AN APPLICATION OF AI TECHNIQUES TO STRUCTURING OJECTS INTO AN OPTIMAL CONCEPTUAL HIERARCHY

by

Ryszard S. Michalski
Robert Stepp

Proceedings of the 7th IJCAI, Vancouver, Canada, August 24–28, 1981.

AN APPLICATION OF AI TECHNIQUES TO STRUCTURING OBJECTS INTO AN OPTIMAL CONCEPTUAL HIERARCHY

Ryszard S. Michalski and Robert E. Stepp

Department of Computer Science
University of Illinois at Urbana-Champaign

ABSTRACT

A method of "learning from observation" is presented which structures a collection of objects into hierarchies of subcategories, such that each subcategory is characterized by a conjunctive description involving relations on selected object attributes. The conjunctive descriptions sprouting from each node are mutually disjoint and optimal as a group according to a flexibly defined criterion. Each level of the hierarchy is determined by an iterative process which repetitively applies a version of the A* search algorithm.

Experiments with the program CLUSTER/PAF implementing the method indicate that the obtained hierarchies represent solutions which have a simple conceptual interpretation and which seem to agree well with the way people structure objects.

I INTRODUCTION

The problem of intelligently structuring a given collection of entities has practical significance not only for applied sciences in general, but also for designing and implementing AI systems. For example, knowledge about the structure underlying given data can help in reducing the search space in problem solving, in organizing large data bases (or rule bases), in dividing knowledge acquisition tasks into useful subcases, or in concisely characterizing a large collection of objects for human understanding.

The problem of data structuring can be viewed as a problem of "learning from observation" ("learning without a teacher"). A simple form of data structuring is clustering, which determines a hierarchy of subcategories ("clusters") within a given collection of objects. In the traditional methods of clustering, developed in cluster analysis and numerical taxonomy [6], the basis for forming subcategories is a "degree of similarity" between objects: the subcategories are collections of objects whose intra-cluster similarity is high and inter-cluster similarity is low.

The traditional clustering techniques have one major disadvantage. Since the only basis for forming clusters is the degree of object similarity (which is a measure dependent only on properties of compared objects), the resulting clusters do not necessarily have any simple conceptual interpretation. The problem of determining the "meaning" of the obtained clusters is simply left to the researcher. This disadvantage is significant because a researcher typically wants not only to find clusters, but also wants to find an explanation of the clusters in human terms.

This paper is concerned with the problem of determining a hierarchical structure underlying a given collection of objects, in which each node corresponds to a subcategory of objects characterized by a conjunctive concept (a logical product of relations on selected object attributes). Structuring objects into such "conjunctive hierarchies" is called conjunctive conceptual clustering.

The idea of conceptual clustering and a general method for determining conjunctive hierarchies was introduced in [3]. This paper discusses in more detail one specific algorithm (implemented in the program CLUSTER/PAF) and illustrates it by a practical problem found in musicology.

II THE SIMILARITY MEASURE VERSUS CONCEPTUAL COHESIVENESS

The similarity between any two objects in the population to be clustered is characterized in the conventional data analysis methods by a single number--the value of the similarity function applied to symbolic descriptions of objects ("data points"). These descriptions are typically vectors, whose components represent scores on selected qualitative or quantitative variables used to describe objects. Frequently a reciprocal of a distance measure is used as a similarity function.

Since the similarity function is solely dependent on the properties of individual objects, the traditional methods are fundamentally unable to capture the "Gestalt properties" of objects that characterize a collection of objects as one whole and are not derivable by considering objects individually. In order to detect such properties, the system must know not only the data points, but also certain "concepts". To illustrate this point, let us consider a problem of clustering data points in Figure 1.

This research was supported in part by the National Science Foundation grant No. MCS-79-06614, the University of Paris IX, and INRIA (France).

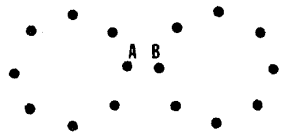

Figure 1. An illustration of conceptual clustering

A person considering the problem in Figure 1 would typically describe it as "two circles". Thus, the points A and B, although being very close, are placed in separate clusters. Here, human solution involves partitioning the data points into groups not on the basis of pairwise distance between points, but on the basis of "concept membership." This means that the points are placed in the same cluster if together they represent the same concept. In our example the concept is a circle.

This idea is the basis of conceptual clustering. From the view of conceptual clustering, the "similarity" between two data points A and B, which we will call the conceptual cohesiveness, depends not only on these points but also on a set of concepts which are available for describing A and B together. In this paper the concepts into which objects are structured are conjunctive descriptions involving relations on selected object attributes.

III TERMINOLOGY

This section gives a brief overview of terminology. A more detailed presentation is contained in [3].

A. Variables and Their Types

Let $x_1,x_2,\ldots,x_n$ denote discrete variables which are selected to describe objects in the population to be analyzed. For each variable a value set (or domain) is defined, which contains all possible values this variable can take for any object in the population. We shall assume that the value sets of variables x_i, i=1,2,...,n are finite. In general, the value sets may differ not only with respect to their size, but also with respect to the structure relating their elements (reflecting the scale of measurement). We distinguish between nominal (qualitative), linear (quantitative), and structured variables, whose domains are unordered, linear, and tree ordered sets, respectively. The structured variables represent generalization hierarchies of related concepts.

B. Event Space and Syntactic Distance

An event e is defined as any sequence of values of variables $x_1,x_2,\ldots,x_n$. The set of all possible events, Σ, is called the event space. The syntactic distance, $\delta(e_1,e_2)$, between two events e_1 and e_2 is the number of variables which have different values in e_1 and e_2.

C. Selectors

A relational statement $[x_i \# R_i]$, where R_i, called the reference, is a list of elements from the domain of x_i, and # stands for one of the relational operators $= \neq \geq \leq$, is called a VL_1 selector or, briefly, a selector.* A selector $[x_i \# R_i]$ is said to be satisfied by an event $e = (x_1,x_2,\ldots,x_n)$, if the value of variable x_i in e, is in relation # with any element of R_i.

D. L-complexes and s-complexes

A logical product of selectors, written as a concatenation of selectors, is called a VL_1 logical complex (or briefly, an ℓ-complex). An event e is said to satisfy an ℓ-complex if values of variables in e satisfy all the selectors in the ℓ-complex. For example, event e = (2,7,0,1,5,4,6,3) satisfies the ℓ-complex $[x_1=2,3][x_3\leq 3][x_5=3..8][x_8=\text{long}]$ (where x_1 is a nominal variable, x_3 and x_5 are linear variables and x_8 is a structured variable) if the value of x_8 in e (i.e., 3) is in the class "long," as defined by the structure of the value set of x_8. An ℓ-complex can be viewed as a symbolic description of the events which satisfy it. For example, the above ℓ-complex is the symbolic description of all events in which x_1 is 2 or 3, x_3 is smaller than or equal to 3, x_5 is between 3 and 8, and x_8 has a value belonging to the category "long" (the values of any other variables are irrelevant).

Any set of events for which there exists an ℓ-complex satisfied by these events and only by these events is called a set-complex or, briefly an s-complex. Henceforth, if α is an ℓ-complex, then by $\hat{\alpha}$ we will denote the corresponding s-complex, i.e., the set of events described by the ℓ-complex. For simplicity, whenever the distinction between an ℓ-complex and an s-complex is not important, then we will use just the term complex.

E. Sparseness

Let E be a set of events in Σ, which represent objects to be clustered. The events in E are called data events (or observed events) and events in $\Sigma \setminus E$ (i.e., events in Σ which are not data events) are called empty events (or unobserved events). Let $\hat{\alpha}$ be an s-complex which covers (includes) some data events and some empty events. The number of data events (points) in $\hat{\alpha}$ is denoted by $p(\hat{\alpha})$. The number of empty events in $\hat{\alpha}$ is called the sparseness and denoted by $s(\hat{\alpha})$. The total number of events in $\hat{\alpha}$ is thus $t(\hat{\alpha}) = p(\hat{\alpha}) + s(\hat{\alpha})$.

The ℓ-complex can be viewed as a generalized description of the data events contained in the corresponding s-complex. The sparseness, as defined above, can be used as a simple measure of the degree to which the ℓ-complex generalizes over (or "fits") the data events. If the sparseness is zero, then the description covers only data events ("zero degree of generalization"). As the sparseness of the complex increases, so does the

*VL_1 is the variable-valued logic system one, which uses such selectors [2].

degree to which it generalizes over the data events. A related but more precise measure of the degree of generalization is the information-theoretic uncertainty of the location of data events in the complex [3].

F. Star

The (theoretical) star G(e|F) of event e against event set F is formally defined [3] as the set of all maximal under inclusion s-complexes covering the event e and not covering any event in F. (An s-complex $\hat{\alpha}$ is maximal under inclusion with respect to property P, if there does not exist an s-complex $\hat{\alpha}^*$ with property P, such that $\hat{\alpha} \subset \hat{\alpha}^*$.) Such maximal complexes, however, have high sparseness and thus are not directly useable in our approach. Therefore, the algorithm produces a reduced star. The reduced star is obtained from the theoretical star by transforming each complex into a new one that covers the same observed events but has the minimum sparseness (or, in general, minimizes a certain criterion).

G. Cover

Let E_1 and E_2 be two disjoint event sets, $E_1 \cap E_2 = \phi$. A cover $COV(E_1|E_2)$ of E_1 against E_2, is any set of s-complexes, $\{\hat{\alpha}_j\}_{j \in J}$, such that for each event $e \in E_1$ there is an s-complex $\hat{\alpha}_j$, $j \in J$, covering it, and none of the complexes $\hat{\alpha}_j$ cover any event in E_2. Thus we have:

$$E_1 \subseteq \bigcup_{j \in J} \hat{\alpha}_j \subseteq \Sigma \setminus E_2 \qquad (1)$$

A cover in which all s-complexes are pairwise disjoint sets is called a disjoint cover. If set E_2 is empty, then a cover $COV(E_1|E_2) = COV(E_1|\phi)$ is simply denoted as $COV(E_1)$. A partition of data events into k subsets, each contained in one set-complex of a disjoint cover is called a conjunctive k-partition. The corresponding k-complexes constitute conjunctive descriptions of these subsets. A simple measure of the "fit" of a k-clustering to the data events is the sparseness of the k-partition defined as the sum of the sparsenesses of the complexes in the partition.

IV THE METHOD AND ITS IMPLEMENTATION

This section describes the algorithm for conjunctive conceptual clustering implemented in program CLUSTER/PAF. The algorithm consists of an inner layer and an outer layer, described in sections IV-A and IV-C, respectively.

A. Inner layer (algorithm PAF)

The inner layer of the algorithm (called PAF) was introduced in [3]. Its function can be described as:

Given:
- a collection of events to be clustered,
- the number of clusters desired (k),
- the criterion of k-clustering optimality,

Find: a conjunctive k-partition of the collection of events that is optimal according to the criterion of k-clustering optimality.

The flow diagram of the algorithm PAF is shown in Figure 2.

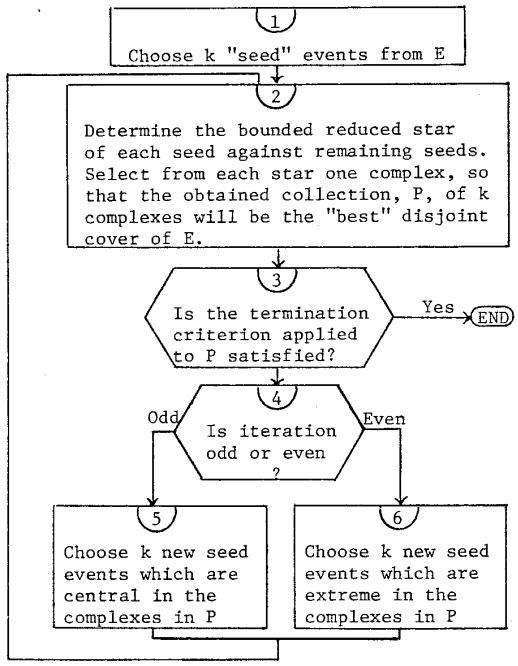

Figure 2. The flow diagram of the inner layer of the PAF algorithm

PAF works iteratively, starting with a set of k initial, randomly chosen seed events ("seeds") from the given collection of events. The seeds are used to determine a set of complexes, which constitute the first conjunctive k-partition of the event set. Subsequent iterations consist of two repeated steps:

1 -- given k complexes, determine the data events (clusters) covered by them,
2 -- given clusters of data events, determine new "seeds," and then a new set of k complexes (a conjunctive k-partition).

The process continues until a termination criterion is satisfied (a local optimum is achieved). The general structure of the algorithm is based on the so-called dynamic clustering method [1].

B. Generating a k-partition from "seeds"

The process of determining a k-partition from seeds involves determining a reduced star of each seed against other seeds, and then selecting complexes from the stars and modifying them in such a way that they constitute a k-partition. The selection is done by a best-first search method.

The existence of such a solution is guaranteed by the "sufficiency principle" proved in [3].

The theoretical star G(e|F) has been defined in section III as the set of maximal complexes covering event e and no events in the set F. Here is a simple algorithm to produce such a star. Assume first that $F=\{e_1\}$, $e_1 \neq e$. To generate the star $G(e|e_1)$ one determines all variables in e that have different values in e_1. Suppose, with no loss of generality, that they are $x_1, x_2, \ldots, x_k$, and that $e_1 = (r_1, r_2, \ldots, r_k, \ldots, r_n)$. Assuming that the variables are nominal, the complexes of the star $G(e|e_1)$ are $[x_i \neq r_i]$, $i=1,2,\ldots,k$, since these are the largest complexes which cover e and do not cover e_1. The number of complexes in a star G(e|F), when F is a single event, is at most n (the number of variables), and at least 1, since $e_1 \neq e$. Assume now that $F = \{e_1, e_2, \ldots, e_s\}$. A star G(e|F) is constructed by building first stars $G(e|e_i)$, $i=1,2,\ldots,s$, and then set-theoretically multiplying these stars by each other, using absorption laws to eliminate redundancy.

This theoretical star is replaced by a reduced star in which complexes cover the same observed events but have the minimum sparseness. To do that, for each complex of the theoretical star observed events contained in it are determined. The list of all values taken by each attribute in the observed events is used as the reference (see section III) of the selector in the corresponding complex of the reduced star (i.e., a refunion operation is performed, as described in [3]). If the reference is equal to the value set (or implies the value set in the case of linear or structured attributes) then the selector is removed.

The upper bound on the size of a star is n^m, where n is the number of variables and m is the number of events in F. Absorption laws will usually eliminate many redundant complexes, but the size of a star may still become unmanageable. Therefore a bounded star is used, which has a specified upper limit, MAXSTAR, on the number of complexes it may contain. Whenever a star exceeds this number, the complexes are ordered in ascending order according to sparseness (or, in general, to any assumed clustering optimality criterion; see end of this section) and only the first MAXSTAR complexes are retained.

The above two steps produce a bounded reduced star. For simplicity, from now on by star and the notation G(e|F) we will mean a bounded reduced star.

At each iteration of algorithm PAF, k stars are produced, each of a single seed event against the remaining k-1 seed events. From each star one complex is selected in such a way that the resulting set will consist of k disjoint complexes (be a conjunctive k-partition), and be optimal according to the assumed criterion. If un-bounded stars were used, each could contain up to $N=n^{(k-1)}$ complexes, and therefore up to N^k sets of complexes would have to be inspected in order to determine the optimal k-partition. To combat this immense search problem the best-first search strategy is used. This search uses a form of algorithm A* (Nilsson [5]).

Assume that k events ("seeds") $e_1, e_2, \ldots, e_k$, have been selected from the collection E and k stars $G_i = G(e_i|\text{remaining seeds})$ have been generated. For simplicity, we will assume that the criterion of clustering optimality is simply to minimize the total sparseness of complexes in the k-partition. At each level of the search tree, a complex is selected from the star corresponding to this level and is added to the partial partition (a sequence of fewer than k complexes). The selected complex is the one which most likely will lead to the optimal k-partition. This procedure avoids testing (possibly very many) clusterings, for which it is possible to predict that they will not be optimal.

Figure 3 illustrates the search process.

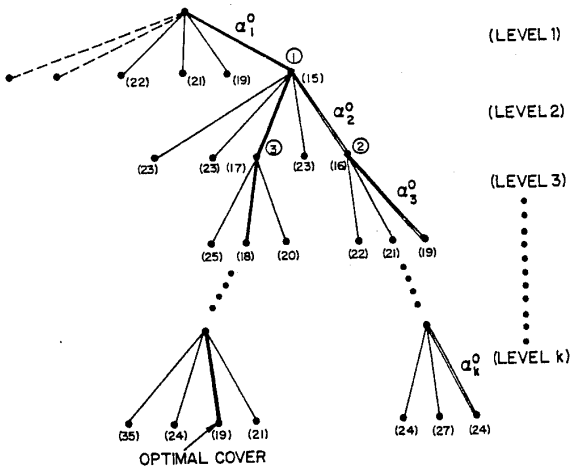

Figure 3. A search tree illustrating algorithm PAF

Branches emanating from a node at level i represent complexes in star G_i. A path from the root to a node at level i represents a partial k-partition with i complexes. When i=k, the path represents a complete k-partition.

In the first step, the sequence of complexes $\alpha_1^0, \alpha_2^0, \ldots, \alpha_k^0$ is determined, where α_1^0 is the complex in star G_i with the smallest sparseness. In the next step, node (1) (Figure 3) is expanded by pairing the "best" complex in G_1, i.e., α_1^0, with every complex in G_2. If the complexes intersect, a special procedure NID modifies them so that they become disjoint. If NID cannot make the complexes disjoint, the path is abandoned (procedure NID is described in detail in [3]). Every so obtained pair of complexes is a partial k-partition with i=2 complexes. This process is repeated for the other complexes in G_1, in the order of their increasing sparseness. Nodes corresponding to all these clusterings (first generation nodes) are assigned a value of the evaluation function: f = h + g, where

h is is the sparseness of the obtained partial disjoint cover and g is the expected cost of the remainder of the k-clustering to be determined (the sum of the sparsenesses of the complexes along the path from node i+1 to leaf node k).

A lower bound for g is determined on the basis of complexes $\alpha_i^?$ generated in the first step. If any of these complexes intersect, procedure NID transforms them into certain "core" complexes, of which it can be proven [3], that the sum of their sparsenesses is a lower bound on the sparseness of the optimal k-partition constructed from complexes of the stars.

According to the algorithm A*, the node to be expanded at the next step is the one which is associated with the lowest value of the evaluation function. The order of expanding nodes in the tree in Figure 3 is shown by numbers in circles. The value of the evaluation function associated with each node is given in parentheses. If complete (not-bounded) stars are used, this algorithm will produce the optimal k-clustering (i.e., in this case, a k-clustering with the minimum total sparseness).

The method can simultaneously use not just one, but several criteria of clustering optimality. In addition to sparseness, these other component criteria include [4]:

- maximizing inter-cluster differences,
- maximizing essential dimensionality,
- maximizing simplicity of cluster representations, and
- maximizing uniformity of cluster populations.

C. Outer layer of CLUSTER/PAF

As described above, the inner layer (PAF) determines an optimal or suboptimal k-clustering of a given collection of events. The outer layer performs two loops, one iterative and one recursive. The iterative loop repeats algorithm PAF for a sequence of values of k (say, k=2,3,...,7) in order to find the value of k for which the most desirable clustering of the given event set is obtained. It is assumed that interesting solutions should have only a few (e.g., less than 7) different clusters.

The recursive loop applies the above process recursively in order to create a hierarchy of clusterings. In the first step, the process is executed for the initial event set E, and a collection of subcategories (clusters) of E is determined. Consecutive steps repeat the same operation for each event set (cluster) obtained in the previous step.

The obtained hierarchy grows in a top-down fashion until a "continuation-of-growth" criterion fails. This criterion requires that the "fit" (measured by sparseness) of the complexes to the events they describe be better by a certain threshold at each next level of the hierarchy. When this criterion is not met, the latest obtained subcategories become leaves of the hierarchy.

The algorithm described above has been implemented in program CLUSTER/PAF, written in PASCAL.

V A MUSICOLOGICAL EXAMPLE

This example illustrates an application of the described method to structuring a collection of one hundred old Spanish folksongs.

The folksongs were characterized by 22 musicological attributes, such as degree of rubato (rhythmic freedom), tonal range, style (monophonic vs. polyphonic), etc. The attributes and the data for the experiment were provided by musicologist Pablo Poveda who studied this problem using traditional methods of numerical taxonomy [6]. The results obtained by those methods, however, were very difficult to interpret because they do not provide any description of the generated clusters.

The top five levels of the conjunctive hierarchy of folksongs produced by CLUSTER/PAF are presented in Figure 4. The criterion of clustering optimality was "minimizing the total sparseness." The branches in the hierarchy have been labeled with the particular characteristic of the folksongs which discriminates between the left and right subcategories. The number of clusters (k) formed at each level was 2 to meet a requirement imposed by the musicologist.

Tips of the hierarchy marked by $\alpha_1, \alpha_2, \ldots, \alpha_{11}$ represent groups of songs (the number of songs is indicated above the tip), whose complete description consists of properties indicated along the path from the root to the tip, and some additional properties not shown in the figure. (These additional properties are less relevant for classifying the songs, as they occur at the lower levels of the hierarchy.) For example, the group denoted by α_4 has the following complete description:

```
[style=monophonic][rubato=low][tonal range=low]
[type=secular][instruments=no]                    (A)
                    Λ
[no. of tones=5..8][panegyric=no][tension=1..3]
[no. of phrases=1..2][melisma=0..2][dance=no]     (B)
```

Part A contains properties shown in the hierarchy (Figure 4) while part B contains additional properties selected by the program from the complete set of attributes.

One interesting aspect of the determined hierarchy is that the value sets of some variables have been split into ranges. These ranges can be considered as new constructed (generalized) values of variables. For example, the range of the degree of "rubato" has been split into two ranges 0..3 and 4..5, which can be described as "low" and "high," respectively (see complex α_4). Similar partitioning of value sets into ranges of values was found for the degree of embellishment, the degree of melisma, the tonal range, and the number of tones in the song. It should be noted that although the nodes in this particular hierarchy are marked by single attributes, the method, in

general, labels the nodes by products of attributes.

VI SUMMARY

The described method for conjunctive conceptual clustering determines a hierarchy of subcategories characterizing a collection of objects. The subcategories are formed in such a way that an appropriate generalization of the description of each subcategory yields a single conjunctive statement. The difference between this method and methods of numerical taxonomy is in extending the concept of the measure of similarity into a more general notion of "conceptual cohesiveness". Such a measure takes into consideration not only the distance between the objects, but also their relationship to other events and, most importantly, their relationship to some predetermined "concepts" (in our case, conjunctive statements). The musicological example described in the paper (as well as other experiments performed with CLUSTER/PAF) indicate that this method has the potential to be a useful new tool for analyzing data.

ACKNOWLEDGEMENTS

The authors thank Mr. Pablo Poveda for providing data used in the musicological experiment and for many helpful comments.

REFERENCES

[1] Diday, E., Simon, J. C., Clustering analysis, Chapter in Communication and Cybernetics 10, Ed. K. S. Fu, Springer-Verlag, Berlin, Heidelberg, New York, 1976.

[2] Michalski, R. S., Variable-Valued Logic and Its Aplications to Pattern Recognition and Machine Learning, Chapter in the monograph: Multiple-Valued Logic and Computer Science, edt. David Rine, North-Holland, 1975.

[3] Michalski, R.S., KNOWLEDGE ACQUISITION THROUGH CONCEPTUAL CLUSTERING: A Theoretical Framework and an Algorithm for Partitioning Data into Conjunctive Concepts, A Special Issue on Knowledge Acquisition And Induction, Policy Analysis and Information Systems, No. 3, 1980.

[4] Michalski, R. S., Stepp, R., Diday, E., A RECENT ADVANCE IN DATA ANALYSIS: Conceptual Clustering, a chapter in the forthcoming book Recent Advances in Pattern Recognition, edt. A. Rosenfeld, L. Kanal, 1981.

[5] Nilsson, Nils, T., Principles of Artificial Intelligence, Tioga Publishing Company, 1980.

[6] Poveda, P., Classification of Folksongs According to the Principles of Numerical Taxonomy, unpublished manuscript, 1980.

[7] Sokal, R. R., Sneath, P. H., Principles of Numerical Taxonomy, W. H. Freeman, 1963.

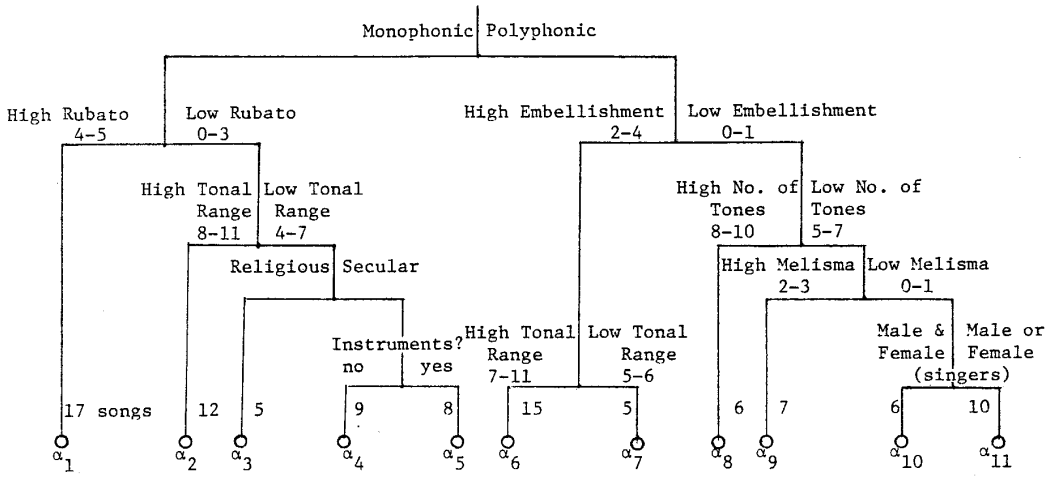

Figure 4. A classification hierarchy of 100 Spanish folksongs found by program CLUSTER/PAF.