A MIXED QUALITATIVE-QUANTITATIVE CLASSIFICATION METHOD APPLIED TO DIAGNOSIS

J. Carlos Aguado Chao
ESAII-UPC, jaguado@esaii.upc.es
C/Pau Gargallo 5
08028 Barcelona, Spain
Joseph Aguilar Martín
LAAS-CNRS, aguilar@laas.fr

Both authors belong to the Associate European Lab. on Intelligent Systems and Advanced Control (LEA-SICA).

Abstract

In this paper a self-learning classification method, able to deal with both qualitative and quantitative data, will be presented. This technique presents important advantages with respect to classical approaches, both numerical and symbolical ones: it is faster, less memory demanding, performs a sequential treatment and can carry out both supervised (non-iterative) and unsupervised learning. Besides, the hybrid connectives based methods yield fuzzy results easy to assess and understand. The first implementation of these classification algorithms, *LAMDA*, exhibits some interesting capabilities, since the fact of being an inductive classification method with numerical processing abilities places it in a somehow hybrid place. Our technique seems very close to the artificial neural network operation, but presents some advantages over it for classification purposes.

This method has been recently implemented as a software tool (offered as shareware), and its capabilities will be illustrated with two examples of process diagnosis, the first one on real data from a pilot plant and the second one on simulated data modeling the behaviour of a communication network.

1. LOGICAL CONNECTIVES BASED ALGORITHMS

Our proposal to Machine Learning is a kind of inductive techniques that combine some of the best characteristics of both symbolic and numerical algorithms. In order to do so, we rely on the generalizing power of Fuzzy Logic [Zadeh 65], and the interpolation capability of logical hybrid connectives [Piera 87].

Hybrid Connectives

By linearly compensated hybrid connective we will mean the interpolation between a t-norm and a t-conorm by means of the β parameter such that $\beta = 0$ represents the intersection and $\beta = 1$ means the union. This parameter will -inversely- determine the exigency level of the classification, so it can be called *tolerance*.

1.1 An Implementation: *LAMDA*

The implementation of this possibilities assumed the form of *LAMDA* algorithm, which was developed by Josep Aguilar in collaboration with a series of authors (Ramon López de Mántaras, Núria Piera, ...) [Aguilar 81] [Aguilar 82][Desroches 87] [Piera 89] who from the eighties have been enhancing this original self learning classifying technique. The input data to this algorithm are in the form of a set of observations or individuals recorded as rows of a text file, each of them divided in different values, one for every descriptor or column. Every descriptor can be either quantitative or qualitative and, in this last case, each possible value is called attribute or modality of the descriptor.

The information treatment is very simple: in a first step every modality is acquired for the qualitative descriptors, and the range of values exhibited in the quantitative ones is determined for normalization purposes. A second stage, as soon as the universe of the data is known, consists in either loading the whole population to be treated or reading and treating the individuals one by one in an indefinite series.

From this moment on, a triple alternative is offered: we can classify the individuals according to a known and fixed set of classes, or we can start from a given classes set which can be modified, or even we can start from zero and construct our own partition. Of course the last possibility is by far the most interesting one, as it comprehends the specific features of the other two and still adds up some other new characteristics.

Anyway, the basic assignment of an individual to a class follows the same process in spite of being or not the prelude to a possible modification or creation of a class. The schema is shown in figure 1.2.1:

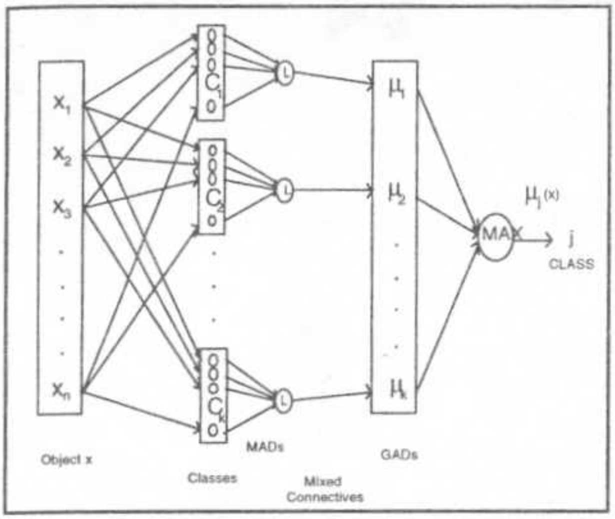

Fig. 1.2.1

where MAD and GAD stand for Marginal and Global Adequacy Degree, respectively, of an individual to a given class. It should be noticed that classifying structure remarkably resembles that of a single neuron in an artificial neural network, as can be seen in section 1.5.

Not amazingly there exist, as in the neural operation, a training stage when classes can be modified and created, and a pure pattern recognition step when we just want to assign individuals to fixed classes. Nevertheless these stages can be compatible and learning could go on forever.

1.2 Assigning individuals to classes.

Obviously, in order to compute the adequacy degree of an individual to a class we need both to be coded as a function of the same descriptors set. That means that in fact a marginal adequacy degree (MAD) will be calculated for each individual, every class and each descriptor, and these partial results will be aggregated in order to get a global adequacy degree of an individual to a class (GAD). The simplest way to build this system would be by using probability distributions and functions, and aggregating them by the simple product, but that would force us to impose a series of hypothesis on the data distribution and independence which are too arbitrary.

As a consequence, fuzzy possibility functions, derived from probability ones, are applied, and we estimate on either a parametric or a non-parametric basis --with or without imposing an *a priori* distribution-- their parameters in order to describe the classes. This means that *LAMDA* is a statistical criteria based inductive technique, but generalizes this kind of algorithms by using fuzzified distribution functions and combining them linearly by means of hybrid connectives.

In the standard implementation of *LAMDA*, our fuzzified functions are real distribution functions, except for a normalization constant that we can leave out. In the case of qualitative descriptors our estimation is exactly the maximum likelihood one for a multinomial distribution, but considering quantitative descriptors we generalize the binomial distribution for a experiment where every intermediate result between "success"(1) and "fail"(0) is possible. In order to estimate these distributions we only need to store as the kernel of a class, when qualitative descriptors are concerned, the list of observed frequencies for each modality and, when the descriptor is quantitative, the average value among the members of the class. From this kernel we obtain the marginal adequacy degree for a given individual, class and descriptor; if the descriptor is qualitative, this degree is just the frequency of the observed modality in the class, on the other hand, for a quantitative descriptor, j, described in the class C_k by the parameter ρ_{kj}, when the observed value is x_j we calculate:

$$MAD(x_j, C_k) = \rho_{kj}^{\,x_j}(1-\rho_{kj})^{1-x_j}$$

where this function represents a non-parametric Bayesian estimation. This is the classical possibility function – derived from a binomial probability– employed in *LAMDA*, but the algorithm is general enough to employ any other one.

A second function was recently proposed by Julio Waissman (personal communication). by observing data around 0.5 and questioning why "success" always had the meant of observations around 1 --by symmetry, observations around 0 were also correctly considered. *LAMDA* builds classes from observed data, and that implies that a true "success" is observing a new individual close enough to the average of the class. Let us assume that we want to classify points in a n-dimensional space from knowing their coordinates; then each class can store the coordinates of the *center* of the class, which are the average of the assigned points. Thus the degree of success can be represented by the inverse of the distance of a new point to this center:

$$d_{kj} = \left|x_j - c_{kj}\right|$$

$$MAD(x_j, C_k) = \rho_{kj}^{\,1-d_{kj}}(1-\rho_{kj})^{d_{kj}}$$

We can consider this new function as a fuzzyfication of the law that already generalized a probabilistic distribution, because a new parameter can now range from 0 to 1. Within the developed software, the user has the possibility of working with the classical function of *LAMDA* or --default option-- this new one, or else, why not, an assumed normal --Gaussian-- distribution:

$$MAD(x_j, C_{kj}) = \frac{1}{\sigma_{kj}} e^{-\frac{1}{2}\frac{(x_j-\mu_{kj})^2}{\sigma_{kj}^2}}$$

Now the hybrid connectives appear, not restricted to the product of independent possibilities, but including it [Zadeh 65] [Zadeh 71] [Schweizer 83]. This means that there exist a wide variety of families of fuzzy connectives from where we can choose, and also we are able to select their tolerance and combine according to it a pair of a t-norm (intersection) and a t-conorm (union) [Sklar 77] [Piera 87] [Piera 88].

By proceeding this way, a global adequacy degree is calculated, for an individual to every class. Of course this GAD will rule the choice of the class for the individual to be assigned to. At this point of the classification process, most standard algorithms employ a minimum threshold which the membership needs to trespass in order to provoke an effective assignment; usually the user is expected to provide this threshold, but *LAMDA* is different, for it holds a Non-Informative Class, NIC, automatically constructed for a given universe, so as to model the maximum entropy, which means returning the same GAD for any individual. Whenever the object that is to be classified does not possess a membership for another class greater than its membership to the NIC it can not be assigned, otherwise the object will be assigned to the class which exhibits the maximum GAD, and the first found one if two or more of them produce the same result. The NIC always remains empty.

1.4 Modifying and creating new classes

Obviously the representation of a class is expected to vary after a new individual has been assigned to it. The part of the kernel which describes the quantitative descriptors parameters will recursively change to collect at least the mean of the observed values, and other parameters if necessary. It is clear that only the class where a new individual is assigned will vary.

The next question is considering which ones could be the more appropriate classes in order to perform a particular classification, and whether *LAMDA* is able to automatically construct them just the same as the NIC is constructed. The answer is yes; our algorithm can either learn its own classes or accept the ones provided by the user. In fact the Non-Informative Class plays a major role in the creation of any other class.

As it has been said before, an individual is assigned to the class for which its Global Adequacy Degree results greater. In the special case when this class turns out to be the NIC, that means that the individual is not close enough to any of the existent classes. Thus, two actions are possible, if we are carrying out a pure pattern recognizing, this individual is to remain unclassified, but, on the other hand, if we are on a learning stage, this situation implies that we need a new class which is suitable for the individual. The creation proceeds by temporally assigning the individual to the NIC and modifying it so as to form a brand new class; this process, however, in order to construct a class which is between the NIC and the new individual, treats the NIC as it held one individual instead of none. Otherwise, the newly created class would be so fixed on its only individual that *probably* will not accept any other, and that is bad politics for a learning stage. The current implementation of *LAMDA* keeps always this virtual element --taken from the NIC-- additional to every class, so as not to arbitrarily reduce only the weight of the first arriving element.

1.5 Parallelisms and differences with respect to other classifying algorithms

Let us recall some of the characteristics of the classifying methods we have reviewed. Symbolic classifiers usually work with qualitative bivalued (true-false) descriptors, need some parameters arbitrarily supplied by the user and rely on some heuristic. Inductive reasoning usually needs supervised learning and is generally slow and always falsehood preserving. Deductive and analogy based reasoning, on the other hand, need to slowly work with a huge amount of data.

Numerical methods produce results difficult to understand and work only with quantitative descriptors. On the other hand, they perform either supervised --much more commonly-- or unsupervised learning for a given structure, but not both of them, and are iterative and slow methods where there is no clear mark of 'learning finished'. Moreover, artificial neural networks, the most popular among them, are highly unpredictable tools.
LAMDA's parallelisms with these techniques are quite obvious. *LAMDA* is an inductive method, with classification criteria generalized from Statistics and can behave as a single neuron in some particular cases. In fact, its classifying structure --see figure 1.2.1-- remarkably resembles that of a single neuron in an artificial neural network, as can be seen in figure 1.5.1.

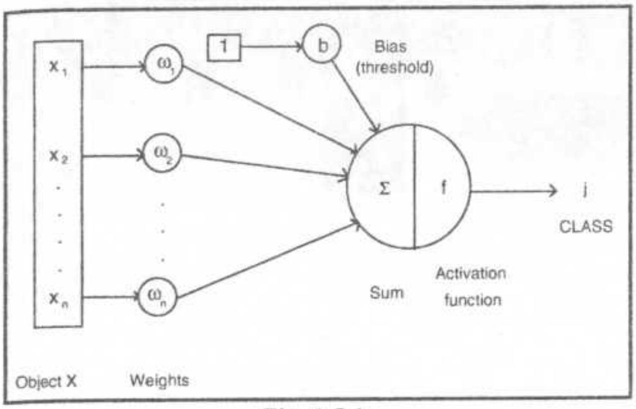

Fig. 1.5.1

Here the class description (kernel) plays the role of the neuron weights and, consequently it is expected to vary in a learning stage. If we use neurons to classify individuals, we can be interested in finding the maximum output signal, which is computed as:

$$\max_j (f(b_j + \sum_i x_{ij} \cdot w_{ij}))$$

where b_j is the bias of every output neuron, f the activation function, x_{ij} the different inputs and w_{ij} the correspondent weights that are determined in the training stage. That can be perfectly equivalent to a particular case of *LAMDA*, which can assign an individual to the class that yields:

$$\max_j (\prod_i \rho_{ij}^{x_i} (1-\rho_{ij})^{(1-x_i)}) \equiv \max_j (\sum_i (1-x_i) \log(1-\rho_{ij}) + x_i \log \rho_{ij})$$

where x_i are the different values for each descriptor and r_{ij} the parameters for every descriptor and class that are determined in the learning process. It is worth noticing than in a neuron the biass is independent from the weights, meanwhile it is not so in *LAMDA*. Nevertheless, by using other possibility functions (Waissman's, Gaussian ...) the additional independent parameter appears. However, the operation is absolutely different. *LAMDA* does not search for minima in unknown functions, our learning process is not the result of "blind" weights combinations, but a coherent modification of the classes description. This description results easily understandable because is nothing more that the list of observed modalities and their appearance frequencies, for every qualitative descriptor, and the average value observed for every quantitative descriptor in the individuals belonging to this class.

On the other hand, our algorithm, not based on some heuristic but on generalized statistical (possibility) functions, does not need a minimum user supplied minimum threshold, which is to be trespassed so as to assign an individual to class, as is the case in most of the statistical classifying algorithms. In *LAMDA* this minimum adequacy level is automatically determined as a function of the database, and its value is that of the global adequacy for any individual to the maximum entropy class, the NIC.

Additionally, the whole assignment process is not a classical statistical one, but fuzzy, thus more general. Additionally the overall exigency can be parametrized in order to produce every possible coherent classification for a given pair of logical connectives and a universe. Of course, when the user has enough information he/she can fix this exigency but *LAMDA* also can automatically explore the different values that yield every possible partition [Piera 91].

Another major advantage of *LAMDA* with respect to usual classifying techniques is its sequential data treatment. We only need to store the present classes and deal with the individual that is to be classified, unlike the case based reasoning (CBR) algorithms which need to store a massive stock of information from the already processed individuals. That means that *LAMDA* is faster and demands but little memory; the counterpart is its dependence of the order in which the observations are provided. Nevertheless, we humans are not less order dependent in our learning processes.

And finally, the implemented algorithm also includes a post classification analysis of the obtained fuzzy partitions. Although the final result always is presented as a classical --crisp-- partition *LAMDA* internally operates on a fuzzy subjacent partition that can provide some valuable information about the conceptual vicinity among either classes or individuals. [Aguilar 90].

2. APPLICATION TO DIAGNOSIS

The possibilities of this method have recently been implemented by the authors of this paper, as a software tool on Windows. There is notice of its existence in the specialized web page www.kdnuggets.com and the software is publicly available at the site ftp.upc.es. This particular program has already been applied to a couple of diagnosis problems to show its capabilities.

2.1 Diagnosis of a tank system

In 1996 a pilot plant consisting of two liquid tanks was built by ALECOP in order to use it as an experimental laboratory platform. In our experiment, we start from an electrical pump that brings liquid in the first tank, an electrovalve that lets it flow to the second tank, and a final electrovalve that allows the liquid exit from this latter tank. We also have sensors to acquire the level on each tank, and a control system able to regulate the level in the second one by means only of measuring it and actuating on the electropump. More specifically, the system can regulate the level in the tank 2 following step-like orders, if their changes are slower than 5 times the time constant of the whole system. That constant turns out to be of 150 seconds, and the control system is robust enough to cope with perturbations of 5 %, simulated in our case by varying the pump tension.

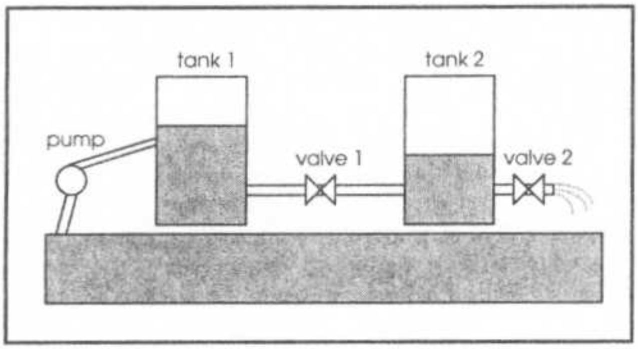

Fig 2.1.2

The normal operation of the system has just been described, but our main interest is to supervise it, and detect when some malfunctioning appears. So as to achieve this ability, we assume two possible breakdowns to be studied: a slight one, that can be handled by our control system, and a critical one, which leads the whole process to an emergency state where all operation should be stopped at once. The former possibility is illustrated by a partial obstruction in the first valve, the one that links together the two tanks, meanwhile the latter is represented by a total obstruction of the second electrovalve, situation that would eventually provoke an overflow of the tanks.

In this context, the aim of our classification algorithm is to distinguish whether this system of liquid level control is in normal operation, or suffering from a small perturbation or in an emergency state. Let us forget everything about the underlying control system, except the overall time constant of the level in tank 2, which is 150 seconds.

The necessary data in order to assess the system operation is acquired by forcing 12 different situations, resulting from the three states we want to distinguish and four different possibilities of orders to the control system: maintain high the level in tank 2, keep it low, shift it from low to high or drop it from high to low. Both latter orders have been scheduled to present their changes after two times the global time constant of the level in tank 2, that is, 300 seconds.

By following these directions we obtain 12600 records of data, since for every of the 12 possibilities we measure the behavior of the system each second and during seven times the overall constant (1050 seconds). The first information that we would think of as valuable to assess the global operation would be the level in tank 2, its desire value and the control signal that the regulator produces. Nevertheless, it seems too poor an amount of information in order to supervise a dynamic system, and everything suggests that we need the time to be involved in our measures.

So as to grasp this fundamental contribution of time, we pick up, by means of a histogram, the approximate derivatives of the tank levels and of the control signal. So, the maximum amount of data that we can measure in this system is expressed by seven descriptors: measures of level in tank 1 and 2, control signal, order of desired level in tank 2, and qualitative derivatives of the first three quantities. Among these measures, those related to the tank 1 were obtained in order to know better the overall functioning, but in a realistic assessment it would be highly preferable to work without them. But let us start by the beginning; in this case this means that when facing a problem with a new technique, it makes sense to start with the easiest case and maximum information, to try later a process of progressive difficulty and less data, as long as successful results allow it.

Thus, our first database consists of 12600 lines divided in seven spaces. The first three of them are numerical quantities between 0 and 10, the fourth is a qualitative descriptor than can be either high or low (H or L), and the three last descriptors, collecting the approximate derivatives, present the values PL, PS, ZE, NS and NL, that is, positive, zero or negative values, either large or small.

There is a question that has to be answered before beginning. A scientific test of this technique would include the comparison of results with other methods, but there are some reasons not to do so. First, as far as we know, there do not exist classification tools able to simultaneously deal with quantitative and qualitative data. The only ones able to approximate this behavior are purely numerical ones, and among them their most famous representatives are the artificial neural networks. On the other hand, even if a neural network could theoretically be found, there are no general methods to do so, and no indication of the needed architecture or the necessary training time. It would be a blind search, and eve if by fortune we found such a network, it would exclusively be useful in supervised or unsupervised learning and not in both modes, and the obtained result could not be interpreted. Finally even a trivial case of pseudo-recognition --where a class is created for every individual and then merged into the desired ones-- could go unnoticed.

Anyway, although with a rather pessimistic attitude, some efforts have been done in order to find a neural network able to recognize the states of the tank system. Trying a three-layered neural network, with half a dozen neurons in the hidden layer and the backpropagation with momentum learning algorithm, has not shown any positive result. Of course the search could go on forever, but this is a typical weakness of neural networks.

It is important to remark that the information is acquired from the real system and not simulated ant that implies the necessity of some filtering and preliminary treatment of the data. As any approach to the study of a real process, it is wise to apply a step by step experimentation. At the beginning, a simple test can determine the potential of the technique: in our case we tried a supervised learning from nearly all the available information and then a pattern recognition in a very reduced number of individuals. The global result had more than a 83% of successful recognitions, and what is more important, every observation in the emergency state was correctly classified.

Once it was clear that the algorithm was adequate to solve the problem, more realistic approaches could be tried. In a second experiment, only a 60% of the individuals were used to build the classes, meanwhile the 40% of them were employed to perform a pattern recognition. The degree of success was in that case higher than 74% and every emergency state observation was correctly recognize, but there were four false alarms of individuals incorrectly considered as belonging to that emergency class.

A third experiment restricted more the available information and work only with data from the pump, desired output and level in the second tank (dropping the level information from the first tank). Now the degree of successful recognitions was a 73.8%, only 1 false alarm was generated in the emergency class, but it took a one-sample delay to recognize the emergency state.

In order to better understand the process, a unsupervised learning was tried, with the results of more classes generated. If we allowed 18 classes the degree of success was nearly 78% with only 2 false alarms in the emergency state, meanwhile by allowing only 6 classes we received a success degree of nearly 76% and 3 false alarms in the third class. Everything seemed to indicate that it was difficult for the algorithm to group together individuals that had an obvious difference: a desired output high or low. So it could be wise to work with 6 classes instead of only 3.

The fifth experiment follow this suggestion and the results could reach again higher than a 74% with only 2 false alarms in the emergency states but maintaining the delay in the detection.

Finally, a last experiment tried to drop the temporal information of the system and working only with real time measures was able to obtain a 76% of successful recognitions, without delay in the emergency state but 7 false alarms, meanwhile the faulty class remained nearly empty. Of course, we could have reached higher success degree for instance trying a 6 classes pattern recognition from all the available information (more than 85%), but it would not be a realistic experiment.

2.2 Diagnosis of a communication network

Thanks to a recent stay in the Temple University of Philadelphia, we contacted Dr. Saroj Biswas, and gained access to one of the diagnosis problems he is currently working in. That research deals with the fault management of communication networks, more specifically the NASA communication network NASCOM. A fault is an unpredictable change in the network that causes a deterioration of performance. In order to maintain reliable data transmission, it is necessary to detect this situation and take corrective actions by reconfiguring the network topology and rerouting he data transmission. More importantly, the detection and isolation of faulty components must be done in real time or near real time so as to minimize the loss of data packets and the network down time.

Preliminary research on failure detection using artificial neural networks has been an ongoing project since the summer on 1994 under the NASA/ASEE Summer Faculty Fellowship program. As part of the previous research conducted at the Goddard Space Flight Center and Temple University, a Failure Detection and Isolation (FDI) neural Network has been developed for real time detection and isolation of link faults. This FDI neural network analyzes the network operational data obtained from the network Management Information Base (MIB), isolates the faulty link and determines the level of severity of the fault. Initial results on the performance indicate a success rate of 70% in detecting and isolating link faults.

Recently the continuation of the previous research has started, with the broad objective of developing a real time failure detection, isolation, and prediction methodology that can be implemented on the NASCOM network. This research is carried out by Dr. Biswas from the Temple University, and Brian Drake, Project Engineer for the FARM project. Because of the nature of the project, the use of any NASA communication hardware is not expected, for the security sake.

As a first step a simulated network that ressembles NASCOM is being utilized in order to generate data from both correct and malfunctioning operations. This model has been developed from queuing theory equations, implemented in MATLAB and validating by COMNET III.

The model of every node is the result of the interaction of different queues to serve the incoming traffic from other nodes, the local input of messages and the final distribution to other nodes:

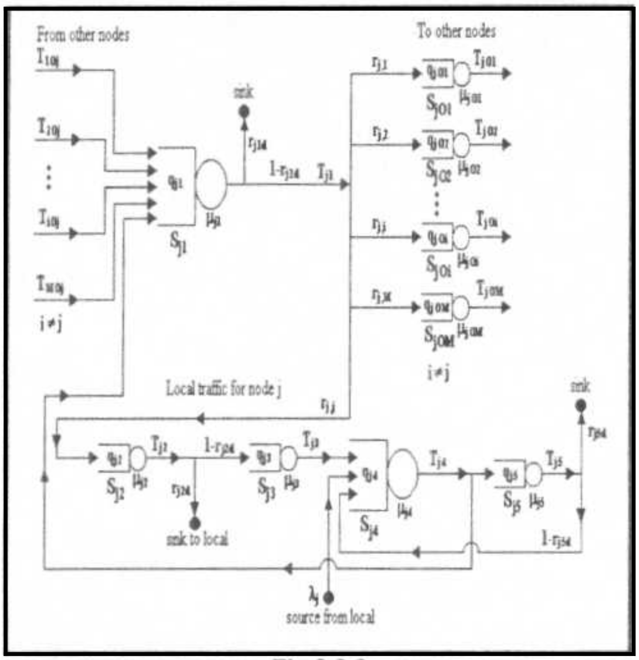

Fig 2.2.2

This structure can raise three classes of failures, which are link, node and local malfunctioning. In order to detect them, there is a considerable amount of information to be monitorized, mainly the packet flow or throughput, the different buffer lengths, the packet transmission requests and maybe their rate of change, time derivatives.

The first generated data from the simulated network corresponded to 75 individuals, each of which collected the state of the network in a moment of time. This network was assumed to have 12 nodes and 40 links between them, and that implies to work with 76 descriptors: the length of the 12 input queues, the 12 transmission requests queues, the 12 local buffers and the state of the 40 links. On the other hand, the solution can be coded with 64 descriptors, collecting the state of each local section, node and link.

Only a first experiment has been tried with *LAMDA*, in order to test its capability to partition the 75 individuals into three classes, corresponding to different states of only the first node, that can be in normal functioning, slightly faulty or completely down. The experiment consisted in employing 4 out of each 5 observations in order to create the 3 classes, and leave the remaining 15 individuals to be classified. Of course this experiment of supervised learning is a simple one, and must be understood only as a first approach to the solution of the problem. Nevertheless, the obtained results have been close to perfection: a similarity degree of 0.933. As it can be seen, only an individual is misclassified, and even it correspond to the slightest possible error, because the node 1 that is in normal state turns out to be erroneously considered as in slightly faulty state.

Some other valuable classifications are obtained. For instance, the distribution quantitative function F_1 and the minimum and maximum connectives, for exigency equal to 0.80327 but without imposing a recognition, has shown a similarity degree of 0.866. On the other hand, the Gaussian distribution function, for any exigency degree and the Frank connectives generates a partition that separates only the fault class and the correct one.

Anyway, these results only mean that the logical connectives based classification algorithms can easily reach or even improve the success of neural networks. Looking for a realistic solution of the network management problem will demand far more time and experimentation, as was the case in the tanks experiment presented in the previous section.

3. CONCLUSIONS

This work was divided in two parts. In the first of them, the logical connectives based classification methods were explained, and then an implementation was tested on two process diagnosis. The first implementation of the hybrid based classification algorithms, *LAMDA*, shows some interesting capabilities, since the fact of being an inductive classification method with numerical processing abilities places it in a somehow hybrid place. The derived advantages are the possibility of simultaneously working with qualitative and quantitative data, and a sequential fast and little memory demanding processing which yields easy to interpret results. But there are more benefits, for instance the fuzzy philosophy that produces more complete results, and the existence of a maximum entropy class that automatically performs some of the functions that other algorithms request from the user. Our technique seems very close to the artificial neural network operation, but presents some advantages over it for classification purposes.

In order to proof so, a modular, flexible and user-friendly learning environment has been designed, for a number of platforms including Windows 98/NT/95, on an object-oriented basis. Once a partition and its parameters are deemed acceptable, the data can be transferred to some real-time pattern recognition modules. For this modules, contrarily to the learning environment, it has been preferred to work on a highly popular programming platform, MatLab, in order to guarantee the maximum extension to this modules and the possibility of any user to modify them. Accordingly to this philosophy, the code of the programs has been provided and commented, so as to shine some light on the inner details of the implementation. The executable code of the learning environment, the real-time pattern recognition modules, some explanations and example data are available in the public domain FTP site ftp.upc.es, and its existence is publicized in the specialized web page www.kdnuggets.com.

In the second and last part of the work, two application processes have been considered. First, real data from a pilot plant were provided: the system consisted in a control system of the level of liquid in the second of two tanks linked together, by means of a pump, a connection valve and a final exit valve. Our goal was the diagnosis of this system, which could be either in normal operation, in a faulty state where the intermediate valve was partially obstructed or an emergency state with the exit blocked. It has to be considered that there existed an inner control system able to correct the partial obstruction, thus making its detection far more difficult. As any real application, the study of the process was carried out step by step, and in this case six consecutive experiments were performed, which lead us to learn more and more about the process. The final, realistic results, using a 60% of the available data to create the classes and 40% of them to carry out a recognition were around a 77 % of global success, but every observation in the emergency state was correctly detected, maybe with a delay depending on the available information, and only a small number of false alarms were generated.

Finally, another diagnosis has just been sketched, since it is still in process. The supervision of a communication network has been proposed, and working on simulated data from a queue model our algorithm has shown a preliminary success rate of 93%, in front of classical neural network approaches that achieved a 70 %.

Acknowledgements

We want to specially thank Ramon Serrate, from the Polytechnical University of Catalonia, and Dr. Saroj K. Biswas, from the Temple University of Philadelphia, for providing the valuable process data that were used in the application part.

REFERENCES

[Aguado 98] *A Mixed Qualitative-Quantitative Self-Learning Classification Technique Applied to Situation Assessment in Industrial Process Control.* Ph. D. dissertation in the Polytechnical University of Catalonia. December 1998.

[Aguilar 81] *Estimation Recursive d'une Partition, Exemples d'Apprentissage et Auto-Apprentissage dans* R^N *et* I^N. J. Aguilar-Martín, M. Balssa and R. López de Mántaras. Qüestió. Vol. 5, nº 3, pp. 150-172. September 1981.

[Aguilar 82] *The process of classification and learning the meaning of linguistic descriptors of concepts.* J. Aguilar-Martín and R. López de Mántaras. Approximate Reasoning in Decision Analysis. pp. 165-175. North Holland, 1982.

[Aguilar 88] "Probabilistic and fuzzy relational semantics systems in propositional approximate reasoning", 18th Intern, Symp. on Multi-Valued Logic, IEE pp. 205-209, Palma de Mallorca (Spain), 1988.

[Aguilar 90] *Conceptual Connectivity Analysis by means of Fuzzy Partitions.* J. Aguilar-Martín, M. Martín and N. Piera. Information Processing and Management of Uncertainty in Knowledge Based Systems. Vol. 1, pp. 250-252 (extended abstract), 1990.

[Aguilar 96] *Data Based Inductive Knowledge Representation,* Aguilar, Aguado, Piera, pp. 1-6, proceedings of the International Panel Conference on Soft and Intelligent Computing, edited by the Technical University of Budapest, October 1996.

[Costes 84] *Application de la classification automatique avec apprentissage a la surveillance d'une centrale solaire.* V. Costes-Albrespique i J. Aguilar-Martín. RAIRO Automatique/Systems Analysis and Control, Vol. 18, nº 4, 1984.

[Desroches 87] *Syclaire: Système de classification avec apprentissage et recoinassaince de formes. Manuel d'utilisation.* P. Desroches. "Research report" 87/9 del Centre d'estudis Avançats de Blanes. November 1987.

[Desroches 90] *Variation points in incremental conceptual clustering.* P. Desroches, N. Piera and J. Aguilar-Martín. "Technical report" of the LAAS nº 90254. July 1990.

[Piera 87] *Connectius de lògiques no estandard com a operadors d'agregació en classificació multivariable i reconeixement de formes.* N. Piera. Doctorate dissertation in the Universitat Politècnica de Catalunya. July 1987.

[Piera 88] *Mixed Connectives of Lineal Compensation.* N. Piera and J. Aguilar-Martín. "Technical report" of the LAAS nº 88301. October 1988.

[Piera 89] *LAMDA: An Incremental Conceptual Clustering Method.* N. Piera, P. Desroches and J. Aguilar-Martín. "Technical report" of the LAAS nº 89420. December 1989.

[Piera 90] *Variation points in pattern recognition.* N. Piera, P. Desroches and J. Aguilar-Martín. Pattern Recognition Letters. Vol. 11 pp. 519-524. August 1990.

[Piera 91] *Controlling Selectivity in Nonstandard Pattern Recognition Algorithms.* N. Piera and J. Aguilar-Martín. IEEE Transactions on S, M & C. Vol. 21, nº 1, pp. 71-82. January-February 1991.

[Sklar 77] *Random variables, joint distributions and copulas.* A. Sklar. Kybernetica. Vol. 9 pp. 449-460, 1977.

[Schweizer 83] *Probabilistic metric spaces.* B. Schweizer i A. Sklar. Elsevier North-Holland, 1983.

[Zadeh 65] *Fuzzy sets.* Information and Control. Vol. 8, pp. 338-353, 1965.

[Zadeh 71] *Fuzzy sets as a base for a theory of possibility.*

Fuzzy Sets and Systems. Vol. 1