

Computer Programs for Performing Iterative Partitioning Cluster Analysis

Roger K. Blashfield
University of Florida

Mark S. Aldenderfer
University of Missouri—St. Louis

Eight programs which perform iterative partitioning cluster analysis are analyzed; they are discussed in terms of versatility of options, accuracy, and cost. These eight programs contain very different heuristic approaches to finding the optimal partition of a data set; the different heuristic approaches are shown to affect both accuracy and cost of clustering solutions. It was not possible to recommend any one program as generally being preferable, however, because of the striking variability in these programs and the lack of knowledge about iterative partitioning methods.

Computer programs for performing hierarchical agglomerative methods of cluster analysis have been discussed in an earlier article (Aldenderfer & Blashfield, 1978). The present article is a companion piece designed to discuss software which contain iterative partitioning methods.

The reviews of iterative methods have been few and, in general, have not been comprehensive (Anderberg, 1973; Ball, 1970; Friedman & Rubin, 1967; Lance & Williams, 1967). Most reviews have been concerned with either presenting a single viewpoint on iterative methods or suggesting a new method to users. Thus, the general comparison of the dimensions of iterative methods, unlike the extensive com-

parison of hierarchical methods, has not yet appeared.

Although there has been little research use of iterative methods, considerable computer software has been developed to perform this type of clustering. Eight popular programs and packages contain iterative methods. The popularity of these programs has been determined by responses to a questionnaire, as summarized by Blashfield (1976a). Of the eight, two are general statistical packages which contain a number of multivariate statistical methods (BCTRY and CLUSTAN); two are from books on cluster analysis (ANDER from Anderberg's [1973] *Cluster Analysis for Applications* and HART from Hartigan's [1975] *Clustering Algorithms*); and the final four (CLUS, HOWD, ISODATA, and MIKCA) are programs which perform only iterative partitioning cluster analysis.

This software, like that for hierarchical clustering, has not been extensively compared; and thus potential users are likely to be unaware of the strengths and weaknesses of each program or package. This ignorance of the quality of the software is intensified for iterative methods because of the relative lack of knowledge about the methods per se.

The purpose of the present article is to provide a partial remedy for these problems. Specifically, it will inform the user about critical dimensions of variability in iterative partitioning

APPLIED PSYCHOLOGICAL MEASUREMENT
Vol. 2, No. 4 Fall 1978 pp. 533-541
© Copyright 1978 West Publishing Co.

methods and provide a consumer-oriented evaluation of popular computer programs, so that interested users can choose the programs appropriate for their needs. Each of the eight programs listed above will be compared for standard options of clustering, accuracy of solution, and cost.

Iterative Partitioning Methods of Cluster Analysis

Because most researchers are unfamiliar with iterative clustering methods, a brief description of the logic and operation of these methods is desirable. The outline of a method proposed by Forgy (1965) illustrates the intent of the iterative partitioning procedures:

1. Begin with an initial partition of the data set into some specified number of clusters and compute the centroid of each cluster.
2. Allocate each data point to the cluster which has the nearest centroid.
3. Compute the new centroids of the clusters; centroids are not updated until there has been a complete pass through the data.
4. Alternate Steps 2 and 3 until no data points change cluster membership (Anderberg, 1973, p. 161).

Unlike hierarchical agglomerative methods, which require the calculation and storage of an $n \times n$ matrix of similarities between cases, iterative partitioning procedures usually only require the storage of the raw data plus a few relatively small additional matrices. Thus, iterative partitioning programs have the potential of handling distinctly larger data sets than hierarchical agglomerative methods.

However, iterative partitioning programs are subject to a different limitation. The most straightforward way to discover the optimal classification of a data set by means of an iterative partitioning method is to form all possible partitions of that data set. Unfortunately, the number of partitions which must be

examined by this simple approach can be enormous. For 15 entities and 3 clusters, the exhaustive approach would require the examination of 217,945,728,000 partitions. Obviously, even with high speed computers, this approach to clustering is not feasible.

Since all possible partitions of even a moderately sized data set cannot be exhaustively analyzed, the authors of iterative partitioning programs have created a wide range of procedures which attempt to sample a small subset of the possible partitions. These heuristic procedures are based upon reasonably plausible notions of how to sample those partitions which are most likely to contain the optimal partition. However, the "rules of thumb" (i.e., heuristic procedures) which different authors have found plausible are quite varied. Even within a program, an author has often considered a number of possible heuristic procedures for efficiently finding the optimal partition and has included these various procedures as options for the interested user. As a result, the difference in methodological procedures within and between programs is large. Most of the following dimensions of variability are concerned with different approaches to creating the best possible partition of a data set.

Dimensions of Heuristic Iterative Partitioning Procedures: Versatility of Program Options

The eight programs vary with respect to the choice of the initial partition, the type of pass, the statistical criterion, and fixed versus variable number of clusters.

Dimension 1: Initial Partition

The first characteristic which can be used to differentiate partitioning methods is the procedure by which the method is started. Some methods use initial estimates of cluster centroids (called "seed points" by Anderberg, 1973), while other methods select a *starting partition*.

If the former are used, the user specifies his/her best estimate of the cluster centroids; and in the first pass through the data, the entities are assigned to the cluster with the nearest centroid. On the other hand, if a starting partition procedure is used, the centroid of each cluster is defined as the multivariate mean of the corresponding partition; in the first pass through the data, entities are assigned to the nearest mean.

Almost all programs use different methods to select the starting partition. ANDER starts by specifying the "seed points." In ANDER the initial cluster centroids can be user-specified or can be some set of k data points which serve as the "seed points" for each of the k clusters. ANDER also allows the user to specify the starting partition. MIKCA starts by analyzing three different sets of randomly chosen seed points for the set which seems most likely to lead to an efficient solution.

For the programs which require the specification of a starting partition, CLUSTAN must be started by a user-specified partition or by assigning every k th entity to the same cluster (where k refers to the number of clusters). ISODATA starts by using cluster centroids which are relatively distant from the centroid of the total data set. CLUS permits four starting options: (1) randomly chosen partition; (2) user-specified partition; (3) a partition in which the first n/k entities are members of the first cluster, the second n/k entities are members of the second cluster, and so forth (where k = number of clusters and n = number of entities); and (4) an option in which the package chooses the starting partition "by its own method." Finally, BCTRY permits the user to specify a starting partition. However, the standard starting partition in BCTRY is dividing the q dimensional space into 2^q equal segments, where q refers to the number of factors found in multiple group factor analysis. The centroids of these segments form the initial "seed points" for the iterative clustering process.

Also, there are two programs which form clusters over a range for the number of clusters.

HOWD, for example, superimposes a hierarchical divisive algorithm onto a basic iterative k -means method. HOWD initially partitions the data set by finding the variable with the largest variance and assigning all entities with a score greater than the mean on that variable to one cluster and all other entities to the remaining cluster. The program uses an iterative k -means procedure until a stable two-cluster solution is found. It then searches for the variable in the two clusters with the largest variance. It subdivides that cluster at the mean of that variable and starts the next iterative k -means pass for a three-cluster solution. The process is repeated until an upper limit on the number of clusters is reached.

CLUSTAN works in the opposite direction by superimposing a hierarchical agglomerative procedure. In CLUSTAN a randomly chosen (or user-specified) partition is set for $k(max)$, where $k(max)$ is the upper limit on the number of clusters as specified by the user. The program uses an iterative k -means procedure to form the clusters. The next step is to merge the two closest clusters. Then, the iterative k -means passes are used to find the $k(max)-1$ solution. This process is repeated until stopped by the user or until a two-cluster solution is found.

Dimension 2: Type of Pass

This second dimension has been the object of frequent discussions; it involves the type of pass used to assign entities to particular clusters. ANDER, BCTRY, HOWD, and ISODATA exclusively utilize k -means passes (also called "nearest centroid" and "reassignment passes"). CLUS and CLUSTAN emphasize the use of hill-climbing passes. The latter type of pass, instead of assigning an entity to the cluster with the nearest centroid, moves an entity from one cluster to another if a particular statistical criterion is better optimized. One program, MIKCA, uses an interaction of both reassignment and hill-climbing passes in order to achieve a solution. Finally, CLUS permits passes

which “force” entities to join new clusters in order to start a new partitioning sequence. With the exception of one FORTRAN subroutine in ANDER (EXEC 5) which follows the original k -means algorithm of MacQueen (1967), all packages iteratively repeat passes until either no membership changes occur or until some iteration termination parameter is reached.

Within k -means passes, there are other distinctions which exist in these programs. For instance, most iterative k -means procedures use *combinatorial* passes in which cluster centroids are updated after a membership move. However, ANDER does contain two iterative k -means procedures in which the passes are *non-combinatorial*; that is, the cluster centroids are not updated until there has been a complete pass through the entire data set. Another distinction which is made by CLUSTAN concerns whether or not an entity is removed from its parent cluster when calculating the centroid of that cluster. CLUSTAN standardly removes an object from its parent cluster when calculating the distance between the object and the centroid of its cluster. Although removing the object guarantees that the algorithm must converge, most programs do not omit the object when calculating the value of the centroid. CLUSTAN thus permits the user to choose this option if so desired.

Dimension 3: Statistical Criterion

The hill-climbing passes in CLUS, CLUSTAN, and MIKCA are concerned with membership changes which lead to better values of a particular statistical criterion. CLUS and MIKCA give the user a choice of four statistics to optimize: $tr\mathbf{W}$, $tr\mathbf{W}^{-1}\mathbf{B}$, $|\mathbf{W}|$, and largest eigenvalue of $\mathbf{W}^{-1}\mathbf{B}$, where \mathbf{W} refers to the pooled within-cluster covariance matrix and \mathbf{B} is the pooled between-cluster covariance matrix. All four statistics are measures often discussed in multivariate analysis of variance (Olsen, 1976).

The user should note that an iterative k -means pass implicitly is concerned with opti-

mizing the $tr\mathbf{W}$ criterion. That is, a k -means procedure attempts to minimize the variance within each cluster. However, an iterative k -means pass and a hill-climbing pass which attempts to optimize $tr\mathbf{W}$ are not identical. In fact, these two different passes can yield distinctly varied solutions on the same data set.

Dimension 4: Fixed vs. Variable Number of Clusters

In ANDER, CLUS, and MIKCA the number of clusters must be specified by the user (fixed number of clusters). The remaining packages contain procedures which affect how many clusters may be found in the final solution (variable number of clusters). The manner in which these packages determine the final number of clusters is different. CLUSTAN agglomeratively collapses clusters across a range specified by the user, while HOWD uses a divisive procedure to form a range of cluster solutions. BCTRY and ISODATA provide procedures for “splitting” and “merging” clusters. ISODATA is the most flexible in this regard, as it permits the user to specify the limits on the diameter of a cluster and/or the limits on the size of the cluster. If clusters are too close, ISODATA may merge them: if a cluster is too heterogeneous, it may be split. In the same way, if clusters have too many members, they can be split; and if too small, ISODATA may assign them to an outlier group.

Choosing the Program Which Will Find the “Best” Partition

Little is known about the effects of these various dimensions on the results from iterative partitioning methods. In order to examine the effects of different combinations of dimensions on data sets, a pilot study was performed to compare these programs.

Method

The data sets used in the study were 20 mixtures created using a procedure described by

Blashfield (1976b). For each data set, k populations were statistically defined. That is, each population had a specified mean vector and covariance matrix. Using monte carlo techniques based upon the multivariate normal distribution, a sample of $n(k)$ entities was drawn. The same process was repeated for each of the k populations. The samples were then scrambled together to form a mixture. The resulting data set had a known classification structure, and the problem for a cluster analysis program was to recover the actual classification structure of the mixture.

Four programs were used in this pilot study: ANDER, CLUS, CLUSTAN, and MIKCA. The reason for choosing these programs was that they are at least somewhat similar in terms of their options. Each program was executed twice. For both ANDER runs, the initial partition was chosen by using the first k data points as "seed points." (The data had been scrambled before clustering began.) The type of pass was an iterative k -means pass, hence the $tr\mathbf{W}$ statistic was being optimized. The only difference between the runs was that for the first run Forgy's (1965) non-combinatorial procedure was used, and for the second run a combinatorial procedure was used.

Like ANDER, CLUSTAN was also run using iterative k -mean passes. Within the types of k -means passes, CLUSTAN used combinatorial passes in which an entity was kept within its cluster during the calculation of the centroid. The only difference between the two CLUSTAN runs was the starting partition. In the first, the every k^{th} entity starting partition was used. Before the second run, a separate cluster analysis was run using Ward's hierarchical agglomerative clustering method. The solution for k clusters from Ward's method was used as the starting partition to the second iterative k -means run of CLUSTAN. The second procedure for choosing the initial partition is recommended by CLUSTAN because Ward's method is considered likely to give a solution

which is relatively similar to the optimal partition.

The third program, MIKCA, was run using hill-climbing passes only. The starting partition was chosen randomly. The two MIKCA runs differed only in terms of the statistical criteria: $tr\mathbf{W}$ and $|\mathbf{W}|$. CLUS was operated in a similar manner to MIKCA. Random starting partitions were chosen. The two CLUS runs differed in terms of the statistical criteria: $tr\mathbf{W}$ and $|\mathbf{W}|$. However, CLUS differed from MIKCA in that CLUS used hill-climbing passes which were followed by occasional forcing passes.

In order to determine how well each program performed, the statistic kappa was used to measure the agreement between the final cluster solution of the program and the actual classification structure of the mixture. Kappa is a measure of the agreement between nominal scales of the same data (Cohen, 1960; Fleiss, 1973). This statistic ranges in value from 0 to 1 with higher values reflecting higher agreement rates (i.e., better solutions).

Results

The results from the pilot study are shown in Table 1. In this table the median kappa values are listed across the 20 mixtures and the number of mixtures for which the particular program had the highest value.

In discussing the results shown in Table 1, the focus will be on making contrasts which are relevant to the dimensions along which iterative partitioning programs vary. The first dimension is the initial partition. The contrast relevant to this dimension is a comparison of ANDER, when executed using combinatorial k -means passes, with the two different CLUSTAN runs. These three runs differ only in terms of the starting partition. In all other ways, the method used in clustering was exactly the same. However, the median kappa values for these three different runs varied from .64 to .71. Thus, varying the starting partition can have an effect on the solution which is obtained. The most ex-

Table 1
 Pilot Study on Iterative Partitioning Programs:
 Results Averaged over Twenty Monte-Carlo Mixtures

	Median Kappa	Number of "Best" * Solutions
ANDER (k-means, noncombinatorial)	.58	0
ANDER (k-means, combinatorial)	.64	0
CLUS (hill-climbing, $tr\ W$)	.54	2
CLUS (hill-climbing, $ W $)	.70	9
CLUSTAN (k-means, every kth)	.71	2
CLUSTAN (k-means, Ward's)	.64	1
MIKCA (hill-climbing, $tr\ W$)	.56	0
MIKCA (hill-climbing, $ W $)	.70	6

* Number of data sets for which the program obtained the largest kappa value.

Note. For each program, the options used are shown in parentheses. Kappa is measure of the correlation of the cluster solution with the actual classification structure of the mixture. The higher the value of kappa, the better the solutions. The number of "best" solutions refers to the number of solutions in which the program had the highest kappa value for each data set.

treme example of this effect is one particular data set on which ANDER, using the first k data points as seed points, obtained a kappa value of .29; CLUSTAN, when started using the cluster solution from Ward's method, had a kappa value of .30; and CLUSTAN, when using its every k^{th} object assignment procedure, obtained a kappa value of .94. Clearly, the results from an iterative partitioning method can be affected by the choice of the starting partition.

The second dimension concerns the type of pass. A number of contrasts are relevant here. First, consider the difference between MIKCA and CLUS using the $tr\ W$ criterion and any of the runs from ANDER and CLUSTAN which were entirely iterative k -means passes. The statistical criterion in both of these instances was the same, and only the type of pass differed. To the extent that these results can be generalized,

it would seem that hill-climbing passes, at least for the $tr\ W$ criterion, did not work as well as iterative k -means passes. Another contrast concerns the difference between MIKCA and CLUS when using the $|W|$ criterion. The major difference in this contrast concerned the addition of forcing passes to the standard hill-climbing k passes. CLUS does contain forcing passes, while MIKCA does not. Although the difference between these programs in terms of median kappa values was not great, there seemed to be a difference in terms of which program more commonly found the better solution.

The third dimension to be contrasted concerns the statistical criterion. CLUS and MIKCA using the $|W|$ criterion were generally found to be the best solution (15 of 20 mixtures), while runs involving iterative k -means and the $tr\ W$ criterion did not perform as well. Some

readers may be skeptical of this result and point out that the mixtures were based upon populations with distinct covariance structures. Thus, it would seem that the data were created in such a way that the $|W|$ criterion was likely to perform well. In fact, this concern is not completely justified. The $|W|$ criterion involves the assumption that the covariance structure for the different populations are identical. This assumption was not met for these data sets; very different covariance structures were randomly selected when defining the populations within a mixture.

Conclusions

To summarize the results from this pilot study, the different dimensions which were discussed earlier can have an important effect on the clustering solutions which are found using iterative partitioning analysis. This study only involved 20 mixtures and clearly was too limited to provide an adequate basis for choosing among these programs and their procedures. This is an intriguing topic for future research (cf. Anderberg, 1973). Without further research, users will have a difficult time rationally choosing which program is most likely to work best for their data.

Thus, no particular programs or particular procedures can be recommended as most likely to find a "good" classification. A user should remember that the iterative partitioning programs are based upon heuristics (i.e., rules of thumb) which may or may not be helpful when classifying the user's data. As a result, the user should be cautious when interpreting a cluster analysis solution. In addition, the user should test the adequacy of any cluster analysis solution which was generated using these heuristic procedures.

Cost

Another important feature to users of cluster analysis programs is the cost of performing an

analysis. Cost is an important dimension in the iterative partitioning programs because it is an explicit limitation on the heuristic approaches of these methods. After all, the most obvious partitioning strategy to determine the best cluster structure would be to test the homogeneity of all possible partitions of the data set and to choose that partition which has the optimal characteristics. As noted earlier, this strategy is not used because it is extremely expensive.

Since the most obvious partitioning strategy is not feasible, all practical iterative partitioning methods contain procedures for selecting partitions which seem relatively likely to yield the optimal value on the statistical criterion. The programs use iterative heuristic processes which are intended to efficiently (i.e., with relatively little cost) find the optimal solution. Hence, cost is an important dimension when considering the iterative partitioning methods.

Table 2 presents some general accounting information concerning five of the iterative partitioning methods used to cluster the Fisher iris data. This data set has 150 different entities measured along 4 dimensions. In order to obtain the data shown in Table 2, the options were set in each program so that the program performed using its standard options. Recorded after each run was the CPU time to execute the analysis (to the nearest .01 seconds), the total system time to run the program including system and input/output time (to the nearest 1 second), the total number of lines of output printed, and the actual cost at Pennsylvania State University on the IBM 370 model 168 to perform the analysis.

In analyzing Table 2, it should be obvious that the costs of the different programs do differ. The most expensive program is CLUS. This program standardly attempts to optimize the $|W|$ criterion. The iterative computation of this statistic is more expensive than the k -means option of ANDER, which does not require the computation of any statistical measure of homogeneity until the final partition has been chosen. On the other hand, the user should note that MIKCA also attempts to optimize $|W|$

Table 2
Cost of Iterative Partitioning Programs: Fisther Iris Data

	CPU Time	Total Time	Number of Printed Lines	Cost at Penn State
ANDER	1.10	5	103	.62
CLUS	9.77	40	917	5.03
CLUSTAN	.76	6	266	.87
HART	3.31	6	342	.87
MIKCA	1.16	5	78	.62

Note. CPU time is expressed to the nearest .01 seconds. Total time includes CPU time, system time and I/O time and is expressed to the nearest integer. The number of printed lines refers to the number of lines printed on output. Cost is in dollars for executing the program at Penn State on an IBM 370/168 based at \$.11 per second of total time and \$.07 per 100 lines printed on output.

and at much less cost than CLUS. Two other relatively expensive features of CLUS are (1) the hill-climbing passes, which require the computation of the statistic being optimized with each possible move, and (2) the fact that CLUS restarts the partitioning analysis in order to avoid the problem of "local maxima." Nonetheless, CLUS consistently is an expensive program to use when compared to the other programs.

Conclusions

As can be seen from the preceding discussion, there is little consensus on how to best use iterative partitioning methods of clustering. The popular programs clearly reflect the confusion of terminology, orientation, and logic which surrounds the use of these methods. The most notable feature of these programs has been the diversity of their algorithms, logic, statistics, output, and terminology. Comparing the various types of iterative partitioning software is much like trying to compare apples and oranges.

It is not possible to complete this report with a rank ordering of the programs. CLUS is clearly the most versatile of the packages, because it has the widest range of options of type of pass, starting partition, and so forth, for the user to choose. Unfortunately, CLUS is the most expensive of the iterative partitioning programs. Cost-efficient programs, such as CLUSTAN and HART, do not have similar versatility. HOWD, ISODATA, and BCTRY have distinctly unique properties which have no direct analogues in the other packages, thus making comparison of them with other programs difficult. The personal preference of the authors is to use ANDER and MIKCA; both are relatively efficient programs which represent the major options that exist in current thinking about the iterative partitioning methods.

References

Aldenderfer, M. S., & Blashfield, R. K. Computer software programs for performing hierarchical

- cluster analysis. *Applied Psychological Measurement*, 1978, 2, 405-413.
- Anderberg, M. R. *Cluster analysis for applications*. New York: Academic Press, 1973.
- Ball, G. H. *Classification analysis*. Menlo Park, CA: Stanford Research Institute, 1970.
- Blashfield, R. K. Questionnaire on cluster analysis software. *Classification Society Bulletin*, 1976, 3, 25-42. (a)
- Blashfield, R. K. Mixture model tests of cluster analysis: Accuracy of four hierarchical agglomerative methods. *Psychological Bulletin*, 1976, 83, 377-388. (b)
- Cohen, J. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 1960, 20, 37-46.
- Fleiss, J. L. *Statistical methods for rates and proportions*. New York: Wiley, 1973.
- Forgy, E. W. Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. *Biometrics*, 1965, 21, 768.
- Friedman, H. P., & Rubin, J. On some invariant criteria for grouping data. *Journal of the American Statistics Association*, 1967, 62, 1159-1178.
- Hartigan, J. A. *Clustering algorithms*. New York: Wiley, 1975.
- Lance, G. N., & Williams, W. T. A general theory of classificatory sorting strategies. II. Clustering systems. *Computer Journal*, 1967, 10, 271-277.
- MacQueen, J. B. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, 1, 281-297.
- Olsen, C. L. On choosing a test statistic in multivariate analysis of variance. *Psychological Bulletin*, 1976, 83, 579-586.

Acknowledgements

This paper was supported by a grant from the National Science Foundation (Grant DCR #74-20007). The computer usage discussed in this paper was performed at The Pennsylvania State University.

Author's Address

Roger K. Blashfield, Department of Psychiatry, J. Hillis Miller Medical Center, University of Florida, Gainesville, FL 32610.