developed by J. K. M. Moody and J. Hollis, based on a suggestion by R. M. Needham. But, they observed, "It is obvious that the algorithm is very ill-suited to being programmed in a language such as FORTRAN."

This abstract describes a FORTRAN version of the Needham-Moody-Hollis algorithm which uses list-processing methods not usually associated with the FORTRAN language. Subroutine CLIQUE and its auxiliary subroutines are mainly concerned with building, searching, and modifying simple list structures called "chains." Each chain consists of a set of linked nodes, and each node consists of three integers: the address of the data item stored at that node, the address of the next higher node, and the address of the next lower node.

The algorithm goes through one cycle for each row of the dissimilarity matrix. Each cycle involves dividing an initial Set C of logical vectors (which are used to represent subsets of the points) into three mutually exclusive subsets, D, E, and F. At the end of a cycle, D, E, and F must be pooled to form a new set, C. At the end of the last cycle, C contains the logical vectors representing clusters at the current level.

By treating the four sets as chains, it is easy to add or delete arbitrary elements, to deal efficiently with continually changing set sizes, and to pool D, E, and F by simply linking the appropriate chains.

INPUT

Arguments passed to Subroutine CLIQUE are DIST, a REAL*4 array dimensioned (60,60) in the calling program; NP, the number of points to be clustered; and CRIT, the REAL*4 cutoff value for the current level. DIST (I,J) is the dissimilarity between Points I and J, with $1 \leqslant I < J \leqslant NP$. For compatibility with the Jardine and Sibson programs, DIST(I,J) = ±1.1E10 is treated as DIST (I,J) = 0.0. DIST(I,J) $\leqslant$ CRIT means Points I and J are connected in the graph for Level h = CRIT.

OUTPUT

When CLIQUE returns to the calling program, the clusters for the current level are stored as a chain local to CLIQUE. A call to UNPACK returns the next cluster from the chain by means of an INTEGER array INDEX. INDEX(I) = 1 if Point I is in the set (= 0 if not). UNPACK also returns an INTEGER flag which is positive until the last cluster has been returned. Then it is set to the negative of the maximum number of list storage locations which were used at the current level.

COMPUTER AND LANGUAGE

All subroutines were written in FORTRAN IV. No special hardware or software is required. The subroutines were written and tested on the IBM 360/91 at Princeton University, using the FORTRAN IV (H) compiler. Core requirements for CLIQUE and all associated subroutines and local storage totaled 8340 hexadecimal bytes.

SIZE LIMITATIONS

CLIQUE was written to be compatible with the clustering programs listed in Jardine and Sibson (1971). A maximum of 60 points can be clustered.

AVAILABILITY

A listing of the subroutines, along with the calling program, is available free of charge from the author.

REFERENCES

Jardine, N., & Sibson, R. *Mathematical taxonomy*. New York: Wiley, 1971.

Johnson, S. C. Hierarchical clustering schemes. Psychometrika, 1967, 32, 241-254.

(Received for publication November 16, 1973; accepted November 23, 1973.)

Statistical programs utilizing dynamic allocation of memory on the CDC 6400

PAUL L. DERBY
Colorado State University, Ft. Collins, Colorado 80521

Statistical routines written for general use often require large data storage arrays. A program using three n x n arrays, for instance, would require 30,000 words of memory for 100 variables. Usually the number of variables in any analysis is less than the maximum number of variables dimensioned in the program. This generally results in a number of unused data storage locations and a waste of computer resources. It would be advantageous to allocate storage for data arrays at execution time rather than at compile time, thereby minimizing storage requirements and turn around time for each program run.

A series of multivariate analysis programs described by Cooley and Lohnes (1971) have been rewritten incorporating a subroutine, CORE, which dynamically allocates memory in blank common during execution according to parameters specified on the first cards preceding the data deck. In multiple problem runs, CORE is capable of either reducing or increasing field length, depending upon the number of variables included in the analysis. As an example, the CORREL program in the Cooley and Lohnes package requires a field length of 43000_8 words to execute on the CDC 6400 under SCOPE 3.3 without the CORE routine. As few as 16000_8 words are needed with the CORE routine.

INPUT/OUTPUT

All input and output remains the same as that described by Cooley and Lohnes.

COMPUTER AND PROGRAMMING LANGUAGE

The Cooley and Lohnes programs are written in FORTRAN IV (modified for the CDC FORTRAN extended compiler). The CORE subroutine is written in COMPASS assembly language. The programs are for use on the CDC 6000 series computers operating under the SCOPE monitor.

RESTRICTIONS

CORE can reserve space up to and including the field length requested by the user's job card at execution time. If more space is requested by CORE than initially reserved, the job aborts with an indication of the necessary field length in the system dump.

AVAILABILITY

The programs may be obtained (free) by sending a magnetic tape to the author.

REFERENCE

Cooley, W. W., & Lohnes, P. R. *Multivariate data anlysis*. New York: Wiley, 1971.

(Received for publication November 16, 1973; accepted November 23, 1973.)