ON SELF-LEARNING PATTERN CLASSIFICATION

by

R.L. De Mantaras

Memorandum No. UCB/ERL M78/63

1 September 1978

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ON SELF-LEARNING PATTERN CLASSIFICATION*

by Ramon L. De Mantaras

Computer Science Division
Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley

Abstract

The ability of learning in Artificial Intelligence systems is the possibility of modification of the model or knowledge representation of the environment.

The information obtained from the environment is processed in order to construct a model of it, that in our case is a partition, by successive modification of some parameters.

In self-learning the system must make decisions by itself, only based on the past information, we give here three algorithms for sequential self-learning classification where the knowledge representation model uses fuzzy and probabilistic concepts. The learning mechanisms consists on the statistical estimation of the parameters defining the membership value for each class.

Several applications are presented where the quality of the recognition can be empirically evaluated from the results.

This type of sequential processing seems particularly useful when the flow of data is continuous and when an instantaneous representation of the data already processed is frequently required. Although it has not been studied here, adaptativity to slow changes in the environment can be easily added to those algorithms.

---

*Research sponsored by Naval Electronic Systems Command Contract N00039-78-C-0013.

ON SELF-LEARNING PATTERN CLASSIFICATION*

by

Ramon L. De Mantaras

Computer Science Division
Department of Electrical Engineering and Computer Science
and the Electronics Research Laboratory
University of California, Berkeley

INTRODUCTION

This report consists of an introduction and four parts A, B, C, and D. It presents two algorithms for the classification of data, based on what we call the self-learning approach.

In part A we examine briefly at the fuzzy approach of Ruspini [36,37,38] to the problem of pattern classification; from there we set our self-learning approach and we present the problem of classification as that of estimating a partition of the data to be classified.

In part B we present an algorithm for classifying data issued from a Gaussian environment; the fundamental tool of this algorithm is the use of numerical filters for estimating a set of parameters which characterize each class. This algorithm has been applied to the recognition of the components of a mixture of Normal distributions [30,31].

In part C we present a second algorithm intended to work with qualitative data. It is very easy to implement and it has been applied to the classification of solid geometrical objects using an artificial hand [3,30].

Finally, in part D, we present an index of dissimilarity and a metric between partitions which are useful for comparing the classification obtained using the algorithmsand the "true" classification [29,30].

---

*Research sponsored by Naval Electronic Systems Command Contract N00039-78-C-0013.

In the field of Pattern Classification, human behavior is the nearest reference model and its advantages and drawbacks are the easiest to evaluate qualitatively. One of the main characteristics common to both men and computers, is the use of a memory. It can be stated that, although the computer memory is more quickly accessible and stores more data without omission, it reaches saturation rather quickly when facing a mass of heterogeneous and unclassified informations. But man, with a memory far less accurate, and in the same environment is able to reach conclusions and take admissible decisions.

Two functions, linked to one another, are the basis of this flexibility in behavior:

1) Use of a capacity of omission, or rather of summarizing the information by remembering important features and forgetting others.
2) Possibility of establishing associations between ideas and of using similarity relations.

In this report, we are going to formulate some of the essential manifestations of this behavior: the classification by self-learning in the absence of initial information.

Because of this ability to forget, man is capable of ignoring some characteristics of an object and can associate it to another object because of the characteristics he remembers. This enables him to establish pseudo-equivalences between objects, and, as we will demonstrate it later in this project, to classify. The mechanisms of forgetting and associating can be performed adaptatively by constant modification of the classification criteria, according to an algorithm.

0.1 General Aspects of Self-Learning

The young child hears sounds emitted by people around him. This emission is not a random one, and a partition emerges from all these sounds. The classes of this partition are the phonems of the linguistic system of his environment. It is self-learning, without teacher and without a priori information, free self-learning, or self-learning with passive environment. Later, the child discovers the communication function of language and he enters into the phase of self-learning with active environment. Sanctions are given according to the degrees of failure of this function. Later, the child's educators will tell him the "truth" on the phonems of the linguistic system which are still obscure for him. It is the type of learning with teacher. We are trying to give here a mathematical model of this aspect of self-learning, called free or with passive environment.

We will notice that this self-learning can be oriented by a "guide" who, without being a teacher who gives sanctions, can still select the data order and influence the transitory period of learning.

We also notice that free learning, that we will call self-learning, is theoretically an elementary situation from which we can introduce oriented self-learning, learning with active environment, and learning with a teacher.

We claim that if this self-learning function is totally absent, there is not a real self-learning, but a simple "conditioning".

0.2 Different Aspects in Pattern Classification

The activities developed in pattern classification have been directed mainly to:

-- either obtaining a best representation of experimental data in order to enable the human being to interpret it

-- or finding identification functions resulting in the classification of data into disjoint classes

We will consider this last asp-ct of Pattern Classification, considering the problem of classification as the process of assigning to each data point a certain degree of belongingness to each class $C_1, C_2, \ldots, C_N$; then the C_i's can be considered as fuzzy sets in the sense of Zadeh [45]. However with our approach these fuzzy sets have a strong probabilistic meaning.

Fuzzy sets as a theoretical basis for pattern classification were first suggested by Bellman, Kalaba, and Zadeh [6]. Subsequently, the papers of Flake and Turner [24], Gitman and Levine [26], Ruspini [36-38], Dunn [20-22], and Bezdek [7-9] concerned various theories of fuzzy pattern classification and fuzzy clustering.

Tamura, Higuchi and Tanaka [41] described for the first time a hierarchical partitioning scheme generated by one parameter family of equivalence relations on a data set representing fuzzy similarity values. At the same time, the notion of similarity relation was developed by Zadeh [46]; subsequently Yeh and Banz [44] suggested the application of fuzzy graphs to clustering analysis.

Some attempts to apply fuzzy automata and fuzzy grammars to Pattern Recognition have been made by Thomason [42], De Palma and Yau [15], and others but we think that a very important problem to consider in the grammatical approach is that of automatic grammatical inference.

Bremmerman [12] introduced the idea of using prototypes for defining the pattern classes, in such a way that the degree of membership of a given object in a certain class could be defined in terms of the amount of deformation to be imposed on the prototype of that class, so that the deformed prototype matches as much as possible the given object. Moreover the method

of deformable prototypes is an effective way to determine the numerical values of the fuzzy membership functions. This problem is central to the fuzzy set approach to empirical problems and has often been discussed, although not very satisfactorily, in literature.

Bezdek and Harris [10] introduced a new definition of transitivity for fuzzy relations that links the triangle inequality to convex decompositions of fuzzy similarity relations in a manner which may generate new techniques for fuzzy clustering.

Finally, the work of Zadeh [47-49] places the connection between fuzzy sets and pattern recognition in a sharp perspective and provides the basis for the application of the fuzzy linguistic approach to the problem of pattern classification.

Our approach is close to that of Ruspini [36-38] in the sense that we look at the problem of classification as the breakdown of the probability density function of the data set into a weighted sum of the probability densities of the component clusters. These densities are interpreted to represent the degree of belongingness of each point to each cluster. Although the concept of fuzzy set is not probabilistic in nature, in this particular case each fuzzy set has a probabilistic meaning and our rules of operation are those of probability.

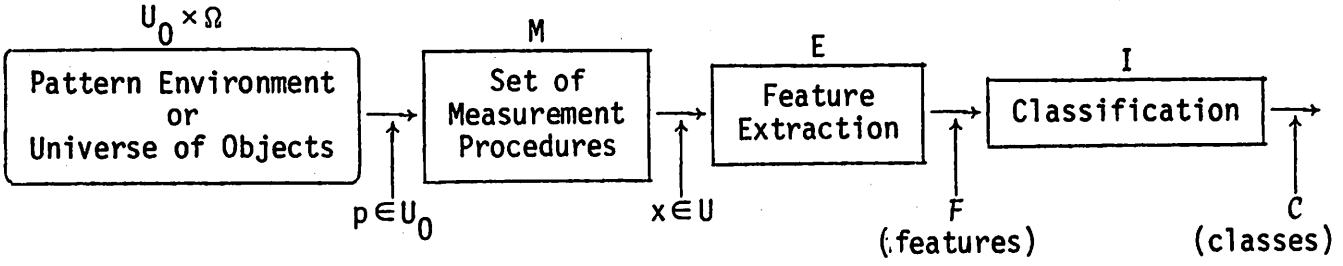
Ruspini [36] suggested the minimization of a meaningful functional defined over all possible fuzzy classifications as a possible technique for decomposing the data set density function into clusters.

Ruspini's approach assumes that the number of classes is known "a priori" and that the data set is available "a priori".

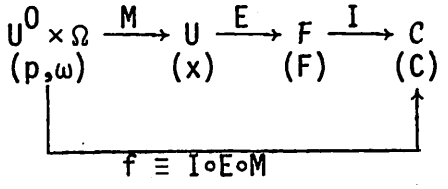
Our approach minimizes the probability of error and is based on the idea of "self-learning"; that means that the algorithm "learns" and classifies

simultaneously (there is no training period before the algorithm starts to classify as in the case of unsupervised learning). Consequently the algorithm itself sets the classes and the differences between the data; there is no external information which would enable the algorithm to discriminate the data "a priori". Furthermore neither the number of samples nor the number of classes is known "a priori" [2,3,30,31].

0.3 General Model



The corresponding mapping diagram is:



Let me give you before some definitions and notations used in the general model above:

0.3.1 Universe Model

Let the pair $(U^0 \times \Omega, A)$ be the universe model where U^0 is the universe of objects to be classified ($U^0 = \{p_1, \ldots, p_n\}$), Ω is the space of perturbations and A is a σ-field such that $(U^0 \times \Omega, A)$ is a measurable space. The space of perturbations enables us to take into account the defects of the objects as well as the defects of the sensors at the perception level.

0.3.2 Perception Model

Our perception model will be assumed to be a triple $(M,U,\mathcal{B})$ where U is the space of mathematical (measured) objects. The pair $(U,\mathcal{B})$ is a measurable space and M is the measurable function:

$$M: U^{0}_{\times\Omega} \rightarrow U \ .$$

0.3.3 Feature Extraction Model

The feature extraction model will be the pair (E,F) where F is the space of features and E is the mapping:

$$E: U \rightarrow F \ .$$

We assume that $\text{card}(F) << \text{card}(U)$.

The methods used for decreasing the dimensionality of the problem are:

- Multivariate Data Analysis methods (principal components, etc.)
- Transform Techniques (Fourier, Karhunen-Loève, etc.)
- Heuristic procedures taking into account the structural properties of U

0.3.4 Classification Model

The classification model will be the pair (f,C) where C is the space of classified objects or "names" or "classes", and f is a mapping from U^0 to C such that one $p_i \in U^0$ corresponds to one $C_i \in C$

$$f: U^{0} \rightarrow C \ .$$

The goal is to find an identification function I (mapping from F to C) such that the "answers" of f and I are equivalent.*

*This is similar to the concepts of opaque algorithms and transparent algorithms defined by Zadeh in [49].

An algorithm that realizes such identification functions is a recognition algorithm (transparent algorithm in the sense of Zadeh). This does not imply that the classification is significant; we can say that the classification is correct only when the interpretation provided by the algorithm coincides with the human interpretation (humans employ an opaque recognition algorithm in their interpretations).

The experience shows that there are roughly two types of identification functions: the characteristic functions and the recursive functions.

0.4 Identification by Characteristic Functions

Let U_i be the set expressed by:

$$U_i = \{x | M(p_i, \omega) = x \text{ and } \omega \text{ covering } \Omega\}$$

If $U_i \cap U_j = \emptyset$ we are in the presence of a deterministic problem and the ideal algorithm which realizes the identification function I should be such that

$$\text{if } x \in U_i \text{ then } (I \circ E)(x) = C_i$$

If $U_i \cap U_j \neq \emptyset$ one can define a membership function for each object:

$$\mu_i : U \to \mathbb{R}^+$$

In this case the recognition algorithm should be such that

$$\text{if } \mu_j(x) > \mu_k(x) \ , \ \forall k$$
$$\text{then } (I \circ E)(x) = C_j$$

This membership function can be grouped into four families:

- Discriminant functions (Rosenblatt 1956) [35]
- Probability measures or density functions (statistical classification)

- Fuzzy membership functions (Zadeh 1965) [45]

- Fuzzy measures (Sugeno 1973) [40]

As has been pointed out earlier, Ruspini's approach regards density functions as fuzzy membership functions. Our approach does the same. Therefore in this case the second and third family are the same. The basic assumption underlying this approach is that there exists a multivariate probability distribution for each class. Members of a pattern class are then treated as population samples which are distributed in a n-dimensional feature space according to the distribution associated with that population. Therefore, for a two-class problem, an observation x is treated as coming from one of two distributions. Then the membership function $\mu_{C_i}(x)$ becomes the probability density function associated to the class C_i.

In this context we can define an optimal procedure, the Bayes procedure.

When $U \equiv \mathbb{R}$ we call $Pr(C_i|x)$ the a posteriori probability of observing an object of the class C_i knowing the measurement x of this object. If these probabilities are known the decision rule at $x \in \mathbb{R}$ is

$$x \in C_j \quad \text{if} \quad Pr(C_j|x) = \max_j [Pr(C_i|x)]$$

By the Bayes rule we have

$$Pr(C_i|x) = \frac{p(x|C_i)Pr(C_i)}{p(x)}$$

where $Pr(C_i)$ is the "a priori" probability of C_i and $p(x)$ is the data set density function.

Assuming that the "a priori" probabilities $Pr(C_i)$ are equal for all i, the strategy at $x \in \mathbb{R}$ becomes

$$x \in C_j \quad \text{if} \quad p(x|C_j) = \max_i [p(x|C_i)]$$

where $p(x|C_i)$ is the conditional density function of x knowing C_i.

Usually these density functions are not known so they must be estimated. Therefore, the classification problem becomes a density estimation problem.

There are two approaches for estimating density functions:

(i) The non-parametric approach: in this case the functional forms of the distributions do not need to be known.

(ii) The parametric approach: in this case the functional forms of the distributions are known but some finite set of parameters characterizing the distribution needs to be estimated (e.g. the mean and the covariance in the case of normal distributions).

PART A

CLASSIFICATION AND ESTIMATION OF A PARTITION

A.1 Classification and Estimation of a Partition

The problem of classification can be regarded as the problem of estimating a partition of the data [30]. Similarly the problem of fuzzy classification can be regarded as that of estimating a fuzzy partition. The works of Ruspini, Bunn and Bezdek all presume a common algebraic framework for fuzzy partitions while our approach does not require that the classes form a fuzzy partition because we think that this condition is too strong when we have noise in the data. In fact with our approach we obtain a partition of the data to be classified, assigning the point to the class to which the degree of membership is the maximum.

The self-learning process consists in building a partition of the data, this partition being modified as long as new data are classified.

Let Ω be a set of elements having the structure of measurable space. A data picked up from Ω will be the observation of an element x_t of Ω.

A sequential S_i is a time ordered set of data $(x_{t_0},\ldots,x_{t_i})$ from t_0 to t_i.

E_i is the subset of Ω $(E_i \subset \Omega)$ observed at the time t_i, i.e. E_i is the set of elements which appear in S_i.

S_i is formed by a set of sets $\{S_j\}_{j=0}^{i}$ such that S_j is the set consisting of the first j elements of S_i.

A self-learning algorithm is the following set of operations which is repeated after incrementation of the index i: at t_i we are given

- the ordered set, S_i
- the subset of Ω, E_i

- a partition of E_i, P_i

- a new element to be classified, $x_{i+1} \in \Omega$

Using a decision (or recognition) rule R, x_{i+1} is classified into a class $C_\alpha \in P_i$. A transformation A is applied to E_i and it becomes E_{i+1}. The partition P_i of E_i becomes the partition P_{i+1} of E_{i+1}, and t_i becomes t_{i+1}.

Remark 1. A partition P_i can be represented in different ways:

non parametric without selection (the complete list of elements forming the classes of the partition is used to represent the partition),

non parametric with selection of the elements of S_i which are the most useful to the decision rule,

parametric. In this case a set θ_i of parameters represents the partition.

Remark 2. Let P_Ω be the set of all possible partitions of Ω and let P_i be a partition of $E_i \subset \Omega$. Then $P_i^* = P_i \cup \{C_\Omega E_i\}$ is a partition of Ω.

A.2 Self-Learning Process of Estimation of a Partition [30]

Let us state the problem of classification as that of estimating a partition. For that we provide the set P_Ω with a metric d [see part C]. Therefore (P_Ω, d) is a metric space.

Let us define a probability measure over Ω characterized by the following density:

$$p[x|\Omega] \quad \forall x \in \Omega$$

Let us define a partition P_{true} which is the "true" partition of the data, and the set $\{p[x|C_k]\}_{C_k \in P_{true}}$ such that

$$p[x|\Omega] = \frac{1}{\text{card}(P_v)} \sum_{C_k \in P_v} p[x|C_k] \quad \forall x \in \Omega \ .$$

Then, a partition P_i^*, built from the sequence S_i and extended to Ω, is called an estimator of P_v. The distance $d(P_i^*, P_v)$ is called the estimation error.

Our goal is to obtain a sequence $\{P_i^*\}_{i=t_0}^{i=t}$ of estimators based on the sequence $\{x_i\}_{i=t_0}^{i=t}$ of observations such that the estimation error decreases when t increases. We call this a <u>self-learning process</u>.

A.3 <u>Parametrization of a Partition</u> [30]

a) <u>Canonical parametrization</u>

A partition P induces a decision rule R that is a mapping from Ω to P

$$R: \Omega \to P$$

such that $R(x) = C_\alpha$ iff $x \in C_\alpha$. The equivalence relation E_q associated with P has the following expression

$$x \ E_q \ y \quad \text{iff} \quad R(x) = R(y)$$

<u>Definition</u>. Let $\Lambda = \{\lambda_{C_i}(\cdot)\}$ be the set of characteristic functions of the classes $C_i \in P$. Then the partition P is equivalent to the set Λ associated with the following decision rule:

$$R(x) = C_\alpha \ ,$$

α being such that $\lambda_{C_\alpha} = \max_i [\lambda_{C_i}(x)]$.

We call Λ a canonical parametrization of P and the associated decision rule will be noted $R(x|\Lambda)$.

b) Fuzzy parametrization

Replacing the characteristic functions $\lambda_{C_i}(x) \in \{0,1\}$ by continuous membership functions $\mu_{C_i}(x) \in [0,1]$ we can define a new decision rule

$$R(x|M) = C_\alpha$$

where $M = \{\mu_{C_i}(\cdot)\}_{C_i \in P}$ and α being such that

$$\mu_{C_\alpha} = \max_i [\mu_{C_i}(x)] .$$

Clearly, we can choose M such that the partition P_M induced by $R(x|M)$ be exactly P (there exists at least the solution of taking $M = \Lambda$).

Similarly every partition P_M induced by a decision rule $R(\cdot|M)$ has a canonical parametrization.

c) Parametrization by kernels

According to the capacity of omission alluded to in the introduction we try to find for each class $C_i \in P$ a finite number n_i of real numbers, composing the vector $\theta_i \in \mathbb{R}^{n_i}$ such that the class C_i can be characterized completely by these numbers according to the decision rule R.

Let us call $\Theta = \{\theta_i\}$ the set of parameters corresponding to the classes $C_i \in P$. Θ is, therefore, a finite set of real numbers and we shall refer to Θ as the set of kernels in the sense of Diday [16].

We denote $R(x|\Theta)$ the decision rule associated with Θ. Clearly if the set M in the fuzzy parametrization can be completely characterized by the set Θ and if Θ can be partitioned into a set of θ_i's such that each θ_i determines uniquely the membership function $\mu_{C_i}(x)$, then $R(x|\Theta)$ is equivalent to $R(x|M)$.

Definition. If for a given P we know that $R(x|M)$ is equivalent to $R(x|\Theta)$, then we shall claim that P is parametrizable by kernels.

Proposition. A partition P of Ω parametrizable by kernels is equivalent to the following pair:

$$\{\Theta, R(\cdot|\Theta)\}$$

where Θ is the set of kernels and $R(\cdot|\Theta)$ is the associated decision rule.

Proof. We have seen that a partition is equivalent to the pair $\{\Lambda, R(\cdot|\Lambda)\}$. Then there exists at least one fuzzy parametrization M such that P is equivalent to $\{M, R(\cdot|M)\}$ and since we know that $R(\cdot|M)$ is equivalent to $R(\cdot|\Theta)$ because it is parametrizable, then we have that P is equivalent to $\{\Theta|R(\cdot|\Theta)\}$.

Concluding Remarks

We have presented the problem of classification as that of estimating a partition. We have also discussed three different ways of parametrization of a partition. In the algorithms that we present in this report we use the fuzzy parametrization in which the membership functions are probabilities. This is similar to Ruspini's approach but the main difference is that we don't obtain a fuzzy partition but a hard partition. The fuzzy partition approach is interesting when there is no noise in the data, because with the fuzzy partition assumption that $\sum_{i=1}^{N} \mu_{C_i}(x) = 1$ where N is the total number of classes will give a high degree of membership of a noise point to at least one of the classes, when, in fact, a noise point whould have a very low degree of membership to each class.

Our approach does not require the sum of the membership functions to be one. Therefore we avoid this problem of noise points. Another important difference between our approach to the classification problem and other existent approaches is the concept of self-learning. This new concept allows the algorithm to start classifying with little "a priori" information.

PART B

DESCRIPTION OF THE ALGORITHM FOR CLASSIFYING GAUSSIAN DATA

Our algorithm uses the parametric approach because we assume that the probability density functions associated with the classes are Gaussian.

The samples to be classified may be viewed as the outcomes of trials governed by a mixture of k probability densities:

$$F(x;\Delta) = \sum_{i=1}^{k} Pr(C_i)p(x|C_i)$$

The "mixing parameters" $Pr(C_i)$ satisfy

$$0 \leq Pr(C_i) \leq 1 \quad \text{and} \quad \sum_{i=1}^{k} Pr(C_i) = 1 \ .$$

Since we assume that all the $Pr(C_i)$'s are equal we have

$$F(x;\Delta) = \frac{1}{k} \sum_{i=1}^{k} p(x|C_i)$$

where Δ is the composite parameter vector:

$$\Delta \equiv \{\delta^1, \delta^2, \ldots, \delta^k\}$$

and $\delta^i \equiv \{\text{mean of } p(x|C_i) = x^i, \text{ covariance of } p(x|C_i) = S^i\}$.

We call δ^i the "kernel" of the class C_i [16].

In this case the classification problem becomes a problem of estimating the parameters of δ^i for each class C_i.

We shall use a Kalman filter [34] for estimating the kernel δ^i of each class. In order to apply this technique we modelize the classes as follows.

Mathematical Model of the Classes

Let us assume that at instant t_n the algorithm has created N classes $(C_1, C_2, \ldots, C_N)$, and that at instant t_{n+1} a new sample is presented to the algorithm. Let us assume that this sample is put into the class C_i. The structure of this class will therefore be modified, since absorbing a new element modifies its mean and its covariance, i.e. the kernel of the class C_i. So we can say that a "dynamics" is associated to each class, taking into account its evolution as the class acquires new elements. For this reason, we denote by $x^i_{k_i}$ and $S^i_{k_i}$ the mean and the covariance of the class C_i at "instant" k_i. The kernel of C_i will be denoted $\delta^i_{k_i}$.

The evolution of the mean $x^i_{k_i}$ of any class C_i is modelized by the following equation:

$$x^i_{k_{i+1}} = x^i_{k_i} + v^i_{k_i} , \quad x^i_{k_i} \in \mathbb{R}^n \tag{1}$$

where $x^i_{k_i}$ is the value of the mean of the class C_i when k_i elements have been absorbed by C_i. $x^i_{k_{i+1}}$ is the mean of C_i after absorbing one more element and $v^i_{k_i}$ is a sequence of independent random Gaussian variables with

$$\mathbb{E}[v^i_{k_i}] = 0 , \quad \mathbb{E}[v^i_{k_i} v^{i\,T}_{j_i}] = Q^i \delta_{k_i j_i} .$$

Now we assume that the mean $x^i_{k_i}$ is imperfectly observed. Then

$$y_t = x^i_{k_i} + w^i_{k_i} , \quad y \in \mathbb{R}^n \tag{2}$$

where $w^i_{k_i}$ is a sequence of independent random Gaussian variables with

$$\mathbb{E}[w^i_{k_i}] = 0 , \quad \mathbb{E}[w^i_{k_i} w^{i\,T}_{j_i}] = R^i \delta_{k_i j_i}$$

and we have the additional hypothesis:

- $\forall k_i, \forall j_i, \ \mathbb{E}[v^i_{k_i} w^{i\,T}_{j_i}] = 0$

- In order to construct a recursive algorithm we assume that the initial state x_0^i is normally distributed with

$$E[x_0^i] = 0 \quad \text{and} \quad E[x_0^i x_0^{i\,T}] = P_0$$

- The initial state x_0^i is uncorrelated with $v_{k_i}^i$ and $w_{k_i}^i$:

$$E[x_0^i v_{k_i}^{i\,T}] = 0 , \quad E[x_0^i w_{k_i}^{i\,T}] = 0 , \quad \forall k_i$$

We see that y and $x_{k_i}^i$ are Gaussian because they are linear combinations of Gaussian variables.

Estimation of the Kernel: $\delta_{k_i}^i = \{x_{k_i}^i, S_{k_i}^i\}$ of Each Class

We know that the mean of the class C_i after absorbing k_i elements is $x_{k_i}^i$.

Because the observed data agree with the equation

$$y_t = x_{k_i}^i + w_{k_i}^i$$

the covariance, which reflects the scattering in the class C_i after absorbing k_i elements is

$$S_{k_i}^i = E[(y-x_{k_i}^i)(y-x_{k_i}^i)^T] = E[w_{k_i}^i w_{k_i}^{i\,T}] = R_{k_i}^i$$

then

$$\delta_{k_i}^i = \{x_{k_i}^i, R_{k_i}^i\} .$$

We shall estimate $x_{k_i}^i$ and $R_{k_i}^i$ recursively because the data are observed sequentially. We shall also estimate $Q_{k_i}^i = E[v_{k_i}^i v_{k_i}^{i\,T}]$ because in most practical situations it is not known.

Having modelized the classes following the equations (1) and (2) our problem becomes a problem of estimating the <u>state</u> of the system, described by the equations (1) and (2), together with the covariances Q^i and R^i.

A recursive solution to this problem is given by a suboptimal adaptive filter based on a Kalman-Bucy filter, in which the unknown covariances $Q^i_{k_i}$ and $R^i_{k_i}$ are recursively and simultaneously estimated with the state $x^i_{k_i}$.

The estimators of $Q^i_{k_i}$ and $R^i_{k_i}$ are the following.

<u>Estimation of Q_k</u>: $$\hat{Q}_k = \frac{1}{k-1} \sum_{j=1}^{k} (q_j - \hat{q}_k)(q_j - \hat{q}_k)^T$$

where $$\hat{q}_k = \frac{1}{k} \sum_{j=1}^{k} q_j$$

and $$q_j = \hat{x}_{j/j} - \hat{x}_{j-1/j-1} \ .$$

$\hat{x}_{j/j} \triangleq$ estimation of the state at instant j, having observed the sequence $\{y_1, \ldots, y_j\}$

$\hat{x}_{j-1/j-1} \triangleq$ estimation of the state at instant j-1, having observed the sequence $\{y_1, \ldots, y_{j-1}\}$

<u>Estimation of R_k</u>: $$\hat{R}_k = \frac{1}{k-1} \sum_{j=1}^{k} (r_j - \hat{r}_k)(r_j - \hat{r}_k)$$

where $$\hat{r}_k = \frac{1}{k} \sum_{j=1}^{k} r_j$$

and $$r_j = y_j - \hat{x}_{j/j-1}$$

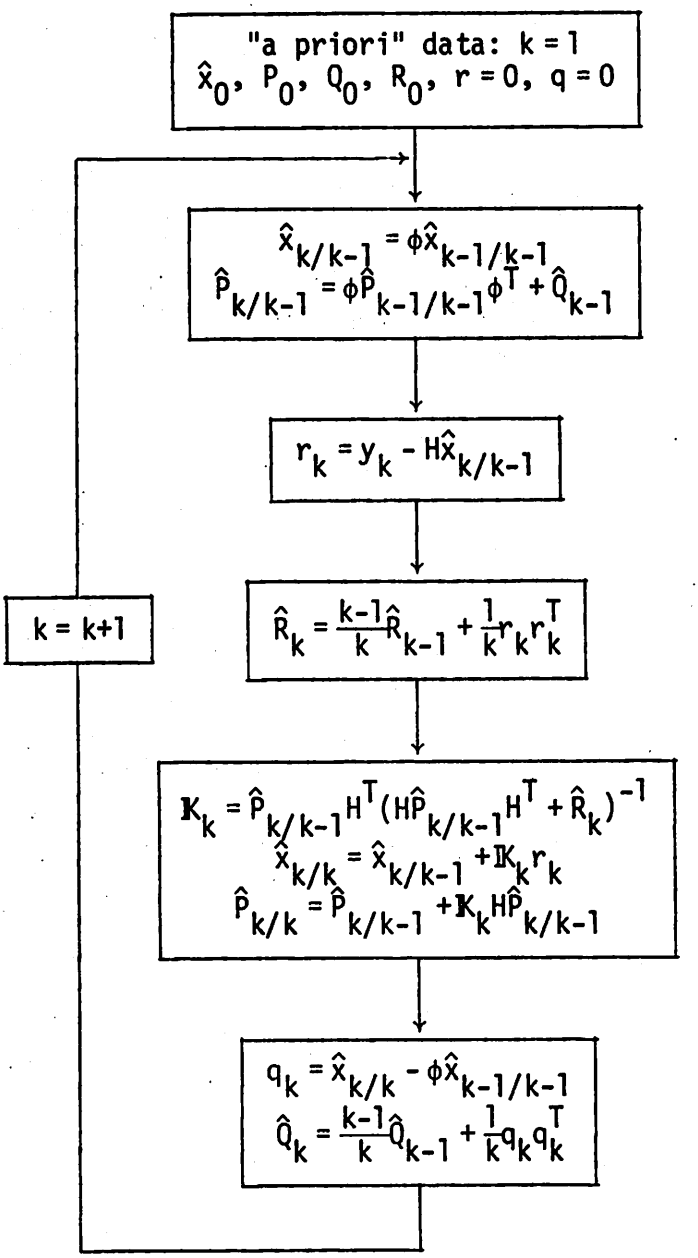
$\hat{x}_{j/j-1} \triangleq$ estimation of the state at instant j having observed the sequence $\{y_1, \ldots, y_{j-1}\}$

<u>Remark 1</u>. It can be proved that if $\lim_{k \to \infty} P_{k/k} = 0$ then the estimators $\hat{Q}_k$ and $\hat{R}_k$ are asymptotically unbiased [30] where P_k is the covariance of the estimation error in the Kalman-Bucy filter defined as [34]

$$P_{k/k} = \mathbb{E}[(x_k - \hat{x}_{k/k})(x_k - \hat{x}_{k/k})^T] \ .$$

So, under this condition, we can say that we are making an asymptotic estimation of the "true" classification of the data.

The algorithm of the suboptimal adaptive filter is the following:



Remark 2. We see that we have one filter for each class but the "evolution instants" are not synchronous because each filter progresses by one step only when its corresponding class accepts one more element.

Progressing by one step is equivalent to updating the estimators from $\hat{x}_{k/k-1}$ to $\hat{x}_{k/k}$ and from $\hat{P}_{k/k-1}$ to $\hat{P}_{k/k}$ using the equations of the filter, i.e. the kernel of the class is modified in order to take into account the new element accepted by this class and that is what we call "learning".

Degree of Membership of One Sample to a Class. Decision Rule

The data observed at any instant t agree with the equation

$$y_t = x^i_{k_i} + w^i_{k_i}$$

where i is to be determined between the existant classes at instant t.

The decision rule for determining the class C_j to which y_t belongs is:

$$y_t \in C_j \quad \text{if} \quad p(y_t|C_j) = \max_{i=1 \text{ to } N} [p(y_t|C_i)] \geq \alpha$$

and we use the suboptimal adaptive filter described before, for estimating the probability density functions:

$$p(y_t|C_i)$$

for $i = 1$ to N, where N is the number of existant classes at instant t.

The threshold α is introduced in order to avoid taking decisions based upon a small probability and at the same time it enables us to increase the number of classes because if:

$$\max_{1=1 \text{ to } N} [p(y_t|C_i)] < \alpha$$

a new class C_{N+1} is created and we decide that:

$$y_t \in C_{N+1}$$

and the initial values of the "kernel" δ^{N+1} of this class will be:

$$x_1^{N+1} = y_t$$
$$s_1^{N+1} = R_0$$

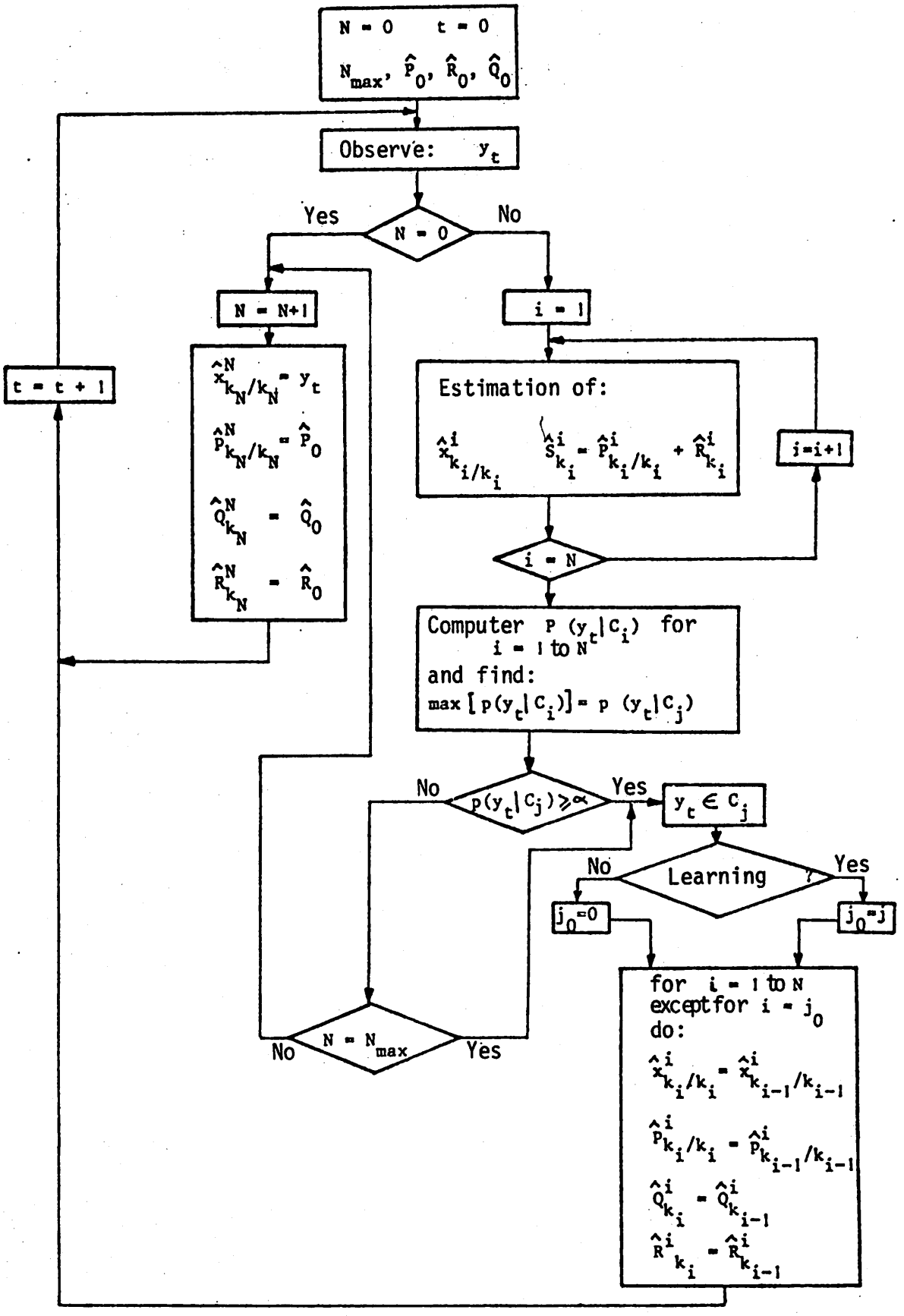
and R_0 is assumed to be known.

Initialization

The algorithm starts by putting the first observation y_1 in the class C_1, so the kernel of C_1 is:

$$\delta_1^1 = \{y_1, R_0\} .$$

This kernel will be modified each time that C_1 will accept one more element.


N = 0 t = 0
N_{max}, $\hat{P}_0$, $\hat{R}_0$, $\hat{Q}_0$
Observe: y_t
Yes
N = 0
No
N = N+1
i = 1
t = t + 1
$\hat{x}^N_{k_N/k_N} = y_t$
$\hat{P}^N_{k_N/k_N} = \hat{P}_0$
$\hat{Q}^N_{k_N} = \hat{Q}_0$
$\hat{R}^N_{k_N} = \hat{R}_0$
Estimation of:
$\hat{x}^i_{k_i/k_i}$ $S^i_{k_i} = \hat{P}^i_{k_i/k_i} + \hat{R}^i_{k_i}$
j=i+1
i = N
Computer $P(y_t|C_i)$ for $i = 1$ to N
and find:
$\max[P(y_t|C_i)] = P(y_t|C_j)$
No
$p(y_t|C_j) \geq \alpha$
Yes
$y_t \in C_j$
No
Learning ?
Yes
$j_0=0$
$j_0=j$
for $i = 1$ to N
except for $i = j_0$
do:
$\hat{x}^i_{k_i/k_i} = \hat{x}^i_{k_{i-1}/k_{i-1}}$
$\hat{P}^i_{k_i/k_i} = \hat{P}^i_{k_{i-1}/k_{i-1}}$
$\hat{Q}^i_{k_i} = \hat{Q}^i_{k_{i-1}}$
$\hat{R}^i_{k_i} = \hat{R}^i_{k_{i-1}}$
No
$N = N_{max}$
Yes

Application to the "Mixture Resolution" Problem

First Example: 150 points picked up from a mixture of 3 Gaussian densities of parameters (see Fig. 1)

$$x^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad x^2 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \quad x^3 = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

$$S^1 = \begin{bmatrix} 4 & 1.7 \\ 1.7 & 1 \end{bmatrix}, \quad S^2 = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}, \quad S^3 = \begin{bmatrix} 4 & -1.7 \\ -1.7 & 1 \end{bmatrix}$$

The estimation results are:

$$\hat{x}^1 = \begin{bmatrix} -0.05 \\ 0.34 \end{bmatrix}, \quad \hat{x}^2 = \begin{bmatrix} 0.07 \\ 3.39 \end{bmatrix}, \quad \hat{x}^3 = \begin{bmatrix} 4.54 \\ 3.11 \end{bmatrix}$$

$$\hat{S}^1 = \begin{bmatrix} 2.5 & 0.81 \\ 0.81 & 0.44 \end{bmatrix}, \quad \hat{S}^2 = \begin{bmatrix} 0.51 & -0.10 \\ -0.10 & 0.23 \end{bmatrix}, \quad \hat{S}^3 = \begin{bmatrix} 2.71 & -1.46 \\ -1.46 & 1.08 \end{bmatrix}$$

In this example all the points have been correctly classified.

Second Example: 150 points picked up from a mixture of 3 Gaussian densities with parameters (see Fig. 2)

$$x^1 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \quad x^2 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \quad x^3 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

$$S^1 = S^2 = S^3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

In this example 6 points have been misclassified. The results concerning the estimation of the kernel of each class are the following:

$$\hat{x}^1 = \begin{bmatrix} 0.24 \\ 3.29 \end{bmatrix}, \quad \hat{x}^2 = \begin{bmatrix} 3.42 \\ 0.38 \end{bmatrix}, \quad \hat{x}^3 = \begin{bmatrix} 3.16 \\ 3.58 \end{bmatrix}$$

$$\hat{S}^1 = \begin{bmatrix} 0.67 & -0.22 \\ -0.22 & 0.66 \end{bmatrix}, \quad \hat{S}^2 = \begin{bmatrix} 1.07 & -0.39 \\ -0.39 & 1.15 \end{bmatrix}, \quad \hat{S}^3 = \begin{bmatrix} 0.64 & -0.03 \\ -0.03 & 1.23 \end{bmatrix}$$

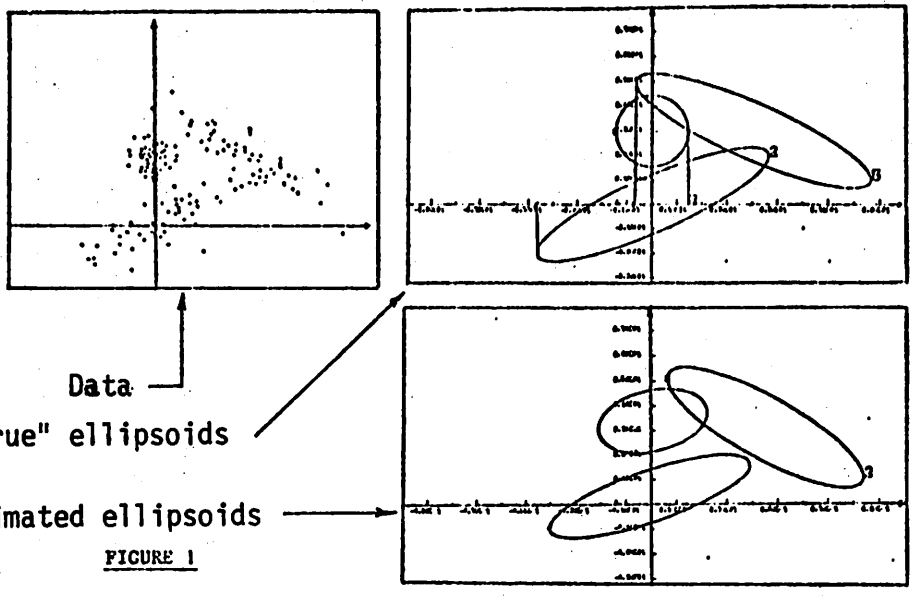


FIGURE 1

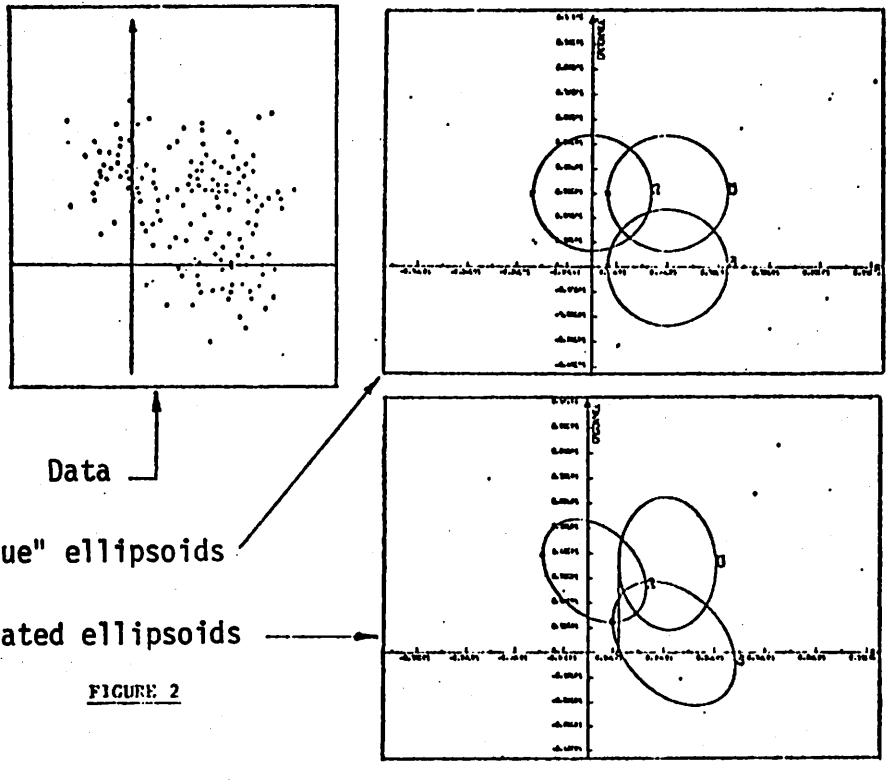


FIGURE 2

Concluding Remarks

The self-learning approach leads to good results in the "Mixture Resolution" problem even when the clusters corresponding to the different distributions are close.

The results concerning the classification itself are better than those concerning the kernel estimation. The last point could be improved by using more elaborated estimators of the covariances.

Our approach is particularly interesting when the "a priori" information is poor (unknown number of classes and unknown number of samples), and when the data are sequentially observed. Because the data are sequentially treated we have at any instant a partition of the data already observed.

PART C

CLASSIFICATION OF QUALITATIVE DATA

In this part we present another algorithm based on the self-learning approach. The decision rule will be the maximum likelihood [3,30].

The probability of an element belonging to a class is estimated by counting; this implies the existence of a similarity measure between qualitative data which gives the possibility of comparing groups of qualitative data.

We have applied this algorithm to the tactile recognition of geometric solid objects using the angular information supplied by potentiometers placed on the finger-joints of an artificial hand [3].

C.1 Probability Estimation by Counting

We consider three types of estimation by counting:

- estimation by natural counting
- estimation by statistical counting
- estimation by weighted counting

For simplicity we are going to consider a vector X having a component quantified into two levels ($v = 2$, i.e. $E = \{0,1\}$) of non equiprobables (the probability of having the level 1 is p and the probability of having level 0 is $q = 1-p$).

i) Estimation by Natural Counting. Let $X_1, X_2, \ldots, X_n$ be a set of independent observations. The estimation of the probability $P[x=i]$ ($i \in \{0,1\}$) is computed by means of the relative frequency. For example, if n_1 is the number of 1's observed among the first n observations, then the estimator of $P[x=1]$ is:

$$p_n = \frac{1}{n} \sum_{i=1}^{n} x_i$$

and we obtain $p_n = \frac{n_1}{n}$

for the X_{n+1} observation

$$p_{n+1} = \frac{n}{n+1} p_n + \frac{1}{n+1} x_{n+1}$$

We have $p_{n+1} = \frac{n_1+1}{n+1}$ if $x_{n+1} = 1$

and $p_{n+1} = \frac{n_1}{n+1}$ if $x_{n+1} = 0$

This estimator is unbiased. It converges almost surely and in quadratic mean. However, it presents the problem of considering zero the probability of an event which has not been observed. For example, if the first n observations are 0, then $p_n = 0$ and $q_n = 1$. This is undesirable when these estimators are used in a product as we shall see later.

To compensate this drawback we can take:

$$p_n^* = p_n \quad \text{if } \exists i: x_i \neq 0$$
$$= \varepsilon << 1 \quad \text{if } \forall i: x_i = 0$$

ε is called the "learning threshold".

ii) Estimation by Statistical Counting. Let $X_1, X_2, \ldots, X_n, X_{n+1}, \ldots$ be a set of independent observations. The estimation of the probability $P[X=i]$ $(i \in \{0,1\})$ is also computed by means of the relative frequency, but in this case we assume that $p_0 = p[X_0=1] = \frac{1}{v}$. Then if n_1 is the number of 1's observed among the first n observations, we have:

$$p_n = \frac{1}{v+n} \sum_{i=1}^{n} X_i = \frac{n_1}{v+n}$$

When the observation of X_{n+1} is added, we have:

$$p_{n+1} = \frac{v+n}{v+n+1} p_n + \frac{1}{v+n+1} x_{n+1} , \quad n = 0,1,\ldots$$

Denoting $$\alpha_n = \frac{1}{v+n}$$

we obtain $$P_{n+1} = \frac{1}{1+\alpha_n} p_n + \frac{\alpha_n}{1+\alpha_n} X_{n+1} \ .$$

Later on, we shall refer to α_n as "statistical α". This estimator is unbiased. It converges to almost surely and also in quadratic mean. We can see that in this case that the "learning threshold" is

$$\varepsilon_n = \alpha_n = \frac{1}{v+n} \ .$$

Therefore, the "learning threshold" depends on n and converges to zero when $n \to \infty$.

iii) Estimation by Weighted Counting. In this case a subjective constant value α is assigned to α_n and we shall prove that the estimation of the probability $P[X=i]$ has an indeterminate bias which is a function of the sequence $\{X_s\}_{s=1}^{n}$.

In order to estimate P let us assume that the observations are independent and that P_0 is known (for example, we can take $p_0 = \frac{1}{v}$ as previously).

Let $X_1, X_2, \ldots, X_n, X_{n+1}, \ldots$ be a set of independent observations. The recursive estimator of P is given by

$$p_{n+1} = \frac{1}{1+\alpha} p_n + \frac{\alpha}{1+\alpha} X_{n+1} \ ; \quad n = 0,1,\ldots$$

In this case the "learning threshold" ε_n is given by

$$\varepsilon_n = \frac{1}{(1+\alpha)^n} \cdot \frac{1}{v} \ .$$

We can see that each observed event artificially modifies its own probability and the greater α, the more important this effect is. Then we can say that this is a subjective method. We shall refer to this α as "subjective α".

We have that P_n strongly depends on the observation X_n for $n \geq 1$. Then P_n is a random variable.

Setting $\delta = 1+\alpha$, we have

$$P_n = \frac{P_0 + (\delta-1)[X_1+\delta X_2+\delta^2 X_3+\cdots+\delta^{n-1}X_n]}{\delta^n}$$

P_n is a random variable that can take 2^n values. Let us compute the expectation of P_n.

$$E[P_n] = \frac{P_0 + (\delta-1)[E[X_1]+\delta E[X_2]+\delta^2 E[X_3]+\cdots+\delta^{n-1}E[X_n]]}{\delta^n}$$

$$E[P_n] = \frac{P_0 + (\delta-1)P(1+\delta+\delta^2+\cdots+\delta^{n-1})}{\delta^n}$$

$$E[P_n] = \frac{P_0 + P(\delta^n-1)}{\delta^n}$$

then

$$\lim_{n\to\infty} E[P_n] = p$$

independently of p_0 and α (if $\alpha > 0$). Unfortunately the variance does not converge to zero:

$$Var[p_n] = \frac{P_0 + p(1-p)(\delta^n-1)}{\delta^n}$$

and

$$\lim_{n\to\infty} Var[P_n] = p(1-p) \quad (\text{if } \alpha > 0)$$

It can be shown [13] that p is estimated with a bias that depends on the sequence $\{X_s\}_{s=1}^n$, on α and on P_0.

This estimator presents the advantage of keeping its adaptativity nature independently of the number of observations that have been made because each new observation has a constant weight (α is constant), while with the "statistical α" this weight tends to zero.

We use the "statistical α" for estimating the probability of an element belonging to a class, in the first version of the classification algorithm that we present in this paragraph. In a second version of the same algorithm we use the "subjective α".

C.2 Description of the Patterns

We have applied this algorithm to the tactile recognition of geometric solid objects using the angular information supplied by potentiometers placed on the finger-joints of an artificial hand. If we have m potentiometers, this information is represented by a vector X of m components $X_1, X_2, \ldots, X_m$ ($\forall i,\ X_i \in \mathbb{R}^+$). We know the maximum and the minimum values that each component can take, which enables us to normalize these components and to quantify the interval in v levels. Then, the pattern to be classified becomes a vector H of m components, each of which can take v values.

This can be represented by a matrix with v rows and m columns. A pattern will represented by this matrix in which each column has only one entry different from zero. For example, let $m = 4$

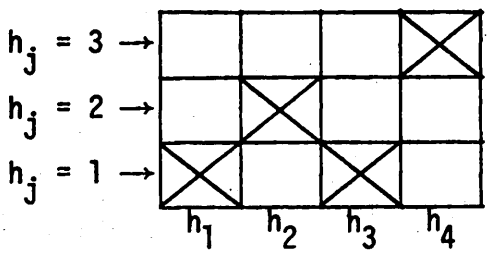
$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3.2 \\ 7.0 \\ 1.3 \\ 4.2 \end{bmatrix}$$

$$x_{max} = \begin{bmatrix} 10.5 \\ 8.2 \\ 2.4 \\ 6.0 \end{bmatrix}, \quad x_{min} = \begin{bmatrix} 0.0 \\ 5.0 \\ 0.2 \\ 0.5 \end{bmatrix}$$

and if $v = 3$ we have

$$H = X(\text{quantified}) = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 3 \end{bmatrix}$$

and the matrix corresponding to this pattern will be



C.3 Characterization of the Classes

A class C_i will be characterized by the probability of the occupation of each entry $f_i(h_j,j)$ in the corresponding matrix F_i. It can be seen that the sum of the probabilities in the same column is

$$\sum_{h_j=1}^{v} f_i(h_j,j) = 1$$

C.4 Degree of Belongingness of a Pattern to a Class

Assuming that the components of the vector X are statistically independent, we can calculate the probability that X belongs to C_i as follows:

$$P_i = \Pr[X|X\in C_i] = \prod_{j=1}^{j=m} \Pr[x_j|X\in C_i] = \prod_{j=1}^{j=m} f_i(h_j,j)$$

and X will be classified into the class C_j if and only if

$$P_j = Pr[x|X \in C_j] = \max_i [P_i]$$

This is called the maximum likelihood decision rule. P_i is the degree of belongingness of X to the class C_i.

C.5 Self-Learning Estimation of the Elements of the Matrices F_i

In order to consider the recursivity of the classification process we shall index by n_i (number of elements classified into the class C_i) the values F_i and P_i. Then we have $F_i(n_i)$, $P_i(n_i)$. Similarly we index X by n_t (total number of elements already classified). Then $n_t = \sum_{i=1}^{N} n_i$ where N is the number of classes that have been created, so X becomes $X(n_t)$. When an element $X(n_t)$ is assigned to a class C_i we modify the corresponding matrix $F_i(n_i)$. This modification consists in increasing the elements $f_i(h_j,j,n_i)$ on a certain value α, i.e.

$$f_i(h_j,j,n_i+1) = f_i(h_j,j,n_i) + \alpha$$

and we normalize the column in such a way that

$$\sum_{h_j=1}^{V} f_i(h_j,j,n_i) = \sum_{h_j=1}^{V} f_i(h_j,j,n_j+1) = 1$$

This is equivalent to the estimation of the probability

$$Pr[h_j(n_t)|X(n_t) \in C_i]$$

by counting (see paragraph C.1). α can be chosen arbitrarily constant or as a function of n_i or n_t. Choosing $\alpha = \frac{1}{n_i}$ the estimation is equivalent to what we called estimation by statistical counting in paragraph C.1. Choosing α arbitrarily constant we do an estimation by weighted counting (paragraph C.1).

C.6 Growing the Number of Classes

An empty class C_k is prepared in such a way that the elements of its corresponding matrix F_k $(n_k = 0)$ are given the same value $f_k(h_j,j,0) = \frac{1}{v}$.

If a new element $X(n_t+1)$ to be classified is such that

$$\forall i \quad P_i(n_i) < P_k(n_k=0)$$

then we assign $X(n_t+1)$ to the empty class C_k. Therefore the structure of C_k is modified by increasing the elements $f_k(h_j,j,0)$, of the matrix $F_k(n_k=0)$, by a certain value α, i.e.

$$f_k(h_j,j,1) = f_k(h_j,j,0) + \alpha$$

and renormalizing the columns.

At this moment we have one more class with one element in it, and we prepare one empty class C_{k+1} with the elements of its corresponding matrix $F_{k+1}(0)$ equal to $\frac{1}{v}$.

We see then that an element will be classified into a non-empty (existant) class with a degree of belongingness greater than that of the empty class, i.e. greater than $(\frac{1}{v})^m$. Then if there exists j such that $p_j(n_j) = \max_i[P_i(n_i)] > (\frac{1}{v})^m$ we classify the element into the existant (non-empty) class C_j and therefore the number of classes is not increased.

C.7 Initialization Without Initial Information

The algorithm starts with just the empty class C_1 in memory. This class is characterized by the corresponding matrix $F_1(n_1=0)$ and the elements of this matrix have the same value, i.e.

$$f_1(h_j,j,0) = \frac{1}{v} \quad \forall j \ .$$

The first element $X(n_t=1)$ is classified into the empty class C_1. Therefore the corresponding matrix $F_1(n_1=0)$ is modified, as described in paragraph C.5, and becomes $F_1(n_1=1)$ and the empty class C_2 characterized by $F_2(n_2=0)$ is prepared.

Remark. If the probabilities were estimated by natural counting instead of weighted counting, the elements $f_1(h_j,j,1)$ of the matrix $F_1(n_1=1)$ would be either 0 or 1 so that the algorithm could not evoluate any more. We avoid this by:

1) The existence of an empty class that has a probability different from zero of accepting an element.
2) The weighted counting of favorable events.

C.8 Flowchart

There exist two program versions of this algorithm depending on the character of α:

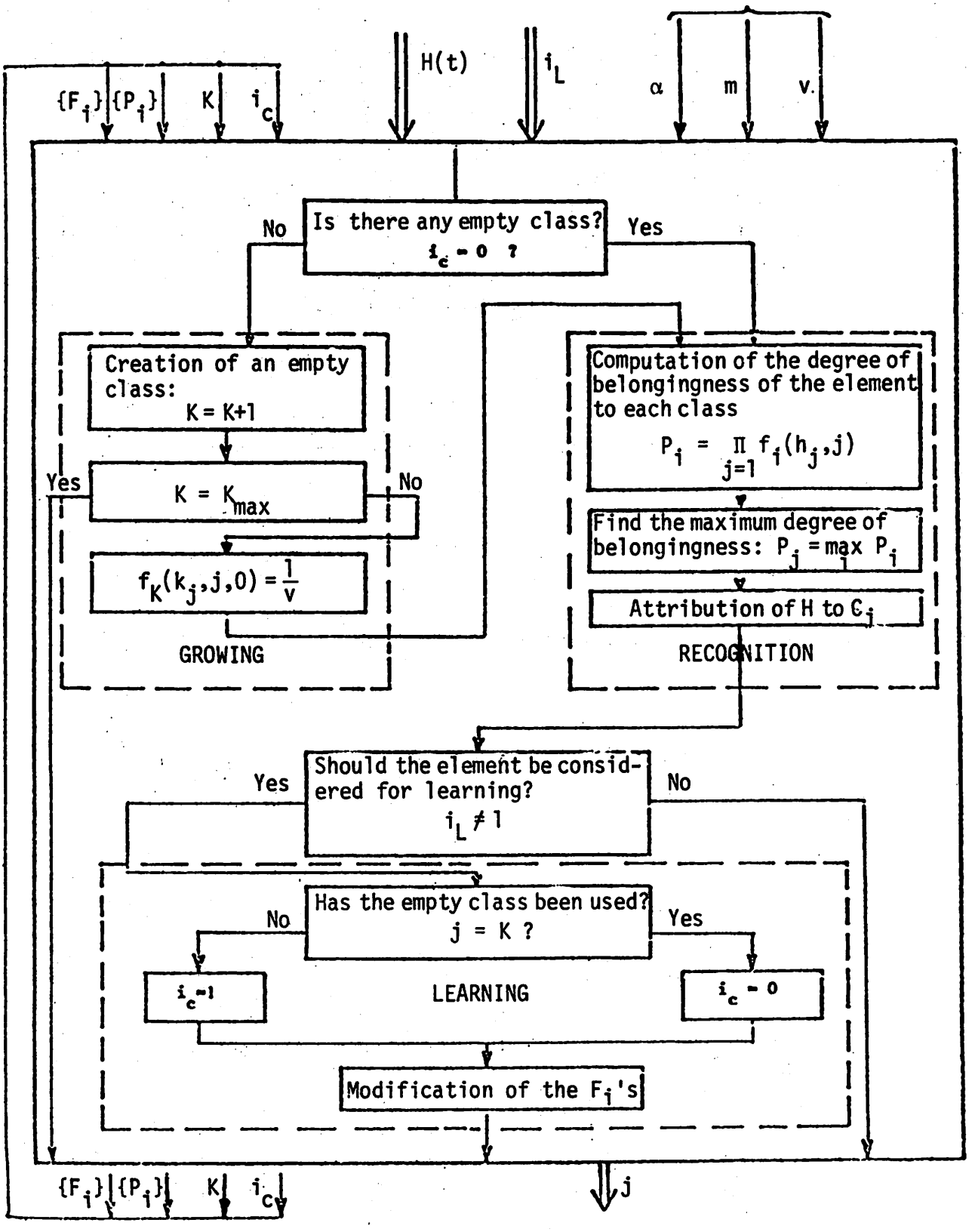
Subjective α: The entries are the element H to be classified and the index i_L. Learning will be used depending on the value of this index.

The output is the number of the class to which the element has been classified.

Other parameters of the program are the matrices $F_i(n_i)$, the vector $P_i(n_i)$, the number k of classes already created, the coefficient α (constant), the number of levels v, the number of components m, and an index i_c that indicates whether or not an empty class exists.

Statistical α: It has the same parameters as the subjective α except the coefficient α, since in this case this coefficient is computed statistically and is different for each class, because its value depends on the number of elements classified in its corresponding class.

These two versions can be represented by the following flowchart.



C.9 Application to the Tactile Recognition of Solid Geometrical Objects

We have studied the possibilities of our self-learning algorithm in recognizing solid geometrical objects using the partial and reduced information supplied by an artificial prehensile organ which has four fingers with three finger-joints each. A linear potentiometer supplying angular information has been placed in each finger-joint.

C.9.1 Measurement Conditions

The different objects that we have considered are:

- a cube
- a sphere
- a tetrahedron

The size of these objects is such that they can be completely embraced by the organ. The objects are positioned in such a way that each phalanx touches the object. 30 measurements for each object are available.

C.9.2 Analysis of the Results

We have considered two cases, either using the information of the four fingers, or using the information of two symmetrical fingers. In the first case the vector H has 12 components whereas in the second case H as 6 components.

Table 1 summarizes the results obtained in the "subjective α" case for different values of α and considering either four fingers or two fingers, the value D is the index of dissimilarity between the classification obtained and the "true" classification. This index of similarity is viewed in detail in Part C of this report.

Table 2 summarizes the results obtained with the "statistical α", also considering either four or two fingers and we have also computed the

dissimilarity between the true classification and the classification obtained by the algorithm.

In Table 3 we see what the output listing of results looks like.

C.10 Conclusions

Our goal was to illustrate our algorithm of self-learning classification and the results appear to be good.

Although the number of classes in the true classification is 3 (cubes, spheres and tetrahedrons), we see that the algorithm creates more than 3 classes. This is because the cube and the tetrahedron can be positioned in different ways in the artificial prehensile organ. This is why we can say that its corresponding classes are multimodal and a multimodal class is interpreted as several classes by the algorithm.

In the case of the sphere we find the 30 corresponding measurements in the same class. This is because the sphere shape is invariant by rotation. Then its corresponding "true" class is unimodal.

For the above reasons we consider good a result that gives a partition of the objects in more than 3 classes if this partition is finer than the "true" 3-class partition, and in this case the index of dissimilarity will be zero.

Summarizing we can say that our approach is interesting when the "a priori" information is poor and the data are observed sequentially. Furthermore, the partition of the data already observed is available at any time.

subjective value of α / experience	α = 0.2	α = 0.5	α = 0.7
Using 4 fingers	D = 0.103 7 classes created very good recognition	D = 0.027 14 classes created perfect recognition but too many classes	D = 0.047 18 classes created very good recognition but too many classes have been created
Using 2 fingers	D = 0.184 8 classes created The spheres have been correctly recognized. The tetrahedron has been well recognized but in its corresponding classes we find some cubes.	D = 0.162 14 classes created The spheres have been correctly recognized. Some cubes are in the classes of the tetrahedrons.	D = 0.148 17 classes created The discrimination between cubes and tetrahedrons is better than with $\alpha = 0.2$ and $\alpha = 0.5$. The spheres are correctly recognized.

TABLE 1

Using 2 fingers	D = 0.1877 7 classes created very good recognition. 14 objects misclassified out of 90 (4 tetrahedra into the class of the sphere, and 10 tetrahedra into the classes corresponding to cubes)
Using 4 fingers	D = 0.1684 8 classes created very good recognition. 13 objects misclassified out of 90 (3 tetrahedra into the class of the sphere and 10 tetrahedra into the classes corresponding to cubes)

TABLE 2

PART D

METRICS BETWEEN PARTITIONS BASED ON THE INFORMATION THEORY

In this part we present an index of dissimilarity and a metric between partitions based on the mathematical theory of information [29]. These distances are useful in analyzing the results of classification algorithms because they are a tool for comparing the partition obtained with the "true" partition or classification.

D.1 Mutual and Conditional Information

Let us consider the following partitions of a set Ω: P_C, of which the classes will be denoted $C_1, C_2, \ldots, C_i, \ldots, C_p$ and P_F, of which the classes will be denoted $F_1, F_2, \ldots, F_j, \ldots, F_r$. Let us consider the following probability-measures:

$$P_i = P(C_i)$$

$$P_j = P(F_j)$$

$$P_{ij} = P(C_i \cap F_j)$$

Clearly, we have:

$$P_i = \sum_j P_{ij}$$

$$P_j = \sum_i P_{ij}$$

$$\sum_i \sum_j P_{ij} = 1$$

The average information of P_C is

$$\bar{H}(P_C) = -\sum_i P_i \log_2 P_i$$

Similarly:

$$\bar{H}(P_F) = -\sum_j P_j \log_2 P_j$$

The mutual average information of $\mathcal{P}_C$ and $\mathcal{P}_F$ is:

$$\bar{H}(\mathcal{P}_C \text{ and } \mathcal{P}_F) = -\sum_{i,j} P_{ij} \log_2 P_{ij}$$

It can be shown [29] and [30] that

$$\bar{H}(\mathcal{P}_C \text{ and } \mathcal{P}_F) \leq \bar{H}(\mathcal{P}_C) + \bar{H}(\mathcal{P}_F)$$

The expression

$$\bar{H}(\mathcal{P}_F/\mathcal{P}_C) = \bar{H}(\mathcal{P}_C \text{ and } \mathcal{P}_F) - \bar{H}(\mathcal{P}_C) = -\sum_{i,j} P_{ij} \log_2\left(\frac{P_{ij}}{P_i}\right)$$

is the conditional information of $\mathcal{P}_F$ knowing $\mathcal{P}_C$ or the supplementary information supplied by $\mathcal{P}_F$ when $\mathcal{P}_C$ occurs. It can be easily seen that

$$\bar{H}(\mathcal{P}_F/\mathcal{P}_C) \leq \bar{H}(\mathcal{P}_F) \ .$$

D.2 The Conditional Information as an Index of Dissimilarity Between Classifications [29]

A classification can be seen as a partition of the set Ω of data to be classified.

Example. Let Ω be $\{x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9\}$. Let $\mathcal{P}_F$ be the partition of Ω in four classes F_1, F_2, F_3, F_4 sharing the data as follows: $F_1 = \{x_1,x_2\}$, $F_2 = \{x_3,x_4,x_5,x_6\}$, $F_3 = \{x_7,x_8\}$, $F_4 = \{x_9\}$ and let us consider another classification of the same set Ω, defined by:

$$\mathcal{P}_C = \{C_1,C_2,C_3,C_4,C_5,C_6\} \ ,$$

the elements of each class being $C_1 = \{x_9\}$, $C_2 = \{x_8,x_7\}$, $C_3 = \{x_3,x_4\}$, $C_4 = \{x_5,x_6\}$, $C_5 = \{x_2\}$, $C_6 = \{x_1\}$. Let us define a matrix $M(C|F)$, the elements of which are

$$P_{ij} = \frac{\text{card}(C_i \cap F_j)}{\text{card}(\Omega)} \ ,$$

i.e. P_{ij} is an estimation of the probability $P(C_i \cap F_j)$.

In the example above we obtain the following matrix:

P_F

P_C \ F_j	F_1	F_2	F_3	F_4
C_1	0	0	0	1/9
C_2	0	0	2/9	0
C_3	0	2/9	0	0
C_4	0	2/9	0	0
C_5	1/9	0	0	0
C_6	1/9	0	0	0

Clearly, we have $\sum_{i,j} P_{ij} = 1$. It is easy to prove that $M(F/C) = M^T(C/F)$ where T stands for transpose.

Remark. Let us assume that we have another classification of Ω represented by

$$P_G = \{G_1, G_2, G_3, G_4\}$$

where

$$G_1 = \{x_7, x_8\}$$
$$G_2 = \{x_9\}$$
$$G_3 = \{x_3, x_4, x_5, x_6\}$$
$$G_4 = \{x_1, x_2\}$$

Let us compute the matrix $M(F/G)$ with $P_{ij} = \frac{\text{card}(F_i \cap G_j)}{\text{card}(\Omega)}$. We obtain

P_F \ P_G	G_1	G_2	G_3	G_4
F_1	0	0	0	2/9
F_2	0	0	4/9	0
F_3	2/9	0	0	0
F_4	0	1/9	0	0

We deduce the following:

Property. If two classifications (partitions) are equal (modulo a permutation of indices) the corresponding matrix is a square matrix that has only one element different from zero in each row and each column. We shall call this matrix "quasi-diagonal".

Remark. In the matrix M(C/F) of the above example we have the following property:

Adding the rows 3 and 4 as well as the rows 5 and 6, we obtain a "quasi-diagonal" matrix; a matrix having this property will be called "quasi-diagonalisable".

Definition. A partition (classification) P_F having less classes than another partition (classification) P_C is compatible with P_C, if and only if P_C is finer than P_F. In this situation the corresponding matrix is "quasi-diagonalisable".

Similarly if the matrix if "quasi-diagonalisable", the partition having less classes is compatible with the other partition.

Now we are going to define an index of dissimilarity between partitions (classifications) that will be zero if and only if the matrix formed with

these partitions is: diagonal, "quasi-diagonal" or "quasi-diagonalisable", i.e. if the partitions are: equal, equal modulo an index permutation or compatible. In any of these three situations, this index of dissimilarity should be zero. The conditional information measure has such properties. Then the distance between two partitions P_F and P_C is:

$$I_d(P_F,P_C) = \bar{H}(P_F|P_C) = -\sum_{i,j} p_{ij} \log_2(\frac{p_{ij}}{p_i}) .$$

D.3 Properties of the Index of Dissimilarity

It can be shown [30] that the index of dissimilarity $I_d(P_F,P_C)$ has the following properties into the set P_Ω of partitions of Ω:

D1 (positivity): $\forall P_F \in P_\Omega$ and $\forall P_C \in P_\Omega$ we have $I_d(P_F,P_C) \geq 0$.

D2: $\forall P_F \in P_\Omega$ and $\forall P_C \in P_\Omega$, if $P_C \subseteq P_F$ then $I_d(P_F,P_C) = 0$.

D3: $Id(P_F,P_C) \neq I_d(P_C,P_F)$ in general.

D4: $\forall P_F \in P_\Omega$, $\forall P_C \in P_\Omega$, $\forall P_G \in P_\Omega$ we have: $I_d(P_F,P_C) + I_d(P_C,P_G) \geq I_d(P_F,P_G)$.

Remark. We will use the following normalization of the above index of dissimilarity

$$I_N(P_F,P_C) = \frac{I_d(P_F,P_C)}{\bar{H}(P_F \text{ and } P_C)}$$

in such a way that $I_N(P_F,P_C) \in [0,1]$. It can be shown [30] that the properties D1 to D4 are preserved.

D.4 A Metric Between Partitions

Let us now define a metric between partitions that will be zero if and only if the corresponding matrix is "quasi-diagonal" or diagonal, i.e. if

the partitions are either equal or equal modulo an index permutation.

This metric can be easily defined by symmetrization of the previous distance, i.e.

$$d(P_F,P_C) = \bar{H}(P_F/P_C) + \bar{H}(P_C/P_F)$$

because symmetry is the only property that does not hold in the case of the index of dissimilarity for being a distance.

Remark. This metric can be normalized as follows

$$d_N(P_F,P_C) = \frac{\bar{H}(P_F/P_C) + \bar{H}(P_C/P_F)}{\bar{H}(P_F,P_C)}$$

such that $d_N(P_F,P_C) \in [0,1]$ and it can be shown [30] that all the metric properties are preserved.

ACKNOWLEDGMENT

I am indebted to IKERLAN, ESCUELA PROFESIONAL POLITECNICA and LIGA DE EDUCACION Y CULTURA of Mondragon (EUZKADI) for making possible my stay in Berkeley. I also would like to thank particularly Lotfi Zadeh and Josep Aguilar-Martin for their help and suggestions.

References

[1] AGUILAR, Martin J., BANON, G. and LOPEZ de MANTARAS, R. (1976). A self-learning algorithm for the classification and recognition of vectorial patterns, IEEE International Symposium on Information Theory, Ronneby, Sweden, June 1976.

[2] AGUILAR, Martin J., BANON, G., BRIOT, M. and LOPEZ de MANTARAS, R. (1976). Tentative de simulation de l'agregation et du classement des informations dans la reconnaissance tactile de solides, Colloque Biomeca II, Toulouse, France, November 1976.

[3] AGUILAR, Martin J. (1978). Learning and self-learning procedures for automatic classification, Memorandum ERL-M78/51, Electronics Research Laboratory, University of California, Berkeley.

[4] BANON, G. (1976). Jeu de hasard comportant une procedure d'apprentissage élémentaire, R.A.I.R.O., pp. 111-117, April 1976.

[5] BELLMAN, R.E. and ZADEH, L.A. (1976). Local and fuzzy logics, Memorandum ERL-M584, Electronics Research Laboratory, University of California, Berkeley.

[6] BELLMAN, R.E., KALABA, R. and ZADEH, L.A. (1966). Abstraction and pattern classification, J. Math. Anal. and Appl. 13, pp. 1-7.

[7] BEZDEK, J.C. (1974). Cluster validity with fuzzy sets, J. Cybernetics 3:3, pp. 58-73.

[8] BEZDEK, J.C. and DUNN, J.C. (1975). Optimal fuzzy partitions: a heuristic for estimating the parameters in a mixture of normal distributions, IEEE Trans. Comp. C-24, pp. 835-838, Autust 1975.

[9] BEZDEK, J.C. (1976). A physical interpretation of fuzzy isodata, IEEE Trans. SMC, SCM-6, pp. 387-390, May 1976.

[10] BEZDEK, J.C. and HARRIS, J.D. (1978). Fuzzy partitions and relations; an axiomatic basis for clustering, International Journal of Fuzzy Sets and Systems.

[11] BONET, E. (1975). En pais de probabilitat finits, Teide Ed. (Spain).

[12] BREMMERMAN, H. (1976). Pattern recognition by deformable prototypes, in Structural Stability, the Theory of Catastrophes and Applications in the Sciences, Springer Notes in Mathematics 25, pp. 15-57.

[13] BRIOT, M. (1977). La stereoguosie en robotique application au tri de solides, Doctoral Dissertation, Université Paul Sabatier, Toulouse, France.

[14] CHANG, R.L. and PAVLIDIS, T. (1977). Fuzzy decision trees, IEEE Trans. SMC, SMC-7:1, pp. 28-35, January 1977.

[15] DEPALMA, G.F. and YAU, S.S. (1975). Fractionally fuzzy grammars with application to pattern recognition, Proc. U.S.-Japan Seminar on Fuzzy Sets and Their Applications, L.A. Zadeh, K.S. Fu, K. Tanaka and M. Shimura (eds.), Academic Press, New York, pp. 449-476.

[16] DIDAY, E. (1972). New methods and new concepts in automatic classification and pattern recognition, Doctoral Dissertation, Université de Paris VI.

[17] DIDAY, E. (1974). Recent progress in distance and similarity measures in pattern recognition, 2nd International Joint Conference on Pattern Recognition, Lungby, Copenhagen, Denmark, August 13-15, 1974.

[18] DIDAY, E. and SCHROEDER, A. (1974). A new approach in mixed distributions detection, R.A.I.R.O., Recherche Opérationnelle 10:6, pp. 75-106, June 1974.

[19] DUDA, R.O. and HART, R.E. (1973). Pattern Classification and Scene Analysis, Wiley, New York.

[20] DUNN, J.C. (1974). Well separated clusters and optimal fuzzy partitions, J. Cybernetics 4:1, pp. 95-104.

[21] DUNN, J.C. (1974). Some recent investigations of a new fuzzy partitioning algorithm and its application to pattern classification problems, J. Cybernetics 4:2, pp. 1-15.

[22] DUNN, J.C. (1974). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, J. Cybernetics 3:3, pp. 32-57.

[23] FAURRE, P. (1973). Réalisations Markoviennes de processus stationnaires, Rapport IRIA No. 13.

[24] FLAKE, R.H. and TURNER, B.L. (1968). Numerical classification for taxonomic problems, J. Theor. Biol. 20, pp. 260-270.

[25] FU, K.S. (1974). Syntactic Methods in Pattern Recognition, Academic Press, New York.

[26] GITMAN, I. and LEVINE, M.D. (1970). An algorithm for detecting unimodal fuzzy sets and its applications as a clustering technique, IEEE Trans. Computers, C-19, pp. 583-593.

[27] KANAL, L. (1974). Patterns in pattern recognition: 1968-1974, IEEE Trans. Information Theory, IT-20, pp. 697-722.

[28] KAUFMANN, A. (1975). Introduction to the Theory of Fuzzy Subsets, Vol. III: Applications to Classification, Pattern Recognition, Automata and Systems, Masson, Paris.

[29] LOPEZ de MANTARAS, R. and NUALART, D. (1976). L'information conditionnelle comme mesure de dissemblance entre classifications, Note of the LAAS (Laboratoire d'Automatique et d'Analyse des Systems) du C.N.R.S., No. LAAS 76I16, Toulouse, France.

[30] LOPEZ de MANTARAS, R. (1977). Autoapprentissage d'une partition: application au classement itératif de données multidimensionnelles, Doctoral Dissertation, Université Paul Sabatier, Toulouse, France, June 1977.

[31] LOPEZ de MANTARAS, R. and AGUILAR, Martin J. (1978). Classification par autoapprentissage a l'aide de filtres numeriques adaptatifs, Congrés AFCET/IRIA Reconnaissance des Formes et Traitement des Images, Paris, France, February 1978.

[32] MYERS, K.A. and TAPLEY, B.D. (1976). Adaptive sequential estimation with unknown noise statistics, IEEE Trans. Aut. Contr., August 1976.

[33] PARRY, W. (1969). Entropy and Generators in Ergodic Theory, W.A. Benjamin, Inc.

[34] RHODES, I.B. (1976). A tutorial introduction to estimation and filtering, IEEE Trans. Aut. Contr. AC-16.

[35] ROSENBLATT (1956). Remarks on some nonparametric estimates of a density function, Annales, Math. Statist. 27

[36] RUSPINI, E.H. (1969). A new approach to clustering, Information and Control 15, pp. 22-32.

[37] RUSPINI, E.H. (1970). Numerical methods for fuzzy clustering, Information Sciences 2, pp. 319-350.

[38] RUSPINI, E.H. (1973). New experimental results in fuzzy clustering, Information Sciences 6, pp. 273-284.

[39] SCHROEDER, A. (1974). Reconnaissance des composants d'un melange, Doctoral Dissertation, Université de Paris VI.

[40] SUGENO, M. (1973). Constructing fuzzy measure and grading similarity of patterns by fuzzy integrals, Trans. SICE 9, pp. 359-367.

[41] TAMURA, S., NIGUCHI, S. and TANAKA, K. (1971). Pattern classification based on fuzzy relations, IEEE Trans. SMC, SMC-1, pp. 937-944.

[42] THOMASON, M.G. (1973). Finite fuzzy automata, regular fuzzy languages and pattern recognition, Pattern Recognition 5, pp. 383-390.

[43] TOU, J. and GONZALEZ, R. (1974). Pattern Recognition Principles, Addison-Wesley, Reading, Mass.

[44] YEH, R.T. and BANG, S.Y. (1975). Fuzzy relations, fuzzy graphs, and their applications to clustering analysis, in Fuzzy Sets and Their Applications to Cognitive and Decision Processes, L.A. Zadeh, K.S. Fu, K. Tanaka and M. Shimura (eds.), Academic Press, New York, pp. 125-149.

[45] ZADEH, L.A. (1965). Fuzzy sets, Information and Control 8, June 1965.

[46] ZADEH, L.A. (1971). Similarity relations and fuzzy orderings, Information Sciences 3, pp. 177-200.

[47] ZADEH, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning, Part I, Information Sciences 8, pp. 199-249; Part II, Information Sciences 8, pp. 301-357; Part III, Information Sciences 9, pp. 43-80.

[48] ZADEH, L.A. (1976). A fuzzy algorithmic approach to the definition of complex or imprecise concepts, Int. J. Man-Machine Studies 8, pp. 249-291.

[49] ZADEH, L.A. (1976). Fuzzy sets and their application to pattern classification and cluster analysis, Memorandum ERL-M607, Electronics Research Laboratory, University of California, Berkeley.