Cluster analysis on the Atlas computer

By M. J. Shepherd* and A. J. Willmott**

This paper discusses the programming implementation of Sneath's Single Link Taxometric Analysis for the Atlas 1 computer and the modification of this method which the authors have developed to overcome any chaining that occurs between otherwise well separated clusters. Two distinct algorithms are presented for Single Link Analysis.

(First received June 1967)

The purpose of Taxometric Analysis is to separate a population into distinct groups or *clusters*, each cluster being defined in terms of the qualities or *attributes* which the members of the cluster have in common. This is achieved by first computing a measure of association, a *similarity coefficient* between each and every member of the population or *operational taxonomic unit* (O.T.U.). The simplest coefficient might be defined as:

$$S = \frac{\text{number of attributes shared}}{\text{total number of attributes}}$$

Many other coefficients have been proposed and a fairly comprehensive list is presented by Sokal and Sneath (1963). For example, if all the attributes were parametric, the correlation coefficient might be employed.

It is assumed in this work that a matrix of similarity coefficients, or *similarity matrix*, can be computed. In the number of practical problems tackled using our programs, we have found that unless unnecessary rigour is enforced upon the format of the source data for analysis, it is easier to define the similarity coefficient for each set of data—usually this consists of a heterogeneous mixture of binary and multistate qualitative attributes together with several parametric attributes. Once the coefficient is defined, the matrix of coefficients can be computed and output onto magnetic tape or perhaps punched paper tape in a format acceptable to our clustering programs. This approach is preferable, we believe, to the one where the source data is required to be in binary form, for example, and the user of the clustering programs needs to convert his data in an arbitrary and perhaps artifical way into this form. It has the added advantage that the user of the programs is called upon to think closely about the measure of similarity he is proposing for the cluster analysis.

In single link clustering, an O.T.U. belongs to a group at specified level of similarity, L, provided that the similarity coefficient between that O.T.U. and at least one other O.T.U. in the group is greater than L. In k-link clustering, an O.T.U. requires to be similar to at least k members of the group. If the number of group members is less than k, the O.T.U. requires to be similar to all members of the group.

A little consideration will lead to the conclusion that the algorithm to be employed for a cluster analysis of this type, because it requires all the coefficients of the similarity matrix to be scanned at least once, must be influenced by the hardware of the computer available. A computer with a relatively small fast store requires that all but the smallest similarity matrices be stored on magnetic tape or magnetic disc.

Under these circumstances, the fewer the inspections of each row of similarity coefficients the better, and in the case of magnetic tape, the rows should be called for sequentially. However, in the case of the Atlas computer at Manchester University a one-level main store consisting of 96K 48-bit words on drums and 16K words in core exists of which a user might call for 60–70K words without disturbing the normal running of the computer's operating system. This is described by Kilburn *et al* (1962). It is possible therefore to have the whole matrix in the main store of the machine and to use an algorithm which generates random accesses to the rows of the similarity matrix. Usually it is sufficient to store the similarity coefficients to 3 significant figures only, in the range 0–1000, and in which case by packing 3 coefficients to an Atlas integer word, quite large matrices can be held in the main store. Further, by storing only the upper triangle of what is a *symmetric* matrix of similarity coefficients, further economies can be made.

If the upper triangle of the similarity matrix is held in packed form, 3 coefficients per word, then a penalty is enforced by the computing time required to unpack the coefficients and to reconstruct each whole row of the matrix. For cluster analysis involving up to 300 O.T.U.s this is tolerable. If populations of 500 are tackled, the time required for coefficient unpacking and row reconstruction becomes excessive and it is preferable to hold the matrix of unpacked coefficients on magnetic tape and to use a different algorithm. It is possible to use an unpacked square matrix stored in main store for up to 200 O.T.U.s without using excessive store.

Single linkage algorithms

1. *Random calls for the rows of the similarity matrix*

This algorithm requires two subsidiary vectors in addition to the row of similarity coefficients currently

* *Department of Mathematics, Computation Division, The University of Manchester Institute of Science and Technology, P.O. Box 88, Sackville Street, Manchester, 1.*
** *Department of Computation, The University of York, Heslington, York.*

being inspected. These might be declared in an ALGOL-like language as

integer array list 1, list 2 (1 : *n*)

where *n* is the number of O.T.U.s to be clustered. The O.T.U.s are given arbitrary reference numbers 1, 2, . . . *n*.

The *address* of a storage element in the vector list 1 is equal to the reference number of an O.T.U. and, at the end of the analysis, the content of that element is equal to the number of the cluster to which that O.T.U. belongs. In list 2, the storage element contains the reference numbers of the currently already clustered O.T.U.s. There are three pointers employed.

Pointer 1 contains the reference number of the O.T.U. around which the clustering is currently proceeding. Pointer 1 therefore also holds the current number of the row of the matrix being examined.

Pointer 2 contains the *address* in the *array* list 2 where the reference number of the next O.T.U. to be clustered is placed.

Pointer 3 contains the *address* in the *array* list 2 in which is contained the reference number of the next O.T.U. around which clustering is next to proceed.

Initially, all pointers are set to 1. The first row of the matrix is scanned and the reference numbers of the O.T.U.s with whom O.T.U.1 *is linked*, that is, with whom O.T.U.1 has sufficiently great similarity, are copied into list 2 and pointer 2 is advanced. At the same time, in the positions, corresponding to these reference numbers, in list 1 is placed the current cluster number.

When this examination of row 1 is completed, pointer 3 is advanced by 1, and row *p* of the similarity matrix is inspected, where *p* is the reference number contained in list 2 (pointer 3). Where possible, further additions are made to the cluster and the necessary entries are made in the arrays list 1 and list 2. Again pointer 3 is advanced, and a new row of the matrix is inspected. This process is exhaustive, that is once a cluster has been formed and the row associated with 'last' member of cluster, examined, the pointer 3 will point to a zero entry in list 2. At this stage pointer 2 will equal pointer 3.

Next the cluster number is advanced one, and pointer 1 is advanced until it points at an *address* in list 1 which corresponds to the reference number of an O.T.U. not yet classified. Clustering starts again and proceeds exhaustively as described above. In **Fig. 1** is set out the layout of the vectors and their pointers at a typical intermediate state of an analysis.

In the implementation of this algorithm in our programs for the Atlas computer, safeguards are incorporated to prevent single O.T.U. clusters being allowed to form.

In Fig. 1, the sequence of the reference numbers in list 2 corresponds to the apparently random sequence in which the rows of the similarity matrix are inspected. If the matrix were held on magnetic tape, this method could involve excessive tape searching and rewinding.

However, the method possesses the advantage that the process is exhaustive, and once a cluster is formed it is known that further inspections of the similarity matrix will not reveal any new members of the cluster. It is

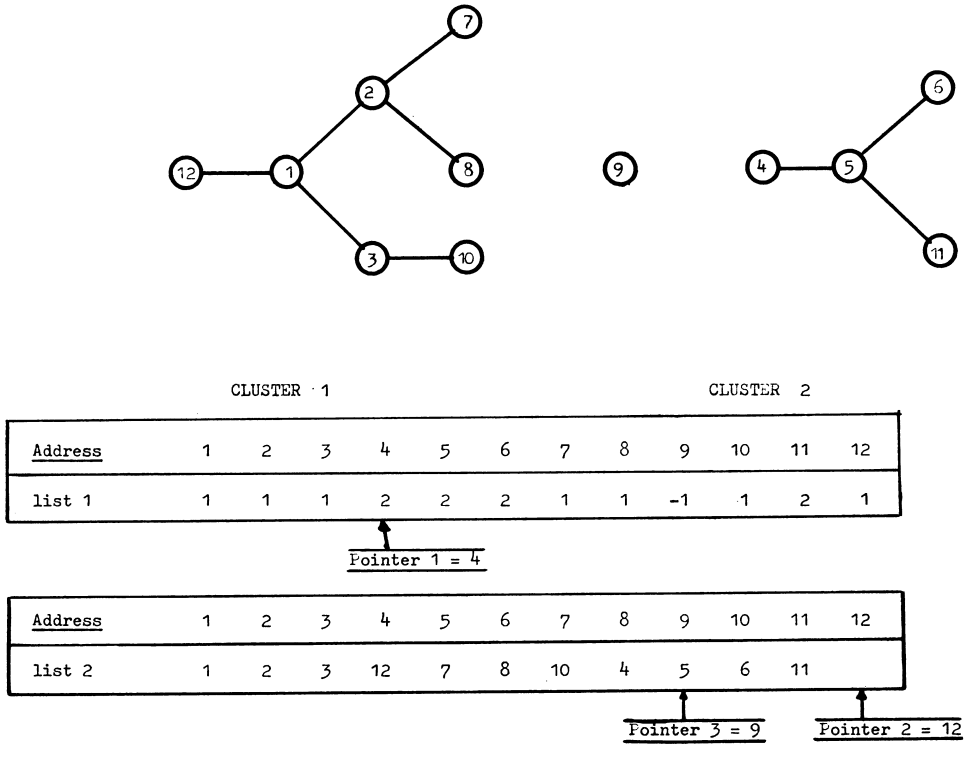

Fig. 1. Illustration of operation of vectors list 1 and list 2 and their associated pointers

also possible to adapt the method to multi-link clustering. Once a single link cluster has been formed, it is possible to re-examine all the rows of the similarity matrix corresponding to the O.T.U.s in the cluster and to reject any members who do not possess a sufficient degree of multi-linkage with the rest of the O.T.U.s. This process is iterative but again exhaustive. If an O.T.U. is rejected, it is necessary to re-examine the rows of the matrix corresponding to O.T.U.s previously retained in the reappraisal of the cluster. This method involves repeated and random accesses to the rows of the matrix and is therefore unsuitable if the matrix is held on magnetic tape.

2. *Sequential calls for rows*

In this method, each row of the matrix is called into the fast store of the computer once and the maximum amount of information is extracted. The rows are called sequentially. The method begins as in the random method by looking at the first row of the matrix and identifying the O.T.U.s which are linked to O.T.U.1 and placing them in Cluster 1.

Next the 2nd row is examined. If O.T.U.2 belongs to Cluster 1, then all O.T.U.s linked to O.T.U.2 (but not picked up as being linked to O.T.U.1) are ascribed to Cluster 1. If O.T.U.2 does not belong to Cluster 1, then a new Cluster 2 is begun. The method proceeds by this orderly inspection of the rows of the matrix. It is necessary, however, to 'back track' from time to time if it is subsequently discovered that a link exists between, for example, an O.T.U. in Cluster 2 and an O.T.U. in Cluster 5. In this case, all the O.T.U.s in Cluster 5 must be reassigned to Cluster 2. It is possible therefore for an O.T.U. to be reassigned to different clusters several times during the execution of the algorithm.

The method has been found to be very fast, and populations exceeding 500 O.T.U.s have been successfully classified using a small amount of computing time.

Chaining

The process of chaining which is inherently possible using single linkage clustering, can be described by consideration of the hierarchical structure of an imaginary data set. Consider two groups present in the data, which are separate at all levels of similarity down to S, below which they coalesce naturally to form one group. This situation is feasible for real data, and corresponds to all inter-group similarities being less than S, and all intra-group similarities being greater than S, for the two groups.

It is more usual with real data to have a less clear cut division between the inter and intra-group similarities, and this leads, where single linkage clustering is employed, to the phenomenon of chaining. A situation can occur in which the similarity coefficients are as described above for the imaginary data with the exception that one inter-group coefficient is above the level S, and in fact is equal to S'. With single linkage clustering, the two groups will coalesce at a level S', and not at level S. This could of course be overcome by using complete linkage clustering, which for this data would yield the desired hierarchy. As mentioned above, it is possible, but rare, to find this type of distribution of similarity coefficients in real data, and it is more usual to have a wide variation in both inter and intra-group coefficients. It will therefore not be possible to say that certain O.T.U.s form a completely separate group, as groups will always overlap to some extent, and it is similarly not possible to say that a group comprises certain O.T.U.s, and only those O.T.U.s, as there will be variations amongst members of a group in the contribution the O.T.U.s make to the group as a whole. In other words, some O.T.U.s are more acceptable in a group than others.

With real data, a process of complete linkage clustering will yield a group comprising only the core or nucleus of the real group, and will discard peripheral objects as outside the group, regardless of the general acceptability of that object. This can be regarded as the reverse of chaining, and can be shown diagrammatically as in **Fig. 2,** where the fusion of two groups is shown. The real fusion point should be at level S, the single linkage fusion point is at S', and the complete linkage fusion point at S''.

Obviously, the ideal classification scheme would cluster this data in such a way that the two main groups coalesced at level S, having retained separate identities up to this point. This leads to the idea of multiple linkage or k-linkage clustering, as the general case, with single and complete linkage being special cases of k-linkage clustering, single linkage corresponding to $k = 1$, and complete linkage corresponding to $k = n$, where n is the total number of O.T.U.s in the group or groups being considered.

The idea of k-linkage clustering can be extended

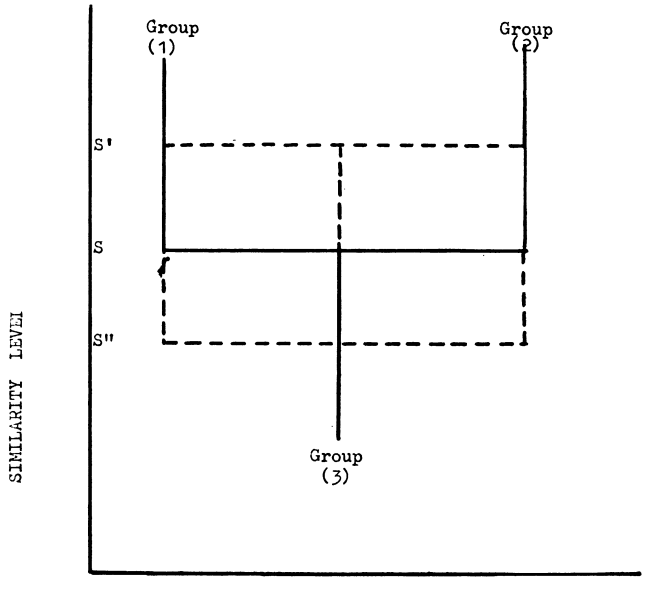

Fig. 2

directly to the initial clustering of O.T.U.s, and will result in groups where membership of a group entails similarity with at least k members of that group. Obviously this is not very convenient, as k will be varying continuously, but a ratio of k to n can be proposed termed N_G which should remain constant.

$$N_G = \frac{k}{n}$$

The formal average linkage clustering method attempts to be a k-linkage method, without actually considering links. The average linkage method of Sokal and Michener (1958) operates by computing the average similarity within a group, the average of all the similarity coefficients between members of that group, and permitting O.T.U.s to join that group if the group average similarity drop after the admission of that O.T.U. is less than a certain value. We will call the drop in group average similarity for the addition of an O.T.U., γ. The method can be implemented in one of two ways: by adding one O.T.U. at a time, and then recomputing the group average similarity (Pair Group Method), or by allowing a number of O.T.U.s to join a group at one stage, provided all the O.T.U.s satisfy the criterion (Variable Group Method).

It will be noted that with Average Linkage methods, the formation of initial groups has been ignored, and this presents the main problem of implementing the method. The final results will depend on the starting point chosen, and Average Linkage is therefore only a group or cluster expansion technique.

It is important to note that chaining is liable to be inherent in any data set, and without using a complete linkage clustering technique the effects of the chaining O.T.U.s cannot be avoided. Indeed, if the chaining is overcome completely, the results will be virtually meaningless, only very compact cluster centres being allowed to form. It is therefore desirable to permit a certain controlled amount of chaining, so that with reference to the hierarchy, a fusion will occur, either between O.T.U.s, or between groups, at the level S, and not at S' or S''.

We can therefore define a degree of chaining required for a given data set by

$$C = \frac{S - S''}{S' - S''}.$$

This represents the fraction of the maximum possible chaining that is desirable for a given fusion. It is not feasible to calculate this value for every fusion in the clustering process, even if it were possible to compute a value for S. It is feasible, though, to set a control on the fusion process which permits a degree of chaining, and to modify the extent of the chaining by trial runs with the clustering algorithm. This leads to the multiple linkage program developed by the authors.

Multiple linkage algorithm

The algorithm is not intended to produce a complete hierarchy at one pass, but is intended to provide a detailed analysis at one level of similarity only. It was intended that an approximate hierarchy could be derived using a purely single linkage analysis, and the multiple linkage algorithm could then be employed to further analyse data for each level of similarity which was of interest. The method is based very broadly on an average linkage cluster expansion method, the initial cluster centres being formed by a modified single linkage method.

Formation of cluster centres

Several methods are available for the formation of cluster centres, but none of these are really acceptable. If severe chaining is present the Centroid method can very easily choose the wrong points as cluster centres, and this is generally true of all geometrical methods. The authors considered that the most acceptable way of forming cluster centres was firstly to separate all O.T.U.s which are likely to be at or near cluster centres, by performing a single linkage classification, and then to consider only those O.T.U.s which have been separated into clusters as worthy of consideration for use as cluster centres. The single linkage method will certainly separate all O.T.U.s that may be useful as cluster centres, by virtue of the fact that it clusters on the basis of only one link being necessary between O.T.U.s within a group.

Once the data has been separated into those O.T.U.s which might possibly be at or near cluster centres, and those which can never be part of a cluster core, attention can be turned to further separating the possibles into those O.T.U.s which will be at cluster centres and those which are peripherals, with perhaps only one or two links within their clusters.

The final retrieval of the O.T.U.s forming cluster centres is performed by consideration of the average similarity each O.T.U. has with the others with which it is supposedly clustered. A criterion is set externally to govern the degree of compactness required within the cluster centres, and on the basis of this, the O.T.U.s which are computed to be within cluster centres are separated. These O.T.U.s are then reclassified using single linkage to obtain the cluster centres.

Extension of cluster centres

The process of extending the cluster centres to form complete groups is based on the pair group average similarity method. The reacceptance criterion is not simply the drop in average similarity, which does not directly take into account the degree of linkage, but a factor of the percentage linkage ($100 \times N_G$) divided by the drop in average similarity for admission of an O.T.U. into a group. This factor, the acceptance ratio can be set externally and can be modified to suit different sets of data. The extension proceeds with the addition to each cluster centre of that O.T.U. which has the highest corresponding acceptance ratio, the process repeating until all acceptable O.T.U.s have been admitted. The residual O.T.U.s are then classified by a single

linkage analysis, to detect any small clusters that were missed by the full classification. The results are then output in the form of a series of groups, each consisting of a list of O.T.U. numbers present.

Program

The program is written in version AB of Atlas Autocode, for use on Manchester University's Atlas computer, or any computer having facilities for COMPILER AB, and adequate fast store. The program will not run using version AA without modification to the main program. The main program is at present available as a 7 track punched paper tape, and utilises multiple channel input to accept data from a separate 5 or 7 track paper tape (input 1). The data takes the form of parameters which control the course of the calculation, possibly followed by the matrix of similarities. Alternatively, if the matrix is large, it can be input from private magnetic tape.

Input

The similarity matrix can be input in two forms:

1. From paper tape, as a square matrix. Each coefficient is in the range 0–1000 and is punched as an integer. At least two spaces or one newline separates each number.
2. From private magnetic tape, stored on the tape as an upper triangular matrix with the leading diagonal present. The similarity coefficients are packed three to an Atlas integer word. This method must be used where the number of O.T.U.s exceeds 200. The program retains the matrix as a vector of packed coefficients unpacking a row of coefficients as and when this is needed in the calculation. The storage needed for the similarity matrix is therefore $((m(m+1))/6)$ words where m is the dimension of the matrix. This must be available in main store.

Use of the program

In order to use the program the following information must be encoded on 5 or 7 track punched paper tape.

1. Job Description (see I.C.T. Document CS 460, 'Preparing a Complete Program for Atlas 1').
2. A list of Control Parameters.
3. The similarity matrix, if this is to be input from paper tape.
4. ***Z terminator as required by the Atlas Supervisor.

The control parameters

The first number on the data tape must always be the number of O.T.U.s, i.e. the dimension of the similarity matrix. This is followed by the three control parameters. These parameters provide information to the main program on the form of the similarity matrix, the input medium, and the form of classification required, either single linkage only, or full classification.

Parameter one

This can take two values, 1 or 2. If it is set to one, a purely single linkage classification is performed at a number of levels of similarity. If it is set to 2, a full classification is performed at one level of similarity only.

Parameter two

This again can take the value 1 or 2. If it is set to 1 it indicates to the program that the data is in the form of a packed upper triangular matrix with the leading diagonal present. A value of 2 indicates that the matrix is square, and not packed.

Parameter three

This parameter indicates the input medium, 1 for paper tape, and 2 for magnetic tape.

The control parameters are followed by several other parameters, but the meaning and values of these will depend on the control parameters.

Similarity levels

For Single Linkage classification only, the range of levels of similarity is determined by three numbers, the initial level, the final level, and a step length between levels. An example of this might be classification at levels 800, 700, and 600. The initial level is 800, the final level is 600, and the step is −100. For full classification one number only is used and this is the level at which the classification is to be performed.

Group reduction criterion

In stage one, the groups are reduced in size to form nuclei. The severity of this reduction process is determined by the group reduction criterion. An initial value of 50 is suggested as being suitable for most data, but the best value will have to be determined by trial and error. An increase in the value will cause more O.T.U.s to be removed from each group.

O.T.U. re-admission criterion

This is used in the second stage to determine how easy re-admission into a group should be. An initial value of 17 has been found to be suitable for medical or biological data, but variations may be advisable for other types of data. Increasing this value will result in a stricter criterion for re-admission. A value of 17 corresponds to a drop in average similarity in the group of 30, with the new O.T.U. linked to at least half the existing members of that group.

Output

The output from the program gives a comprehensive description of the results, and is adequately captioned to enable the user to see exactly what the results mean. For each level of similarity, a list of groups is given. The first of these, captioned *group minus one*, is the residual O.T.U.s, i.e. those which have not been accepted into any group. The order of the actual groups has no significance, usually group one contains the lowest

numbered grouped O.T.U., group two contains the next lowest, etc.

For full classification, the results are only given at the level specified, but this is given on the print-out as a caption. The complete process is given, including the results of the initial single linkage classification, the O.T.U.s forming the nuclei, with a classification of these to show the actual nuclei, a list of additions made to each of the nuclei, in the order in which these additions were made, and a final full classification, captioned as for the single linkage case.

The amount of output required will depend on the data, and the ease with which it classifies, but a request for 500 lines will be adequate for most problems.

Computing time

It is not possible to estimate the amount of computing time required, as this will depend to a very great extent on the number of iterations needed in the second stage. For small data matrices (< 100) a request for 2 minutes on Atlas should suffice. For larger matrices, for which a full classification is required, the amount of time needed will have to be determined specifically for that data.

Conclusions

The programs described in this paper have been used to classify biological and medical data, in studies of the 'twilight zones' of a large city by town and country planners, and in population movement studies by social geographers. Each of these pieces of work warrants separate description. However, our general observations have been that in many cases, single link analysis is more than adequate and no gains are to be made by continuing the analysis by using the de-chaining algorithms. Where a large number (for example 500) of attributes are used to describe each O.T.U., the risk of chaining increases, and in one of our medical applications the de-chaining process has proved invaluable.

References

KILBURN, T., EDWARDS, D. G. B., LANIGAN, M. J., and SUMNER, F. H. (1962). *I.R.E. Transactions on Electronic Computers*, Vol. 11, No. 2, p. 223.

SOKAL, R. R., and MICHENER, C. D. (1958). A Statistical Method of Evaluating Systematic Relationships, *University Kansas Sci. Bull.*, Vol. 38, pp. 1409–1438.

SOKAL, R. R., and SNEATH, P. H. A. (1963). *Principles of Numerical Taxonomy*, *Freeman.*

Book Review

Cybernetic Modelling, by J. KLIR and M. VALACH (Tr. P. DOLAN), 1967; 437 pages. (London: *Iliffe Books Ltd.*, 63s.)

This long, ambitious, and rather strange book was first published in Czechoslovakia in 1965. Its characteristics are the authors' obvious lack of advanced computing facilities, their ignorance of, or lack of interest in, current Western research and their adherence to a completely materialist viewpoint. In consequence the book has a somewhat archaic flavour, with much time spent putting forward philosophical and terminological arguments and introducing research ideas now of largely historical interest.

The book begins with the sentence—'The basis of our world is *matter*, which is in continuous motion in space and time in the widest sense of the word'. The vague generality of this statement is fairly typical. The authors then distinguish between *inanimate* and *animate* matter, and remain fascinated by this distinction.

The first one hundred and twenty pages are devoted primarily to introducing and defining concepts and to discussing the elements of what may ultimately become a useful cybernetic theory. By 'cybernetics' the authors mean, roughly, the study of the structure and behaviour of collections of elements which interact with one another, and with their environment. They formulate a definition of what they mean by one system being a model of another in terms of identity of structure and behaviour at a given level of observation.

The next seventy pages of the book are devoted to practical methods of modelling, primarily using analogue and digital computers and using logical networks. The discussion of computers concentrates on the organisation of the machines themselves and the basic ideas of programming. Programming languages more complex than machine code are given only a single paragraph, and Monte Carlo methods about a page. A simple form of time-sharing is briefly described, as is the concept of a Turing machine.

This is followed by a long discussion in general terms of the meaning of such words and phrases as 'decision making', 'goal seeking', 'communication', and 'consciousness', when applied to machines, and then two specific topics are treated at some length. The first is machine understanding of natural language. The authors propose graphical representations of sentence and text structure and indicate how a machine might answer questions and form abstracts by operations over these graphs. The second topic considered is the detection of moving objects in a visual field. The authors propose a simple method involving the matching of successive views. I feel that the authors' treatment of each of these topics contains little of interest for the informed research worker, and yet is too special to serve as an introductory text. No actual experimentation is mentioned.

The final chapter of the book is an unimpressive excursion into what one might call 'Science Fiction philosophy' concentrating on the difference between animate and inanimate systems, and on the future of man and robots.

In sum, I find this book more suitable for the collector of curios and for the graduate student with time and interest to spare than for the research scientist, or the person requiring an introduction to cybernetics. However, it is often thought-provoking, and has a large bibliography with many useful references to work in the U.S.S.R. and elsewhere.

JAMES DORAN (Edinburgh)