CLUSTERING

by

R. S. Michalski
R. Stepp

In AI Encyclopedia, John Wiley & Sons, New York, pp. 185-194. January, 1986.

ENCYCLOPEDIA OF ARTIFICIAL INTELLIGENCE

VOLUME 1

Stuart C. Shapiro, *Editor-in-Chief*
David Eckroth, *Managing editor*
George A. Vallasi, *Chernow Editorial Services, Developmental Editor*

Wiley-Interscience Publication
John Wiley & Sons
New York / Chichester / Brisbane / Toronto / Singapore

BIBLIOGRAPHY

1. J. McCarthy, "Circumscription—A form of non-monotonic reasoning," *Artif. Intell.* **13**, 27–39 (1980).
2. J. McCarthy, "Applications of circumscription to formalizing common sense knowledge," Workshop on Nonmonotonic Reasoning, New Paltz, NY, sponsored by AAAI, October 17–19, 1984.
3. D. Etherington, R. Mercer, and R. Reiter, "On the adequacy of predicate circumscription for closed-world reasoning," *J. Comput. Intell.* **1**, 11–15 (1985).
4. D. Etherington, personal communication, Comp. Sci. Dept., Univ. of British Columbia, Canada, 1984.
5. M. Davis, "The mathematics of non-monotonic reasoning," *Artif. Intell.* **13**, 73–80 (1980).
6. J. Minker and D. Perlis, "Protected circumscription," Workshop on Nonmonotonic Reasoning, New Paltz, NY, Oct., 1984.
7. D. Kueker, "Another failure of completeness for circumscription," Week on Logic and Artificial Intelligence, University of Maryland, October 22–26, 1984.
8. J. Doyle, "Circumscription and implicit definability," Workshop on Nonmonotonic Reasoning, New Paltz, NY, Oct., 1984.
9. D. Perlis and J. Minker, "Completeness results for circumscription," *Artif. Intell.* **28**, 29–42 (1986).
10. V. Lifschitz, "Some results on circumscription," Workshop on Nonmonotonic Reasoning, New Paltz, NY, Oct., 1984.
11. B. Grosof, "Default reasoning on circumscription," Workshop on Nonmonotonic Reasoning, New Paltz, NY, Oct., 1984.
12. R. Reiter, "A logic for default reasoning," *Artif. Intell.*, 13, 81–132 (1980).
13. R. Reiter, "Circumscription implies predicate completion (sometimes)." Proc. Nat'l. Conf. on Art. Intell., Pittsburgh, PA, 1982.
14. K. Clark, "Negation as failure," in H. Gallaire and J. Minker (eds.), *Logic and Data Bases*, Plenum, New York, 1978.
15. M. A. Papalaskaris and A. Bundy, "Topics for circumscription," Workshop on Nonmonotonic Reasoning, New Paltz, NY, Oct., 1984.

D. Perlis
University of Maryland

CLUSTERING

Clustering is usually viewed as a process of grouping physical or abstract objects into classes of similar objects. According to this view, in order to cluster objects, one needs to define a measure of similarity between the objects and then apply it to determine classes. Classes are defined as collections of objects whose intraclass similarity is high and interclass similarity is low. Because the notion of similarity between objects is fundamental to this view, clustering methods based on it can be called similarity-based methods. Many such methods have been developed in numerical taxonomy, a field developed by social and natural scientists, and in cluster analysis, a subfield of pattern recognition (qv). Various similarity measures and clustering algorithms utilizing them are presented below (see also Concept learning; Region growing.)

Another view recently developed in AI postulates that objects should be grouped together not just because they are similar according to a given measure, but because as a group they represent a certain conceptual class. This view, called *conceptual clustering,* states that clustering depends on the goals of classification and the concepts available to the clustering system for characterizing collections of entities. For example, if the goal is to partition a configuration of points into simple visual groupings, one may partition them into those that form a T-shape, an L-shape, and so on, even though the density distributions and distances between the points may suggest different groupings. A procedure that uses only similarities (or distances) between the points and is unaware of these simple shape types clearly can only accidently create clusterings corresponding to these concepts. To create such clustering, these descriptive concepts must be known to the system. Another example of conceptual clustering is the grouping of visible stars into named constellations. Conceptual clustering is contrasted with the classical view in the next section and described in more detail in the section Conceptual Clustering.

Clustering is the basis for building hierarchical classification schemes. For example, by first partitioning the original set of entities and then repeatedly applying a clustering algorithm to the classes generated at the previous step, one can obtain a hierarchical classification of the entities (a divisive strategy). A classification schema is obtained by determining the general characteristics of the classes generated.

Building classification schemes and using them to classify objects is a widely practiced intellectual process in science as well as in ordinary life. Understanding this process, and the mechanisms of clustering underlying it is therefore an important domain of research in AI and other areas. This process can be viewed as a cousin of the "divide and conquer" strategy widely used in problem solving (qv). It is also related to the task of decomposing any large-scale engineering system into smaller subsystems in order to simplify its design and implementation.

The Classical View versus the Conceptual Clustering View

In the classical approach to clustering mentioned above, clusters are determined solely on the basis of a predefined measure of similarity. To define such a measure, a data analyst determines attributes that are perceived as relevant for characterizing objects under consideration. Vectors of values of these attributes for individual objects serve as descriptions of these objects. Considering attributes as dimensions of a multidimensional description space, each object description corresponds to a point in the space. The similarity between objects can thus be measured as a reciprocal function of the distance between the points in the description space.

Let V_A and V_B denote the attribute vectors representing objects A and B, respectively. The distance of object A to object B is defined as a numerical function of the attribute vectors of A and B and is written as $d(V_A, V_B)$. For example, assuming that vector descriptions of objects A and B are $V_A = (x_1(A), x_2(\mathrm{A}), \ldots, x_n(\mathrm{A}))$ and $V_B = (x_1(B), x_2(B), \ldots, x_n(B))$, respectively, where $x_1, x_2, \ldots, x_n$ are selected object attributes, a simple measure of distance is:

$$d(V_A, V_B) = \sum_{i=1}^{n} |x_i(A) - x_i(B)|$$

Because distance is a function of only the attributes of two compared objects, the similarity-based clustering can be performed relatively easily and without a need for knowledge about its purpose. The similarity-based approach has produced a number of efficient clustering algorithms, which have been useful in many classification-building applications.

The classical approach suffers, however, from some significant limitations. The results of clustering are clusters plus information about numerical similarities between objects and object classes. No descriptions or explanations of the generated clusters are supplied. The problem of cluster interpretation is simply left to the data analyst. Data analysts, however, are typically interested not only in clusters but also in their explanation or characterization.

To overcome this, one may postscript the similarity-based clustering process with an intelligent interpretation that tries to learn the conceptual significance of each cluster through the use of AI techniques. Such a process, however, is not easy. In fact, it may be even more difficult than that of generating clusters themselves. This is because it requires inducing category descriptions from examples, which is a complex inferential task. Even if one ignores this difficulty, this process may not produce desired results. Clusters generated solely on the basis of some predefined numerical measure of similarity may in principle lack simple conceptual explanations.

One reason for this is that a similarity measure typically considers all attributes with equal importance and thus makes no distinction between those that are more relevant and those that are less relevant or irrelevant. Consequently, if there is coincidental agreement between the values of a sufficient number of irrelevant attributes, objects that are different in a conceptual sense may be classified as similar. Even if one assigns some a priori "weights" to attributes this will not change the situation very much, because the classical approach has no mechanisms for selecting and evaluating attributes *in the process* of generating clusters. Neither is there any mechanism for automatically constructing new attributes that may be more adequate for clustering than those initially provided.

Another reason for the difficulty of the postclustering interpretation is that in order to generate clusters that correspond to simple concepts, one has to take into consideration concepts useful for characterizing clusters as a whole in the process of clustering and not after clustering.

The following example illustrates this point. Consider the problem of clustering the points in Figure 1. Typically, a person looking at this figure would say that it is a letter S inte
secting with a letter M. One should observe that points A an
B, which are closer to each other than to any other points, a
classified into conceptually different clusters. The reaso
seems to be that people are equipped with concepts such a
letter shapes, straight lines, and so on to help them recogniz
certain concepts in the figure. Thus, clustering in this case i
not based on local closeness of points but on global concep
characterizing collections of points together. A conceptua
clustering program would solve this problem by matching th
descriptions of the letter shapes (contained in its memory a
background knowledge) against the given collection of point
The best match would be obtained for shapes "S" and "M."

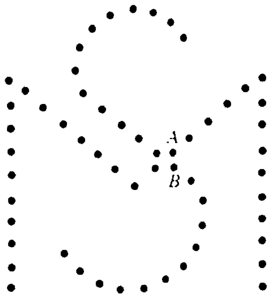


Figure 1. How would you cluster these points?

One may add that, in general, classical techniques do no
seem to be much concerned with the ways humans cluste
objects. They do not take into consideration any Gestalt con
cepts or linguistic constructs people use in describing objec
collections. Observations of how people cluster objects sugges
that they search for one or more attributes (out of many poten
tial attributes) that are most relevant to the goal of clusterin
and on that basis cluster the objects. Objects are put to th
same cluster if they score similarly on these attributes. A de
scription of the objects in the same cluster can therefore b
expressed as a single statement or a conjunction of statement
each specifying one common property (attribute value) of th
objects in the cluster. The above remark does not mean, how
ever, that individual statements could not include a disjunc
tion of values of the same attribute (the so-called *internal dis
junction*). For example, a cluster may be characterized as "
set of large boxes, made of cardboard, and colored either *blu
or yellow*." Different clusters are expected to have description
with different values of the relevant attributes.

Conceptual clustering has been introduced as a way to over
come the above-mentioned limitations of classical methods. It
basic premise is that objects should be arranged in classes tha
represent simple concepts and are useful from the viewpoint o
the goal of clustering. Thus, objects in the same cluster do no
necessarily have to be similar in some mathematically define
sense but must as a group represent the same concept. In orde
to cluster objects into conceptual categories, the notion of simi
larity must be replaced by a more general notion of conceptua
cohesiveness (1) (see also Learning, machine).

The conceptual cohesiveness (CC) between two objects
and B depends on the attributes of these objects, the attribute
of nearby objects, and the set of concepts available for describ
ing object configurations. Thus, it is a function CC(V_A, V_B, E
C), where V_A and V_B are vectors of attribute values for A an
B, respectively, E denotes objects in the environment of A an
B, and C is the set of available concepts. Thus, the conceptua
cohesiveness is a four-argument function in contrast to a two
argument distance or similarity function.

In conceptual clustering there is a constant duality betwee
category descriptions and cluster membership. Specificall
the result of conceptual clustering is not only a set of cluster
(a classification of the initially given objects) but also a set o
concepts characterizing the obtained clusters (a classificatio
scheme).

One may say that from the viewpoint of AI, the similarit
based approach represents the so-called weak method, that i
a general method that uses little problem domain knowledg
Such a method can be called domain-general knowledge-poo

In contrast, the conceptual clustering approach that is dependent on the background concepts and clustering goals can be called domain-generic knowledge-modular. It requires an interchangeable module of knowledge defined for the problem at hand. A *goal-dependency network* (GDN) (27) may be used to indicate which attributes are relevant to which goals of classification. Various algorithms for classical methods and conceptual clustering methods are presented below.

A Classification of Clustering Problems

From the viewpoint of applications, it is useful to classify clustering problems on the basis of the dimensionality of objects to be clustered. Three classes of problems can be distinguished:

1. *One-dimensional clustering (quantization of variables).* For continuous variables or discrete variables with ranges of values that are significantly larger than necessary for a given problem, one wants to reduce the number of distinct values of the variables by identifying equivalence classes of values. Clusters of values of individual variables are then treated as single units. For example, in image processing the scanners usually distinguish between a large number of gray levels, but only a few levels may be needed for solving a given problem (see Image understanding). Rosenfeld (2) has shown that clustering methods can be used for making such a reduction. Nubuyaki (3) proposed a clustering algorithm for this purpose in which the clusters have minimal sums of squares of intracluster distances. Clustering techniques have also been used to analyze LANDSAT images (4).
2. *Two-dimensional clustering (segmentation).* This type of clustering occurs most often in image processing, where one searches for segments of an image in which all picture elements share some common properties. For example, they may have a similar gray level or similar texture. Coleman (5) defined region segmentation as a problem of clustering (which he calls nonsupervised learning) and used the k-means algorithm of MacQueen (6). Haralick and Shapiro (7) have used clustering to analyze object shapes.
3. *Multidimensional clustering.* In multidimensional clustering objects are partitioned into clusters in a description space spanned by many attributes characterizing the objects. As mentioned earlier, the basis for clustering is typically a similarity measure. Traditional clustering techniques may assume different geometric distributions of the points in the space by the use of different normalization, transformation, and statistical treatments of the attributes. The next section gives more details on the similarity-based methods. In conceptual clustering the concept of description space is also useful; however, here the space is not fixed but may change as new attributes are generated by background knowledge heuristics. [illegible] addition, the method is equipped with a set of concepts [illegible] can be used to characterize object configurations.

Classical Methods of Clustering

The thrust of research in cluster analysis and numerical taxonomy has been toward determining various object similarity or proximity measures and developing clustering techniques utilizing them. A large number of such measures and corresponding clustering methods have been developed to date. Comprehensive surveys can be found in Sokal and Sneath (8), Cormark (9), Anderberg (10), Gower (11), and Diday and Simon (12). A summary of various distance measures is described in Ref. 13.

Clustering techniques can themselves be clustered in many interesting ways. One classification partitions the techniques on the basis of the type of control used in building the clusters. The categories of clustering techniques according to this classification are agglomerative, divisive, and direct.

Agglomerative Techniques. Agglomerative techniques are often used in numerical taxonomy. These techniques form clusters by progressive fusion, that is, by recursively joining separate entities and small groups together to form larger and larger groupings. Eventually a single universal group is formed and the process halts, leaving a record of the merges that took place. The history of merges is often displayed in the form of a dendrogram (see Fig. 2*c*) that shows, by the position of the horizontal location of the merge, the between-group similarities. As the groups encompass more and more entities, the between-group similarity scores decrease.

By adopting a threshold of minimum similarity, the agglomeration process can be halted before all entities are merged into a single group. Conversely, the complete dendrogram may be "cut" apart across some similarity boundary. This yields a number of clusters, each containing those entities that were merged at a similarity score above the given threshold.

During the agglomerative clustering process it is necessary to calculate the similarities between groups of entities. There are three standard ways to compute between-group similarities (measured as the reciprocal of distances). Suppose two groups are identified as X and Y. The *single-linkage* methods calculate between-group distance between one entity in group X and another entity in group Y. The *complete-linkage* methods use the maximum distance between one entity in group X and another entity in group Y. The *average-linkage* methods use the average of the distances between all possible pairs of entities with one taken from group X and the other from group Y.

Divisive Techniques. Divisive techniques form a classification by progressive subdivision, that is, by repeatedly breaking the initial set into smaller and smaller clusters until only single entities exist in each cluster. The result is a hierarchy of clusters. The divisive technique of Edwards and Cavalli-Sforza (14) examines all $2^N - 1$ partitions of N objects and selects the one that gives the minimum intracluster sum of the squared interobject distances. The computational cost of the method limits its use to cases involving the clustering of only a few objects.

Direct Techniques. The direct techniques neither merge entities into clusters nor break large clusters into smaller ones. A direct technique is given the number (usually denoted k) of clusters to form and proceeds to find a partitioning of the enti-

1. MP (Microprocessor)
Type: structured
Domain: 13 values
8080a
8502
Z80
1802
6502C
6502A
68000
6800
6805
6809
8048
Z8000
HP (Hewlett-Packard
Co. proprietary)

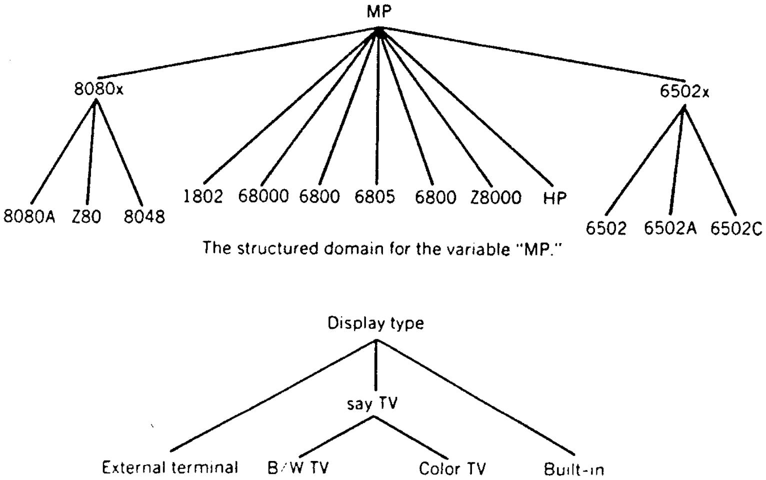
2. RAM memory size
Type: linear
Domain: 4 values
16,000 bytes
32,000 bytes
48,000 bytes
64,000 bytes

3. ROM memory size
Type: linear
Domain: 7 values
1000 bytes
4000 bytes
8000 bytes
10,000 bytes
11,000–16,000 bytes
26,000 bytes
80,000 bytes

4. Display type
Type: structured
Domain: 4 values
Terminal
B/W-TV
Color-TV
Built-in

5. Keys on keyboard
Type: linear
Domain: 5 values
52 keys
53–56
57–63
64–73
92

(a)



The structured domain for the variable "MP."

The structured domain for the variable "display type."

(b)

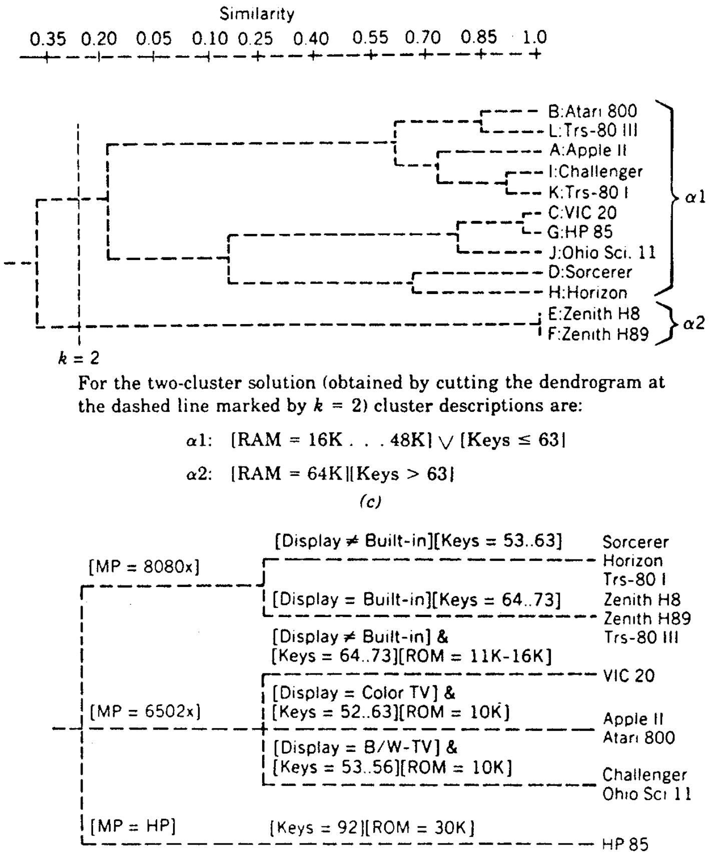
Figure 2. *(a)* Variable used to describe microcomputers. *(b)* The structure of domains of variables "MP" and "Display type." *(c)* A dendrogram generated by NUMTAX with descriptions generated by Aq. *(d)* A conceptual clustering of microcomputers.

ties into *k* clusters that optimizes some measure of the goodness of the clusters. Two early direct clustering techniques are *k*-means developed by MacQueen (6), and the center adjustment method developed by Meisel (15). A generalization of the *k* means and center adjustment techniques called the dynamic clustering method has been developed by Diday (16).

Another classification of clustering methods separates the monothetic techniques from the polythetic ones. A monothetic clustering algorithm divides the set of objects into clusters that differ in the value of one attribute. For example, such a technique might form one cluster in which attribute X_i has the value 1 and another cluster in which attribute X_i has the value 0. A polythetic clustering technique forms clusters in which the values of several attributes differ for different classes.

Traditional clustering relies on measures of similarity and the requisite need to "fold" the attribute values together to measure object-to-object similarities. When this occurs in a multidimensional space, the question of attribute weighting comes up, and there is much controversy over what weighting scheme is best for various purposes.

Weights on attributes have to be given a priori by the researcher. Problems with such an approach are that it is usually difficult to define such weights, and that some attributes may be dependent on other attributes. For example, attributes *B* and *C* may be important only if attribute *A* has the value 1. A similarity metric uses some static weights for attributes *A*, *B*, and *C*. The attributes *B* and *C* are weighted too high when attribute *A* takes the value 0 (since they should receive zero weight in that case), and they may be weighted too low when attribute *A* takes the value 1.



For the two-cluster solution (obtained by cutting the dendrogram at the dashed line marked by $k = 2$) cluster descriptions are:

α1: [RAM = 16K . . . 48K] ∨ [Keys ≤ 63]

α2: [RAM = 64K][Keys > 63]

(c)

A description of the class α1: [MP = 8080x] & [Display ≠ Built-in] & [Keys = 53..63]

(d)

Figure 2. *(Continued)*

Conceptual Clustering

As described above, conceptual clustering arranges objects into clusters corresponding to certain conceptual classes, for example, classes characterized by conjunctive concepts (i.e., concepts defined by a simple conjunction of properties). The basic theory and an algorithm for conceptual clustering have been developed by Michalski (17). Implementation and experimentation with the algorithm has been performed by Michalski and Stepp (1,18) and Stepp (19) and has produced the programs CLUSTER 2 and CLUSTER S. Other programs that work differently but provide conceptual clustering features include DISCON (20), RUMMAGE (21), and GLAUBER (22).

From the viewpoint of AI, clustering is a form of learning from observation (or learning without a teacher). It is a process that generates classes (conceptually defined categories) in order to partition a given set of observations. It differs from concept learning (qv) in that the latter creates descriptions of teacher-provided classes by generalizing from the examples of the classes.

Below, one method for conceptual clustering is briefly outlined. The method is based on the idea that conceptual clustering can be conducted by a series of conceptual discriminations similar to those used in learning concepts from examples. The method uses the extended predicate calculus proposed by Michalski (17). Such a language is used to describe objects, classes of objects, and general and problem-specific background knowledge. The method employs a general-purpose criterion for measuring the quality of generated candidate classifications. Finding classifications that score high on the quality criterion is the most general goal of the method. Additional problem-specific goals may be supplied by the user or inferred by the system from a general goal dependency network. Goal dependency is important to reduce the space of hypothetical classifications the method investigates.

Creating a classification is a difficult problem because there are usually many potential solutions with no clearly correct or incorrect answers. The decision about which classification to choose can be based on some perceived set of goals as described by Medin, Wattenmaker, and Michalski (23), a goal-oriented, statistic-based utility function as described by Rendell (24), or some other measure of the quality of the classification.

One way to measure classification quality is to define various elementary, easy-to-measure criteria specifying desirable properties of a classification, and to assemble them into one general criterion. Each elementary criterion measures a certain aspect of the generated classifications. Examples of elementary criteria are the relevance of descriptors used in the class descriptions to the general goal, the fit between the classification and the objects, the simplicity of the class descriptions, the number of attributes that singly discriminate among all classes, and the number of attributes necessary to classify the objects into the proposed classes.

Building a meaningful classification relies on finding good classifying attributes. The method presented below uses background knowledge in the search for such attributes. Background knowledge rules enable the system to perform a chain of inferences to derive values for new descriptors for inclusion in object descriptions. The new descriptors are tested by applying the classification quality criterion to the groupings formed by them.

Concept Formation by Repeated Discrimination. This section explains how a problem of concept formation (here, building a classification) can be solved via a sequence of controlled steps of concept acquisition (learning concepts from examples). Given a set of unclassified objects, *k* seed objects are selected randomly and treated as representatives of *k* hypothetical classes. The algorithm then generates descriptions of each seed that are maximally general, form a good match with a subset of the objects given, and do not cover any other seed. These descriptions are then used to determine the most representative object in each newly formed class (where the newly formed class is defined as the set of objects satisfying the generated class description). The *k* representative objects are then used as new seeds for the next iteration. The process stops either when consecutive iterations converge to some stable solution or when a specific number of iterations pass without improving the classification (from the viewpoint of the quality criterion).

This approach requires that the number of classes is specified in advance. Since the best number of classes to form is usually unknown, two techniques are used: varying the number of classes and composing the classes hierarchically.

For most purposes, it is desired that the classification formed be simple and easy to understand. With this in mind, the number of classes that stem from any node of the classification hierarchy can be assumed to be in some modest range such as from 2 to 7. With this small range, it is computationally feasible to repeat the whole clustering process for every number in the range. The solution that optimizes the score on the classification quality criterion (with appropriate adjustment for the effect of the number of classes on the score) indicates the best number of classes to form at this level of the hierarchy.

The above method of repeated discrimination for performing clustering has been implemented in the program CLUSTER/2 for a subset of extended predicate calculus (see Logic, predicate) involving only attributes (zero-argument functions). Besides its relative computational simplicity, this approach has other advantages stemming from use of quantifier-free descriptions (for both objects and classes). It should be noted that classifications normally have the property that they can unambiguously classify any object into its corresponding class. To have this property, the class descriptions must be mutually disjoint.

For conjunctive descriptions involving relations on attribute–value pairs, the disjointness property is easy to test and easy to maintain. For the more complex problems that require object representations involving quantified variables, predicates on these variables, and function–value relationships over quantified variables, the test for mutual disjointness of descriptions is much more complex. To cope with this difficulty, the problem of clustering of structured objects is decomposed into two steps. The first step finds an optimized characteristic description of the entire collection of objects and then uses it to generate a quantifier-free description of each object. The second step processes the quantifier-free object descriptions with the CLUSTER/2 algorithm to form optimized classifications. These two processes are combined in the program CLUSTER/S.

Example 1: Microcomputers. The problem is to develop a meaningful classification of popular microcomputers. Each microcomputer is described in terms of the variables shown in Figure 2*a*. Variables "MP" and "Display type" are structured, i.e., their value set forms a hierarchy (Fig. 2*b*). Two programs were applied to solve this problem: NUMTAX, which implements several techniques of numerical taxonomy, and CLUSTER/2, which implements conjunctive conceptual clustering. A representative dendrogram produced by NUMTAX is shown in Figure 2*c*. The dashed lines indicate where the dendrogram is cut apart to form two clusters ($k = 2$). Accompanying the dendrogram is a logical description of the clusters. These descriptions were produced by an inductive learning program that accepts as input a collection of groups (clusters) of objects and generates the simplest discriminant description of each group. For example, the first cluster is described as

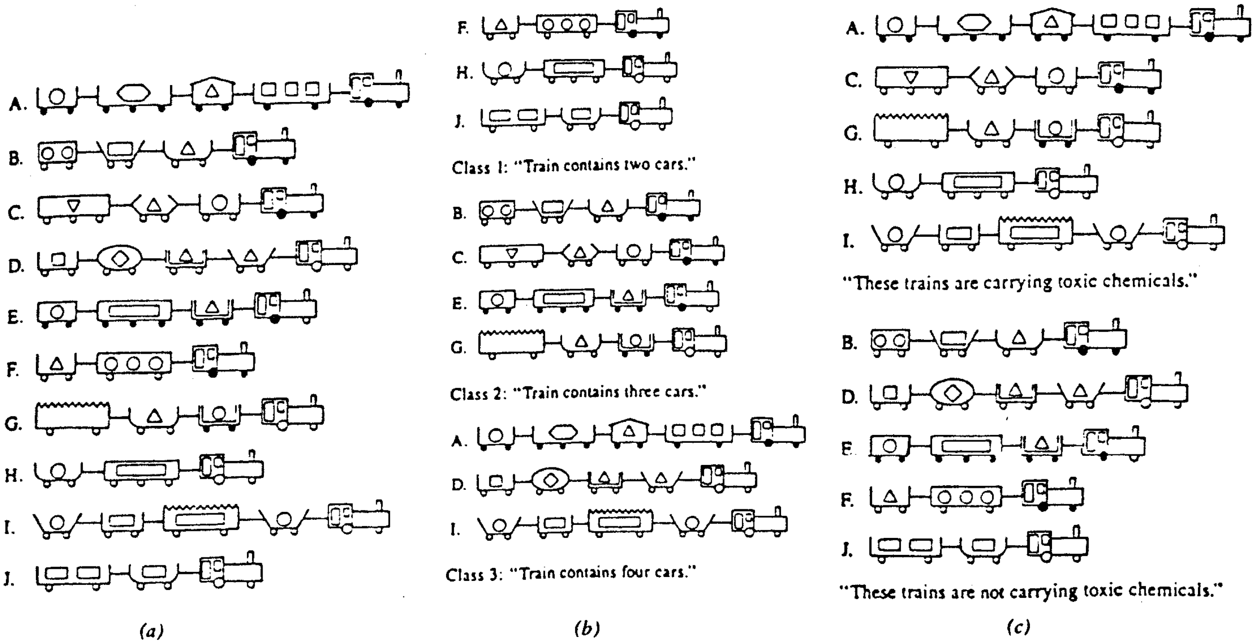
$$[\text{RAM} = 16\text{K} \ldots 48\text{K}] \vee [\text{Keys} \leq 63]$$

This description suggests that the cluster is composed of two kinds of computers, one that has [RAM = 16K . . . 64K] and the other that has [Keys ≤63]. The presence of disjunction raises the question of why these computers are in the same cluster.

The program CLUSTER/2 was given the same data and was told to use a classification quality criterion that maximizes the fit between the clustering and the objects in the cluster and then maximizes the simplicity of category descriptions. The clustering obtained is shown in Figure 2*d*. The first-level clustering is done on the basis of type of microprocessor.

Example 2: Trains. Consider a problem of classifying structured objects, for example, the problem of finding a classification of trains shown in Fig. 3*a*. The trains are structured objects, each consisting of a sequence of cars of different shapes and sizes. The individual cars carry a variable number of items of different shapes.

Human classifications of the trains shown in Figure 3*a* have been investigated by Medin, Wattenmaker, and Michalski (23). The 10 trains were placed on separate index cards so they could be arranged into groups by the subjects in the experiment. The experiment was completed by 31 subjects who formed a total of 93 classifications of the trains. The most popular classification (17 repetitions) involved the number of cars in the trains. The three classes formed were "trains con-



Class 1: "Train contains two cars."

Class 2: "Train contains three cars."

Class 3: "Train contains four cars."

"These trains are carrying toxic chemicals."

"These trains are not carrying toxic chemicals."

Figure 3. (*a*) Trains to be classified. (*b*) The most frequent human classification of trains. (*c*) Conceptual clustering of trains carrying toxic chemicals.

taining two cars," "trains containing three cars," and "trains containing four cars." This classification is shown in Figure 3*b*.

This problem is an example of a class of problems for which the implicit classification goal is to generate classes that are conceptually simple and based on easy-to-determine visual attributes. When people are asked to build such classifications, they typically form classes with disjoint descriptions, as in the above-mentioned study by Medin. For this reason methods that produce disjoint descriptions are of prime interest.

The problem of classifying trains represents a general category of classification problems in which one wants to organize and classify observations that require structural descriptions, for example, classifying physical or chemical structures, analyzing genetic sequences, building taxonomies of plants or animals, characterizing visual scenes, or splitting a sequence of temporal events into episodes with simple meanings.

One problem of concern here is to develop a general method that when applied to the collection of structured objects, such as trains, could potentially generate the conjunctive concepts occurring in human classifications or invent new concepts having similar appeal.

An extension of the trains problem illustrates the use of a goal dependency network and problem-specific background knowledge. Suppose that the knowledge base includes an inference rule that can identify trains carrying toxic chemicals and that the general goal "survive" has a subordinate goal "monitor dangerous shipments." This background knowledge can be used to help build a classification.

In the illustrations of the trains a toxic chemical container is identified as a single sphere (circle) riding in an open-top car. A background-knowledge rule supplied to the program is

[contains(train,car)][car-shape(car) = opentop]
[cargo-shape(car) = circle][items-carried(car) = 1]
⇔ [has_toxic_chemicals(train)]

In the above rule, equivalence is used to indicate that the negation of the condition part is sufficient to assert the negative of the consequence part. After this rule is applied, all trains will have descriptions containing either the toxic chemical predicate or its negation. The characteristic description generated by the program will now contain the additional predicate "has_toxic_chemicals(train)" (or its negation). By recognizing that this predicate is important to the goal "survival" through use of a GDN, the program produced the classification shown in Figure 3c.

Concept Formation by Finding Classifying Attributes. This section describes an alternative approach for building classifications. This approach searches for one or more classifying attributes whose value sets can be split into ranges that define individual clusters. The important aspect of this approach is that the classifying attributes can be derived through a goal-directed chain of inferences from the initial attributes. The classifying attributes sought are the ones that lead to classes of objects that are best according to the classification goal and the given classification quality criterion.

The "premise" of a descriptor to serve as a classifying attribute is determined by relating it to the goals or derived subgoals of the problem and by considering how many other descriptors it implies. For example, if the goal of the classification is "finding food," the attribute "edibility" might be a good classifying attribute.

The second way of determining the premise of an attribute can be illustrated by the problem of classifying birds. The question of whether "color" is a more important classifying attribute than "is-waterbird" is answered in favor of "is-waterbird" because the latter leads to more implied attributes than does the attribute "color" in a given GDN (e.g., "is-waterbird" implies can swim, has webbed feet, eats fish, and so on), as described by Medin, Wattenmaker, and Michalski (23).

There are two fundamental processes that operate alternately to generate the classification. The first process searches for the classifying attribute whose value set can be partitioned to form classes such that the produced classification scores best according to the classification quality criterion. The second process generates new descriptors by a chain of inferences using background knowledge rules. Descriptors that can be inferred are ordered by relevancy to the goals of the classification.

The search process can be performed in two ways. When the number of classes to form (k) is known in advance, the process searches for attributes having k or more different values in the descriptions of the objects to be classified. These values are called the *observed* values of the attribute. Attributes with the number of observed values smaller than k are not considered. For attributes with observed value sets larger than k, the choice of the mapping of value subsets to classes depends on the resulting quality criterion score for the classification produced and the type of the value set. When the number of classes to form is not known, the above technique is performed for several different values of k. The best number of classes, k, is indicated by the classification that best satisfies the quality criterion and goals.

The generate process constructs new attributes from combinations of existing attributes. Various heuristics of attribute construction are used to guide the process. For example, two attributes that have linearly ordered value sets can be combined using arithmetic operators. When the attributes have numerical values (as opposed to symbolic values such as small, medium, and large), a trend analysis can be used to suggest appropriate arithmetic operators, as in the BACON system by Langley and his associates (25). Predicates can be combined by logical operators to form new attributes through background knowledge rules. For example, a rule that says an animal is a reptile if it is cold-blooded and lays eggs can be written as

$$[\text{cold-blooded}(al)][\text{offspring birth}(al) = \text{egg}] \Rightarrow [\text{animal-type}(al) = \text{reptile}].$$

The application of this rule to the given animal descriptions yields the new attribute "animal-type" with the specified value "reptile." Using this rule and similar ones, one might classify some animals into reptiles, mammals, and birds even though the type of each animal is not stated in the original data.

Summary

Clustering objects or abstract en[illegible] meaningful categories is an important form of lea[illegible] from observation. This entry has described a cla[illegible]milarity-based" approach and the more recent concep[illegible]stering approach to this problem. The fundamental not[illegible] conceptual cohesiveness that groups together objects that correspond to certain concepts rather than objects that are similar according to a mathematical similarity function.

BIBLIOGRAPHY

1. R. S. Michalski and R. E. Stepp, Learning from Observation: Conceptual Clustering, in R. S. Michalski, J. Carbonell, and T. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, CA, pp. 331–363, 1983.
2. A. Rosenfeld, Some Recent Developments in Texture Analysis, *Proceedings of the Conference on Pattern Recognition and Image Processing*, Chicago, 1979.
3. O. Nubuyaki, Discriminant and Least Squares Threshold Selection, *Proceedings of the Fourth International Conference on Pattern Recognition*, Kyoto, Japan, p. 592, 1978.
4. P. H. Swain, "Image and Data Analysis in Remote Sensing," in R. M. Haralick and J. C. Simon (eds.), *Issues in Digital Image Processing*, Sijthoff and Noordhoff, Amsterdam, 1980.
5. G. B. Coleman, Scene Segmentation by Clustering, University of Southern California Image Processing Institute, Report USCIPI, 1977.
6. J. MacQueen, "Some methods for classification analysis of multivariate observations," *Proc. 5th Berkeley Symp. Math. Stat. Prob.*, 281, 1967.
7. R. M. Haralick and L. Shapiro, Decomposition of Polygonal Shapes by Clustering, *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, Troy, NY, p. 183, 1977.
8. R. R. Sokal and R. H. Sneath, *Principles of Numerical Taxonomy*, W. H. Freeman, San Francisco, 1963.
9. R. M. Cormark, "A review of classification," *J. Roy. Stat. Soc.*, Series A, P, 134–321 (1971).
10. M. R. Anderberg, *Cluster Analysis for Applications*, Academic Press, New York, 1973.
11. J. C. Gower, "A comparison of some methods of cluster analysis," *Biometrics* **23**, 623–637 (1967).
12. E. Diday and J. C. Simon, "Clustering analysis," *Communication and Cybernetics*, Springer-Verlag, New York, 1976.
13. R. S. Michalski, R. E. Stepp, and E. Diday, "A Recent Advance in Data Analysis: Clustering Objects into Classes Characterized by Conjunctive Concepts," in L. N. Kanal and A. Rosenfeld (eds.), *Progress in Pattern Recognition*, Vol. 1, North-Holland, Amsterdam, 1981.
14. A. W. F. Edwards and L. L. Cavalli-Sforza, "A method for cluster analysis," *Biometrics* **21**, 362–375 (1965).
15. W. Meisel, *Computer Oriented Approaches to Pattern Recognition*, Academic Press, New York, 1972.
16. E. Diday, "Problems of clustering and recent advances," *Eleventh Congress of Statistics*, Oslo Norway, 1978.
17. R. S. Michalski, "Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts," *J. Pol. Anal. Inform. Sys.* **4**, 219–244 (1980).
18. R. S. Michalski and R. E. Stepp, "Automated construction of classifications: Conceptual clustering versus numerical taxonomy," *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-5** (4), 396–410 (July 1983).
19. R. E. Stepp, Conjunctive Conceptual Clustering: A Methodology and Experimentation, Ph.D. Thesis, Department of Computer Science, University of Illinois, Urbana, IL, 1984.
20. P. Langley and S. Sage, Conceptual Clustering as Discrimination Learning, *Proceedings of the Fifth Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, London, Ontario, 1984, pp. 95–98.
21. D. Fisher, A Hierarchical Conceptual Clustering Algorithm, Technical Report, Department of Information and Computer Science, University of California, Irvine, 1984.
22. P. Langley, J. Zytkow, H. Simon, and G. Bradshaw, The Search for

Regularity: Four Aspects of Scientific Discovery, in R. S. Michalski, J. Carbonell, and T. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol. II, Morgan Kaufmann, pp. 425–469, 1986.

23. D. L. Medin, W. S. Wattenmaker, and R. S. Michalski, "Constraints in inductive learning: An experimental study comparing human and machine performance," ISG Report 86-1, UIUCDS-F-86-952, University of Illinois, 1986.
24. L. A. Rendell, "Toward a unified approach for conceptual knowledge acquisition," *AI Mag.* 4, 19–27 (Winter 1983).
25. P. Langley, G. L. Bradshaw, and H. A. Simon, "Rediscovering chemistry with the BACON system," in R. S. Michalski, J. Carbonell, and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach,* Tioga, 1983, pp. 307–329.
26. D. Fisher and P. Langley, Approaches to Conceptual Clustering, *Proceedings of the Ninth International Joint Conference on AI,* Los Angeles, CA, pp. 691–697, (August 1985).
27. R. E. Stepp and R. S. Michalski, Conceptual Clustering: Inventing Goal-Oriented Classifications of Structured Objects, in R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach,* Vol. II, Morgan Kaufmann, pp. 331–363, 1986.

R. S. Michalski and R. E. Stepp
University of Illinois

COGNITION. See Reasoning.

COGNITIVE MODELING

A cognitive simulation model is a computer simulation of mental or cognitive processes. Such a model is normally constructed by cognitive psychologists, who are members of the branch of experimental psychology that is concerned with the scientific and empirical study of human behavior, with an emphasis on understanding the internal mental mechanisms that underlie behavior (see Cognitive psychology). The purposes of cognitive modeling are to express a theory of mental mechanisms in precise and rigorous terms, to demonstrate the sufficiency of a set of theoretical concepts, and to provide an explanation for observed human behavior.

Because cognitive models use many techniques and ideas from AI, they are similar to AI programs. But the goals of cognitive modeling and AI tend to be substantially different (see Ref. 1). Briefly put, the goal of AI is to build intelligent machines, whereas the goal of cognitive modeling is to build models of human mental mechanisms. These activities are very similar, but they differ mainly in the criteria for success. Again briefly put, the quality of a piece of AI work is measured in terms of how well the machine is able to perform the task. In a cognitive modeling effort the question is not only whether the computer program is able to perform the task but also the extent to which it behaves like a human performing the same task and whether the mechanisms involved are plausible theoretical explanations for human mental processes. Notice that in AI terms these mechanisms may be inefficient or unnecessarily complex for the task.

This entry touches on the contribution of cognitive modeling to AI. It is not a commonly accepted idea, but cognitive modeling work is relevant to AI in that some of the mechanisms in cognitive models are applicable to AI problems.

Purposes of Cognitive Modeling

The rationale for cognitive modeling is best seen in terms of the history of theoretical development in cognitive psychology. Except for the temporary aberration of behaviorism, the goal of experimental psychology over the last century has always been to construct an adequate theory of the mental processes that underlie behavior. An adequate theory of the human mind would explain the observed behavioral data in terms of plausible internal mechanisms. The traditional mode for describing such mechanisms has been in the form of verbal statements. As the ideas get more complex, such verbal theories become difficult to handle. Thus, there is a need to express psychological theory precisely and to demonstrate that theoretical concepts are actually sufficient to explain the behavior and to derive testable predictions about data in a rigorous fashion.

The idea of rigorous theoretical models in experimental psychology is a fairly old idea; an excellent early example is the work of Hull during the 1940s, who constructed one of the first large-scale mathematical theories of behavior. During the fifties and sixties mathematical models of psychological processes were developed. These models represented perceptual and learning situations as stochastic processes, which were very successful in accounting quantitatively for many details of human behavior. See Ref. 2 for a summary of these approaches.

This combination of verbal and mathematical theory has produced what might be termed the "standard" theory of cognition, which is based on a decomposition of the human mind into major components. These consist of structures such as short-term memory and long-term memory and processes such as recognition, memory storage, and memory retrieval, which process and manipulate the information stored in the structures. This theory is the basic framework for most current cognitive models.

As interest in cognitive psychology moved from simple learning (qv) and perception (see Vision, early) to complex behavior such as reasoning (qv) and reading comprehension (see Natural-language understanding), the mathematical models seemed to be inadequate because they characterized behavior in terms of a small number of continuous mathematical variables; it seemed that complex qualitative, or symbolic, systems were needed instead, especially in order to represent knowledge (see Representation, knowledge). In addition, many researchers came to feel that a psychological theory or model should describe the processes going on in the mind rather than simply providing a characterization of the statistical properties of the behavior (3). Thus, computer programs, in which these complex entities can be represented directly, became the ideal mode for expressing theory (4).

Perhaps the most important event in symbolic cognitive modeling was the adoption of semantic networks (qv) from AI. For cognitive psychologists the significance of the semantic network representation was that it provided a representation of knowledge in a form that tied into the classical concept of association very well (see Ref. 5 for a comprehensive review of this topic). Semantic networks were so appealing theoretically that AI quickly became of intense interest to cognitive psychologists, and cognitive simulation models were the best way to incorporate AI concepts into cognitive theory. Currently, there seems to be a consensus that cognitive simulation models best represent the core theoretical concepts in cogni-