R. E. Bonner

On Some Clustering Techniques

Abstract: The problem of organizing a large mass of data occurs frequently in research. Normally, some process of generalization is used to compress the data so that it can be analyzed more easily. A primitive step in this process is the "clustering" technique, which involves gathering together similar data into a cluster to permit a significant generalization.

This paper describes a number of methods which make use of IBM 7090 computer programs to do clustering. A medical research problem is used to illustrate and compare these methods.

Introduction

The clustering problem as considered in this paper may be stated as follows. *Given*: A set of objects, each of which is defined by the values of a set of attributes associated with it. This attribute set is the same for each object. *Find*: "Clusters" of objects (subsets of the original object set) such that members of a cluster "look like" each other but do not look much like objects outside the cluster.

The definitions of the terms *cluster* and *look like* are deliberately left unspecified since none of the many specific definitions that might be given seems "best" in any general sense. The value judgment of the user is the ultimate criterion for evaluating the meaning of these terms. If using them produces an answer of value, no more need be asked of them.

Some characteristics of clustering problems can be cited as follows: First, the given set of objects can be a sample taken from an even larger set or else the set can be complete in itself. Second, the experimenter might be interested in knowing the individual members of a cluster, the members in an over-all description of the cluster, or in both. Third, an object may be allowed in only one cluster or in more than one, or it may be required to be a member of at least one cluster or not allowed to be in any cluster at all. To exemplify these characteristics we can consider a taxonomy problem where the object set is complete, there is interest in knowing the individual members of a cluster, and it is required that an object be in at least one cluster. Here the emphasis is on "structuring" the objects. On the other hand, in another important class of problems the objective is to identify possible "causality" underlying the data as suggested by multivariate dependence. Usually, the given set of objects represents a sample from a larger population. The goal is to obtain an over-all description of the cluster which hopefully would be reproduced if another sample from the same population were chosen. Objects can be in more than one cluster. As the function of these objects is to add to the cluster descriptions, they are devoid of any individual importance.

In this paper, a description is given of two IBM 7090 computer programs—Clustering Programs I and II—which are intended mainly for "structuring" problems, and one other, Clustering Program III, which is intended for "causality" problems. The technique employed in the latter is compared with factor analysis, a method whose goal is to determine "factors" which account for the correlations existing between all pairs of attributes. All three programs deal only with binary attributes, although much of their conceptual foundation is applicable to multistate attributes and work is now in progress to exploit this fact.

The programs were tested on a problem in nosology, the process of classifying diseases, which was supplied by Dr. Hans Zinsser of the Columbia University College of Physicians and Surgeons. From a medical point of view, the techniques yielded a set of hypotheses which can now be clinically evaluated, for example, by seeing whether or not a particular therapy has different effects on people in different clusters.

The similarity measure

Since the input to Programs I and II is a similarity matrix, a definition of the term and a brief discussion of how a

Table 1 **Example of the formation of a similarity matrix. T = similarity threshold.**

Binary description of sample object set							*$S_{\alpha\beta}$ matrix:*								*Similarity matrix for T = 0.45*							
Object No.	*Attribute No.* 1	2	3	4	5	6	*Object No.*	1	2	3	4	5	6	7	8	*Object No.*	1	2	3	4	5	6	7	8
1	1	0	0	1	0	0	1	x	2/3	1/5	0	2/3	0	1/4	1/4	1	1	1	0	0	1	0	0	0
2	1	1	0	1	0	0	2		x	1/6	1/5	2/4	0	2/4	2/4	2	1	1	0	0	1	0	1	1
3	0	0	1	1	1	1	3			x	2/5	2/5	2/4	2/5	1/6	3	0	0	1	0	0	1	0	0
4	0	1	1	0	0	1	4				x	0	1/4	2/4	2/4	4	0	0	0	1	0	0	1	1
5	1	0	0	1	1	0	5					x	1/4	1/6	1/5	5	1	1	0	0	1	0	0	0
6	0	0	1	0	1	0	6						x	0	1/4	6	0	0	1	0	0	1	0	0
7	0	1	0	1	0	1	7							x	1/5	7	0	1	0	1	0	0	1	0
8	1	1	1	0	0	0	8								x	8	0	1	0	1	0	0	0	1

similarity matrix might be found will be useful. In its specific form, the clustering problem requires a particular definition of similarity for it to be possible to determine whether two objects look alike. An intelligent choice of this definition is quite dependent on the specific problem. A few sample definitions follow to acquaint the reader with some possibilities. For example, each attribute can be considered a dimension in N-dimensional space and a distance measure can be used as a measure of similarity D between objects α and β (the smaller the distance, the greater the similarity). To illustrate such a measure where the k_i might be normalizing or value-judgment coefficients, we have

$$D_{\alpha\beta} = \sum_{i=1}^{i=N} k_i(x_{i\alpha} - x_{i\beta})^2. \quad (1)$$

In the case where all the attributes are binary variables, a number of similarity measures have been proposed by various authors[1-5] and are discussed more fully in Ref. 5. All of these measures involve $C_{\alpha\beta}$, defined as the number of attributes which are "one" for both object α and object β. For example, Tanimoto[3] has defined similarity as

$$s_{\alpha\beta} = (C_{\alpha\beta})/(C_{\alpha\alpha} + C_{\beta\beta} - C_{\alpha\beta}). \quad (2)$$

If a judgment is made that valid clusters arise because of dependence between attributes in the original object set, then the measures mentioned previously are not very satisfactory because they put equal weight on matches between correlated or uncorrelated attributes.

An example of a similarity measure for binary variables which takes pairwise correlation into account is given by

$$S_{\alpha\beta} = \sum_{i=1}^{i=N}\sum_{j=1}^{j=N} r_{ij}[1 - |x_{\alpha i} - x_{\beta i}|] \\ [1 - |x_{\alpha j} - x_{\beta j}|][1 - 2|x_{\alpha i} - x_{\alpha j}|], \quad (3)$$

where $S_{\alpha\beta}$ is the similarity between objects α and β and r_{ij} is the correlation coefficient for attributes i and j.

A term contributes to the sum if (a) attribute i is the same for both samples α and β; and (b) attribute j is the same for both samples α and β. If attribute i is the same as attribute j, r_{ij} is added to the sum; if they are different, r_{ij} is subtracted from the sum. This procedure effectively adds a positive increment if the correlation present agrees in sign with r_{ij} and a negative one if not.

All of the measures under discussion here are normally subjected to a "threshold" value, on one side of which objects are judged "similar" and on the other side, "dissimilar." This results in a "similarity matrix" of ZEROS and ONES where the dimensions are objects versus objects and ONE means two objects are similar.

The matrix is then used as the starting point for various clustering techniques. Table 1 shows an example of the formation of a similarity matrix using the similarity measure of Eq. (2). (In this Table S_{12} is calculated as follows: $C_{12} = 2$; $C_{11} = 2$; $C_{22} = 3$; therefore, $S_{12} = 2/3$.)

- *Clustering Program I*

This program takes a similarity matrix (whose dimensions are N_T objects by N_T objects), considers it as a set of N_T objects each having N_T binary attributes, and forms its similarity matrix using the measure of Eq. (2). Essentially this is taking the similarity matrix of a similarity matrix; since the result is another similarity matrix, the procedure can be iterated as many times as desired. The reason for doing this is to give better definition to clusters which are loosely connected internally and to better separate those which overlap. For the maximum allowable sample size of 350 objects, it takes about one minute of computer time per iteration. The result of taking the similarity matrix of the similarity matrix of Table 1 is given in Table 2.

- *Clustering Program II*

The input to this program is either the original similarity matrix or the matrix that is the output from Program I. The purpose of the program is, first, to find all clusters where all members of the cluster are similar to each other and no nonmember is similar to all members. (In graph

Table 2 **Similarity matrix of similarity matrix of Table 1 for similarity threshold T = 0.45.**

$S_{\alpha\beta}$ *matrix*:

	Attribute No.							
	1	2	3	4	5	6	7	8
1	1	3/5	0	0	3/3	0	1/5	1/5
2		1	0	2/6	3/4	0	2/6	2/6
3			1	0	0	2/2	0	0
4				1	0	0	2/4	2/4
5					1	0	1/5	1/5
6						1	0	0
7							1	2/4
8								1

Similarity matrix for $T = 0.45$

	Attribute No.							
	1	2	3	4	5	6	7	8
1	1	1	0	0	1	0	0	0
2		1	0	0	1	0	0	0
3			1	0	0	1	0	0
4				1	0	0	1	1
5					1	0	0	0
6						1	0	0
7							1	1
8								1

theory, these are called *maximal complete subgraphs* of the similarity matrix graph; for simplicity they will be referred to as "tight" clusters.) The second purpose is to find, using the clusters so identified, a set of clusters where no object is in more than one cluster and all objects in a cluster are similar to each other. The entire procedure can be viewed as finding a set of "core" clusters to use as input to a later "cluster adjustment" program which attempts to build around these "cores."

• *Algorithms for finding tight clusters*

There are at least two other algorithms[7,8] for finding tight clusters. An advantage of the present method over that described in Ref. 7 is that it does not print out subsets of clusters or the same cluster more than once. Although it is difficult to compare utility more generally among the three because of the differing requirements in output and memory and because of the dependence on the input data, the method discussed here is sufficient to handle a useful set of inputs in a reasonable time. It should be observed that all methods have difficulty as the number of clusters becomes very large. Another factor which time-limits their use is that it is difficult to avoid finding the same cluster or its subsets over and over again. The ability of Program I to keep these difficulties within bounds is important because it permits solution of problems that had been "impracticable" and provides a much broader base of the concept of a tight cluster. Figure 1 shows a diagrammatical representation of the similarity matrix of Table 1 and a list of the tight clusters which are present. A detailed description of the steps in the procedure used is given in Appendix I.

• *Procedure for picking disjoint sets using tight clusters*

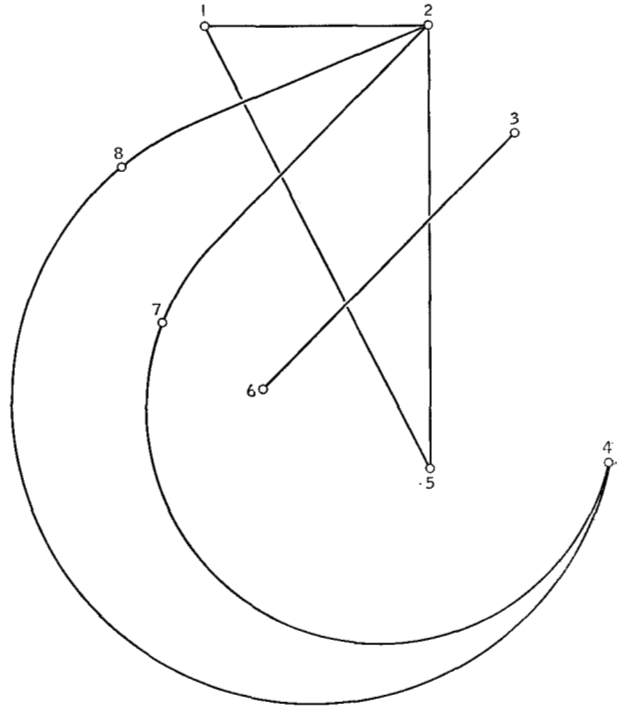
In a typical case, many of the tight clusters will contain almost the same set of members. It is desirable that only one cluster represent the core of this cluster set. It is also desirable to have a core for each of the other cluster sets that are reasonably different from one another. A question then arises as to whether an object should be allowed in only one core or in more than one core. Both situations are certainly admissible depending on the requirements of the specific problem under consideration. The procedure to be described forms disjoint core clusters. The program builds up a set of cores one at a time. At a given level of buildup (certain number of cores already chosen), it can find a number of ways of choosing the next core which are all equally satisfactory. These choices are called the alternative set for that level of buildup. Table 3 presents a detailed description of the buildup procedure.

Difficulty is encountered when the number of alternatives becomes too large. Occasionally, this event occurs in the trivial situation caused by attempting to choose between many small clusters at the end of the core-forming process. This problem can be greatly alleviated by picking an arbitrary alternative when the difference set reduces to a certain specified size.

Step 5 is interesting in that it illustrates an unsuspected situation which arose within the program, in which it was

Figure 1 **"Tight" clusters for similarity results of Table 1.**

"Tight" Clusters	*"Core" Clusters*
1, 2, 5	1, 2, 5
3, 6	3, 6
2, 8	4, 7
2, 7	8
8, 4	
7, 4	



necessary to solve a clustering problem in order to break a tie. Such a case arises in the application of the algorithm to the tight clusters of Fig. 1; here a tie between (3, 6), (8, 4) and (7, 4) was resolved by picking (3, 6) and (7, 4). The complete result is given in Fig. 1. Figure 2 shows the result of applying the algorithm to the similarity matrix of Table 2. Note that because of the use of Program I, the clusters have become disjoint.

Cluster adjustment program

The set of output core clusters from the program just described can now be used as input to a cluster adjustment program or interpreted in their own right. If Cluster Program I was used before Program II, it is desirable to use the cluster adjustment program.

This program attempts to integrate the members of the smaller core clusters into the larger and to relocate misplaced members of the larger. It does this on the basis of information from the original similarity matrix (before the manipulation in Program I). The user specifies the size of the smallest cluster he wishes to consider as "large." The program then proceeds as outlined in Table 4.

A measure of "value" for the cluster can be constructed by subtracting from I_{xx}, which is a measure of internal clustering strength, the average external interactions

$$V_x = I_{xx} - \frac{1}{N_R} \sum_{y=1}^{y=N_R} I_{xy},$$

where N_R is the number of clusters other than cluster x. A measure of value for the whole set of clusters is the average "value"

$$V_{av} = \frac{1}{N_R + 1} \sum_{x=1}^{x=N_R+1} V_x.$$

Figure 2 **"Tight" clusters for similarity matrix of Table 2.**

"Tight" Clusters
1, 2, 5
4, 7, 8
3, 6

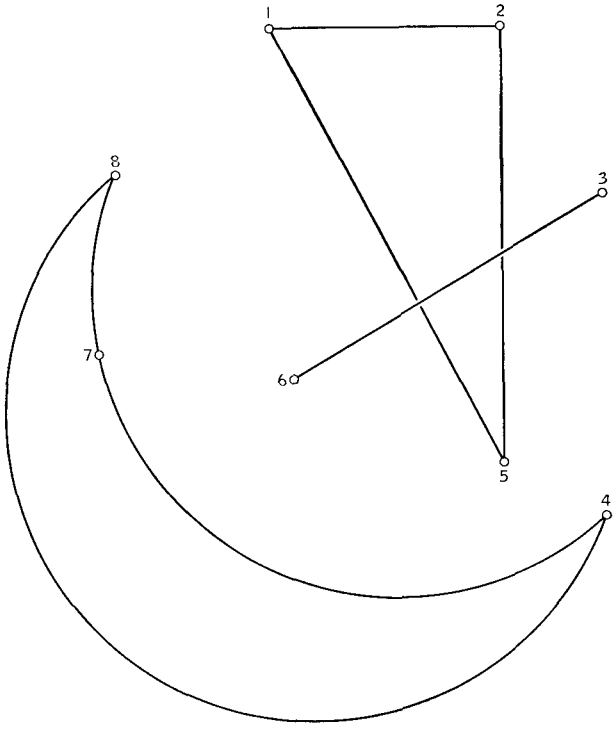


Table 3 **Steps in forming disjoint core clusters.**

i = Alternative index
j = Build up level index

Step No.	*Step Description*
1.	Find the "tight" cluster having the largest number of members and store it as the first "core" cluster. Set $j = 1$. If there is a tie for the largest cluster, go to Step 9.
2.	Set $i = 1$.
3.	Find the "tight" cluster having the most members different from the total set of members in all stored "core" clusters of alternative i of build up level j. Call this its "difference set." Call the cluster itself a "maximum distance" cluster.
4.	If this difference set is larger than that of any of the other alternatives of build up level j yet considered, drop these alternatives; consider only the present alternative and go to Step 5. If it is smaller, drop the present alternative and go to Step 6. If it is the same as that of other alternatives of build up level j, consider all still as possible alternatives and go to Step 5.
5.	If there is only one "maximum distance" cluster, store its "difference set" as the next "core" cluster for alternative i and go to Step 6; if there is a tie, go to Step 8.
6.	Have all alternatives of build up level j been considered? If yes, go to Step 7. If no, add one to i and go to Step 3.
7.	For any given alternative, are all possible objects in one of the stored "core" clusters? If so, print out the core clusters for all alternatives and terminate procedure; if not, add 1 to j and go to Step 2.
8.	Of the set of clusters involved in the tie, pick the smallest and store its "difference set" as a "core" cluster for alternative i, and go to Step 6. If there is still a tie, go to Step 9.
9.	Form a "dissimilarity" matrix for the clusters in the tie, where two clusters are considered dissimilar if their "difference sets" contain no common member. Find all the "tight" clusters for this matrix. Each "tight" cluster here will represent a set of the original "tight" clusters from the input similarity matrix whose "difference sets" are disjoint. Store the largest such set of "difference sets" as a set of "core" clusters. If there is a tie for the largest set, all alternatives will be followed in the hope that subsequent choices of "cores" will favor some alternatives over others. They are therefore added to the alternative list of the next level of build up. Note that it is possible that more than one core will be added to each alternative by Step 9. By convention, this addition is still treated as one level of build up. Go to Step 6.

Table 4 **Steps in the cluster adjustment program.**

Step No.	*Step Description*
1.	Set $i = 1$.
2.	Set $j = 1$.
3.	Consider the j^{th} member of cluster i: Compute from the similarity matrix the number of objects in the first large cluster to which this j^{th} object is similar. Divide this by the number of objects in the first large cluster to produce a percentage "match" of the j^{th} object to the first large cluster. Compute such a percentage "match" of the j^{th} object with each of the large clusters and with each of the small clusters already considered.
4.	Determine whether any of these matches are above some threshold (such as 0.8). If yes, go to Step 5; if no, go to Step 6.
5.	Delete the j^{th} object from its small cluster and put it into the cluster offering the best match.
6.	Add 1 to j; determine if all members of cluster i have been considered? If no, go to Step 3; if yes, go to Step 7.
7.	Add 1 to i: determine if all clusters have been considered? If no, go to Step 2, if yes, go to Step 8.
8.	Iterate this entire procedure as many times as desired, with the hope that stability will be eventually obtained.
9.	Compute for all remaining pairs of clusters, x and y, a measure of their interaction, which is given by $$I_{xy} = \frac{1}{N_x N_y} \sum_{\alpha=1}^{\alpha=Nx} \sum_{\beta=1}^{\beta=Ny} S_{\alpha\beta},$$ where N_x is the number of members in cluster x, N_y is the number of members in cluster y, $S_{\alpha\beta}$ is 1 if member α of cluster x is similar to member β of cluster y; is 0 if they are not similar. I_{xy} is the percentage of possible similarity "links" which are actually present between the members of cluster x and the members of cluster y.

These measures can be used to judge the value of the cluster set. If it has been decided that dependence between variables is the only valid "cause" of clusters, another test can be made for this similar to the one described in the next section.

As an option, any set of binary variables associated with an object can be entered into the computer for the entire object set. It will then compute for each cluster the average value of each of these variables over the set of objects in that cluster. This procedure is useful for identifying characteristics of each cluster and for noting the effect of clustering on variables that are not used in the clustering procedure.

Causality clusters

In an important class of clustering problems, it is desired that the clusters be chosen to reflect the multivariate dependence existing in the data. Each cluster would supposedly represent the effect of one cause, so that the goal of the clustering process would be to separate and identify the causes which produced the observed object set. To do this, it is necessary to postulate the characteristics of a cluster produced by only one cause. Of course, such a choice must be somewhat arbitrary and the results will therefore depend on how the model fits the particular problem. It should also be understood that statistical methods can never determine whether any observed multivariate dependence was produced by an actual cause; rather they only suggest that a search for such a cause might be productive.

Here it is postulated that a cause results in a particular "state" of the attributes: in other words, if only this cause were present, the attributes $X_1, X_2 \cdots X_t$ of each object would theoretically be some fixed set of numbers $(X_1)_0$, $(X_2)_0$, $\cdots$ $(X_t)_0$. In addition, it is postulated that the actual value of an attribute of an object can be different from the theoretical value because of random fluctuations. These fluctuations are such that the attributes are independent random variables, with the means given by the theoretical values. On this basis, the clustering problem now becomes the problem of finding sets of objects where the attributes are estimated to be independent within a set.

• *Test of cluster validity*

To test cluster validity it is first postulated that the given object set represents only one cluster. In testing this assertion, a hypothetical object population is set up which has attribute means and variances given by estimates of those of the actual population. However, in this hypothetical population, the attributes are independent and their means are normally distributed. The philosophy of the clustering procedure to be followed involves finding sets of objects which do *not* come from this population. If none can be found, the original object set is judged to be one cluster. If any is found, it is removed from this set as a cluster. The statistical test used to judge whether a cluster will be removed will now be described.

A cluster of N_k objects are drawn from the actual population and the attribute means $(\bar{x}_i)_k$ estimated from the objects in the cluster. A statistic G_k is then computed by (4)

$$G_k = \sum_{i=1}^{T} \frac{[(\bar{x}_i)_k - \bar{x}_i]^2}{s_i^2 / N_k}, \tag{4}$$

where i = attribute index
j = object index
T = number of attributes
N = number of objects in total given population
N_k = number of objects in cluster

$$\bar{x}_i = \left(\frac{1}{N}\right) \cdot \sum_{j=1}^{N} x_{ij}$$

$$(\bar{x}_i)_k = \left(\frac{1}{N_k}\right) \cdot \sum_{j \text{ over } j_0} x_{ij},$$

where j_0 is a set of objects in cluster

$$s_i^2 = \sum_{j=1}^{N} \frac{(x_{ij} - \bar{x}_i)^2}{N - 1}.$$

For the hypothetical population, G has a χ^2 distribution which allows us to calculate the probability P that $G \geq G_k$. Here P is the probability that N_k objects *picked at random* from the hypothetical population will have a G greater than or equal to G_k. However, we have not picked N_k objects at random but instead have employed some clustering technique. Let us assume we have used a perfect clustering technique, one which finds that set of objects for which G is maximum (let us say G_k). Then it is important to know the probability P_d of the event $G \geq G_k$ occurring at least once in a sample set of size N drawn from the hypothetical population, for if the event happened even once, our perfect clustering technique would have found it. The probability P_d is difficult to calculate, but it is easy to calculate a probability which is suspected to exceed P_d and therefore will hopefully give a pessimistic estimate of the situation. This latter probability, P_k, is obtained in Eq. (5) by first calculating the probability that the event will never happen in all $\binom{N}{N_k}$ ways of drawing N_k objects from an N object set and then subtracting this from 1:

$$P_k = 1 - (1 - P)^{\binom{N}{N_k}}. \tag{5}$$

If P_k is less than some small number, such as, say, 0.01, then we have equivalent assurance that our cluster was not drawn from the hypothetical population. Such an occurrence could arise because the attributes in the real population are either not independent or their means are not normally distributed or some of both. If P_k is close to 1, we have some assurance that our cluster could have been drawn from the hypothetical population but we cannot say that it could not have been drawn from a correlated population. Calculation of P_k must therefore be viewed as a test which admittedly combines some weaknesses but has the advantage of physical realizability.

Let us assume now that we have found a cluster and through (5) have shown that it is proper to separate it from the total object set. What can we say about the independence of attributes within the cluster? If we define the set of objects in the cluster as a new total population, we can then attempt to find clusters in this population by use of our original method. If we find none, we assume that the attributes are independent and it is therefore only one cluster. If we find some, we repeat the procedure using each of these in turn as the whole population. A simple example will now be given, in which the attributes are binary. This will provide the reader with a more intuitive understanding of the situation.

Table 5 **Ten objects with three binary attributes.**

Object	*Attribute* 1	2	3
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	1	1	1
7	1	1	1
8	0	0	0
9	0	0	0
10	0	0	0

We will assume that we have ten objects of three binary attributes each, as shown in Table 5.

These attributes are perfectly correlated. Calculations yield

$$G_k = \sum_{i=1}^{i=3} \frac{[(\bar{x}_i)_k - 0.7]^2}{0.233/N_k}. \tag{6}$$

If we choose Objects 1 through 7 for association in Cluster 1 we get

$$G_1 = \frac{7}{0.233} \times 3 \times (0.3)^2 = 8.1$$

$$P_1 \cong 1 - (1 - 0.018)^{\binom{10}{3}} \cong 0.2.$$

If we choose as Objects 8 through 10 for association in Cluster 2 we get

$$G_2 = \frac{3}{0.233} \times 3 \times (0.7)^2 = 18.9$$

$$P_2 < 1 - (1 - 0.0001)^{\binom{10}{3}} \cong 0.012.$$

The value of P_1 is too high to reject possibility that the cluster was chosen from the hypothetical population. P_2, however, is sufficiently low to allow Cluster 2 to be judged as valid. The object set is therefore split into two parts.

The value P_2 is also minimum with a choice of Cluster 2, which is the only set of objects which would be judged as a valid cluster. Within both Cluster 1 and 2, the attributes are independent by best estimate since $P_a(b) = P(b)$ for all attribute pairs a and b.

This is always true for a deterministic attribute (an attribute that has only one value) since knowledge of other events are of no help in predicting its value. Therefore, no further clustering need be done; indeed, no valid subclusters of Cluster 1 or 2 would be found using the technique.

This example indicates how this method of determining value for a cluster splits the original object set in such a way that correlations between attributes contributing to large values of G_k tend to be removed and replaced by independence of these attributes within the chosen cluster. This tends to take place because attributes contributing to large values of G_k usually have cluster means of 0 or 1 and are therefore deterministic attributes within their cluster.

Factor analysis

Factor analysis[9] is a statistical technique for finding a certain kind of organization in data. It starts with an object set, forms a correlation matrix for all attribute pairs, and proceeds to produce a structure consisting of a set of factors where each factor is described by a set of loadings, one for each attribute. If the factors are uncorrelated, one requirement of this structure is that the mathematical relation of Eq. (7) hold true:

$$r_{xy} = \sum_{k=1}^{k=F} f_k(x) f_k(y), \tag{7}$$

where F = number of factors
k = factor index
x, y = attributes indices
r_{xy} = correlation coefficient between x and y
$f_k(x)$ = factor loading for attribute x in factor k.

The resultant factors from a factor analysis cannot be interpreted as over-all descriptions of the clusters. However, it is specifically shown in Appendix II that, under certain assumptions, over-all descriptions of clusters can be found which satisfy Eq. (7). In other words, given clusters which satisfy the assumptions of Appendix II, values of $f_k(x)$ can be calculated from Eq. (A8). This procedure can be viewed as use of clustering techniques to ultimately calculate "factor loadings" [only in the sense of satisfying Eq. (7)]. Since for any set of objects there will be a finite number of ways to produce clusters satisfying the assumptions of Appendix II, there will only be a finite number of sets of $f_k(x)$ values. This is in contrast to factor analysis which, through continuous rotation, allows an infinite number of possible solutions, a direct consequence of the fact that the correlation matrix does not contain enough information to allow further resolution. In clustering techniques which use a test for cluster validity as described in the previous section, advantage is taken of complete knowledge of the object set and of its multivariate dependence information to further restrict allowable solutions. The situation depicted in Table 6 illustrates this point.

The three attributes are pairwise independent but are found to be dependent when considered as a triplet. A factor analysis of the situation yields three factors, one for each variable. A clustering algorithm, which finds the clusters that minimize the measure of cluster validity of Eq. (5), yields four clusters (Objects #1 to #3, Objects #4 to #6, Objects #7 to #9 and Objects #10 to #12). The result of calculating "factor loadings" for these clusters using Eq. (A8) of Appendix II is given in Table 7.

Table 6 **Sample object set.**

Object	*Attribute* 1	2	3
1	0	0	0
2	0	0	0
3	0	0	0
4	1	1	0
5	1	1	0
6	1	1	0
7	1	0	1
8	1	0	1
9	1	0	1
10	0	1	1
11	0	1	1
12	0	1	1

Table 7 **Calculated "factor" loadings.**

Attribute Number	*Cluster or "Factor" Number* 1	2	3	4
1	−0.5	0.5	0.5	−0.5
2	−0.5	0.5	−0.5	0.5
3	−0.5	−0.5	0.5	0.5

This example shows how cluster analysis can produce a result consistent with the mathematical structure of factor analysis [as expressed by Eq. (7)] which could not have been obtained from knowledge of the correlation matrix alone.

The conclusion is that both clustering (using the test for cluster validity) and factor analysis are techniques which are aimed at discovering information about multivariate dependence; factor analysis infers this dependence from pairwise correlations, whereas clustering observes it directly. To compare them further, a problem was attacked using each, as will be described later.

Clustering Programs I and II deal with information contained in a similarity matrix. The number of elements in such a matrix depends on the square of the number of objects. This squaring effect plus a fixed amount of high speed memory in the computer limits the allowable object set to a maximum of 350. The number of attributes per object can be large, however, because the size of the similarity matrix does not depend on this parameter.

A required calculation in a factor analysis is a correlation matrix between attributes. Here the square of the number of attributes determines memory limitation, while

Table 8 **Program III.**

Step No.	*Step Description*
1.	Pick an object to act as a cluster center.
2.	Find the similarity between this object and all other objects using Eq. (1) as the measure. All objects more similar to the center than T are considered to be in the crude cluster. T is an arbitrary threshold.
3.	Compute the typical member of this cluster. Compute the expected number of clusters rarer than this to be found in an uncorrelated population, as given by $\binom{N}{N_k}P$. This quantity is a good approximation to P_k when it is very small and it is easier to calculate. If this number is greater than a preset number K, go to Step 7; otherwise, hill climbing will be done in Step 4.
4.	Find the similarity between the typical member and all other objects using the following measure: Add up the weights [as given by appropriate individual terms of Eq. (4) as calculated for the last cluster] of all attributes where there is bit match between an object and the "typical" member. If this sum is greater than a certain percentage Y of the total possible [G_k in Eq. (4)], then this object is judged "similar" to the typical member. This measure weights highly attribute matches that contributed most to making the last cluster as rare as it was. All objects similar to this typical member are now members of the new cluster. This method is intended to be a crude approximation to the slower but better procedure of recalculating $\binom{N}{N_k}P$ for each object under consideration and accepting only those that lower this measure from its value in the previous cluster.
5.	Is this cluster the same as the last? If so, go to Step 6; if not, go to Step 3. This procedure is a check to see if stability has been reached. Note that stability does not signify that the rarest cluster in the vicinity has been found. A better but more complicated procedure, used whenever a cluster is found that is less rare than the previous one in the iteration, would be to raise the value of Y and recompute the cluster.
6.	Store the stable cluster as a final cluster. Delete each member of this cluster from consideration as a future cluster "center".
7.	Have all allowable objects been used as cluster centers? If no, pick one and go to 2; if yes, terminate the program.

the number of objects does not affect the size of the correlation matrix.

Both methods resort to matrices, the use of which limits the size of the problem that can be handled. To circumvent this, and also to take advantage of the test of cluster validity, another program was developed.

Clustering Program III

The input to this program is an object set with binary attributes. The algorithm picks a random "center," builds a crude cluster around this, and then "hill climbs" to a better cluster. The yardstick of cluster "goodness" is the measure of rarity given by Eq. (5). A detailed description is shown in Table 8.

It can be seen that an object can be in more than one final cluster when this technique is used. With some changes in the similarity measures, the same clustering philosophy can also be used where the attributes are continuous variables. Such a program is now being written.

The program can handle a problem in which the input set contains a maximum of 720,000 bits, i.e., 2000 objects of 360 binary variables each. The problem described in the experimental results required three minutes of computer time.

Experimental results

A problem in the medical field was chosen to test these programs. The set of objects was a set of 350 patients, all of whom had been diagnosed by physicians as having pyelonephritis. Each patient was described by the presence or absence of each of eighteen symptoms. It is recognized that a binary symptom description is sometimes less satisfactory than a multinomial or continuous representation; however, it was felt in this case that sufficient information was retained by the binary representation to justify the study.

Pyelonephritis is an inclusive term used by physicians to classify cases that fall in a certain broad area but do not much resemble each other. It is a poorly defined condition which Dr. Hans Zinsser, of the Columbia Presbyterian Medical Center in New York, suspected might be a combination of better defined diseases. Finding definitions for these subdiseases is, then, a clustering problem.*

Note that this is not the medical diagnosis problem† where each given patient has a known disease and the task is to produce a logic which will place this disease into the proper category. The task here is to help define the disease categories. Although the emphasis in this problem is on "causality" rather than "structure" (for the sense of the quotes, see the Introduction), Clustering Programs I and II were tried to judge their effectiveness in such a problem.

A similarity matrix was formed using Program I and the similarity definition of Eq. (2). The similarity matrix of the similarity matrix was then taken five times, also using Program I. Cluster cores were now found using Program II and these were introduced into the Cluster Adjustment Program. A number of runs were made to adjust parameters to yield a reasonable result as judged by the probability measure of the last program. The characteristics of the valid resultant clusters are given in Table 9. This table lists the percentage of patients in each cluster having each of the 18 symptoms.

It should be mentioned that a measure different from Eq. (5) had been used here. The expected number of clusters, all having attribute means farther from the cor-

* Results of the test from a medical point of view can be found in Ref. 10.

† An example of medical diagnosis by computer is given in Ref. 11.

Table 9 **Subdiseases from Clustering Programs I and II.** *Percentage of patients for which variable is present in each cluster.*

Cluster Number	*Total Set*	I	II	III	*Residue Set*
No. of Patients in Cluster	350	58	32	25	235
1. Bacteria	0.75	0.95	0.84	0.80	0.68
2. Obstruction	0.57	0.95	0.59	0.48	0.48
3. Chills	0.18	0.17	0.94	0.16	0.09
4. Fever	0.33	0.09	1.00	0·12	0.31
5. Pain	0.49	0.36	0.94	0.80	0.44
6. Nausea	0,25	0.19	0.28	0.80	0.21
7. Decreased Output	0.06	0.04	0.09	0.12	0.06
8. Abdominal/Back Signs	0.33	0.22	0.22	0.92	0.30
9. Urinary WBC	0.41	0.85	0.31	0.24	0.34
10. Urinary Bacteria	0.26	0.45	0.13	0.08	0.25
11. Urinary RBC	0.72	0.85	0.81	0.92	0.66
12. WBC	0.50	0.52	0.38	0.80	0.49
13. Sediment Rate	0.43	0.85	0.50	0.88	0.26
14. Dilatation	0.11	0.09	0.03	0.04	0.13
15. Blunting	0.11	0.02	0.09	0.04	0.14
16. Infundibula narrowed	0.11	0.09	0.06	0.04	0.33
17. Uremia/Toxemia	0.05	0.09	0.00	0.04	0.04
18. Chronicity	0.44	0.48	0.38	0.40	0.40

Table 10 **Subdiseases from Clustering Program III.**

Sub-disease	I	II	III	IV
No. of Patients in Set	56	50	42	29
$\binom{N}{N_k} P$	0.2×10^{15}	0.2	0.8×10^{15}	0.2×10^{10}
1. Bacteria	0.95	0.86	0.76	0.00
2. Obstruction	0.66	0.62	0.57	0.45
3. Chills	0.14	1.00	0.10	0.03
4. Fever	0.27	1.00	0.33	0.35
5. Pain	0.39	0.72	0.71	0.55
6. Nausea	0.25	0.26	1.00	0.28
7. Decreased Output	0 05	0.10	0.12	0.10
8. Abdominal/ Back Signs	0.20	0.32	1.00	0.38
9. Urinary WBC	1.00	0.38	0.43	0.10
10. Urinary Bacteria	1.00	0.18	0.19	0.03
11. Urinary RBC	0.79	0.76	0.74	0.00
12. WBC	0.57	0.52	0.62	0.21
13. Sediment Rate	0.61	0.46	0.55	0.24
14. Dilatation	0.18	0.10	0.19	0.14
15. Blunting	0.07	0.10	0.10	0.07
16. Infundibula narrowed	0.25	0.10	0.19	0.07
17. Uremia/Toxemia	0.13	0.02	0.00	0.00
18. Chronicity	0.50	0.42	0.43	0.52

responding attribute means in the total set than the observed cluster, was calculated using a binomial distribution. All three observed clusters had a probability of less than 0.01 using this measure.

Use of this method of clustering proved unwieldy in the test and is not normally recommended for finding clusters based on multivariate dependence.

In the next phase of the experiment, the problem was attacked using Clustering Program III and four clusters were found, as shown in Table 10. Three of these had about the same characteristics as those from the other clustering method. The fourth (No. IV in Table 10) seems to represent the condition of "hardly any symptoms" rather than a valid subdisease. It is interesting, however, that it was found here and missed before, since the value of Eq. (5) gives it as much right to be considered as Clusters I and III. The reason for this is that the similarity measure (Eq. 2) used in the first method considers zero matches unimportant.

A weakness of the use of Eq. (5) can be seen by observing the large values of P_k obtained for Clusters I, III and IV. One interpretation is that they could have been chosen from the null population, but they could also have been chosen from a weakly correlated real population, as was the actual case. The presence of actual correlations in the correlation matrix gives justification for keeping them. It is unfortunate that knowledge from a correlation matrix had to be used to determine this, since it introduces the very kind of thing the program attempts to avoid. A saving grace, however, is that the entire correlation matrix need not be calculated, only those correlations of interest.

Another idiosyncrasy of Program III is the possibility that almost the total value of G_k will be contributed by one variable. This situation, combined with a violation of the assumption of normality of the distribution of the variable mean, can result in a small value of P_k for the cluster. To guard against calling such clusters valid, the contribution of each variable to G_k is indicated by the program.

Finally, a factor analysis using a maximum likelihood method was carried out, resulting in three factors whose loadings are shown in Table 11. These are quite similar in principle to the clusters of Table I (since the assumptions of Appendix II are not accurately met, the loadings can be interpreted only approximately as clusters). It is interesting to note that rotation to simple structure reduces the number of factors to two, essentially combining I and III. Yet cluster analysis shows two distinct sets of people (only 5 of 93 in common) with characteristics like these two factors.

This points up a fundamental problem in factor analysis: What criterion should govern the rotation of axes? Here the choice of simple structure leads to an unwar-

ranted combination of factors from a clustering point of view.

In summary, the three techniques yielded similar results. Use of Clustering Programs I and II for this kind of problem, although possible, is unwieldy and not recommended. Program III gave the best results but showed some of the shortcomings of the test for cluster validity. Because only pairwise correlations are used in factor analysis, this method combined two factors corresponding to clusters, which cluster analysis had separated.

Conclusions

The major point to be made is that clustering methods, as represented by Program III, can be used for problems now done by factor analysis. It is not implied that such a cluster analysis should replace factor analysis, but that both methods applied to the same data should yield a deeper understanding than either method alone.

In a factor analysis, the goal is to explain the observed correlation matrix using as few factors or "underlying causes" as possible. In a cluster analysis, the goal is to determine the presence and nature of multivariate dependence and use this information to suggest the underlying causes.

Much work needs to be done to improve the clustering techniques in the areas of: 1) the clustering algorithm, 2) the measure of cluster validity, 3) extension to multistate attributes.

For the more conventional type of clustering problems where structure is of primary interest or where special value judgments require an unconventional definition of similarity, Programs I and II are applicable.

Table 11 **Subdiseases from factor analysis.**

	Factor I	*Factor II*	*Factor III*
1. Bacteria	0.44	0.08	−0.08
2. Obstruction	0.24	0.08	0.05
3. Chills	0.10	0.67	−0.12
4. Fever	0.01	0.65	−0.09
5. Pain	−0.04	0.33	0.31
6. Nausea	0.11	0.12	0.37
7. Decreased Output	−0.03	0.04	0.15
8. Abdominal/Back Signs	0.08	0.11	0.41
9. Urinary WBC	0.55	−0.06	−0.07
10. Urinary Bacteria	0.36	−0.21	−0.25
11. Urinary RBC	0.20	0.00	−0.01
12. WBC	0.29	0.07	0.13
13. Sediment Rate	0.32	0.03	0.13
14. Dilatation	0.18	0.00	0.13
15. Blunting	−0.09	−0.04	−0.15
16. Infundibula narrowed	0.17	0.00	0.00
17. Uremia/Toxemia	0.26	−0.05	−0.01
18. Chronicity	0.10	−0.08	−0.02

The method of taking the similarity matrix of the similarity matrix, as represented by Program I, makes finding all tight clusters practical. Program II finds these tight clusters using the algorithm of Appendix I, which is felt to be an improvement over previous methods. The cluster adjustment program is useful to reassign objects to better clusters based on the original similarity matrix.

The ultimate value of this set of programs can be ascertained only after use on a number of problems. Such an evaluation is a goal of future work.

Acknowledgments

I owe a special debt to Rolf Bargman for performing the factor analysis and for many hours of interesting and productive conversation, although any opinions expressed are necessarily my own. I also would like to thank S. Ghosh and R. Casey for useful comments and L. Cohen for programming help.

Appendix I. Algorithm for finding "tight" clusters

The algorithm builds up a cluster, one object at a time. It keeps track of three things at each level i of buildup:

1. The set of objects (A_i) in the cluster up to this point.
2. The set of objects (C_i) which could possibly be added to A_i to further increase the cluster.
3. The number (L_i) of the last object of C_i to be considered for addition to the cluster.

These three things are stored for each i which is smaller than or equal to the present i. Also needed is the similarity matrix where the set of all members similar to object L_i is called S_{L_i}

Table A1 **Cluster-building algorithm.**

Step No.	*Step Description*
1.	Set $i = 1$, C_1 = all objects, A_1 = no objects, $L_1 = 1$.
2.	Consider C_i for the presence of object L_i: if it is present, go to Step 3; if absent, add 1 to L_i and go to Step 5.
3.	Store objects common to C_i and S_{L_i} as C_{i+1}, deleting L_i (from C_{i+1}): Store objects in set A_i, plus object L_i, as set A_{i+1}.
4.	Add 1 to L_i and store as L_{i+1}: then add 1 to i.
5.	Is L_i greater than the number of the last possible object? If so, go to Step 6; if not, go to Step 2.
6.	Determine whether C_i is empty. If so, store A_i as a cluster; if not, it means either the cluster A_i has been found before or it is a subset of a cluster found before. In this case, do not store A_i. In any event, store A_i as T.
7.	Subtract 1 from i: Determine whether $i = 0$; if yes, all clusters have been found; if no, go to Step 8.
8.	Form the set of all objects in C_i with numbers greater than L_i: Determine whether this set is a subset of T. If so, it means that there is no point attempting to add these objects to A_i as the result will be the same as or a subset of T. Therefore, go to Step 7: If not, go to Step 2.

Appendix II. Calculation of "pseudo-factor" loadings using clustering parameters

F = number of clusters
r_{xy} = correlation coefficient between attributes x and y.
$j \equiv$ object index
$k \equiv$ cluster index
$x_j \equiv$ value of attribute x in object j.
$y_j \equiv$ value of attribute y in object j.
$N \equiv$ number of objects
$N_k \equiv$ number of objects in cluster k
$j_k \equiv$ set of objects in cluster k

$$\bar{x} \equiv \frac{1}{N} \cdot \sum_{j=1}^{j=N} x_j$$

$$(\bar{x})_k \equiv \frac{1}{N_k} \cdot \sum_{jk} x_j$$

$$s_x^2 \equiv \frac{1}{N} \cdot \sum_{j=1}^{j=N} x_j^2 - \bar{x}^2$$

$$\overline{xy} = \frac{1}{N} \cdot \sum_{j=1}^{j=N} x_j y_j$$

By definition:

$$r_{xy} = \frac{\overline{xy} - \bar{x}\bar{y}}{s_x s_y}. \tag{A1}$$

If all clusters are disjoint then Eqs. (A2) and (A3) are valid:

$$1 = \sum_{k=1}^{k=F} \frac{N_k}{N} \tag{A2}$$

$$\bar{x} = \sum_{k=1}^{k=F} \frac{N_k}{N} (\bar{x})_k. \tag{A3}$$

If in addition all attributes are assumed independent within each cluster then

$$\overline{xy} = \sum_{k=1}^{k=F} \frac{N_k}{N} (\bar{x})_k (\bar{y})_k. \tag{A4}$$

Define now $M_k(x)$ as the mean value of attribute x in cluster k normalized by the total population means and variance.

$$M_k(x) \equiv [(\bar{x})_k - \bar{x}]/s_x. \tag{A5}$$

Then form the sum of Eq. (A6)

$$\sum_{k=1}^{k=F} \frac{N_k}{N} M_k(x) M_k(y) = \sum_{k=1}^{k=F} \frac{N_k}{N} \left(\frac{(\bar{x})_k - \bar{x}}{s_x}\right)\left(\frac{(\bar{y})_k - \bar{y}}{s_y}\right). \tag{A6}$$

Expansion and use of Eqs. (A1) to (A4) produces Eq. (A7):

$$r_{xy} = \sum_{k=1}^{k=F} \frac{N_k}{N} M_k(x) M_k(y). \tag{A7}$$

If we then define $f_k(x)$ as

$$f_k(x) \equiv \sqrt{N_k/N}\, M_k(x), \tag{A8}$$

Eq. (A7) will have the same structure as Eq. (7) in the text.

It can be shown that $| f_k(x) \leq 1 |$, a requirement for a loading.

References

1. H. E. Stiles, *Journal ACM* **8**, 271–279 (1961).
2. P. B. Baxendale, *IBM Journal* **2**, 354–361 (1958).
3. D. Rogers and T. Tanimoto, "A Computer Program for Classifying Plants," *Science* **132**, 1115–1118 (October 1960).
4. W. Brandenburg, H. C. Fallon, C. B. Hensley, T. R. Sorage, and A. J. Sowarby, "Selective Dissemination of Information, SDI-2 System," unpublished report.
5. M. Kochen, "Techniques for Information Retrieval Research: State of the Art," presented at IBM World Trade Corporation Information Retrieval Symposium at Blaricum, Holland, November, 1962; to be published in the Proceedings of the Symposium.
6. C. T. Abraham, "Evaluation of Clusters on the Basis of Random Graph Theory," unpublished report.
7. R. M. Needham, "Theory of Clumps II," report M. L. 139, issued by Cambridge Language Research Unit, Cambridge, England, 20–22.
8. J. Kuhns, "Work Correlation and Automatic Indexing," report issued by the Ramo Wooldridge Corporation, Canoga Park, California, Appendix D (December, 1959).
9. H. H. Harman, *Modern Factor Analysis*, University of Chicago Press, Chicago, 1960.
10. H. Zinsser, R. Bonner, A. Lemlich, L. Roots, "Pyelonephritis: Study of a Disease in Depth," Proceedings, Fourth IBM Medical Symposium, October, 1962.
11. R. S. Ledley and L. B. Lusted, "The Use of Electronic Computers in Medical Data Processing," *IRE Trans. on Medical Electronics*, **ME-7**, 31 (1960).

Received February 12, 1963