A PROBABILISTIC APPROACH WHICH PROVIDES A MODULAR AND ADAPTIVE NEURAL NETWORK ARCHITECTURE FOR DISCRIMINATION.

by Christophe Monrocq

Thomson-CSF LCR , France

We are concerned with the supervised discrimination of a vector x ($\in$ R^p) between K classes ($C_i; i = 1 \dots K$). The discrfimination consists in learning a discriminant function from a training set of N examples. In a Bayesian context, the discriminant function is a probability function which is the probability of having the class C_i knowing the pattern to classify is x, denoted $P(C_i/x)$ (or equivalently $P(C_i, x)$).
It is well known that MultiLayer Perceptrons (MLP) with a single hidden layer are universal classifiers in the sense that they can approximate decision surfaces of arbitrary complexity, provided the number of hidden neurons is large enough. However this number is unknown and limitations in terms of hardware requirements or learning time may limit the complexity of the network.
Sometimes it is possible to decompose the classification problem, which requires a big network, into subproblems which are efficiently solved by simple modules (with a few or no hidden neurons). To each subproblem corresponds a cluster within the data set on which a module acts like an expert.
If back-propagation [10] is used to train a single MLP to solve the global discrimination, and thus to perform these different subproblems, there will generally be strong interference effects which could lead to slow learning and poor generalization; so for these many reasons the modular approach seems to be preferable.
A number of authors [3, 4, 8, 7] have suggested to use a system composed of several different "experts" : one "expert" for each subproblem. We give theoretical justification for this approach by constructing the global discriminant functions $P(C_i/x)$ from outputs of the "experts" which perform local discriminations within the previous clusters. In a Bayesian context, this means that we are able to construct the global discriminant functions $P(C_i/x)$ by means of the discriminant functions for each subproblem.
Two main hypothesis are posed :

- the experts have probabilities as outputs : e.g. we treat the outputs of the networks as probabilities of alternatives (e.g. patterns' classes), conditionned on the input pattern x.
- the information about clusters is available.

In this paper, we look for appropriate output non-linearities and for an appropriate criterion for the update of parameters of the neural networks. Two approaches are studied : with or without cooperation between modules during the learning.

Terminology :

- the term "cluster" refers to a group of data which belong to different classes; but in the examples presented, a class will belong only to one cluster;
- the terms "module" and "expert" refer to a system which does a discrimination within a cluster; in this paper we are concerned with neural networks.
- the expression "the usual approach" means to solve problem of discrimination with a single network and without the information of clusters.

1 THE MODULAR APPROACH

In a probabilistic framework, our goal is to construct discriminant functions which give estimations of a posteriori probabilities as outputs. In this context and by means of the Bayes' formula $P(A/B)\ P(B) = P(A \cap B)$ these probabilities can be derived as a product between : the probabilities of the classes within clusters and the probabilities of these clusters. All these probabilities are conditionned by the pattern x.

In this paper, a cluster contains patterns of different classes. Three cases can appear :

- Disjoint clusters : Let us study the following example. We have a training set composed of six classes : $(C_i)_{i=1}^6$, which can be split into two disjoint clusters containing classes (C_1, C_3, C_4) and (C_2, C_5, C_6) respectively. By using the Bayes theorem, we have :

$$P(C_1/x) = P([C_1, C_3, C_4]/x) * P(C_1/[C_1, C_3, C_4], x)$$

So, if we want to estimate $P(C_1/x)$ and $P(C_2/x)$, we have to build three "expert-networks": the first does discrimination between clusters (C_1, C_3, C_4) et (C_2, C_5, C_6), the second within cluster (C_1, C_3, C_4) and the last within cluster (C_2, C_5, C_6).

- Non disjoint clusters : A class belongs to many clusters; then the a posteriori probability of a class C_i is obtained by a sum on the clusters to which this class belongs :

$$P(C_i/x) = \sum_j P(\text{cluster } j/x) P(C_i/\text{cluster } j, x)$$

- Hierarchical clustering : The existence of nested clusters means that exist some hierarchical structure in the data base. Let two clusters A et B be such that $A \subset B$. By hypothesis the class C_i is in the cluster A : $C_i \subset A \subset B$. The a posteriori probability of the class C_i is :

$$\begin{aligned} & P(C_i/x) \\ & \quad = P(\text{cluster } A/x) \times P(C_i/\text{cluster } A, x) \\ & \quad = \left[P(\text{cluster } B/x) \times P(\text{cluster } A/\text{cluster } B, x)\right] \\ & \qquad \times P(C_i/\text{cluster } A, x) \end{aligned}$$

The aims of this modular approach are :

1. to obtain more insight into the structure of the data base;
2. to obtain many levels of discrimination.

   These two points mean that we have more information than in the usual approach. This new information can be very relevant if the discrimination is one stage in a complex decision system. Also the discrimination system becomes more robust : if a module is failing, it is possible to have information at another level.
3. to have more chances to solve the problem of discrimination by the decomposition in less complex subproblems.
4. this modular approach allows a better interpretation of the learning. For each cluster we have a local learning, thus the problems during the training phase can be localized : we look for the cluster which causes problem and then we modify the structure of the expert which does discrimination within this cluster. In the usual approach, a complex network has to be built to overcome a local learning problem; this increase in complexity can result in a decrease of the speed of convergence and of the generalization rate.

2 NEURAL LEARNING OF A CLUSTERING

Our only restriction is that modules of discrimination have a posteriori probabilities as outputs. In this paper, the study of the modular approach is restricted to the case of neural networks as modules (= experts) of discrimination.
In this section we propose two approaches for the training of the neural network experts : with or without cooperation between the experts. The paragraph 2.1 details these two approaches; the paragraph 2.2 presents the neural networks that have been used. Our results regroup on one hand theoretical foundations and on the other hand simulations which valid the theory.

2.1 Two approaches : with or without cooperation

Approach 1 : a big network is built from modules : one expert by cluster. The outputs of this big network are combinations of the outputs of the experts, so the outputs of one expert can have influence on the learning of the parameters of the others modules which compose this big network.
Approach 2 : Each expert is learned independently. At the end of the training phase, we combine the outputs to obtain the final probabilities of each class.

2.2 Description of the learning conditions

Criterion to be minimized : for testing the similarity between the targets $d_i(x)$ and the final outputs $\pi_i(x) = \hat{P}(C_i/x)$ as estimates, many criteria can be used :

1. The relative entropy ([5],[1]) :

$$C_{RE} = \sum_{x=1}^{N} C_{RE}(x) = -\sum_{x=1}^{N}\sum_{i=1}^{K-1} d_i(x) \ln \frac{d_i(x)}{\pi_i(x)} \quad (1)$$

2. The log-likelihood ([1]) :

$$C_{LL} = \sum_{x=1}^{N} C_{LL}(x)$$
$$= -\sum_{x=1}^{N} \ln \left(\prod_{i=1}^{K-1} \pi_i^{d_i(x)}(x) \, (1 - \pi_i(x))^{1-d_i(x)} \right) \quad (2)$$

3. The squared error :

$$C_{SE} = \sum_{x=1}^{N} C_{SE}(x) = \sum_{x=1}^{N}\sum_{i=1}^{K-1} \left[d_i(x) - \pi_i(x) \right]^2 \quad (3)$$

Output non-linearities : Our only restriction is that for each module, the outputs are estimations of probabilities. So, the activation functions have to satisfy some constraints such that the outputs satisfy the probabilities' axioms : positivity and sums to one. The functions that have been used, are :

- the logistic : the outputs are included in [0,1] and the experience had proved that the outputs sum nearly to one at the convergence :

$$f(x_1, x_2, \ldots, x_n) = \left(\frac{1}{1+\exp(-x_1)}, \ldots, \frac{1}{1+\exp(-x_n)} \right)$$

- the Gibbs function (= softmax) : the outputs are positive and sum to one.

$$g(x_1, x_2, \ldots, x_n) = \left(\frac{e^{x_1}}{\sum_j e^{x_j}}, \ldots, \frac{e^{x_n}}{\sum_j e^{x_j}} \right)$$

The four sets of conditions under study are :

1. logistic + squared error
2. logistic + log-likelihood
3. Gibbs + squared error
4. Gibbs + relative entropy

Below we shall refer to these choices with the term : "conditions 1, 2, 3 or 4".

Our work and others papers [9, 1] have proved that these modifications may improve the speed of convergence and generalization.
Moreover, in the case of the training of a single network, the choices 2 and 4 give a error term for the back-propagation algorithm which is less complex that for 1 and 3.
So, we are interested to know if these modifications can improve the results of the modular approaches.

Remark 1 : These functions are used on the output layer. On the hidden layers, the output non-linearities are between [-1,1] : the reasons are related to the dynamic of the algorithm for the learning (the back-propagation algorithm) [6].

3 RESULTS

The algorithm used for the learning is the back-propagation of the gradient [2], in its stochastic form : after the presentation of each pattern, the error $C(x)$ between the target and the output is measured. This error is used for the update of the weights (formula (4)). Our aims are :

- for a fixed set of conditions (criterion, output non-linearities), to test the learning and generalization rates w.r.t. the approach;
- for a given approach, to test the learning and generalization rates w.r.t. the set of conditions.

APPROACH 1 : COOPERATIVE LEARNING
In this paragraph the experts are regrouped to form an single network like the one represented by the figure 1. So, we have a single data base which all classes. All experts have the same inputs and the attribution of a class to a cluster is made by the multiplication nodes represented on the figure 1. For example, if the class C_6 belongs only to the second cluster only, then it's a posteriori probability is :

$$\pi_i = P(C_i/x) = P(\text{cluster 2}) \times P(C_i/\text{cluster 2}, x)$$

The back-propagation algorithm is appliqued to a single network like the one on figure 1.

We want to know if the learning and generalization rates of this network depend of the conditions : (criterion, output non-linearities).

- the type of non-linearity on the experts' outputs to obtain P_i and $P_{j/i}$;
- and the criterion used ((1), (2), (3)) on the final outputs π_i.

Simulations : Figure 2 and the first line of the table 1.
The recognition rates and the value of the criterion for the test set are plotted on the figure 2. As we have discussed, it is possible to use an other criterion than the squared error; but for homogeneity and to permit comparisons, we have only plotted the squared error even when the criterion that is minimized is (1) or (2).
The data base, on which the results of the figure 2 and table 1 are obtained, is constituted from artificial data in two dimensions (gaussians) which form eight classes : 800 patterns in the training set and 1200 in the test set. Identical results was obtained with simulations made on real data.
On the figure 2 we can see that the conditions 2,3 and 4 give recognition rates that are nearly similar (for the test set, the incertitude on the results is about 1.4%). However case 1 (logistic + squared error) does not converge and the squared error increases. The reasons of this situation are linked to the back-propagation algorithm. This algorithm uses a gradient descent, where the iterative adjustment to a weight w_i is :

$$\Delta w_i = -\epsilon \frac{\partial C_{SE}(x)}{\partial w_i} \quad (4)$$

On the figure 1, the following weights have to be adjusted :

- the weights W_i make the discrimination between clusters,
- the weights W_{ji} make the discrimination between the class C_j and the others within the cluster i.

Suppose that the current pattern belongs to the class C_1 which is included within the cluster 1; then the squared error associated to this pattern is :

$$C_{SE}(x) = \frac{1}{2}\left\{(\pi_1 - 1)^2 + \sum_{j=2}^{K} \pi_j^2\right\}$$

The term $\frac{\partial C_{SE}(x)}{\partial w_i}$ for the weights W_i and W_{ji} are :

$$\left\| \begin{array}{l} \frac{\partial C_{SE}(x)}{\partial W_{1,1}} \propto -\pi_1(1-\pi_1)(1-P_{1,1}) \\ \\ \frac{\partial C_{SE}(x)}{\partial W_{ji}} \propto \pi_j^2(1-P_{ji}) \quad , j \neq 1 \end{array} \right.$$

$$\left\| \begin{array}{l} \frac{\partial C_{SE}(x)}{\partial W_1} \propto \left[P_1\left(\sum_{i \in \text{cluster1}} P_{i1}^2\right) - P_{1,1}\right] P_1(1-P_1) \quad (5) \\ \\ \frac{\partial C_{SE}(x)}{\partial W_i} \propto P_i^2(1-P_i)\sum_{j \in \text{cluster i}} P_{ji}^2 \quad , i \neq 1 \end{array} \right.$$

We find that the term (5) has a not constant sign, so there is no reason that the back-propagation will minimize the squared error C_{SE} as shown on figure 2.b .
The same computationfor the types of networks 2,3 and 4 show that this problem is overcome as shown on figure 2 and on table 1.

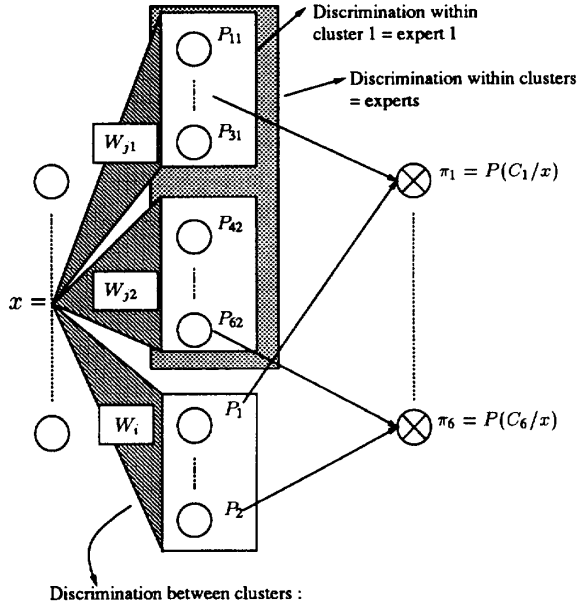


Figure 1 : Case of two disjoints clusters composed of classes $(C_1, C_2, C_3), (C_4, C_5, C_6)$ respectively.
$P_i = P(\text{cluster i}/X)$
$P_{ji} = P(\text{class } C_j/\text{cluster i}, X)$
$P(C_6/X) = P(C_6/\text{cluster 2}, X) \times P(\text{cluster 2}/X)$

APPROACH 2 : NON COOPERATIVE LEARNING
In the precedent paragraph, we had a single network to train : it was composed of many experts (see figure 1). Now the experts are trained independently. So in the place of the big network on figure 1, we have three little netwoks to train independently; after that their outputs are combined to obtain the final output $\pi_i(x)$:

- first expert : discrimination into cluster (C_1, C_2, C_3);
- second expert : discrimination into cluster (C_4, C_5, C_6);
- third expert : discrimination between the two clusters.

With the non cooperative learning, the training within a cluster is independent of the learnings in the others clusters : this means that we have to create as many subsets of training data as there are clusters. But on the test set, all experts have the same inputs. Thus, we have just increase the complexity of the training phase.
One interest to have clusters is that the discriminations within different clusters do not influence each other, so the modular approach without cooperation is very appealing. By chance, the use of the Gibbs function as output non-linearities and the use of the relative entropy as criterion, permit to construct a single network with the experts (like for the cooperative approach) and we have the advantage of the independance between experts.

Remark 2 : for the association 4 (Gibbs + relative entropy), when we compute the terms $\frac{\partial C_{RE}(x)}{\partial w}$ for the weights W_i and W_{ji}, we note that these terms are independent of the approach. Thus this association has two advantages :

- permits to use a network like the one used for the cooperative approach (figure 1) : thus we have only one data base;
- the experts which composed the network on figure 1, are independent.

As before, we are looking for appropriate output non-linearities and for an appropriate criterion for the experts.

Simulations : the second line of the table 1.
It appears that :

- first : the association (logistic + squared error) exhibits recognition rates which depend on the approach (1 or 2) : a modular approach with cooperation between modules gives bad rates; whereas a modular approach without cooperation gives good rates;
- the others recognition rates are nearly independent on one hand of the approach (1 or 2) and the other hand of type of network (2,3 or 4).

Thus for the data bases we have considered, except for the association (logistic + squared error), the main advantage of the non cooperative approach over the cooperative one, is to have experts which learn independently. But we have to create many data bases.

MODULAR OR NOT ? It is well known that the quality of the learning is very dependent upon the normalization of the data : centering and variance [6]. So we have studied the recognition rates and the speed of convergence for different normalizations :

1. the data base on its whole is centered and has unitary variance;
2. each module does a discrimination that is limited to a cluster which is centered;
3. like 2, but each cluster has unitary variance.

On figure 3, we have represented the recognition rate and the squared error on a test set (artificial data) when we use the Gibbs' function as output non-linearities and the relative entropy as criterion (see remark 2). The curves on the figure 3 represent three cases :

1. the usual approach : a single MLP, without hidden layer. The data base is considered on its whole and is centered;
2. like 1, but with a hidden layer (12 neurons).
3. the modular approach with clusters which are not centered but the data base on its whole is centered;
4. the modular approach with centered clusters.

We see that the fourth approach gives the best results on the test set as we expected : on the one hand convergence is faster, on the other hand the recognition rate on the test set is equal or better.
Moreover, in the example (figures 3 and 4), the usual approach (a network with 12 hidden neurons) gives worse results than the modular approach with experts without hidden cells.

When the discrimination within clusters are of very different complexity, the modular approach permits to adapt the complexity of the experts to the complexity of the local discrimination. So, if there is a local problem of discrimination, in the usual approach, it is necessary to use a complex single neural network to solve this local problem : then overfitting may appear elsewhere in data space.

4 CONCLUSION

We have shown that the modular approach leads to recognition rates which can be better than the results obtained by the usual approach (a single network without information about clusters) : this difference may become large (for example when the classes are far away from the centroid of the data base (figures 3 and 4)).

To the question about the coopperation between the experts we have seen that :

- if the output non-linearities are the logistic function and the criterion is the squared error, then the recognition rates become very poor in the case of a modular approach with cooperation between modules. However the use of other output non-linearities and other critera permit to overcome this problem in the case of cooperation between experts;
- in the case of no cooperation between experts the recognition rates are nearly similar;
- if there are not cooperation between the experts, it is necessary to create many data base (one for each cluster). But with cooperation we have only one data base because these experts compose a single network that has to be trained.
- with the conditions (Gibbs + relative entropy) the two approaches (with or without cooperation between experts) are identical (see remark 2);

Further research : In this paper we have supposed that the clusters are known. For some applications, it is not the case; so a new question appears : what is the best clustering ? Or equivalently, how to make the clustering for simple modules give classification's error near the Bayes' error ?

TABLE 1 - L : learning rate. T : test rate.
Comparisons of 3 learnings (1,2 and the usual one)
For the four types of networks.
For the test set (1200 patterns), the confidence on the results is about 1.4%
for the training set (400 patterns), the confidence on the results is about 1.7%

output non linearities	logistic				Gibbs			
cost	squared error		log-likelihood		squared error		relative entropy	
	L	T	L	T	L	T	L	T
approach 1	50.5	49.6	86.5	82	87.5	82.3	87	83.1
approach 2	87.5	83	87.2	82.75	86	82.75	87	83
usual approach	85.5	82	85.5	82.2	85.7	82	85.5	82.2

References

[1] J. Bridle. 1989 Probabilistic interpretation of feedforward classification network outputs, with relationship to statistical pattern recognition. In J. Herault F. Fogelman, editor, Neurocomputing: algorithms, architectures and applications. NATO ASI series on systems and computer science. Springer-Verlag.

[2] G. E. Hinton, J. L. McClelland, and D.E. Rumelhart. 1986 Distributed representations.
In Parallel distributed processing: Explorations in the microstructure of cognition., volume I. Bradford Books, Cambridge, MA.

[3] R.A. Jacobs and Jordan. A competitives modular connectionist architecture. 1991 In Touretzky Lippman, Moody, editor, Advances In Neural Information Processing 3, 767–773. Morgan-Kauffman.

[4] R.A. Jacobs, M. Jordan, S.J. Nowlan, and G.E. Hinton. 1991 Adaptives mixtures of local experts. Neural Computation, 3, 79–87.

[5] S. Kullback. 1959 *Information Theory and Statistics*. John Wiley and Sons, New York.

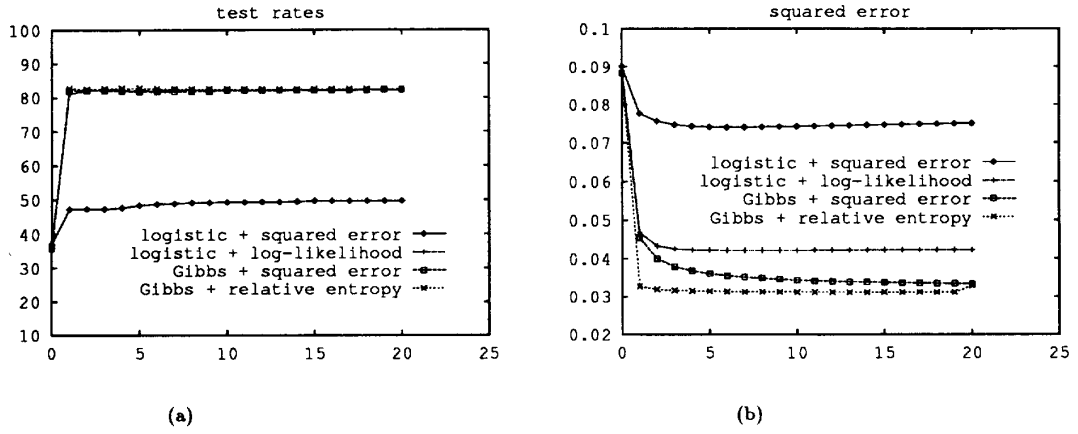
[6] Y. Le Cun, Ido. Kanter, and S. Solla. 1991 Second order properties of error surfaces : Learning time and generalisation. In Touretzky Lippman, Moody, editor, Advances In Neural Information Processing 3. Morgan-Kauffman.

[7] J. Moody and C.J. Darken. 1989 Fast Learning in Networks of Locally-Tune Processing Units. Neural Computation, 1, 281–294.

[8] S.J. Nowlan and G.E. Hinton. 1991 Evaluation of adaptative mixtures of competing experts. In Touretzky Lippman, Moody, editor, Advances In Neural Information Processing 3, 774–780. Morgan-Kauffman.

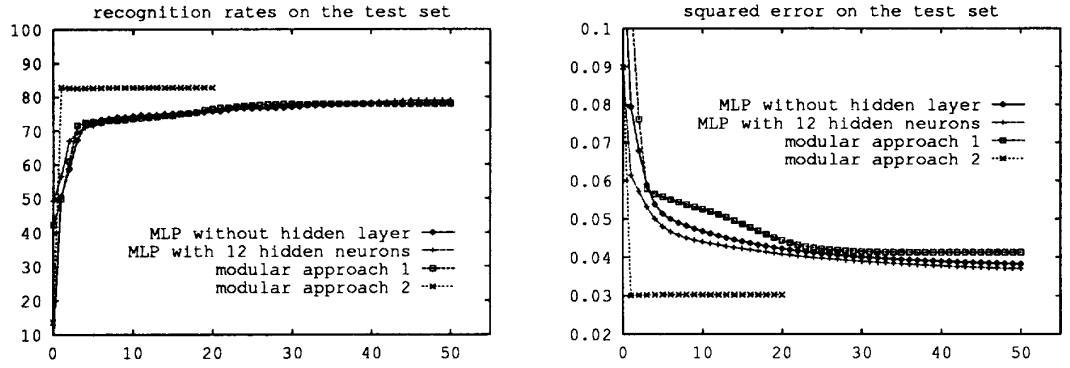
[9] M.D. Richard and R.P. Lippmann. Neural network classifiers estimate bayesian a posteriori probabilities. Neural Computation, 3, 461–483, 1991.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986 Learning internal representations by error propagation. In Parallel distributed processing: Explorations in the microstructure of cognition, volume I, 318–362. Bradford Books, Cambridge, MA.



(a) (b)

Figure 2 : Results on the test set for the approach 1 : modular approach with cooperation.
We see that the configuration (logistic as non-linearity for the outputs of the modules and squared error as criterion on the final outputs) leads to worse case : no learning.



(a) (b)

Figure 3 : Recognition rates on test set : comparisons between the modular and the usual approachs.
Modular approach 1 : the clusters are not centered
Modular approach 2 : the clusters are centered
We see that the use of the modular approach and the use of clusters which are centered improve greatly the generalisation. The data base used is plotted on the figure 4.

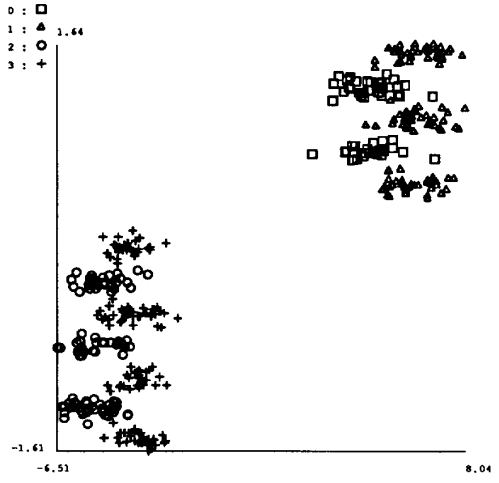


Figure 4 : Base of 4 multi-modales classes which form 2 clusters far away from the centroide of the data base.
This data base has been used for the simulations plotted on the figure 3.