COMBINING NUMERIC AND SYMBOLIC TOOLS: A CASE STUDY IN PATTERN RECOGNITION

J. Nicolas, I.C. Lerman

IRISA / INRIA Rennes
Campus de Beaulieu
35042 Rennes Cedex France
Tel (33) 99 847100 Fax (33) 99 383832 e.mail jnicolas@irisa.fr

ABSTRACT
The paper describes a symbolic/numeric approach for the resolution of an identification task, i.e. the characterization of some concepts from a set of examples of these concepts and the assignation of new instances to one of them from the characterizations.
We adress the problem of combining a combinatorial machine learning algorithm with statistical and data analysis algorithms. We argue that the cooperation between these two kind of algorithms leads to better results than if each one is applied independently.
The main conclusions are that a numerical analysis allows a good selection of the descriptive features, but also the creation of new features and the selection of clusters of instances, whereas a symbolic search allows the detection of more robust characterizations of the concepts.

Key words
Interaction between artificial intelligence and data analysis, Machine-learning, Numeric-symbolic identification, Pattern-recognition

1 Introduction

An identification task is the characterization of some concepts from a set of examples of these concepts and the assignation of new instances to one of them from the characterizations.

Algorithms that learn from examples are looking for good characterizations of a given concept to be learned, from positive and negative descriptions of instances of this concept and the knowledge of a language of characterization [Mitchell 82]. We refer such studies as symbolic methods, because their results are elaborated from a combinatorial search, basically involving a symbolic derivation operator. On the other hand, a statistical study of the same set of positive and negative instances, such as a discriminant analysis, lies upon a numerical analysis and synthesis, basically involving a selection of counting operations.

Symbolic methods tend to take into account the user's language and/or knowledge while evaluating their results. Therefore, they are able to produce these results with very detailed representations. But the treatment becomes quickly intractable when the volume of data increases. On the other hand, numerical methods allow the treatment of imperfect data and are generally much faster, but they only allow relatively poor representations of the results (e.g. conjunctions of attribute-value pairs or linear equations). Recently, the moving closer of two disciplines, namely data analysis and machine learning, put in concrete form the symbolic-numeric treatment of data [Gascuel 90]. Symbolic-numeric approach tries to combine numerical and symbolic methods in order to move beyond the limitations of each approach.

We report in this paper and within this framework the improvement of a symbolic characterization task, based on a preliminary data analysis of a set of instances to be characterized. We illustrate our purpose, in case of a large data set, on an identification task in letter recognition.

In the following, we will use indifferently variables, attributes, features or descriptors to denote the primitives of descriptions or characterizations. A characterization will be also named generalization (of the instances), this last term being used also to denote the learning process itself, producing the generalizations. Finally, concepts are sometimes called classes.

The paper is divided into four parts. We first introduce two available tools for the identification task and we compare them through a letter recognition application. The third section describes the benefits of a symbolic-numeric cooperation, with a preliminary data analysis and a modified generalization algorithm. The conclusions of the experiment on the letter recognition task are provided in the last section.

2 A symbolic and a numeric tool for the identification issue

2.1 CEA: a general generalization algorithm

A generalization algorithm takes as input a set of instances of a concept (i.e. class) to be learned (i.e. characterized) and produces a general description of it, covering the positive instances (examples) and not covering the negative instances (counter-examples). It is particularly useful in identification tasks, where new objects have to be assigned to a particular concept. It assumes a given language for the generalizations (characterizations), which contains generally the instance language. The only restriction about this language is that it is always possible to compute the most specific common generalizations of two characterizations and it is always possible to compute the most general common specializations of two characterizations. Mitchell has designed a now classical incremental algorithm [Mitchell 82], managing two sets of generalizations (called boundary sets): the set S of most specific and the set G of most general generalizations. Initially, the most general generalization is set to a characterization covering every possible instances, and the most specific generalization covers none. Then, each example or counter-example is treated in turn. In case of an example (resp. counter-example), the algorithm generalizes (resp. specializes) if necessary the elements of the S set (resp. G set), in order to cover (resp. reject) the new instance. The process halts when the S and G sets are reduced to a same characterization. The algorithm proceeds by a double constrained search between the two boundary sets.

The interest of maintaining these particular sets is that every solution of the generalization task is bounded by one element in each boundary set. Thus, they fully characterize the concept to be learned (and the boundary sets give a usable characterization of the concept at each step), and the result does not depend on the presentation order. The recognition function is built from the boundary sets. Basically, if a new instance is covered by all the elements in the S set, then it is classified as belonging to the concept. On the contrary, if a new instance is not covered by any element in the G set, then it is classified as not belonging to the concept. It remains a range of interpretations for elements "between" these two extrema. A pessimistic interpretation rejects all such elements. An optimistic view accepts all of them. Since discussing the different possible interpretations is beyond the scope of this paper, we will restrict ourselves to these two interpretations.

The algorithm is linear w.r.t. the number of instances and quadratic w.r.t. the size of the boundary sets. The main limitations of the algorithm is that it exhibits a

very slow convergence and that the size of the boundary sets can grow exponentially large in the number of instances (or more exactly, according to the number of attributes). Haussler [Haussler 88] has presented an example showing that this occurs even with purely conjunctive concepts, and such a behaviour is actually observed on real data.

2.2 SPSS DISCRIMINANT: A discriminant analysis

The classical approach for our discrimination problem in the framework of supervised learning, is given by the elaboration of a statistical decision function [Tatsuoka 71], [Duda 73]. With respect to the previous approach, the language of characterization is restricted to score functions, linear combinations of variables. The recognition function is built from the score function using Bayes' rule. The probability that a new instance with score s belongs to the concept C is estimated by

$$P(C/s) = \frac{P(s/C)\ P(C)}{\sum_{C_i \in Concepts} P(s/C_i)\ P(C_i)}$$

This assumes the prior probability of each possible concept to be given and the conditional probabilities $P(s/C_i)$ to be estimated on the sample of training instances. The key concept of a discriminant analysis is variability. Coefficients of the score function are chosen such that its value varies as much as possible between the concepts (classes) and as little as possible within a same concept, i.e. the ratio of within concepts to between concept variability is minimum. SPSS DISCRIMINANT uses the sum of squares, i.e. the variance to measure the variability. In case of two classes, the previous ratio is called Wilks' lambda or U statistic.

3 A comparison of CEA and DISCRIMINANT

3.1 A letter recognition application

We have chosen to run our algorithms on the letter recognition database, created by David J. Slate, and kindly available from University of California at Irvine (May be obtained by anonymous ftp to ics.uci.edu in `machine-learning-databases`). It contains a set of 26 capital typographic letters in several fonts and styles, which have been distorted in various ways. Sixteen primitive numerical attributes (statistical moments and edge counts) have been scaled into a range of integer values from 0 to 15. Thus, each instance is a vector of integers. The interest of such an application is that a discriminant analysis is a priori typically suited for these data. Machine learning

algorithms have a more general application field, but may be expected to behave relatively poorly, compared to the statistical approach.
For the experiment described in this paper, we restrict ourselves to the following task : learning to recognize the letter A. The training set is a set of 50 'A' and 50 'B'. There are two test sets of new instances for the evaluation of the quality of the recognition rules: one with 250 'A' and 250 'B', the other with 250 'A' and 2500 instances with the occurences of all the other letters. The second test set is designed to test the robustness of the learned concepts, i.e. to verify if the recognition rules are not too degradated on entirely new instances.

3.2 A first experiment

The generalization algorithm requires the definition of the language of concepts. We have chosen a simple one. A generalization is a vector of intervals of integers. We also allow disjunctions of two vectors. Thus, geometrically, the possible concepts are all the hyper-rectangles or disjunctions of two hyper-rectangles in the space of features. Note that only simplicity has guided this choice and that it is likely that more interesting languages exist to describe the clusters.
The discriminant analysis has been run with defaults values, that is equal prior probabilities for class A and class B. The first result we have obtained is that the generalization algorithm becomes intractable as soon as more than 4 variables are involved. Indeed, the following table shows an average 20 factor for the increase of complexity w.r.t. the number of variables! (SPSS takes on the order of 1 s, independently of the number of variables on such a task). Although the code is written in Quintus Prolog and not specially optimized, the increase is sufficiently fast to be problematic.

one variable	two variables	three variables	four variables
0.3 sec	15 sec	270 sec	4255 sec

Table 1: Average timing of CEA on a SPARC station 2

The second result deals with the recognition rate of CEA and DISCRIMINANT. In all subsequent tables, each square contains two numbers. The first one (resp. second) is the recognition rate on the first (resp. second) test set. We present a "best effort" value computed by each algorithm in various cases. Interestingly enough, and when the comparison was possible, these values have been reached on the same subset of variables. Since there are 2^{16} subsets of variables, we have

not exhaustively tested the possible combinations. However, we have empirically noticed that the result does not vary much for subsets of length greater than six and even slightly decreases. The important point we wish to emphasize is that if results are slightly better for DISCRIMINANT on a test set containing the same kind of objects ("A" and "B" letters), the results of CEA are far more robust, as it is illustrated by the recognition rate on the second test set. In order to have an idea of the best performance of DISCRIMINANT, we have also run an experiment where the training set is made of 50 "A" and a sample of 150 letters belonging to the rest of the alphabet. We have also set the prior probability (see §2.2) of the class labeled "A" to 9.1 % and the prior probability of the class of counter-examples to 90.9 % (it corresponds to the known frequency of A in the second test set). The resulting recognition rate is 96.5 % on the second test set, lower than our result. Furthermore, it must be noticed that the "real" discrimination power of the function is even worse since only 88.4 % of "A" are indeed recognized. The next step

method	test three variables 500 (A/B)| 2750 (A/X)	test four variables 500 (A/B)| 2750 (A/X)	test all variables 500 (A/B)| 2750 (A/X)
DISCR.	94.6 % | 92.6 %	94.8 % | 91.0 %	96.4 % | 76.9 %
CEA	93.8 % | 96.9 %	93.8 % | 97.7 %	untractable

Table 2: Recognition rates for CEA and DISCRIMINANT

in our study consists of reducing the computing time of CEA, while keeping its good robustness qualities. We have seen that the reduction of the number of variables is crucial in order to avoid the expansion of the boundary sets. But we have also studied two other possibilities: reducing the number of instances and changing the initial description language.

4 Benefits of a symbolic-numeric approach

The field of pattern recognition has long known about the importance of feature selection. Artificial Intelligence and Data Analysis have also emphasized the crucial role of the choice of a good representation for the resolution of problems, that is not only the selection, but also the creation of interesting features. Thus, it seems reasonable to expect improvements from a preliminary analysis of the set of variables. Such an analysis is typically numerical, because what matters is a *global* view of the set of variables. The next section describes how this can be automated with

statistical and data analysis tools. The selection principle consists of retaining the most discriminant variables. The creation principle is to replace a subset of variables with a linear combination of them.

In fact, the analysis of the behaviour of CEA reveals another parameter which influences the performance of the algorithm. We have explained that the cost of the computation is increasing linearly with the number of instances. The issue is that the recognition rate is also a function of the number and quality of the instances. So we cannot reduce the set of instances in the same way as we have reduced the variable set. However, a concept generally presents several facets (in our case, different styles or fonts for a same letter) with different associated instances. Recognizing these several groups of instances by means of an automatic classification may be of great help for their characterization. For this purpose, the Candidate Elimination algorithm has been modified such that *two* levels of generalization are performed. At a first level, instances belonging to a same facet (class) are generalized within a simplified language. At a second level, the algorithm proceeds as usual, but with the previous generalizations as input. Thus, a given instance class comprising close observations of the same concept (e.g. letter "A" of font "Times") can be summarized into a single instance within a new representation language, relevant for the generalization task.

Finally, we have extended this idea of several levels of generalization to benefit from the knowledge of a clustering of the set of *variables* this time. This requires a deeper modification of CEA that is described at the end of the next section. The principle consists to run at a level the basic algorithm on various independent subsets of variables and to compose their results at a higher level, with a focus of the combinatorial search around approximations computed at the previous level.
The two previous points require new data, namely a classification of the variables and a classification of the instances, which will be evaluated by an automated classification also described in the next section.

To summarize, the benefits of a preliminary data analysis for the generalization task are the possibility of

- pruning irrelevant descriptors (keeping the most discriminant ones) or irrelevant instances ("noisy" instances, appearing isolated in the classification);
- selecting sub-concepts, i.e. subsets of instances easier to characterize;
- selecting various clusters and combinations of variables, i.e various abstraction levels to represent the data.

5 Discrimination of a set of variables and two new tools for the identification issue

5.1 Partition discrimination by a set of variables

We wish to define a discrimination degree of a set of variables $\mathcal{W}$ with respect to a given partition Π of the instances. Such a coefficient can be founded on an explained variance (inertia) notion. Assume there are n instances weighted by 1, m classes in Π and the i^{th} class of Π is weigthed by a scalar ν_i (the cardinal of the class). Let $m_e[\mathcal{W}(i)]$ denotes the empirical mean of the values of the sum of the variables belonging to $\mathcal{W}$. The discrimination degree reads:

$$\mathcal{D}(\Pi, \mathcal{W}) = \frac{\sum_{1 \leq i \leq j \leq m} \nu_i \nu_j \ (m_e[\mathcal{W}(i)] - \ m_e[\mathcal{W}(j)])^2}{\sum_{1 \leq k \leq l \leq n} (x_k - x_l)^2} \tag{1}$$

SPSS uses various parameters to estimate the discrimination degree, amongst them being the Wilk's lambda already presented in section 2.2, which is very similar to this one. The variable selection method employed is stepwise selection [Tatsukoa 75], combining the characteristics of forward selection and a backward elimination.

5.2 LLA: a data analysis tool

The general principles of the Likelihood Linkage Analysis (LLA) classification method have been given elsewhere [Lerman 91]. The general parameter of this approach in hierarchical classification deals with the mathematical structure of the data table and the set O to be classified.

5.2.1 Classification of the attribute set

A given column in the data table represents the values of an attribute on the elements of the described set of instances. Thus, each numerical attribute (variable) may be considered as a weigthed unary relation on O.
The raw association index between two variables v and w can be written:

$$s(v, w) = \sum_{i \in Instances} v(i) w(i). \tag{2}$$

A random raw association index can be put in the following form:

$$s^{rand}(v, w) = \sum_{i \in Instances} v(\sigma[i]) w(\tau[i]), \tag{3}$$

where σ and τ are random (with a uniform probability measure) independent permutations on the set of instances. The standardized association coefficient

$$Q(v,w) = \frac{s(v,w) - \mathcal{E}[s^{rand}(v,w)]}{\sqrt{var[s^{rand}(v,w)]}}, \tag{4}$$

where $\mathcal{E}$ and var indicate respectively the mathematical expectation and the variance, is nothing else than $\sqrt{n-1}\rho(v,w)$, and $\rho(v,w)$ is the correlation coefficient between v and w. Then, we may determine the globally standardized coefficient between v and w, m_e and var_e denoting respectively the empirical mean and variance of (4) on the set of unordered pairs of attributes:

$$Q_s(v,w) = \frac{Q(v,w) - m_e(Q)}{\sqrt{var_e(Q)}}. \tag{5}$$

The probability scale for measuring the associations in the spirit of the LLA method is now reachable by (see [Lerman 81],[Lerman 91] for a justification)

$$P_s(v,w) = \Phi[Q_s(v,w)], \tag{6}$$

where Φ designates the cumulative normal distribution function $\mathcal{N}(0,1)$. Table (6) is handled by means of the "maximal link likelihood criterion", in order to build, in an ascendant way, a classification tree on the variable set [Lerman 91].

5.2.2 Classification of the instance set

A given row in the data table represents the vector description $\vec{x} = (x_1, x_2, ...)$ of an instance x through the different variables. For the raw comparison of two instances x and y, we have first considered the cosine between $\vec{x}$ and $\vec{y}$:

$$C(x,y) = cos(\vec{x},\vec{y}) = \frac{\sum_{i \in Variables} x_i y_i}{\sqrt{\sum_{j \in Variables} x_j^2}\sqrt{\sum_{j \in Variables} y_j^2}} \tag{7}$$

In our application, all numbers are positive. Thus, C(x,y) lies in the range [0,1], the maximum being obtained when x and y share a homothetic relation.
In the framework of the LLA method, as in the previous section (equation 5), this raw similarity index is then standardized and rapported to a probability scale. It has been experimentally established, through the treatment of many real examples, the good behaviour of the index.

We have also tested a very different kind of index, comparing the objects with each variable playing an equal role, that is, having the same discrimination power. For this purpose, we have started from the following raw similarity index. Let m be the number of variables. For *each* variable v, we set:

$$s_v(x,y) = \frac{1}{m} - \frac{1}{2}\left(\frac{x_v}{\sqrt{\sum_{i \in Variables} x_i^2}} - \frac{y_v}{\sqrt{\sum_{i \in Variables} x_i^2}}\right)^2. \tag{8}$$

Then, the index is standardized w.r.t. its empirical distribution on the set OxO of ordered instance pairs, and the final similarity index is the mean of the normalized contribution of each variable.

$$S(x,y) = \frac{1}{m} \sum_{i \in Variables} \frac{s_i(x,y) - m_e(s_i)}{\sqrt{var_e(s_i)}} \tag{9}$$

The basic properties are the same than for the cosine index. The new aspect conveyed by S is relative to equal discrimination by the different descriptive variables. Nevertheless, we must notice that the new property is obtained by disconnecting the respective roles of the different descriptive variables, and thus, does not take into account possible interactions between variables.

5.2.3 Significant levels and nodes of the classification tree

A classification tree on O may be interpreted in terms of an ordered sequence of partitions of the set of instances. It is important to retain only the most significant elements (levels of the tree) in this sequence. For this purpose, we introduce a raw association coefficient between a given partition Π and the partition corresponding to the equivalence classes of the globally reduced similarity index S_{ind} used to build the classification (for instance 5):

$$S(\Pi, S_{ind}) = \sum_{(x,y) \text{ in a same class of } \Pi} S_{ind}(x,y) \tag{10}$$

This index is standardized w.r.t. an independence hypothesis: a random partition of the same type as Π is defined to be a partition with the same cardinalities of its classes. The resulting association coefficient gives the basic criterion to compare the partitions of the classification tree.

$$\mathcal{S}(\Pi, S_{ind}) = \frac{S(\Pi, S_{ind}) - \mathcal{E}[S((\Pi^{rand}, S_{ind})]}{\sqrt{var[S((\Pi^{rand}, S_{ind})]}} \tag{11}$$

Under these conditions, the most significant nodes are detected, corresponding to the local maxima of the increase of the previous function $\mathcal{S}$ on the increasing sequence of partitions in the classification tree.

5.3 ICE : generalization with approximations

We have modified CEA with an additional pretreatment of instances. Instances are presented in clusters, each cluster being generalized with a simplified operator, i.e., being characterized within a simplified language. In case of our application, we do not allow disjunction in this language. Thus, clusters are simply characterized by hyper-rectangles.
We have also worked on a generalization of the CEA, ICE [Nicolas 91], allowing a new parameter: the introduction of approximative characterizations of the concept to be learned. These approximations may be considered as a kind of empirical bias, focusing the search for generalizations around the given approximations. Technically speaking, these approximations are generalizations which are "between" an element of the S set and an element of the G set. Focusing the search means that only the pathes lying between an element of a boundary set and an element of the set of approximations are followed when managing the boundary sets (recall that the original algorithm follows *all* pathes between two elements of the boundary sets). The more accurate the approximations are, the more the search is focused.

Now, the issue has shifted towards the construction of good approximations of the concept to be learned. Assume one searches in a smaller description space than the full version space, by selecting a few descriptors in the set of descriptors and by selecting or reducing in a relevant way the instances themselves. Then, the result of CEA would be an approximation of the correct one, using all the available information. We can provide ICE with this approximation and run the algorithm with more descriptors, thus incrementally refining the boundary sets. The last difficulty is to build the approximations in a higher dimensionality space than the space they come from. In case of our application, assume we first select m variables and build the boundary sets of approximations Sm and Gm. In the same way, we choose n other variables and build the sets Sn and Gn. We want to compute the boundary sets in the space of m+n variables. We have to represent Sm, Gm, Sn and Gn in a space of dimensionality m+n, i.e. to compute a set of disjunction of two hyper-rectangles of dimension m+n from a set of disjunctions of two hyper-rectangles of dimension m and a set of disjunctions of two hyper-rectangles of dimension n. The basic idea is to select in one subspace the default values of the missing dimensions in the other subspace. However, taking all possible combinations would result in a huge number of focus points, and the focalization would be very inefficient. Following the general philosophy of the boundary set, we have chosen to only retain for each element in a S set (resp. G set), the hyper-rectangles of minimum (resp. maximum) hyper-volume in the S set (resp. G set) in the other subspace. For instance if m=n=1 and Sm={ [1,2], [2,4] or [6,7]} and Sn={ [3,5], [4,8]} , the approximations

in dimension m+n=2 built on the S sets are Sm+n={ ([1,2], [3,5]), ([2,4], [3,5]) or ([6,7], [3,5]), ([6,7], [3,5]), ([1,2], [4,8]), ([6,7],[4,8])} .

6 A second experiment

Given a set of instances and a set of attributes, we have produced the three analysis we need for the modified generalization algorithm.
First, a tree classifying the instances, and a set of significant nodes in the tree has been produced with the Likelihood Linkage Analysis, as it is described in the previous section. At a given node in the tree, one can associates a partition on the instances. We choose this partition at a meaningfull node, such that it contains less than a given number of classes (12 in this experiment). This partition is then refined such that each class in the partition corresponds to a single type of instances (either examples or counter-examples). In other words, the final partition is the intersection (infimum) between the partition set up at a given level of the classification tree and the one splitting the instances into a positive and a negative class. The following table (best result shown) and additional experiments clearly demonstrate that the clustering process does not deteriorate much the characterizations. Moreover, it appears that generalizations may be better, due to instances pruned for global inconsistency (if the boundary sets reduce to empty sets for a given instance, then the instance is rejected). Thus, a surprising side effect of the clustering of instances is to allow some kind of filtering of "noisy" (low quality) instances. The learning time varies much for various set of attributes, but the gain is always greater than the initial reduction of the set of instances (9) as soon as three variables or more are involved.

size	1 variable	2 variables	3 variables	4 variables	6 variables
timing CEA	0.3 sec	6.6 sec	101.3 sec	466.1 sec	untractable
error CEA	91.8|90.3 %	93.8|94.0 %	92.6|96.9 %	93.8|96.7 %	untractable
timing ICE	0.4 sec	0.7 sec	1.0 sec	1.3 sec	11.6 sec
error ICE	91.8|90.3%	93.8|94.0 %	93.8|96.9 %	93.8|97.6 %	93.8|96.7 %

Table 3: Comparison of CEA and ICE with the clustering of instances

A second series of tests have tried to determine the better subsets of variables for the generalization task. We have computed the discrimination power of each variable with the coefficient described in section 5.1, on three different partitions of

the training set of instances. The first partition was obtained using the cosine index, the second one using the "disconnecting" index, and the third one being just made of two classes ("A" and "B" letters). The stepwise selection of DISCRIMINANT gives a fourth selection. We give the 7 most discriminant variables by decreasing value for each method. Variables 2, 5, 7, 8, 9, 11, 15 have at least three occurences in these lists, thus showing a good coherency of the results.
Cosine : 2, 9, 11, 7, 5, 15, 8
Disconnecting : 9, 2, 7, 8, 5, 12, 11
Two classes : 9, 11, 15, 7, 12, 8, 14
DISCRIMINANT : 9, 7, 4, 15, 5, 16, 2
We have produced generalizations corresponding to the three and four most discriminant variables of these lists. In all cases, the best results are obtained for the set of variables best discriminating the simplest partition into two classes. Also in all cases, the stepwise selection of DISCRIMINANT produces the worst results. Clustering the instances with the cosine index gives sligthly better results than the disconnecting index (however, one must take into account for an objective comparison the broader scope of the latter).

3 variables	Cosine	Disconnecting	Two classes	DISCRIMINANT
ICE+Cosine	93.8| 94.3 %	91.6| 91.1 %	93.8| 96.3 %	67.0| 84.7 %
ICE+Discon.	92.0| 94.7 %	91.0| 91.1 %	93.8| 92.3 %	91.6| 92.6 %
DISCR.	93.8| 85.3 %	93.2| 86.9 %	94.0| 86.6 %	92.8| 85.8 %
4 variables	Cosine	Disconnecting	Two classes	DISCRIMINANT
ICE+Cosine	93.8| 96.7 %	89.8| 93.1 %	93.8| 97.7 %	95.4| 95.5 %
ICE+Discon.	92.0| 96.7 %	88.2| 92.2 %	93.6| 93.3 %	95.4| 93.5 %
DISCR.	94.8| 91.0 %	93.2| 87.1 %	94.6| 91.2 %	94.6| 79.9 %

Table 4: Recognition rates for the 3 and 4 most discriminant variables

A last experiment has investigated the possible contribution of the replacement of the original variables with new ones, linear combinations of the former. The transformation of each "raw" instance is based on the unstandardized coefficients produced by DISCRIMINANT to classify each instance. We have extracted some significant results from several experiments, to illustrate the potential and limitations of such an approach. It appears that the combination of the clustering of instances and the replacement of variables generally decreases the quality of the result. Furthermore, if more than three variables are involved in the combination, the result is generally not acceptable. The better results are observed with few variables

and is sufficient to take a single representative of each class in order to characterize the instances with a good approximation. Thus, variables are chosen according first to the discrimination criterion and second, according to their classification. For instance, the best four variables for the discrimination index on the two classes partition is 9,11,15,7. Taking into account the fact that 9 and 15 are very close in the classification tree, we build the new selection 9,11,7,12. The corresponding recognition rate changes from 93.8/97.7 % to 96.4/97.2 %, which may be considered as a better overall result.
Other indices such as the role of a given attribute in a cluster formation [Geffrault 82] are interesting research directions.

References

[1] M. Tatsuoka "Multivariate analysis." *Wiley, N.Y. 1971*

[2] R. Duda, P. Hart "Pattern Classification and Scene Analysis." *Wiley, N.Y. 1973*

[3] O. Gascuel, A. Guénoche "Aspects de l'interface entre symbolique et numerique", *Actes des 3eme journees nationales du PRC Intelligence Artificielle* Hermes 1990 pp 90-112.

[4] J.P. Geffrault "Discrimination de classes et détermination d'ensembles minimaux de mesures pour la classification automatique de formes. Application à des données en archéologie", *Thèse de 3ème cycle, Université de Rennes I, 15 mars 1982.*

[5] D. Haussler " Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework", *Artificial Intelligence 36*, p. 177-221, 1988

[6] I.C. Lerman "Classification et analyse ordinale des données" *Dunod, Paris 1981*

[7] I.C. Lerman "Foundations of the Likelihood Linkage Analysis (LLA) Classification Method" *Applied Stochastic Models and Data Analysis,* Vol 7 , pp 63-76 Wiley 1991.

[8] T. Mitchell "Generalization as search" *Artificial Intelligence 18* pp 203-226, 1982.

[9] J. Nicolas " Empirical Bias for Version Space", *IJCAI-91*, p. 671-676, Sydney, august 1991