B} = \underline{PAP}^T = \begin{array}{cc} \begin{array}{cc} r & n-r \end{array} & \\ \left[\begin{array}{c|c} \underline{X} & \underline{0} \\ \hline \underline{0} & \underline{Y} \end{array}\right] & \begin{array}{c} r \\ n-r \end{array} \end{array}$$

where $\underline{X}$ and $\underline{Y}$ are square. Such a matrix is block-diagonal. It is obvious that the eigenvectors of $\underline{B}$ fall into two classes: those with the last (n-r) components equal to 0, and those with the first r components equal to 0. The Jordan canonical form

$$\underline{B} = \underline{S}\underline{\Lambda}\underline{S}^T$$

with

$$\underline{S} = \left(\left[\underline{s}^1\right]\left[\underline{s}^2\right]\left[\underline{s}^3\right] \cdots \left[\underline{s}^n\right]\right)$$

(where each $\underline{s}^i$ is a normalized eigenvector) reflects the property of reducibility by taking on the special form

† $a_{ij} \neq 0$ in the matrix if and only if an arc connects nodes i and j in the graph.

$$\begin{array}{cc} & \begin{array}{cc} r & n-r \end{array} \\ \underline{S} = & \left[\begin{array}{c:c} \underline{S}_{(1)} & \underline{0} \\ \hdashline \underline{0} & \underline{S}_{(2)} \end{array}\right] \begin{array}{c} r \\ n-r \end{array} \end{array}$$

The eigenvectors of $\underline{A}$ may be expressed as follows:

$$\underline{A} = \underline{P}^T\underline{BP} = \underline{P}^T\underline{S\Lambda S}^T\underline{P} = \underline{Q\Lambda Q}^T \text{ where } \underline{Q} = \underline{P}^T\underline{S}$$

By writing $\underline{\Lambda}' = \underline{P}^T\underline{\Lambda P}$ we obtain

$$\underline{A} = (\underline{P}^T\underline{SP})\underline{\Lambda}'(\underline{P}^T\underline{SP})^T$$

which involves only a reordering of the eigenvalues. It follows that the eigenvectors of $\underline{A}$ fall into two classes: one with zero in r positions, the other with zero in the remaining (n-r) positions.

Let us now make the further assumption that all elements of $\underline{A}$ are nonnegative, and that each of the matrices $\underline{X}$ and $\underline{Y}$ is irreducible. Even more generally, assume that $\underline{A}$ has been completely reduced to the form

$$\underline{B} = \underline{PAP}^T = \left[\begin{array}{c:c:c} \underline{X} & \underline{0} & \underline{0} \\ \hdashline \underline{0} & \underline{Y} & \underline{0} \\ \hdashline \underline{0} & \underline{0} & \underline{Z} \ddots \end{array}\right]$$

where each of the diagonal blocks is irreducible.

The eigenvalues of $\underline{A}$ are clearly those of $\underline{B}$; $\underline{B}$, however, has as its set of roots

$$\{\lambda(\underline{X})\} \cup \{\lambda(\underline{Y})\} \cup \{\lambda(\underline{Z})\} \cup \ldots$$

The classical theorem of Perron and Frobenius, applied to the matrices $\underline{X},\underline{Y}$, etc., yields the result that each of the irreducible matrices along the main diagonal has a positive maximum eigenvalue to which corresponds an eigenvector, all of whose components are nonnegative. The other eigenvalues are all strictly smaller, and the other eigenvectors do not, in general, possess the property of nonnegativity. This fact implies that if the eigenvalues of the entire matrix are obtained, one at a time, in descending order according to magnitude, the largest eigenvalue will be the maximum eigenvalue of one of the submatrices. The corresponding eigenvector has positive elements in the positions corresponding to the members of the index set identifying the submatrix. Adopting the terminology of the Iverson programming language (see Iverson[8] or Brooks and Iverson[5] for details), we regard the logical vector $(\underline{s} > \underline{0})$[†] as a selection vector[‡] for membership in one of the clusters.

The next largest eigenvalue may belong to the same submatrix (in which case the corresponding eigenvector either has negative elements, or else selects the same cluster again), but it is much more likely that this second eigenvalue is the maximum root for one of the other submatrices; in this case, its eigenvector selects the elements of a second cluster in the same manner as the first. In this way the eigenvectors corresponding to the

[†] A logical vector $\underline{u}$ such that $\underline{u}_i = 1$ if and only if $\underline{s}_i > 0$.

[‡] In the sense that $\underline{u}$ selects those elements of the vector $(1,2,3,\ldots,n)$ corresponding to its nonzero elements.

eigenvalues obtained in descending order of magnitude identify systematically the clusters originally present.

For the binary case, where each $a_{ij} = 0$ or 1, the submatrix

$$\begin{array}{cc} & r \\ r & \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & & & \vdots \\ \vdots & & & \vdots \\ 1 & \cdots & & 1 \end{bmatrix} \end{array}$$

has an eigenvalue r, and an (r-1)-fold root of 0. If there are p clusters, each with n_i members:

$$n_1 + n_2 + \dots + n_p = n$$

then there are exactly p nonzero eigenvalues:

$$n_1, n_2, \dots, n_p$$

to which correspond eigenvectors selecting the p clusters.

The <u>fully-reducible</u> matrices just discussed provide a model of perfect or ideal clustering, where each member of a cluster is linked to other members of the same cluster but to no nonmember of the cluster. To bridge the gap between this ideal model and those actual cases of strong clustering in which there are large links and small, but not zero, links, consider a matrix in which the zero elements of the fully reducible case are replaced by small quantities ϵ. The result will be called an <u>ϵ-reducible</u> matrix. Its properties are deducible from those of the reducible matrix by an application of matrix perturbation theory, for a systematic development of which the reader is referred to Bellman,[2] pp. 60 ff.

Let the original reducible matrix be $\underline{R}$. The matrix $\underline{P}$ describes the pattern of perturbations. Thus

$$\underline{R}_{(\epsilon)} = \underline{R} + \epsilon \underline{P}$$

is the perturbed, or ϵ-reducible, matrix. Let λ_i be the eigenvalues of $\underline{R}$; the corresponding normalized eigenvectors (which form a complete orthonormal set as a consequence of the earlier assumption that $\underline{R}$ be symmetric) are $\underline{x}^{(i)}$; finally, let μ_i be the eigenvalues of $\underline{P}$. Two cases are considered: the special case in which $\underline{R}$ and $\underline{P}$ commute, and then the general case.

If $\underline{R}$ and $\underline{P}$ commute, then they have a common basis; that is, any eigenvector of one is an eigenvector of the other. From this it follows immediately that the eigenvectors of the perturbed matrix $\underline{R}_{(\epsilon)}$ are the same as those of the original matrix $\underline{R}$; to each eigenvalue λ_i of $\underline{R}$ corresponds an eigenvalue

$$\lambda_i + \epsilon\mu_i$$

of $\underline{R}_{(\epsilon)}$.

The general case is treated by expressing the eigenvalues and eigenvectors of the perturbed matrix as power series in the quantity ϵ. Retaining only first-order terms, we obtain these results:

$$\text{eigenvalues of } \underline{R}\text{:} \quad \lambda_i + \epsilon(\underline{x}^{(i)^T} \underline{P}\underline{x}^{(i)})$$

$$\text{eigenvectors of } \underline{R}\text{:} \quad \underline{x}^{(i)} + \epsilon \sum_{\substack{j=1 \\ j\neq i}}^{n} c_j \underline{x}^{(j)}$$

where

$$c_j = \frac{\underline{x}^{(j)^T} \underline{P}\underline{x}^{(i)}}{\lambda_j - \lambda_i}$$

Note that these reduce to the results of the special case if the $\underline{x}^{(i)}$ are also eigenvectors of $\underline{P}$.

We conclude that, to first order in ϵ, the elements of any eigenvector are changed (if at all) by a term of order ϵ, thus preserving the structure of the eigenvectors noted above.

3. Spectral Analysis and Factor Analysis

The key procedure in the computer program for determining clusters is the extraction of the characteristic roots of a matrix $\underline{R}$ in decreasing order of magnitude. The most appropriate algorithm in this case is the so-called power method, using the Rayleigh quotient at each iteration to estimate the root. (See Fadeeva,[6] Chapter 3.)

To describe the process briefly, let us assume that the first p eigenvalues have been determined. We desire to evaluate λ_{p+1}. An initial vector $\underline{y}^{(0)}$ is constructed, orthogonal to each of the first p eigenvectors. This is done in practice by subtracting the vector

$$\sum_{i=1}^{p} (\underline{\epsilon}, \underline{x}^{(i)}) \underline{x}^{(i)}$$

from an initial vector ϵ, each component of which is unity. Here $\underline{x}^{(i)}$ is the normalized eigenvector corresponding to λ_i. The resulting $\underline{y}^{(0)}$ is clearly orthogonal to all of the $\underline{x}^{(i)}$. One now forms the sequences:

$$\underline{y}^{(0)}$$

$$\underline{y}^{(i+1)} = \underline{R}\underline{y}^{(i)}$$

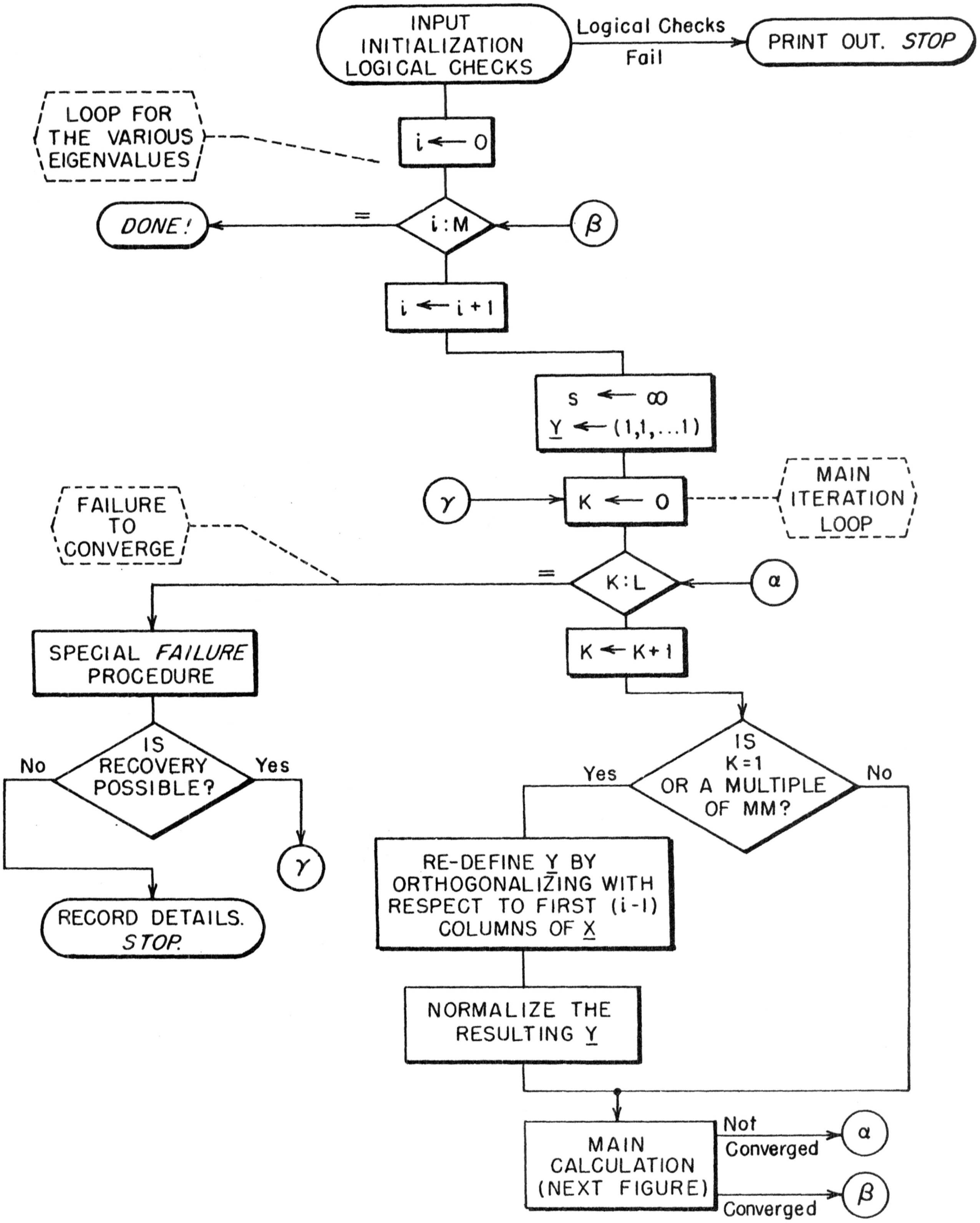
and

$$s^{(0)} = +\infty$$

$$s^{(i+1)} = \frac{(\underline{y}^{(i+1)},\underline{y}^{(i)})}{(\underline{y}^{(i)},\underline{y}^{(i)})}$$

where the usual notation $(\underline{x},\underline{y})$ for dot product is used. When the sequence s has converged to the desired degree of accuracy, we take the final value of s as λ_{p+1}, and the final vector $\underline{y}^{(i+1)}$, suitably normalized, as the eigenvector corresponding to λ_{p+1}. In actual computations, numerical rounding errors reintroduce into $\underline{y}^{(i)}$ components along one or more of the vectors $\underline{x}^{(1)},\ldots,\underline{x}^{(p)}$; this may cause the process to converge back to an earlier eigenvalue (usually λ_1) instead of λ_{p+1}. Such parasitic behavior is inhibited by occasional reorthogonalization of $\underline{y}^{(i)}$ to remove any such components. At the same time, it is desirable to renormalize the $\underline{y}^{(i)}$ from time to time, to prevent possible computer overflow in case λ_1 (for example) is quite large.

The basic process, for the case where a fixed number M of roots is to be extracted, is outlined in Flowcharts 1 and 2.



Determination of Dominant Eigenvalues
and Corresponding Eigenvectors
(Program Organization)

Flowchart 1

Legend to Flowchart 1

1. Parameters

M: number of eigenvalues to be determined

L: maximum number of iterations

MM: how often to reorthogonalize and normalize

2. Variables

i: control index; the ith eigenvalue is being determined

k: iteration counter

s: previous value of the Rayleigh quotient

3. Arrays

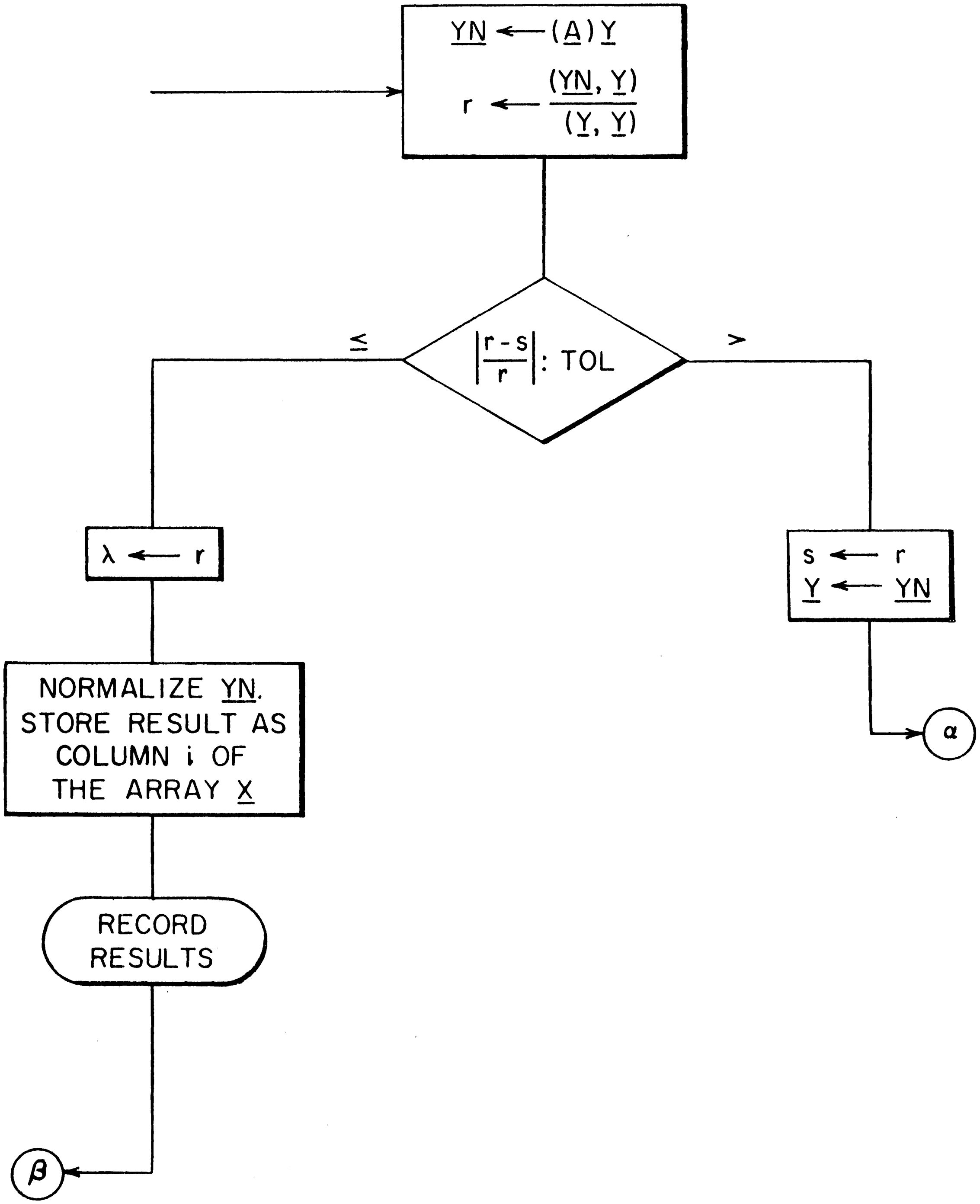
Y: current vector iterate

X: the ith normalized eigenvector, as soon as it has been determined, is stored as the ith column of the matrix X.

The close relation between this algorithm and the principal factor solution, or method of principal axes, in factor analysis will be discussed briefly. In essence, the present algorithm, if applied to a matrix of product-moment correlations, will yield the principal factor solutions to the factor analysis problem.

Factor analysis, developed originally as a mathematical model for certain psychological theories of human abilities and behavior, has become a standard tool of multivariate statistical analysis. Although no attempt can be made here to discuss the development of methods or their applications[†]

[†] For this, refer to Harmon.[7]


$\underline{YN} \leftarrow (\underline{A})\underline{Y}$
$r \leftarrow \frac{(\underline{YN}, \underline{Y})}{(\underline{Y}, \underline{Y})}$
$\left|\frac{r-s}{r}\right|$: TOL
≤
>
$\lambda \leftarrow r$
$s \leftarrow r$
$\underline{Y} \leftarrow \underline{YN}$
NORMALIZE YN. STORE RESULT AS COLUMN i OF THE ARRAY X
RECORD RESULTS
α
β

Determination of Dominant Eigenvalues
and Corresponding Eigenvalues
(Main Calculation)

Flowchart 2

Legend to Flowchart 2

1. Parameters

TOL: convergence criterion for successive iterates

2. Variables

s: previous value of the Rayleigh quotient
r: current value of the Rayleigh quotient

3. Arrays

A: the matrix to be analyzed
Y: the previous vector iterate
YN: the current vector iterate
X: the eigenvector just determined, after normalization, is stored as the ith column of the matrix X.

a brief outline of the fundamental factor analysis problem may make clear the relation between factor analysis and spectral analysis.

Consider the following: given the symmetric, nonnegative definite matrix $\underline{R}(n \times n)$, find an $(n \times m)$ matrix $\underline{A}^{(m)}$, with m as small as possible, for which

$$\underline{R}' = \underline{A}\underline{A}^T$$

is sufficiently close to $\underline{R}$. If we express $\underline{R}$ in Jordan canonical form as

$$\underline{R} = \underline{S}\underline{\Lambda}\underline{S}^T$$

where $\underline{S}$ is orthogonal and $\underline{\Lambda}$ is a diagonal matrix displaying the eigenvalues of $\underline{R}$, ordering the elements of $\underline{\Lambda}$ and the corresponding columns of $\underline{S}$ (which are the normalized eigenvectors of $\underline{R}$) so that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$$

then we may write

$$\underline{R} = \underline{T}\underline{T}^T$$

where

$$\underline{T} = \underline{S}\underline{\Lambda}^{1/2}$$

and

$$\underline{\Lambda}^{1/2} = \begin{pmatrix} \lambda_1^{1/2} & & & 0 \\ & \lambda_2^{1/2} & & \\ & & \ddots & \\ 0 & & & \lambda_n^{1/2} \end{pmatrix}$$

Each matrix $\underline{A}^{(m)}$ is formed, clearly, by taking the first m columns of $\underline{T}$. If the matrix $\underline{R}$ is of rank p (less than n), then at m = p the product

$$\underline{R}' = \underline{A}^{(m)}\underline{A}^{(m)^T}$$

reproduces $\underline{R}$ exactly, and the process terminates. In the usual applications of factor analysis, the given matrix $\underline{R}$ differs slightly, because of sampling errors, from an ideal matrix $\underline{\tilde{R}}$ of rank p, less than n (in fact, usually $p << n$). Both p and $\underline{\tilde{R}}$ are initially unknown. When successive factors cease to improve the representation, it may be assumed that p has been reached, and the current $\underline{R}'$ is taken as the best estimate of $\hat{\underline{R}}$.

Finally, it will be seen that the method of spectral analysis described here may be considered as "using factor analysis to determine clusters" if, and only if, two conditions are satisfied:

(1) the pairwise similarities are expressed by the product-moment correlation, and

(2) the factors turn out to have the properties assumed for the eigenvectors of "almost separable" matrices.

4. Organization of Programs for Finding Eigenvalues and Eigenvectors

The fundamental matrix and vector manipulations which form the innermost loops of the program were coded (for the IBM 7090 computer) in the FAP language, to ensure greater speed and accuracy. The remaining subprograms are coded in FORTRAN. NORM, ORTHOG, RAQUO, and SMPY are the entries to the FAP-coded vector-matrix package (DOT), based on a double precision accumulation of dot products.

Subprograms

NORM: normalizes the input vector to length one.

ORTHOG: orthogonalizes the input vector with respect to any desired number of stored normalized vectors.

RAQUO: computes the Rayleigh quotient of two input vectors.

SMPY: does matrix-vector multiplication.

Main: the calling program which uses this routine to extract factors, determine clusters, etc.

BOSS: controls the basic logic of the successive determination of eigenvalues and vectors.

CALC: performs one iteration of the "main calculation"

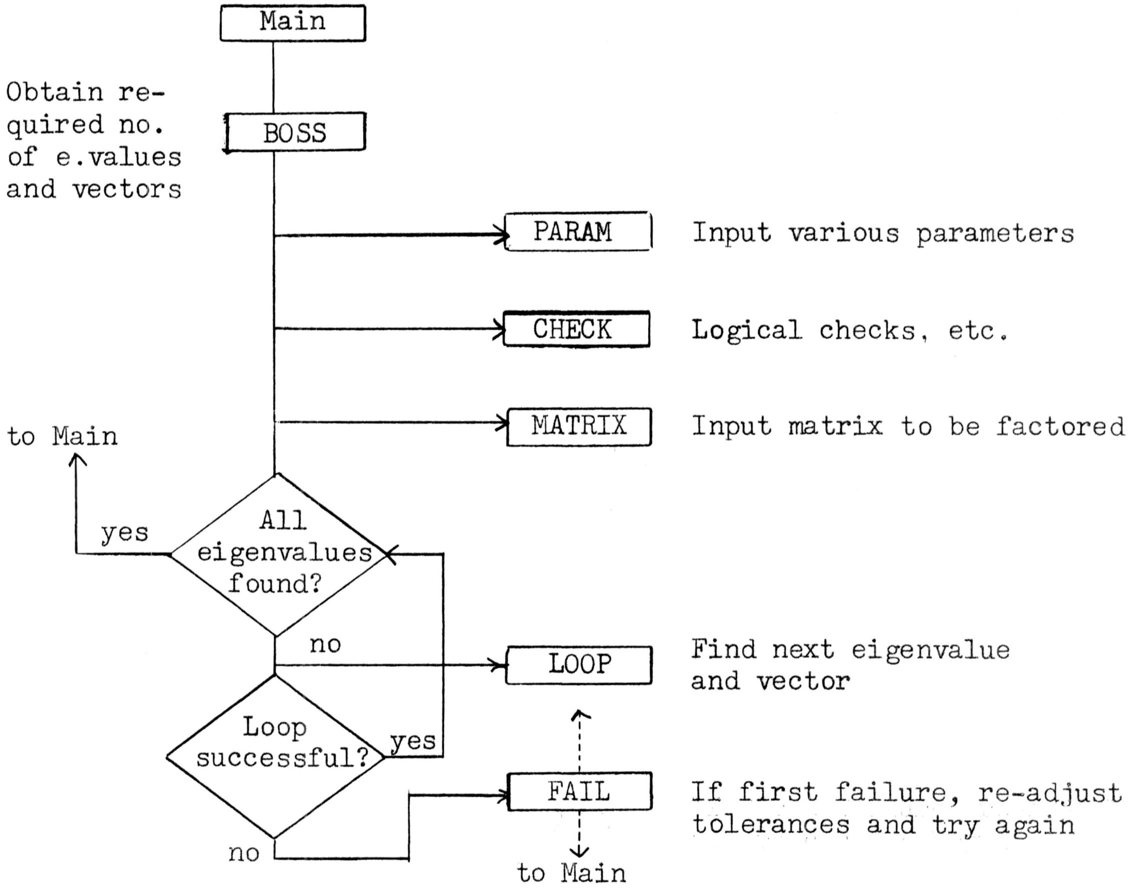
PARAM: inputs control parameters for the run

CHECK: makes logical and consistency checks on parameters and input values

LOOP: controls the iterations to find one eigenvalue

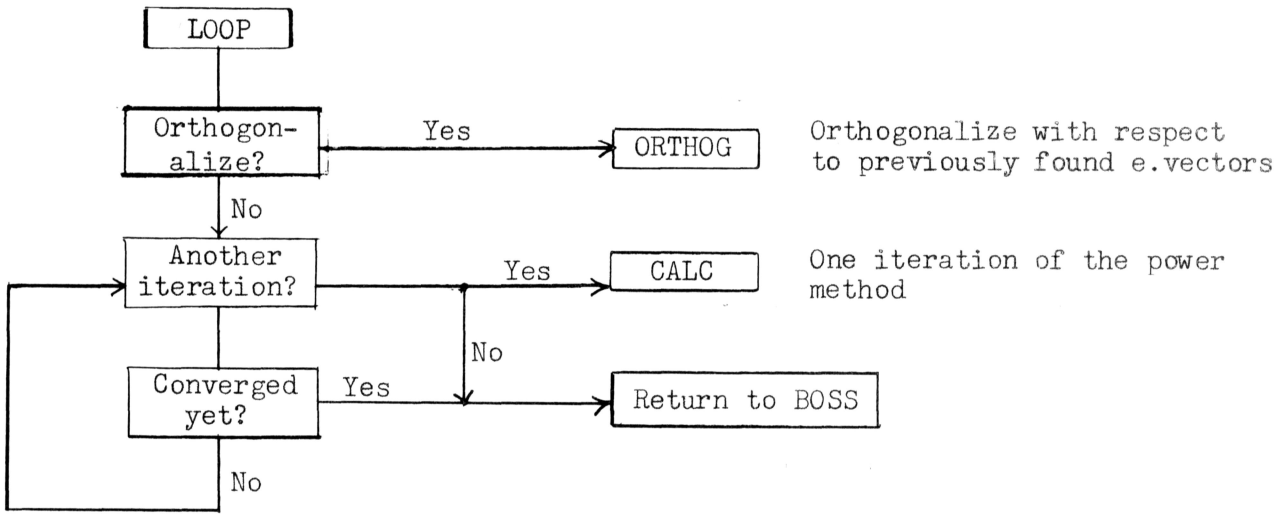
FAIL: attempts corrective action in case iterations fail to converge

MATRIX: computes or stores the matrix to be analyzed


Main
Obtain required no. of e.values and vectors
BOSS
PARAM
Input various parameters
CHECK
Logical checks, etc.
MATRIX
Input matrix to be factored
to Main
yes
All eigenvalues found?
no
LOOP
Find next eigenvalue and vector
Loop successful?
yes
FAIL
If first failure, re-adjust tolerances and try again
no
to Main

Logical Organization Eigenvalue-vector Program

Flowchart 3



Logical Organization
Eigenvalue-Vector Program (Inner Loop)

Flowchart 4

5. Program Test

The program was tested by using it to determine, to six-figure accuracy, the first several eigenvalues and eigenvectors of three matrices which have been used as examples in the literature of factor analysis. Each of the matrices was factored once with unities on the main diagonal, once with estimates (or exact values) of the communalities, giving six tests in all. Results agreed in every case with published values. The following summarizes the tests.

Test Matrix	Size	Number of Factors
Harmon, Table 8.5, p. 142,[†] a classical example involving interrelationships among results of a battery of psychological tests for children	13 X 13	5 factors (unities) 5 factors (communalities)
Harmon, Table 5.6, p. 91, a contrived "textbook example"	6 X 6	3 factors (unities) 2 factors (communalities)
Harmon, Table 9.1, p. 164, an actual example involving correlations among various physical measurements	8 X 8	3 factors (unities) 3 factors (communalities)

The following statistics, for the first of the cases listed above, are typical of the performance of the program.

Eigenvalue Number	Eigenvalue	Number of Iterations
1	5.066688	7
2	1.801387	27
3	1.445443	12
4	0.857069	24
5	0.719914	66

6. Spectral Analysis to Determine Clusters

Two examples were chosen to test the process of determining clusters by eigenvalue-eigenvector analysis. The first is based on a

[†] The references are to Harmon.[7]

classical example in the literature of factor analysis, involving an analysis of results of psychological testing. This is the 13×13 matrix used in the first two tests described in the previous part. A number of authors, studying this matrix and the underlying data by a variety of methods, have determined, on the basis of certain patterns of strong correlations, that the 13 tests may be divided into three groups, with the members of each group strongly interrelated. It was expected that an examination of the eigenvectors belonging to the dominant eigenvalues would reveal this known "clustering" directly.

The second test, not contrived, but actual, involved a matrix of word-word correlations generated by the document analysis program under development by the Information Retrieval Group at the Computation Laboratory of Harvard University. Certain tendencies – fairly weak, to be sure – toward clustering of the words had been identified using the simple methods of the type described by Lesk in a previous report from this Laboratory.[9] It was hoped that the present techniques would be capable of detecting clusters, if any, in this case.

A. The 13-variable Case (Psychological Tests)

Two runs were made, one using unities, the other using known estimates of the communalities, on the main diagonal of the matrix. As expected from theory, the eigenvectors in the two cases were not significantly different. The values quoted here are for unities on the main diagonal, since this is felt to be a more realistic test.

First eigenvalue:		5.067
Corresponding eigenvector:	1	0.271
	2	0.172
	3	0.201
	4	0.232
	5	0.344
	6	0.337
	7	0.340
	8	0.333
	9	0.331
	10	0.205
	11	0.248
	12	0.209
	13	0.295

Examination reveals five elements of approximately the same magnitude (0.33 or 0.34) which are larger than any other elements. The logical vector $\underline{u}^{(1)} = (\underline{x}^{(1)} > 0.33)$ extracts this cluster:

$$\underline{u}^{(1)}/(1,2,3,\ldots,13) = (5,6,7,8,9).$$

We now extract the second root.

Second eigenvalue:		1.801
Corresponding eigenvector:	1	0.075
	2	-0.014
	3	-0.070
	4	-0.083
	5	-0.176
	6	-0.271
	7	-0.254

8	-0.110
9	-0.300
10	0.449
11	0.363
12	0.517
13	0.334

It will be noted, first of all, that all the elements of $\underline{x}^{(2)}$ corresponding to the cluster (5,6,7,8,9) are negative and relatively large in magnitude; this fact further sequesters these five elements and confirms their grouping as a cluster. There are four relatively large and positive components of $\underline{x}^{(2)}$. The logical vector ($\underline{x}^{(2)} > 0.33$) selects this second cluster:

(10,11,12,13).

A third root is now extracted:

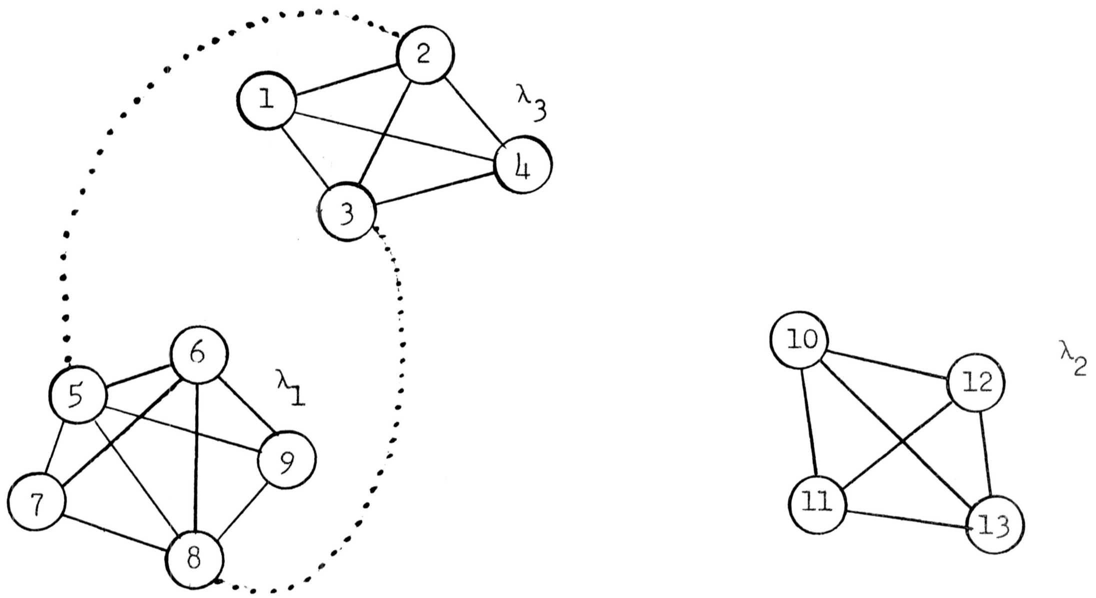
Third eigenvalue:		1.445
Corresponding eigenvector:	1	0.412
	2	0.387
	3	0.491
	4	0.357
	5	-0.214
	6	-0.164
	7	-0.214
	8	-0.052
	9	-0.213
	10	-0.311
	11	-0.169
	12	-0.025
	13	0.136

The criterion $\underline{u}^{(3)} = (\underline{x}^{(3)} > 0.33)$ selects the third cluster:

$$\underline{u}^{(3)}/(1,2,3,\ldots,13) = (1,2,3,4)$$

The fourth eigenvalue ($\lambda_4 = 0.857$) is somewhat smaller in magnitude; the corresponding eigenvector has only one relatively large component, the second. This tends to set off element "2" in a class by itself, agreeing with the observation that "2" is more weakly bound in cluster (1,2,3,4) than the other elements. The further eigenvalues of the matrix, and their corresponding eigenvectors, provide no useful indications.

The clustering as summarized in Fig. 1 agrees exactly with that determined previously and reported in the literature.



Links Among the 13 Elements, Showing Clustering

Figure 1

B. The 25-variable Case (Word Correlations)

An original unedited text, a section of a journal article on the subject "extensions of ALGOL," consisted of 37 sentences, the total length being somewhat under 1000 words. Processing of the text identified occurrences of 51 distinct terms, each potentially "content-rich," which are found in the automatic thesaurus of computer-science technical terms built into the processing system. A word-sentence incidence matrix:

$$\begin{array}{c|ccccc}
\text{Word No.} \backslash \text{Sentence No.} & 1 & 2 & 3 & \cdots & 37 \\ \hline
1 & & & & \vdots & \\
2 & & & & \vdots & \\
3 & & & & \vdots & \\
\vdots & \cdots & \cdots & \cdots\cdots\cdots & \underline{R}^i_j & \\
51 & & & & &
\end{array}$$

where $\underline{R}^i_j$ is the number of occurrences of word i in sentence j, describes the distribution of these words among the sentences in the text. From this list of 51 terms all were deleted which occurred only once in the text; after this, any surviving term was deleted if it did not have at least two co-occurrences with other undeleted terms. At the conclusion of this selection process, 25 terms remained. These are listed according to their thesaurus numbers in Table 1, but the actual terms are not listed to avoid prejudicing the evaluation of the results.

Index Number	Thesaurus Number	Index Number	Thesaurus Number
1	16	14	103
2	26	15	117
3	27	16	121
4	29	17	134
5	35	18	137
6	49	19	143
7	53	20	147
8	57	21	156
9	60	22	178
10	68	23	181
11	77	24	191
12	80	25	208
13	102		

Terms Included in Clustering Study

TABLE 1

The next step is the preparation of a (25 × 25) term-term association matrix by row-wise correlation, using the cosine measure described by Salton.[7] Thus the matrix element

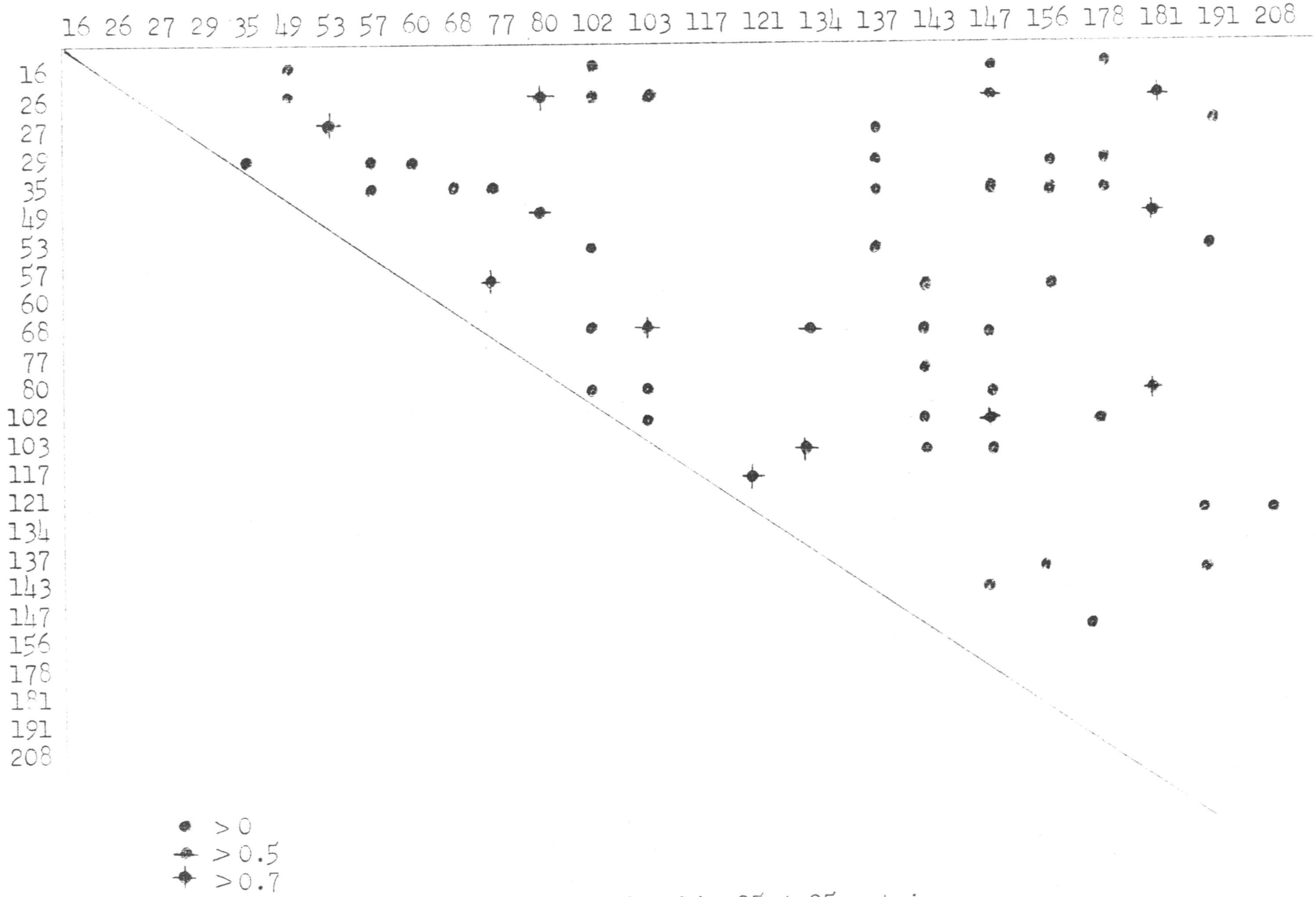
$$\underline{S}^i_j = \cos(\underline{R}^i,\underline{R}^j)$$

measures the putative similarity between term i and term j, based on their tendency to co-occur within sentences of the text. There were only 116 nonzero terms in this symmetric (25 × 25) matrix as shown in Fig. 2.

Index Numbers		Similarity Coefficient
1	6	.5000
1	13	.2500
1	20	.2673
1	22	.5000
2	6	.5000
2	12	.8944
2	13	.5000
2	14	.2887
2	20	.5345
2	23	.7071
3	7	.7071
3	18	.5000
3	24	.5000
4	5	.2860
4	8	.1741
4	9	.2132
4	18	.2132
4	21	.4264
4	22	.2132
5	8	.1826
5	10	.1826
5	11	.2236
5	18	.4472
5	20	.1195
5	21	.4472
5	22	.2236
6	12	.6708
6	23	.7071
7	13	.1768
7	18	.3536
7	24	.3536
8	11	.8165
8	19	.4082
8	21	.4082
10	13	.2041
10	14	.7071
10	17	.5774
10	19	.4082
10	20	.2182
11	19	.5000
12	13	.2236
12	14	.1291
12	20	.2390
12	23	.9487
13	14	.4330
13	19	.2500
13	20	.8018
13	22	.2500
14	17	.8165
14	19	.2887
14	20	.4629
15	16	.7559
16	24	.2673
16	25	.2673
18	21	.5000
18	24	.5000
19	20	.2673
20	22	.2673

Nonzero Elements of the (25 × 25) Similarity Matrix S

TABLE 2



2 × 58 nonzero off-diagonal **entries** in this 25 × 25 matrix

Location of Nonzero Similarities in the (25 × 25) Term-term Matrix

Figure 2

The nonzero similarities range in magnitude from 0.1195 to 0.9487, with ten entries equal to or greater than $\frac{\sqrt{2}}{2}$ (that is, $\measuredangle(\underline{R}^i,\underline{R}^j) \leq 45^\circ$).

The matrix $\underline{S}$ is now used as input to the eigenvalue-eigenvector analysis program. The dominant eigenvalue is $\lambda_1 = 3.659$ and the corresponding eigenvector:

(1)	0.139451	(9)	0.001784	(17)	0.123847
(2)	0.452920	(10)	0.176368	(18)	0.024163
(3)	0.014859	(11)	0.042998	(19)	0.134669
(4)	0.022021	(12)	0.427966	(20)	0.327092
(5)	0.051981	(13)	0.315435	(21)	0.023593
(6)	0.313962	(14)	0.278046	(22)	0.094985
(7)	0.029859	(15)	0.000390	(23)	0.356421
(8)	0.042727	(16)	0.001324	(24)	0.011611
				(25)	0.000138

Although it would be difficult to justify a clear-cut distinction between "large terms" and "small terms," it was felt that a reasonable threshold would be

$$t_1 = 0.278$$

since there is, after this point, a relatively large gap in the magnitudes of the elements. Then the selection vector

$$\underline{u}^{(1)} = (\underline{x}^{(1)} > t_1\underline{\epsilon})$$

selects the cluster

$$C1 = (2,6,12,13,14,20,23).$$

The second eigenvalue is $\lambda_2 = 2.843$, with corresponding vector:

(1)	0.004183	(9)	0.017721	(17)	0.228352
(2)	-0.169649	(10)	0.308153	(18)	0.254108
(3)	0.172282	(11)	0.208934	(19)	0.271674
(4)	0.152761	(12)	-0.264186	(20)	0.169747
(5)	0.240429	(13)	0.165931	(21)	0.216144
(6)	-0.254443	(14)	0.298267	(22)	0.095270
(7)	0.160453	(15)	0.011246	(23)	-0.297750
(8)	0.239106	(16)	0.027254	(24)	0.151109
				(25)	0.003977

It will be noted first that elements 2,6,12 and 23 are negative, confirming the adherence of these four items to cluster C1 and expressing their remoteness or disjointness from cluster C2. The setting of a threshold for membership in C2 is, once again, somewhat arbitrary. If we take

$$t_2 = 0.2$$

then

$$\underline{u}^{(2)} = (\underline{x}^{(2)} > t_2\underline{\epsilon})$$

selects the cluster

$$C2 = (5,8,10,11,14,17,18,19,21).$$

It is of interest that clusters C1 and C2 are not disjoint; element 14 is a member of both subsets.

Had we wished to impose stricter criteria, in order to obtain smaller and more tightly bound clusters, we would have chosen

$$t_1 = 0.325 \text{ and } t_2 = 0.250$$

to obtain, respectively,

$$\widetilde{C1} = (2,12,20,23) \text{ and } \widetilde{C2} = (10,14,18,19)$$

which are now disjoint.

The analysis was halted at this point; a third root was not extracted.

These results have been compared with those obtained by the clustering routines currently incorporated in the document analysis system. The first three clusters of interest (these are actually No. 1, No. 3, and No. 5 of the computer output) include the following items (in addition to a few others, not listed, which had not been retained in our set of 25).

$$K1 = (8,10,11,14,17,19)$$
$$K2 = (1,2,6,12,13,20,22,23)$$
$$K3 = (3,7,18,24)$$

There can be no doubt of the identification of C1 with K2; in fact, the similarity between the selection vector $\underline{u}^{(1)}$ for cluster C1 and the corresponding vector $\underline{v}^{(2)}$ associated with K2, is

$$s = \cos(\underline{u}^{(1)},\underline{v}^{(2)}) = 0.802$$

using the cosine measure of similarity. To answer the question of whether this agreement is to be regarded as significant, we compute (from the truncated hypergeometric distribution) the probability of a chance agreement this high or higher:

$$p = 0.001$$

Now C2 may be identified with K1. In this case the similarity measure is

$$s = 0.8165$$

with an even higher significance level ($p = 0.0008$).

The set K3 has only a single item (number 18) in common with the union of C1 and C2; this points to a third cluster lying in the complement of C1 $\cup$ C2.

One final test was undertaken to examine the validity of the clusters selected by the eigenvector analysis. This was a hand simulation of a simple clustering technique based on methods discussed by Needham[10] and by Parker-Rhodes and Needham.[11] The first step is sorting the nonzero elements of the matrix $\underline{S}$, as shown in Table 3.

Only the final results will be quoted here. Incorporating links of strength 0.5 and greater, we build the following clusters:

(2,6,12,23)
(13,20)
(10,14,17)

The correspondence with clusters C1 and C2 is seen immediately.

It may now be concluded that the clusters identified by eigenvector analysis correspond closely to those detected by the other approaches. The attempt to seek a final confirmation by determining whether the actual terms associated with the thesaurus entry numbers are conceptually related

Index Numbers		Similarity Coefficient
5	20	.1195
12	14	.1291
4	8	.1741
7	13	.1768
5	8	.1826
5	10	.1826
10	13	.2041
4	9	.2132
4	18	.2132
4	22	.2132
10	20	.2182
5	11	.2236
5	22	.2236
12	13	.2236
12	20	.2390
1	13	.2500
13	19	.2500
13	22	.2500
1	20	.2673
16	24	.2673
16	25	.2673
19	20	.2673
20	22	.2673
4	5	.2860
2	14	.2887
14	19	.2887
7	18	.3536
7	24	.3536
8	19	.4082
8	21	.4082
10	19	.4082
4	21	.4264
13	14	.4330
5	18	.4472
5	21	.4472
14	20	.4629
1	6	.5000
1	22	.5000
2	6	.5000
2	13	.5000
3	18	.5000
3	24	.5000
11	19	.5000
18	21	.5000
18	24	.5000
2	20	.5345
10	17	.5774
6	12	.6708
2	23	.7071
3	7	.7071
6	23	.7071
10	14	.7071
15	16	.7559
13	20	.8018
8	11	.8165
14	17	.8165
2	12	.8944
12	23	.9487

Nonzero Elements of $\underline{S}$ in Ascending Order

TABLE 3

reveals no particular relation among the items in any of the clusters. This is, however, no criticism of the methods used to infer the presence of clusters from the correlation matrix, but merely an indication that the raw correlation data themselves (the matrix $\underline{R}$ from which $\underline{S}$ was derived) probably express little more than chance co-occurrence of words in sentences.

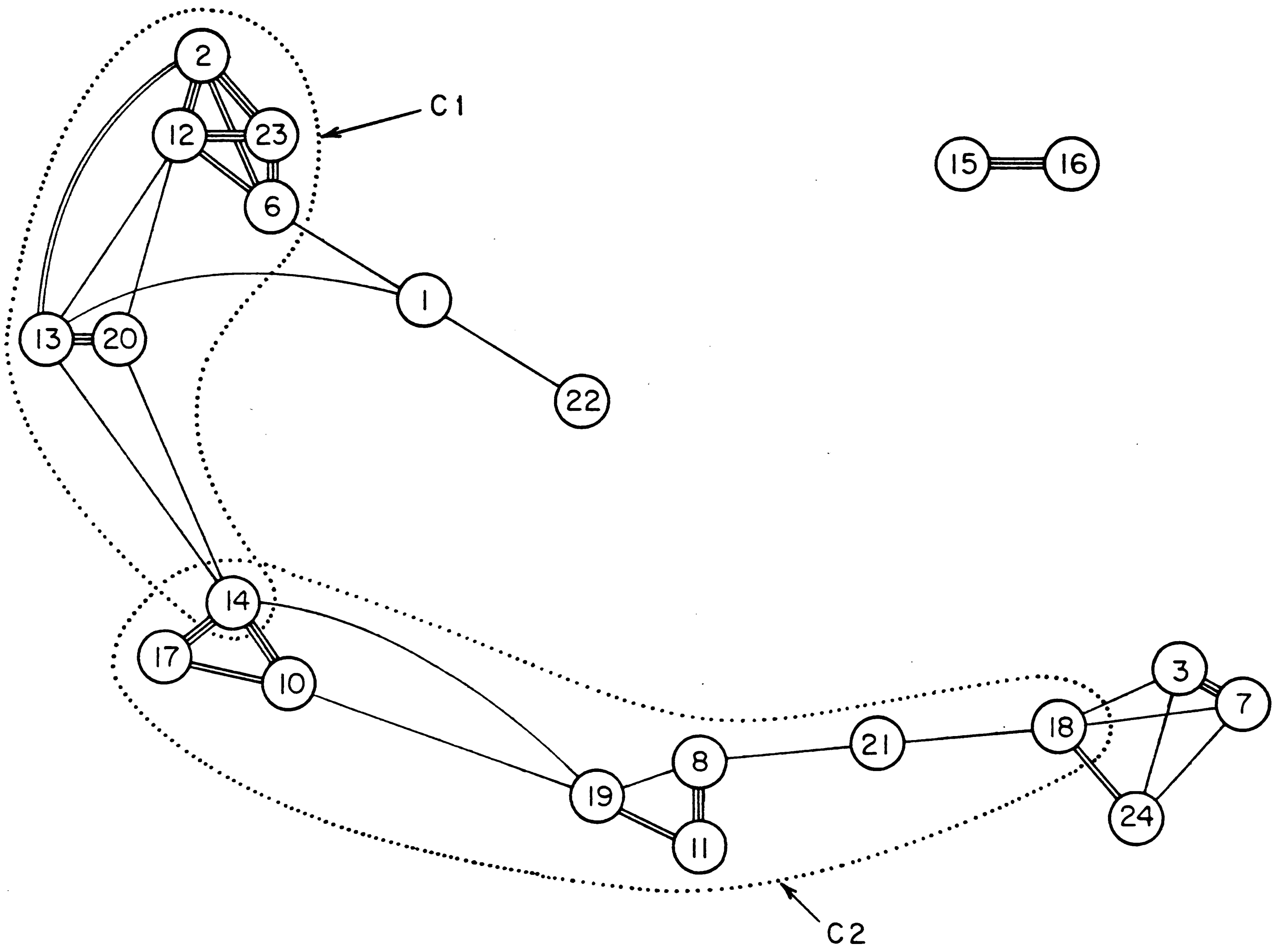
The results of the clustering attempts are displayed in Figs. 2 and 3, each of which may be regarded as a portion of a labeled symmetric graph showing the pairwise correlations among the 25 items. The graph is drawn to represent the strength of a link by the relative nearness of the vertices and/or the thickness of the line joining them, according to these conventions:

$s > 0.7071$	≡
$0.5 < s \leq 0.7071$	=
$s \leq 0.5$	—

An examination of these graphs suggests a few more remarks. Cluster C1 may seem somewhat unsatisfactory because it includes item 14, but breaks the strong links which join "14" with "10" and "17". Had the threshold t_1 been lowered only slightly (to $t_1 = 0.176$), then item 10 would have been included along with "14". On the other hand, had t_1 been raised slightly to 0.3, then item 14 would have dropped out, leaving a more compact cluster

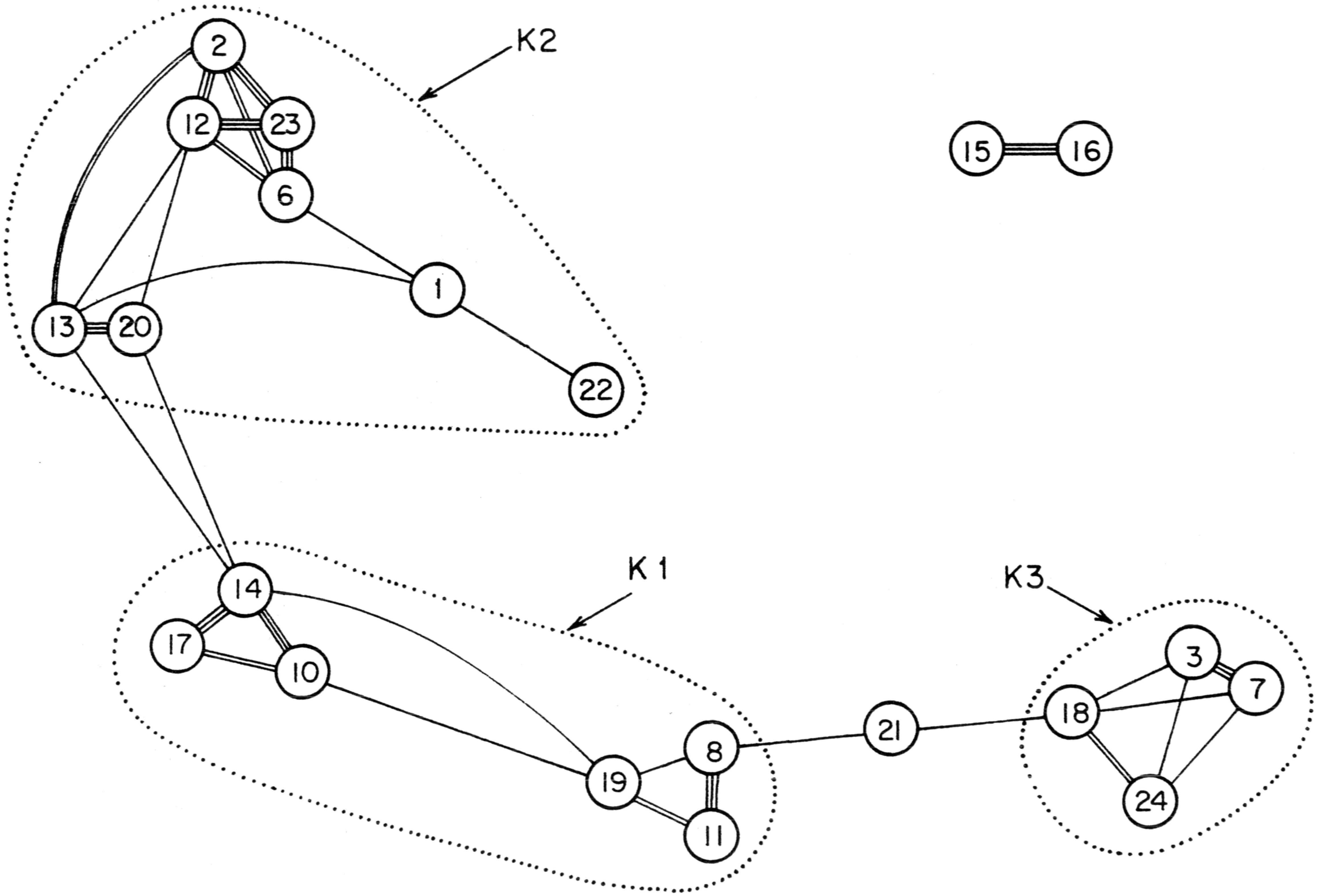
$$(2,6,12,13,20,23)$$

which would have been even better correlated with cluster K2.



Correlations Among the 25 Items, Showing Clusters Found By Eigenvector Analysis

Figure 2



Correlations Among the 25 Items, Showing Clusters Found By
Methods Described in Text

Figure 3

Similar considerations apply to cluster C2. A slight lowering of threshold t_2 would have picked up items 3, 7, and 24 (which are strongly linked with "18").

7. Evaluation and Conclusions

An evaluation of the clustering technique presented here must depend on these three questions:

(1) Are clusters found which are intuitively reasonable, and which match clusters found by other methods?

(2) Is the algorithm rapid enough to be of any practical significance?

(3) Can the decision procedure implicit in the selection of the "large elements" of the normalized eigenvectors be mechanized fully?

The first question is answered affirmatively, it is felt, by the theoretical analysis of Part 2 and the results presented in Parts 5 and 6.

The tests have not been extensive enough to permit a direct evaluation of the running time of the clustering programs. The actual running times were so short, in fact, that timing was not feasible. A theoretical estimate of the efficiency of the algorithm is presented here.

It is reasonable to assume that the bulk of the time will be expended in the actual determination of eigenvalues and eigenvectors. The computing time for this process depends critically on the rate of convergence of the basic iterative method. This, in turn, is a function of the distribution of the eigenvalues – in other words, of the structure of the data matrix under analysis. Assume that the average number of iterations is M. In the course of each iteration, $(n+2)$ dot products are computed, each of which involves n multiplications and additions. For the first eigenvalue we expect, then, an operation count proportional to

$$Mn(n+2);$$

for succeeding eigenvalues, a somewhat greater count (because of orthogonalization). Assuming this to be absorbed in an adjusted factor M', we require an operation count proportional to

$$kM'n(n+2)$$

to extract k eigenvalues and eigenvectors. Both k and M' depend strongly on the structure of the matrix: k, the number of eigenvectors required to approximate the structure adequately, being equal to the approximate rank of the matrix; M', on the other hand, being related to the separation between eigenvalues, but not necessarily to its size. Therefore, all things being equal, the computing time should increase as

$$n(n+2)$$

or approximately as n^2, with increase in n.

This estimate compares favorably with running times for procedures involving complete factor analysis[4] or latent class analysis,[1] which are expected to grow at least as fast as $n.^3$ For the processing of large matrices, however, it appears that if truly practical methods are ever found, their computing time should increase no faster than n.

Using the eigenvectors of the similarity matrix to determine membership in clusters reduces what is basically a two-dimensional problem to a one-dimensional one. Each of the components of a normalized eigenvector may be regarded as a measure of the adherence of the corresponding element to the cluster associated with that vector. Thus a scan over the two dimensions of the original matrix (to find the large elements which indicate that certain members of the index set belong together) is replaced by a one-dimensional scan to determine cluster membership. This does not, however, eliminate the element of human intervention or personal choice in the process of deciding which elements are large enough to warrant inclusion in the cluster; in other words, in deciding what the cut-off point should be between the "large" elements and the "small" elements of the eigenvector.

An automatic decision procedure which has been proposed is described in the following paragraphs, and the results of a few tests, by no means extensive or definitive, are now presented.

The two steps in the procedure are the following:

(1) permute the normalized eigenvector $\underline{x}$ into a vector $\underline{y}$, where

$$\underline{y}_1 \geq \underline{y}_2 \geq \underline{y}_3 \geq \cdots \geq \underline{y}_n$$

by a sorting process, and

(2) find that value of k which maximizes

$$F(k) = \frac{\sum_{i=1}^{k} \underline{y}_i^2}{k^p} \qquad 0 < p < 1$$

where only nonnegative $\underline{y}_i$ are eligible for inclusion.

Since F(k) first increases, then decreases, with increasing k, the maximum is determined sequentially by evaluating F(1), F(2), etc. until F no longer increases. The parameter p, which may be considered a shaping factor, influences the cut-off between "large" and "small" elements. At the extremes, $p = 0$ will cause inclusion of all elements; $p = 1$ will include only the first (or the first q, if $\underline{y}_1 = \underline{y}_2 = \ldots = \underline{y}_q$).

The method works well, as expected, for contrived cases in which the distinction between large and small elements is clear-cut and unambiguous. Here, however, much simpler procedures could be programmed. A more critical test has been based on attempts to reproduce the results of the 13 × 13 case described in Part 6.

The maximum attained by F(k), for each of the five values of p used in the test, is underlined in Table 4. The corresponding cluster would contain all elements above the horizontal line. Thus, for

$$p = 0.8$$

the same cluster is selected as before:

$$(5,6,7,8,9).$$

When the procedure was applied to the second normalized eigenvector of the 13 X 13 test matrix, it was found that the value $p = 0.8$ did not select precisely the same cluster as our previous methods, although a slightly larger value of p did. It is not known whether one value of the parameter p can be found which will make the correct selection for all three eigenvectors of this matrix. Further tests will be necessary to settle

1	2	3	4	5	6	7	8
1	5	0.344	0.1183	0.1183	0.1183	0.1183	0.1183
2	7	0.340	0.2339	0.1654	0.1473	0.1343	0.1170
3	6	0.337	0.3475	0.2006	0.1671	0.1443	0.1158
4	8	0.333	0.4584	0.2292	0.1819	0.1512	0.1146
5	9	0.331	0.5680	0.2540	0.1943	0.1567	0.1136
6	13	0.295	0.6550	0.2674	0.1984	0.1562	0.1092
7	1	0.271	0.7284	0.2753	0.1991	0.1536	0.1041
8	11	0.248	0.7900	0.2793	0.1975	0.1494	0.0988
9	4	0.232	0.8437	0.2812	0.1946	0.1455	0.0937
10	12	0.209	0.8874	0.2806	0.1912	0.1406	0.0887
11	10	0.205	0.9295	0.2802	0.1875	0.1365	0.0845
12	3	0.201	0.9699	0.2800	0.1850	0.1328	0.0808
13	2	0.172	0.9999	0.2772	0.1808	0.1284	0.0769

Legend

Column 1: current index i
2: original index
3: original component of normalized eigenvector
4: F(k), with p = 0
5: p = 0.5
6: p = 0.67
7: p = 0.8
8: p = 1.0

Normalized and Sorted Vectors

TABLE 4

this question, and to explore the more general problems of the proper choice of p and the efficacy of the algorithm in general.

The difficulty here in identifying the "large" components of a vector is felt to be, in large part, a reflection of the more basic difficulty in deciding cluster membership in those cases where the clustering cannot be regarded as strong. This is a problem inherent in the data and in the criteria adopted for cluster selection.

REFERENCES

1. Baker, F. B., "Information Retrieval Based on Latent Class Analysis," JACM, Vol. 9, No. 4 (October 1962).
2. Bellman, R., Introduction to Matrix Analysis, McGraw-Hill, New York (1960).
3. Bonner, R. E., "On Some Clustering Techniques," IBM Journal of Research and Development, Vol. 8, No. 1 (January 1964).
4. Borko, H., and Bernick, M. D., "Automatic Document Classification," JACM, Vol. 10, No. 2 (April 1963).
5. Brooks, F. P., Jr., and Iverson, K. E., Automatic Data Processing, Wiley, New York (1963).
6. Faddeeva, V. N., Computational Methods of Linear Algebra (Translated by Benster, C. D.), Dover Publications, New York (1959).
7. Harmon, H. H., Modern Factor Analysis, University of Chicago Press (1960).
8. Iverson, K. E., A Programming Language, Wiley, New York (1962).
9. Lesk, M., "Attempts to Cluster Documents with Citation Data," Information Storage and Retrieval, Report ISR-3, Sec. VI, The Computation Laboratory of Harvard University (1963).

10. Needham, R. M., "A Method for Using Computers in Information Classification," Proc. IFIP Congress 1962, Vol. VIII, No. 3 (1962) pp. 121-123.

11. Parker-Rhodes, A. F., and Needham, R. M., "The Theory of Clumps: A New Concept of Classification and Selection," Report ML 126, Cambridge Language Research Unit, Cambridge, England (1960).

12. Salton, G. A., "Associative Document Retrieval Techniques Using Bibliographic Information," JACM, Vol. 10, No. 4 (1963), pp. 440-457.