Fuzzy Descriptive Models: An Interactive Framework of Information Granulation

Giovanni Bortolan, *Member, IEEE,* and Witold Pedrycz, *Fellow, IEEE*

***Abstract*—In this paper, we introduce and discuss an important class of endeavors of fuzzy modeling, such as fuzzy descriptive models. In a nutshell, the objective of fuzzy descriptive models is to provide with a sound, comprehensible, and relevant description of experimental data at a general level of relationships revealed there. The elements of such models called descriptors are inherently information granules as the notion of granularity goes hand-in-hand with the interpretability of the resulting constructs (information granules). This paper elaborates on the use of the language of fuzzy sets that are viewed as generic models of information granules. The development of the information granules is carried out in an interactive manner in which a designer can inspect a structure in a data set in a visual fashion. Such visualization is possible through a suitable visualization vehicle provided by self-organizing maps. The role of the designer is to choose from some already visualized regions of the self-organizing map characterized by a high level of data homogeneity. We provide a new algorithm of constructing membership functions of the information granules (fuzzy sets). In addition to some synthetic data, the study includes a comprehensive descriptive modeling of highly dimensional electrocardiogram data.**

***Index Terms*—Computerized electrocardiogram (ECG) signal analysis and classification, fuzzy descriptive model, information granulation, self-organizing maps, user-interactive model development.**

I. Introduction

FUZZY modeling comes today with a plethora of architectures, algorithms, and hybrid design methodologies; cf. [10]–[12], [17], [19], [24], [25], [27], and [34]. The omnipresent and strong visibility of various mechanisms of computational intelligence (CI) [20], [40], such as evolutionary optimization [29] and neurofuzzy modeling and optimization [30], [35], [41], is the dominant feature of the area. The use of clustering techniques becomes also more visible nowadays; cf. [1], [2], [6], [9], [21], [26], [31], and [32]. In spite of this diversity, most of the resulting fuzzy models become surprisingly close each other as far as the underlying objective is concerned: all of them tend to approximate data (predominantly numeric) to the highest possible extent. The original agenda and an evident strength of fuzzy sets that comes with the transparency of their granular constructs (that is fuzzy sets and fuzzy relations) do not seem to be fully explored. To the contrary: we tend to compete with purely numeric models especially neural networks are going to do a superb job. One may even wonder what would make fuzzy sets superior in this type of competition. Having noticed the evident quest for accuracy of the models, one may consider two classes of fuzzy models and the ensuing modeling principles and development methodologies, that is

- *Fuzzy descriptive models*: These models are aimed at a description of data in the language of well-defined, semantically sound and user-oriented information granules [37]–[39]. The objective is to understand the data where a certain point of view there is carefully articulated in the language of fuzzy sets (*viz.* the relationships deal with fuzzy sets). Obviously a dynamic perspective can be included that is the fuzzy sets along with their granularity modified in order to produce descriptors that are meaningful and legitimized from the standpoint of experimental evidence (data). We regard the models in this category to be a prerequisite to other fuzzy models of higher level of details. They tend to be *data noninvasive*, meaning that we do not attempt to impose a specific detailed structure on the data.
- *Fuzzy predictive models* are the models to carry out some sort of prediction. The term prediction is used in a general sense as such models can encompass a prediction of time series, output variable or a class in a classification problem. As a matter of fact, an evident majority of fuzzy models fall under this category; cf. [6], [9], [17], [29], [33]–[35], and [41].

These two categories of fuzzy models interact in the sense outlined in Fig. 1, yet their role in the entire modeling environment is quite different.

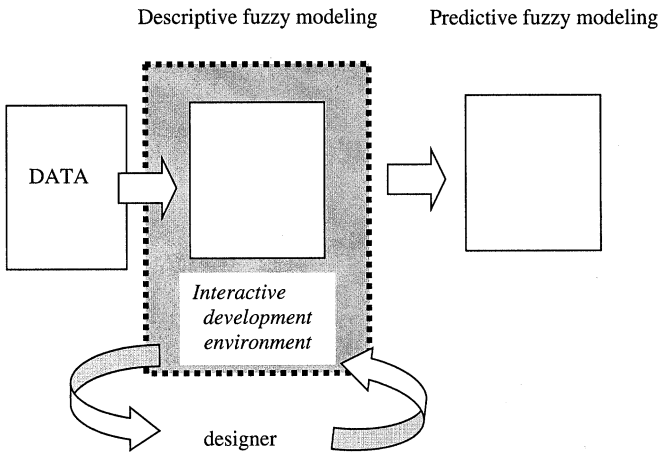


Fig. 1. Descriptive and predictive fuzzy modeling.

Manuscript received December 5, 2001; revised February 26, 2002 and April 11, 2002. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by Alberta Software Engineering Research Consortium (ASERC).

G. Bortolan is with LADSEB-CNR, 35020 Padova, Italy.

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada, and also with the Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland (e-mail: pedrycz@ee.ualberta.ca).

Digital Object Identifier 10.1109/TFUZZ.2002.805891

The data and some domain knowledge hints are combined together to form a descriptive model. Here, an active role of the designer is a must. To accomplish that one need a highly interactive, visual development environment. The results of descriptive modeling are a prerequisite to efficient predictive fuzzy modeling. As a matter of fact, we can view some constructs of the descriptive modeling to be used as building blocks of predictive modeling. In particular, this is visible when using fuzzy clusters in the design of fuzzy models; cf. [21], [27], and [32]. It is important to stress that the specific research agenda of descriptive and predictive models is very different so is the list of the fundamental pursuits. In descriptive models, we question whether the information granules (fuzzy sets) are relevant, descriptive, and suitable from the standpoint of their granularity (which helps capture the required level of details of the data). In predictive models, the primary questions deal with the accuracy of the model, its generality, robustness, etc.

It is interesting to note that this type of categorization of the models (in general sense, not necessarily fuzzy models) has become visible quite strongly in data mining and intelligent data analysis (IDE); cf. [5], [22], and [28].

This paper is devoted to the descriptive fuzzy models and attempts to address the crucial design issues, especially those concerning a formation of a highly interactive, efficient and user-friendly visualization environment. The material is arranged in 8 sections, each of them focusing on a separate facet of the descriptive modeling. In Section II, we revisit a concept of self-organizing maps (SOMs) regarded as a backbone of the user-oriented development environment. The basic idea of SOMs is augmented here by an introduction of some auxiliary maps that help visualize the structure in the data and describe its characteristics (Section III). In Section IV, we discuss a way of constructing membership functions of fuzzy sets that are a direct product of the homogeneous data regions defined by the designer in the SOM. The experimental part of the study is covered in Sections V and VI. Here, we deal with a synthetic data set and use two data sets used in the machine learning studies. Next, Section VI includes a comprehensive development of the descriptive model for electrocaridogram (ECG) data. Section VII moves the ECG data analysis further by elaborating on the links between the descriptive model and the ensuing predictive models. Conclusions are covered in Section VIII.

II. SOMs—An Insight Into a Structure of Data

The concept of an SOM was originally coined by Kohonen [14]–[16] and is currently used as one of the generic neural tools for structure visualization. There are a number of augmentations of the generic version of SOM; see [18] as well as other generalizations such as growing SOMs [8]. In this study, we concentrate on the use of the generic version of the SOM; our selection is motivated by a wealth of theoretical studies and experimental evidence collected in practice.

As usually reported in the literature, SOMs are regarded as regular neural structures (neural networks) composed of a rectangular (squared) grid of artificial neurons. The intent of SOMs is to visualize highly dimensional data in a low-dimensional structure, usually emerging in the form of a two- or three-dimensional map. To make this visualization meaningful, an ultimate requirement is that such low-dimensional representation of the originally high-dimensional data has to preserve *topological* properties of the data set. In a nutshell, this means that two data points (patterns) that are close each other in the original highly-dimensional feature space should retain this similarity (or resemblance) when it comes to their representation (mapping) in the reduced, low-dimensional space in which they need to be visualized. And, reciprocally: two distant patterns in the original feature space should retain their distant location in the low-dimensional space. Being more descriptive, SOM performs as a *computer eye* that helps us gain insight into the structure of the data set and observe relationships occurring between the patterns being originally located in a highly dimensional space. In this way, we can confine ourselves to the two dimensional map that apparently helps us to witness all essential relationships between the data as well as dependencies between the software measures themselves. In spite of the existing variations, the generic SOM architecture (as well as the learning algorithm) remains basically the same. In what follows, we summarize the essence of underlying self-organization algorithm that realizes a certain form of unsupervised learning.

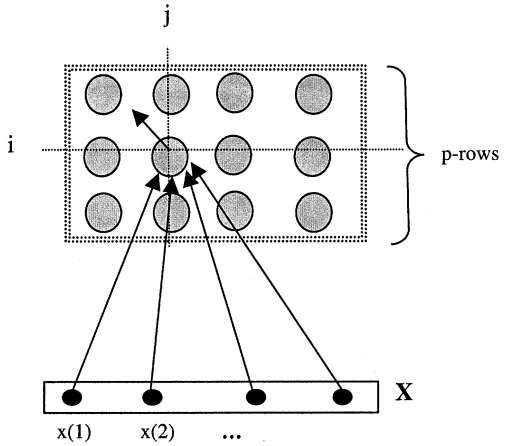


Fig. 2. A basic topology of the self-organizing map constructed as a grid of identical processing units (neurons).

Before proceeding with the detailed computations, we introduce all necessary notation. "n" feature of the patterns (data) are organized in a vector $\mathbf{x}$ of real numbers located in the n-dimensional space of real numbers, $\mathbf{R}^n$. The SOM comes as a collection of linear neurons organized in the form of a regular two-dimensional grid (array), Fig. 2.

In general, the grid may consist of "p" rows and "r" columns; quite commonly we confine ourselves to the square array of "p" $\times$ "p" elements (neurons). Each neuron is equipped with modifiable connections $\mathbf{w}(i,j)$ where the connections are arranged into an n-dimensional vector that is $\mathbf{w}(i,j) = [w_1(i,j)\, w_2(i,j) \cdots w_n(i,j)]$. The two indexes ($i$ and j) identify a location of the neuron on the grid. The neuron calculates a distance (d) between its connections and a certain input $\mathbf{x}$

$$y(i,j) = d(\mathbf{w}(i,j), \mathbf{x}). \tag{1}$$

The same input $\mathbf{x}$ affects all neurons. The neuron with the shortest distance between the input and the connections becomes activated to the highest extent and is declared to be a winning neuron—we also say that it matched the given input ($\mathbf{x}$). Let us denote its coordinates by ($i0$, $j0$). More precisely, we have

$$(i0, j0) = \arg\min_{(i,j)} d(\mathbf{w}(i,j), \mathbf{x}). \quad (2)$$

As a winner of this competition, we reward the neuron and allow it to modify the connections so that they are getting even closer to the input data. The update mechanism is governed by the expression

$$\mathbf{w\_new}(i0, j0) = \mathbf{w}(i0, j0) + \alpha(\mathbf{x} - \mathbf{w}(i0, j0)) \quad (3)$$

where α denotes a learning rate, $\alpha > 0$. The higher the learning rate, the more intensive updates of the connections. In addition to the changes of the connections of the winning node (neuron), we allow this neuron to affect its neighbors (*viz.* the neurons located at similar coordinates of the map). The way in which this influence is quantified is expressed via a neighbor function $\Phi(i, j, i0, j0)$. In general, this function satisfies two intuitively appealing conditions: 1) it attains maximum equal to one for the winning node, $i = i0$, $j = j0$, and 2) when the node is apart from the winning node, the value of the function gets lower (in other words, the updates are less vigorous). Evidently, there are also nodes where the neighbor function zeros. Considering this, we rewrite (1) in the following form:

$$\mathbf{w\_new}(i,j) = \mathbf{w}(i0,j0) + \alpha\Phi(i,j,i0,j0)(\mathbf{x} - \mathbf{w}(i,j)). \quad (4)$$

Commonly, we use the neighbor function in the form

$$\Phi(i, j, i0, j0) = \exp(-\beta((i - i0)^2 + (j - j0)^2))$$

with the parameter β (equal to 0.1 or 0.05 depending upon the series of experiments) modeling the spread of the neighbor function.

The update expression (4) applies to all the nodes (i, j) of the map. As we iterate (update) the connections, the neighbor function shrinks: at the beginning of updates we start with a large region of updates and when the learning settles down, we start reducing the size of the neighborhood. For instance, one may think of a linear decrease of its size.

The number of iterations is either specified in advance or the learning terminates once there are no significant changes in the connections of the neurons.

The distance $d(\mathbf{x}, \mathbf{w})$ can be defined in many different ways. A general class worth considering is that of the Minkowski distance. Practically, three of representatives of this family are commonly used, that is Hamming, Euclidean, and Tchebyschev. From the experimental end, the choice between these three distances is not critical and the Euclidean distance is a legitimate choice.

Dealing with raw measures poses the risk that one software measure may become predominant, simply because its domain includes larger numbers (that is the range of the measure is high). Therefore, the distance function is computed for normalized rather than raw data. In the sequel, the SOM exploits these transformed software measures. Two common ways of normalization are usually pursued, the linear and statistical normalization. In the linear normalization, the original variable is normalized to the unit interval [0, 1] via a simple linear transformation:

$$x_{\text{normalized}} = \frac{x_{\text{original}} - x_{\min}}{x_{\max} - x_{\min}}$$

where $x_{\min}$ and $x_{\max}$ are the minimal and maximal value of the variable encountered in the data. The statistical normalization uses the mean $\overline{x}$ and the standard deviation σ_x of the variable

$$x_{\text{normalized}} = \frac{x_{\text{original}} - \overline{x}}{\sigma_x}.$$

Finally, the logistic normalization involves a nonlinear transformation of data that follows a logistic transformation, namely $1/(1 + \exp(-x))$. In addition, when observing the activity of the individual neurons in the grid, some of them may be excessively "active" and winning most the time. The other neurons tend to become "idle." This uneven activity pattern is undesired and should be avoided. In order to promote more even activity across the network, we make the learning frequency-sensitive by penalizing the frequently winning nodes and increasing the distance function between the patterns (inputs) and the connections of the winning node. For instance, instead of the original distance $d(\mathbf{x}, \mathbf{w})$, we use the expression $(1+\varepsilon)*d(\mathbf{x}, \mathbf{w})$ where ε is a positive constant modeling the effect of intentionally increased distance between $\mathbf{x}$ and $\mathbf{w}$. The higher the value of ε, the more substantial the increase in the effective distance between the pattern and the neuron.

When designing a self-organizing map, the following are essential design parameters: the size of the map, initial learning rate and its temporal decay, type of distance function, and a form of data normalization.

Overall, the developed SOM is fully characterized by a matrix of connections of its neurons, that is $\mathbf{W} = [\mathbf{w}(i,j]]$, $i = 1, 2, \ldots p$, $j = 1, 2, \ldots p$ (note that we are now dealing with a squared p by p grid of the neurons). The simplest visualization scenario one can envision is to map the original data on the map so in this manner we get a certain insight into the structure of the data in a highly-dimensional space. For instance, we can state that $\mathbf{x_k}$ and $\mathbf{x_l}$ are similar because they "activate" two neighboring neurons on the map. A visualization of the relative position of the patterns is a main advantage of the SOM. Moreover, by a careful arrangement of the weight matrix into several planes (arrays) we can produce a variety of important views at the data. We introduce such concepts as a weight, region (cluster), and data density map.

III. Associated Self-Organizing Maps

The associated maps come as a result of a different organization of $\mathbf{W}$ or some slight modifications of the original connections.

A. Weight Maps

Obviously the weight matrix $\mathbf{W}$ can be viewed as a pile of layers of p by p maps indexed by the variables, see Fig. 3. That is we regard $\mathbf{W}$ as a collection of two-dimensional matrices each corresponding to a certain feature of the pattern, say

$$[w_1(i,j)]\ [w_2(i,j)]\ \ldots\ [w_n(i,j)]$$

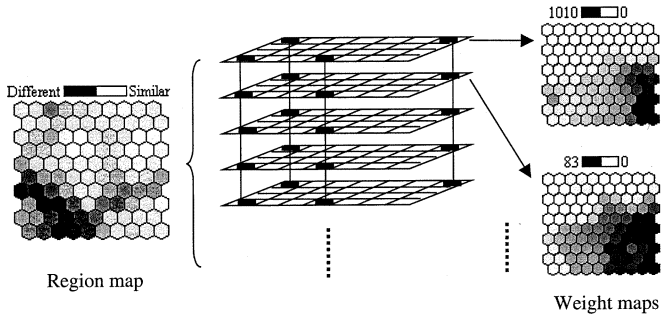


Fig. 3. A concept of associated SOM maps: region and weight maps.

Each of these matrices contain information about the weights (or the features of the patterns as the weights tend to follow the features once the self organization has been completed).

The most useful information we can get from these weight maps deals with an identification of possible associations between the features. If the two weight maps are very similar, this implies that the two corresponding features they represent are highly related. If two weight maps are very dissimilar, this means the two variables they represent are not closely interrelated. In addition, we can also determine the feature association for a data subset. For example, two weight maps can be very similar in the upper right corner, but are very dissimilar in other area of the map. This means only the data located in the upper-right corner are highly related. One should note that the identification of relationships is carried out in a visual mode and we do not allude to any measure of association such as a correlation coefficient. Nevertheless, this aspect is highly supportive in a descriptive data analysis and helps the designer understand the essence of the relationships between the features. In the sequel, it may lead to the identification of possible redundancies of some features (e.g., we state that two features are redundant if their weight maps are very close each other).

B. Region (Clustering) Map

A slight transformation (summarization) of the original map $\mathbf{W}$ allows us to visualize homogeneous regions in the map, *viz.* the regions in which the data are very similar. Furthermore we should be able to form boundaries between such homogeneous regions of the map. Owing to the character of this transformation, we will be referring to the resulting areas as clusters and calling the map a region (or clustering) map. The calculations leading to the region map are straightforward: for each location of the map, say (i, j) we compute distances between the weight vectors of its closest neighbors, such as $(i-1, j)$, $(i, j-1)$, $(i, j+1)\ldots$, etc. That is

$$\begin{array}{c} d(\mathbf{w}(i,j), \mathbf{w}(i-1,j)) \\ d(\mathbf{w}(i,j), \mathbf{w}(i,j+1)) \\ \cdots \end{array}$$

and take a median of these differences, that is, $\text{med}_{\mathtt{I}}\ [\Delta(i, j)]$ with $\mathtt{I}$ treated as a neighborhood of this particular location of the map. The neighborhood $\mathtt{I}$ functions in a same manner as commonly encountered in image processing (not surprising, the SOM is a digital image). The neighborhood consists of eight cells of SOM (pixels) surrounding the given neuron of the map. This median is regarded this as a measure of homogeneity of the nearest neighbor of the (i, j) location of the map. At the visual end, we map the median on a certain level of brightness to each of these results that gives us a useful vehicle of identifying regions in the map that are highly homogeneous. Likewise, the entries with dark color form a boundary between the homogeneous regions. Clusters can be easily identified by finding areas of higher level of brightness being surrounded by these dark boundaries. For some data set, there are distinct clusters, so in the clustering map, the dark boundaries are clearly visible. There could be cases where data are inherently scattered, so in the clustering map, we may not see clear dark boundaries. In Fig. 3, the region (clustering) map reveals two clusters: the one quite extensive that covers an upper part of the map and the smaller one. The clustering map is an important vehicle for a visual inspection of the structure in the data. It delivers a strong support for descriptive modeling: the designer can easily understand how structure looks like in terms of clusters. In particular, one can analyze the size of the clusters, their location in the map (that tells about closeness and possible linkages between the clusters). By looking at the boundaries between the clusters, the region map tells us how strongly these clusters are identified as separate entities distinct from each other. Overall, we can look at the region map as a granular signature of the data. These visualization aspects of SOMs underline their character as a user-friendly vehicle of descriptive data analysis. In this context, it also points out at the essential differences between SOMs and other popular clustering techniques driven by objective function minimization (say FCM and alike). Note that while FCM solves an interesting and well-defined optimization problem but does not provide with the same interactive environment for data analysis.

It is worth stressing that the homogeneous regions of the SOM could be detected in an automatic manner (as discussed in [7]). While attractive *per se*, the formation of the regions is affected by the values of some parameters (quite often difficult to adjust) that are not transparent to the user. The position promoted in this study is that the user/designer should play a dominant role in the determination of the regions in the map.

C. Data Distribution Map

The previous maps were formed directly from the general map $\mathbf{W}$ produced through self organization. It is advantageous to supplement all these maps with a data distribution (density) map. This map shows (again on a certain brightness scale) a distribution of data as they are allocated to the individual neurons on the map.

Following the assumed visualization scheme, the darker the color of the neuron, the more patterns invoked the neuron as the winning one; see Fig. 4.

The data density map can be used in conjunction with the region map as it helps us indicate how much patterns are behind the given cluster. In this sense, we may eventually abandon a certain cluster in our descriptive analysis as not carrying enough experimental evidence. Moreover, the data density map helps reveal some evident learning problem related to a few

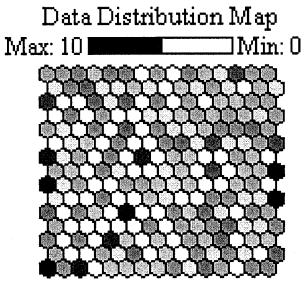


Fig. 4. An example data density map; the darker neurons identify groups of higher data density. These regions should be analyzed in conjunction with the region (clustering) map.

frequently winning nodes (neurons); to alleviate this discrepancy, we can introduce a frequency sensitivity component in the learning process.

Overall, the sequence realized so far can be described as follows:

highly dimensional data→SOM→region, data density maps
feature maps→interactive user-driven descriptive analysis.

In the sequel, we discuss how the regions (clustering areas) identified in the map can be described in terms of information granules—fuzzy sets.

IV. Estimation of Membership Functions

There are a number of existing methods of membership function estimation whose origin stems from different ways of interpreting fuzzy sets; cf. [13], [23], and [24]. In what follows, we are concerned in the notion of membership cast in the framework of pattern recognition. We also share an opinion that membership grades should relate in some way to experimental data.

We start with experimental data $\mathbf{X} = \{x(k)\}$, $k = 1, 2, \ldots, N$ belonging to several classes where the degree of belongingness of $x(k)$ to any class is binary (that is we adopt the yes–no class assignment). The dominant class is determined; say ω_0 and the intent is to compute a membership function A of the concept (feature) describing ω_0. Without any loss of generality, we consider a two-class problem, meaning that we distinguish between the data points belonging to this most frequent class we are interested in (ω_0) and other class ω_1 that may represent all remaining classes.

The underlying idea is to assign the high membership grade (1) to the regions where there are only patterns belonging to ω_0. When moving to the regions where we encounter some patterns belonging to ω_1, we gradually start reducing the corresponding membership grades. Furthermore, we consider the membership function to be unimodal and distributed around a prototypical value of the concept governing the dominant class ω_0.

The procedure outlined can be formalized as a two-step algorithm

- determination of the prototypical value of the fuzzy set;
- determination of membership grades assumed by the fuzzy set around the prototype based on the experimental data.

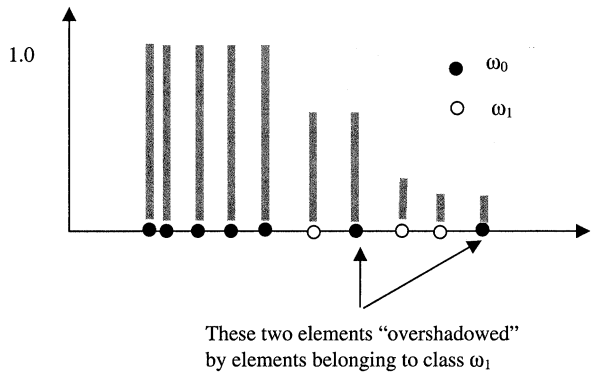


Fig. 5. Computing the membership function of A with the use of overshadowing principle; note that the point belonging to ω_1 "overshadows" other elements belonging to ω_0 that are located more distantly from m.

The prototypical value can be found in many different ways (say, as a median or mean). The simplest one is to take a mean value of the elements in $\mathbf{X}$ that belong to ω_0, that is

$$m = \frac{\sum_{k:\, x(k) \in \omega_0} x(k)}{\text{card}(x(k) | x(k) \in \omega_0)}$$

The determination of the membership grades of other points is guided by the following rationale; refer also to Fig. 5. The calculations are carried out separately for the data points to the left and to the right from the mean value. Furthermore we order the elements larger than the prototype in an increasing order. The elements lower than the prototypical value are order in decreasing fashion.

We start moving from the prototype up toward the higher values of the data with the initial membership grade equal to one, $A(m) = 1$. In this move, two cases may occur: a) either the next data point belongs to ω_0, or b) it belongs to ω_1. In the first case, we maintain the previous membership grade. If the point belongs to ω_1, we assign lower membership grade whose value is computed as

$$A(x(k)) = \max(0, A(x(k-1)) - \delta)$$

where

$$\delta = 1 - \exp\left(-\frac{x(k) - m}{x_{\max} - m}\right)$$

and $x_{\max}$ is the largest element in the data set. This means that if this new point $x(k)$ is close to the prototype, the reduction in membership value (δ) becomes substantial. Interestingly, the existence of elements that belong to the other class leads to an irreversible drop in the membership values of the fuzzy set. Note that for the next data points, say $x(k+1)$, $x(k+2)$ that may belong to the dominant class, we end up having the lower membership grade. In other words, the data point belonging to ω_1 "overshadows" the remaining larger data points in the sequence. Overall, the calculations of the membership values can be succinctly described by the expression

$$A(x(k+1)) = \begin{cases} A(x(k)) & \text{if } x(k+1) \in \omega_0 \\ A(x(k)) - \delta & \text{if } x(k+1) \in \omega_1. \end{cases}$$

The membership grades computations for the data points lower than the prototype (m) is carried out in an analogous

TABLE I
COLLECTION OF A TWO-CLASS DATA

x(k)	-2.5	-1.8	-0.6	-0.5	-0.1	0	4.2	0.9	1.5	1.9	2.1	3.3	2.8	3.1
Class	1	2	1	1	1	1	1	1	1	1	1	1	2	2

TABLE II
MEMBERSHIP GRADE OF THE FUZZY SET A CONSTRUCTED WITH THE OVERSHADOWING PRINCIPLE; THE MEMBERSHIP GRADES EQUAL TO 1 ARE HIGHLIGHTED

-2.5	-1.8	0.6	0.5	0.1	0.1	0.0	0.9	1.5	1.9	2.1	2.8	3.1	3.3	4.2
0.45	0.45	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.56	0.08	0.08	0.08

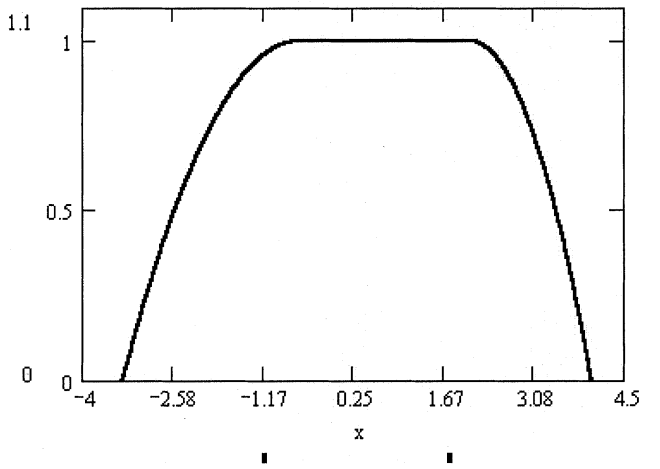


Fig. 6. Parabolic membership function of the feature of class ω_0 defined in the feature space x.

manner. The only difference is that now we order these data points in a decreasing order.

The membership function determined in this way exhibits a stairwise effect meaning that the changes of membership grades are confined to the discrete data points. One can approximate a continuous membership function (such as linear, parabolic, Gaussian, etc.) using these specific values.

As an example, let us consider a data set in Table I.

The dominant class (1) is denoted by ω_0. Following the above algorithm, the resulting membership grades are summarized in Table II.

If we proceed with a further parabolic approximation of the membership grades

$$A(x; m_-, m_+, \alpha_1, \alpha_2) = \begin{cases} 1 - \alpha_1 (x - m_-)^2, & \text{if } x < m_- \\ 1, & \text{if } x \in [m_-, m_+] \\ 1 - \alpha_2 (x - m_+)^2, & \text{if } x > m_+ \end{cases}$$

the resulting membership function is shown in Fig. 6.

In spite of the type of approximation developed on the basis of some discrete membership grades, there is an interesting and general observation as to the underlying form of the fuzzy set. If we do not encounter any elements belonging to class ω_1, then we consider a complete membership to the class of interest (ω_0). This way of looking at the fuzzy sets promotes existence of cores of the information granules (namely, 1-cuts of the respective fuzzy set). By endorsing this point of view, we often end up with regions of the feature space in which we have a high confidence as a reliable descriptor of the class. In contrast, in all models of probabilistic pattern recognition where we work with Gaussian probability density functions, these functions assume only a single prototypical value (mean) and do not show up any plateau around it.

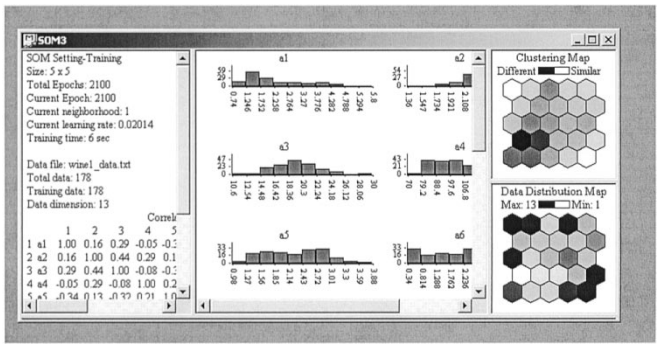


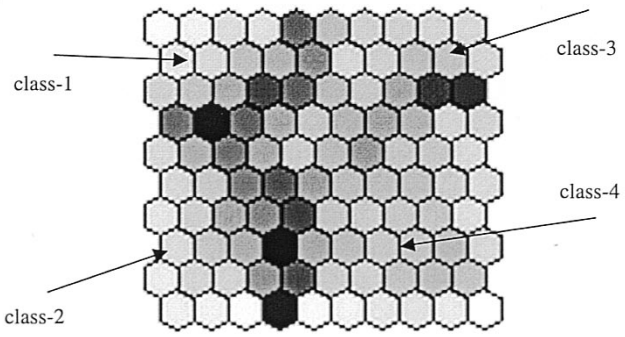
Fig. 7. A graphic environment of the development and interpretation of self-organizing maps.

V. SOFTWARE ENVIRONMENT AND EXPERIMENTING WITH A SYNTHETIC DATASET

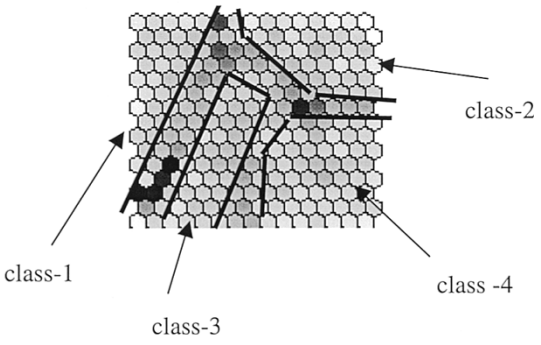
To highlight and exemplify a way in which the self-organizing map is used in data analysis, below we show a few snapshots illustrating how the software operates and what type of user interaction is involved. The developed software (C++, running on a PC platform) is user friendly with a significant interaction facilitated by the graphical user interface. Not only does this interface supports a visualization environment, but it helps the developer make decisions as to finding the structure in the data set and interact with the data in such process. The development environment provides the user with some generic statistical characterization of the data, Fig. 7, and navigates him through a detailed setup of the self-organization process (involving all necessary details such as the size of the map, normalization schemes, types of distance function, number of learning epochs). During the learning a map being formed is continuously updated and this dynamics is visualized in Fig. 7. Finally, the analyst can highlight (select) any region on the map and look at the corresponding subset of experimental data. The regions on the map are identified through a visual inspection. As one can immediately look at the corresponding data, the overall process becomes highly interactive. Obviously, such regions could be formed in an automatic fashion (as a matter of fact, SOMs are just images and there are a lot of image processing tools of edge detection and contour forming that could be found useful here). This option has not been pursued as being too restrictive and biased toward a specific algorithm of edge detection and edge tracking.

It is worth stressing that when running the same data set through a self-organizing map, we may end up with a different configuration of the regions. This is not essential as the regions themselves are quite repeatable in terms of their data content as well as mutual distribution on the map.

The intent of this section is to take a relatively simple low-dimensional example in which we know the structure in the data set and observe what structure is revealed by the self-organizing map. A four-dimensional synthetic data set is generated



(a)



(b)

Fig. 8. The structure revealed by the self-organizing map after the training; neurons marked in darker color delineate regions of high homogeneity both for the (a) 10 by 10 map and (b) 15 by 15 map.

by a uniform random generator [random ( )]where each group is shifted as follows:

$$x_1 = \text{random}() + m_1 \quad x_2 = \text{random}() + m_2$$
$$x_3 = \text{random}() + m_3 \quad x_4 = \text{random}() + m_4$$

Four groups were generated, each of them consisting of 2000 data points. The shift parameters describing each group are listed as follows:

Group no.	m_1	m_2	m_3	m_4
1	0.1	0.5	0.9	0.0
2	0.9	0.1	0.0	0.4
3	0.5	0.9	0.8	0.5
4	0.2	0.4	0.2	1.0

We experimented with two sizes of the map; in the first case it has a 10 by 10 grid of neurons, in the second structure the grid was increased to the size 15 by 15. The learning took 1000 epochs. The experimental finding was that after that no substantial changes in the connections of the neurons (and equivalently the structure of the data revealed by the map) have been observed. After the training, the structure of the data was revealed quite profoundly as illustrated in Figs. 8 and 9.

The differences are even more profound and the clusters are clearly delineated when the size of the map was increased to 25 neurons per column/row.

The distribution map (Fig. 10) provides us with the qualitatively the same picture as before yet now it comes with more

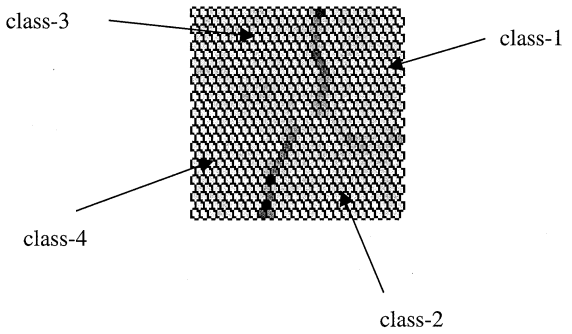


Fig. 9. Homogeneous regions in the 25 by 25 SOM.

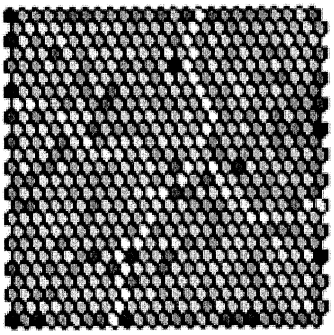


Fig. 10. The density distribution of data (associated self-organizing map); the darkest entries correspond to 33 patterns allocated to the respective neuron, the brightest points allocate zero patterns. Note that the boundaries exhibit a very low density of data points.

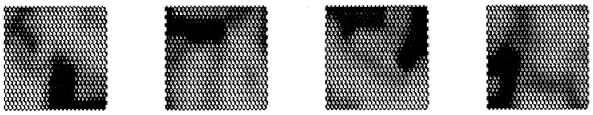


Fig. 11. Distribution of features in the SOM (darker regions correspond to higher values of the respective feature of the patterns).

details. Note that the boundaries are formed by a few data points that are different from the rest of the patterns.

Finally, the distribution of the features on the map is shown in Fig. 11. This provides us with another option to investigate relationships between the features (variables). By visual inspection, we immediately learn that features 2 and 3 are more "related" (in a visual sense) than the first and second feature. On the other hand, there is a relationship between the first and fourth feature that occurs only for a lower portion of the map (where the high values of these features coincide).

It should be mentioned that for smaller maps, their ability to distribute the classes in disjoint regions was very limited. For instance, in the 5 by 5 map, we were not able to delineate clearly separable regions. Furthermore far more overlap between the classes occurred even for the same neuron. This was a clear indicator of the size of the network not being fully adequate to the size of the data set.

In the sequel, we visualize the performance of SOMs on two data sets quite often used in machine learning.

Wine Data: This dataset consists of 178 data points (patterns) belonging to three classes. We start with a small 5 by 5 SOM which was trained for 2000 epochs. The patterns are normalized linearly. The results are shown in Fig. 12. By visually

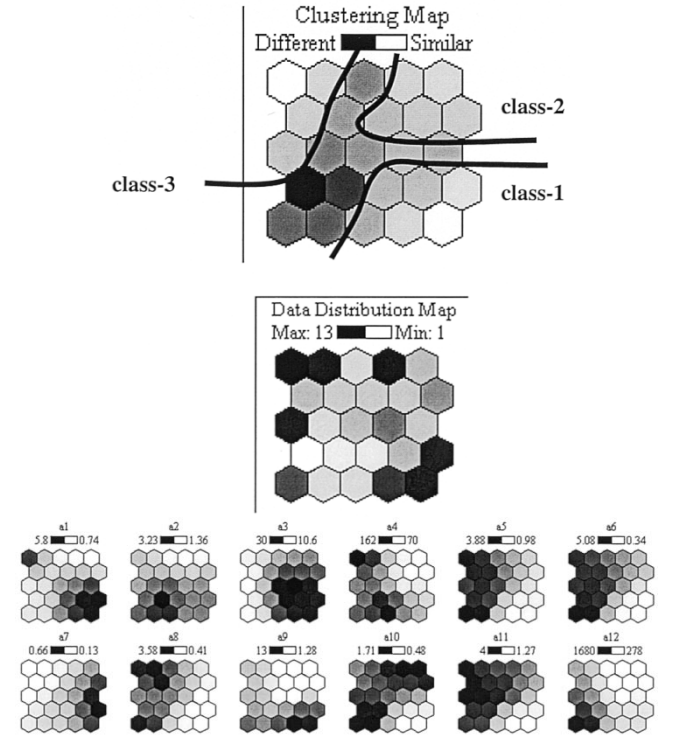


Fig. 12. Visualizuation of data in the SOM with inspection of classes. (Top) Distribution of patterns across the map. (Middle) Individual features distributed across the maps.

inspecting the map, we see clearly identified boundaries that potentially delineate the patterns belonging to different classes. This indeed has happened. As the software environment is highly interactive so that we can directly look under the "hood" of the map and inspect the patterns associated with the selected nodes of the map. The homogeneous regions (the nodes with light shadowing) and their correspondence with the classes is included in Fig. 12. Interestingly, the homogeneous regions correspond quite well with the areas of the map of high density of data points [again, this effect is visualized in Fig. 12(a)].

This analysis gives us an immediate visual insight into the complexity of the problem treated as a potential classification task. The weight maps reveal dependencies between the features in a graphical form. It can be viewed as a generalization of the standard correlation analysis when we characterize a (linear) dependency between features by a single numeric value (correlation coefficient) while now we are provided by a series of maps one can visually inspect and "correlate." The details are shown in Fig. 12(c). It becomes apparent that some features (shown in the maps denoted by a5, a6, a10, and a11) exhibit the same behavior while others are quite distinct.

Glass Data: In this study, we are concerned with 214 instances of glass belonging to seven classes. The classification was motivated by criminological investigation. Each pattern is characterized by a number of features dealing with the chemical content of glass (sodium, aluminum, silicon, barium, etc.).

We start with a 5 by 5 SOM trained for 2000 learning epochs. Linear normalization is used to preprocess the data; see Fig. 13.

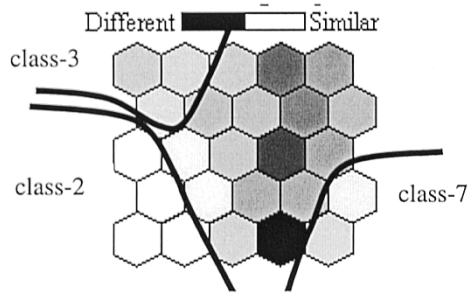


Fig. 13. Visualization of glass data (classes) supported by SOM.

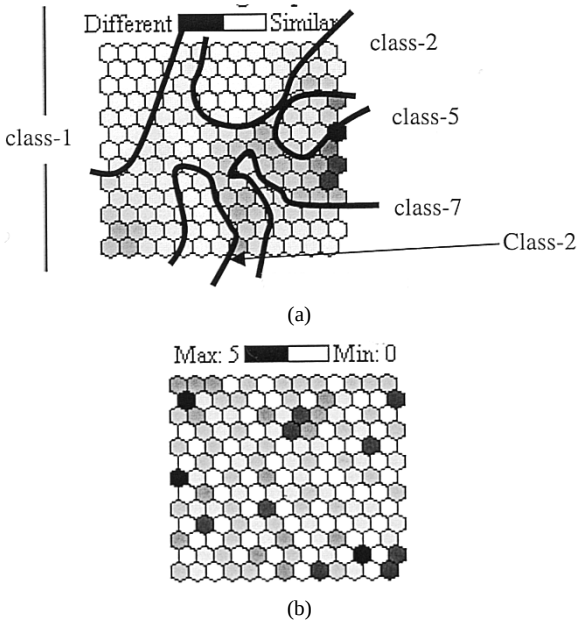
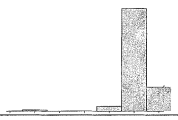
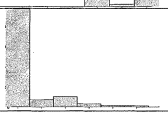
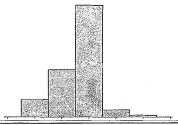
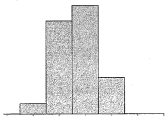
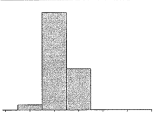
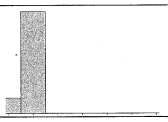
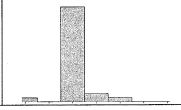


Fig. 14. (a) Distribution of classes in the SOM. (b) Data density across the map.

Few classes can be delineated, namely 2, 3, and 7. The rest are difficult to distinguish. The increase of the size of the map helps alleviate the problem. With the increase of the size of the map to a 13 by 13 grid, Fig. 14, after 7000 learning epochs we end up with several homogeneous regions identifying most of the classes existing in the problem (the size of the map was made quite large on purpose with an intent to see how far the discrimination between the classes can be realized). The results point out that some classes are easy to discriminate (those are the classes we were able to find in the map) while others such as class require more attention when building their classifiers. Still, at this size of the map, we were not able to find a clearly distinguished region occupied by class-6. The distribution of the classes in the map reflects the diversity of the patterns belonging to the corresponding class; apparently class-5 is more "compact" than class-1. The mutual distribution of the classes is another interesting indicator as to the relationships between the classes; for instance class-1 and class-2 are neighbors while class-7 is located quite distant from these two. Interestingly, the distribution of patterns across the map is quite uniform, Fig. 14(b) meaning that all nodes of the map were involved in the organization of the data to a similar extent.

TABLE III
CHARACTERISTICS OF THE GRANULAR DESCRIPTORS OF THE ECG CLASSES

Region of the map	Number of patterns	Homogeneity (number of patterns across classes)	Description of the region
A	248		Occupied by class "AMI" with some patterns from class "MIX"
B	93		Class "IMI" and "MIX " are represented in almost equal mixture
C	119		Clas "N" dominates this cluster
D	106		Class"BVH" with some data coming from class"RVH"
E	48		Class 'BVH" and "RVH" in an almost equal mix with a few patterns in class"LVH" and "AMI"
F	28		Class"RVH" dominates here with some patterns coming from class 'BVH"
G	23		Class"LVH" with a few patterns in class"N"
H	28		Class"RVH" with few patterns belonging to class "BVH"

VI. GRANULAR ANALYSIS OF THE ECG DATA

The complex problem of computerized diagnostic classification of the ECG signal has been considered as a real example. A consistent ECG database characterized by a clinical validation has been investigated.

The CORDA database, developed by J. Willems at the Medical Informatics Department of the University of Leuven, Leuven, Belgium [36], consists of 3253 12 lead ECGs (2140 men and 1113 women with a mean age of 49 ± 12 years). There were 12 standard leads (that is I, II, III, AVR, AVL, AVF, V1, V2, V3, V4, V5, V6). It consists only of single-disease cases with normal QRS duration and no conduction abnormality. Seven diagnostic classes have been considered: normal (N), left ventricular hypertrophy (LVH), right ventricular hypertrophy (RVH), biventricular hypertrophy (BVH), inferior (IMI), anterior (AMI), and combined (MIX) myocardial infarction.

From the original ECG signal (12 standard leads acquired at 500 Hz for a period of about 10 s), a set of 540 (45 for each lead) primary measurements were computed with a computerized system, obtaining a first consistent data reduction. A second data reduction, according to a clinical selection and a statistical selection, has been performed obtaining a set of 39 ECG features. They include amplitudes and duration of the QRS and T waves, QRS and T axes, ST-segment elevation or depression, and the area under QRS and T waves.

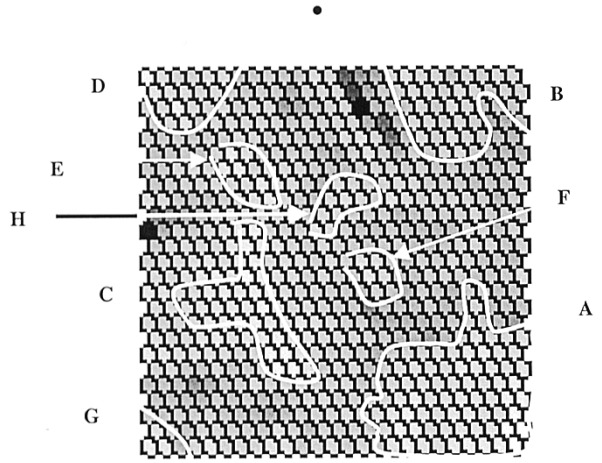


Fig. 15. The self-organizing map (size of 25 by 25) and several clusters identified for further analysis.

The same dataset has been used to establish the performance of statistical classification models [36], and to validate the performance of different architectures of neural networks [3], [4].

One can envision a certain hierarchy of the classes of the signals that could be helpful in understanding the results of self-organization. The diagnostic class of biventricular hypertrophy (BVH) then includes both LVH and RVH, and consequently the three classes BVH, LVH, and RVH are not completely independent. This means that a classification of LVH + RVH is equivalent to BVH, and that a BVH patient classified as only LVH or RVH represents a partial discrepancy. Analogous considerations are valid for the diagnostic class of combined myocardial infarction MIX with respect with AMI and IMI.

The size of the map was experimented with. Finally, the size of 25 by 25 is a reasonable choice considering the size of the data set as well as the interpretation results one can derive (it is worth noting that this description of data is an interactive process so the user has control over the granularity of the descriptors visible through the map). The data were normalized with the use of a logistic transformation.

There are of immediate and important observations one can make on the basis of a visual inspection of the self-organizing map (especially the region map and the maps of the individual features). We may quantify the groups in a more quantitative manner as summarized in Table III. These groups of data are described in terms of class homogeneity, total size, and fuzzy sets—information granules capturing the data beneath the selected portion of the map.

Several interesting observations can be drawn.

- The homogeneous regions in the SOM, their size and location *vis-à-vis* other regions help identify relationships between the classes of ECG signals.

  In particular, a region (cluster) capturing normal signals [region C in Fig. 15] is quite compact and shows a high level of homogeneity. The region denoted by A [that involves AMI] is quite extended and is quite distant from other regions. Similarly, region B (that captures a mixture

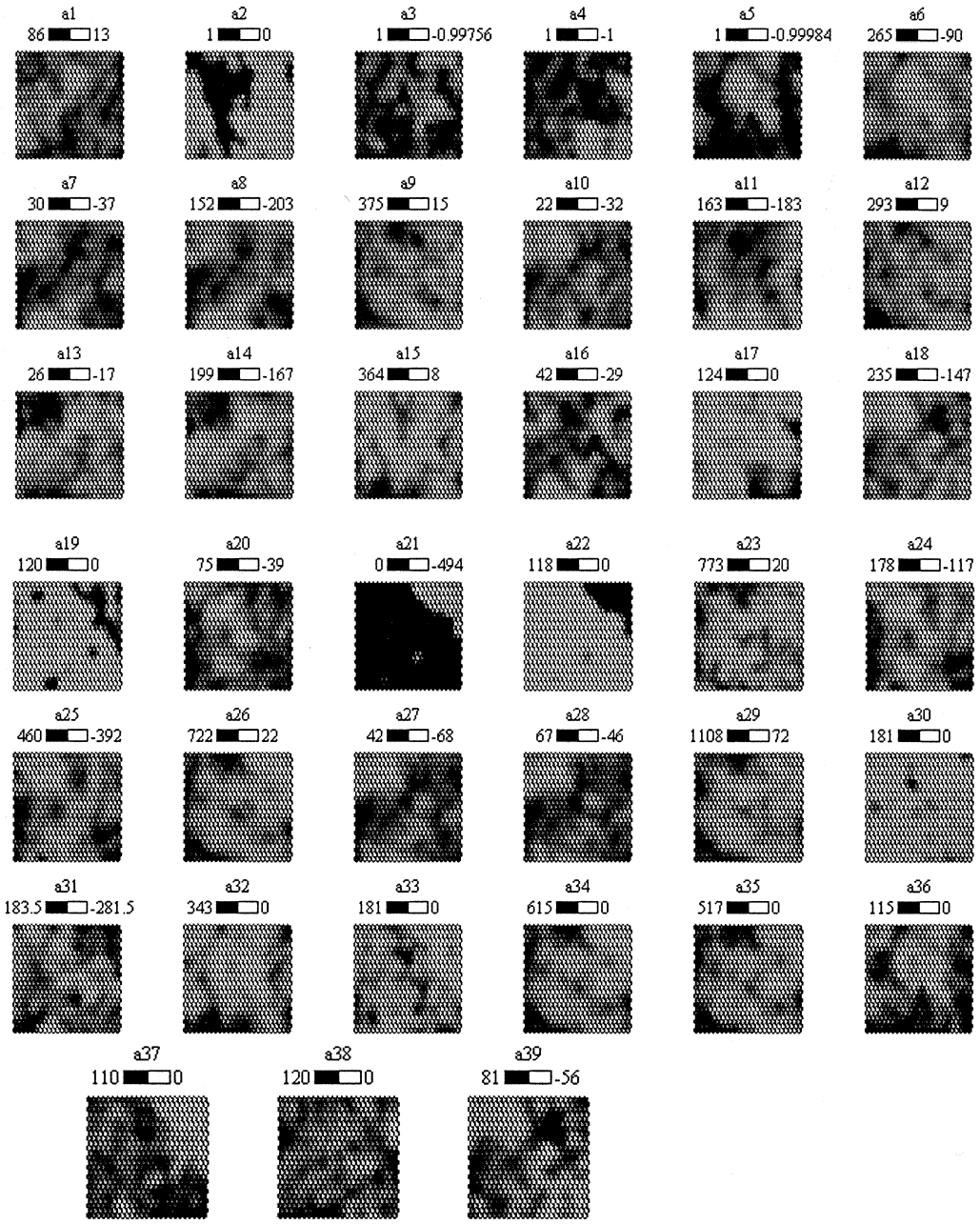


Fig. 16. Feature maps of the ECG patterns; the features are denoted as a1, a2, …, a39. The brightness scale shows the values of the features as they are distributed across the map.

of IMI and MIX) is apart from the other regions and occupies an entire region on the upper right corner of the map. A very different behavior can be observed for the three other regions, that is E, H, and F. These are close neighbors and all of them capture two classes RVH and BVH but in a different mix. When moving along the map and starting from the first one (E), there is an evident mix of BVH and RVH. In the sequel, the next group (H) is dominated by RVH while the group identified as F has a similar dominance by RVH with some BVH.

- The identification of the groups in the map can be viewed as a descriptive data analysis with an ultimate goal to capture the essence of the data. In this case we are interested in building concise and homogeneous descriptors of the ECG classes. The map tells us what is most likely as to the occurrence of "plain" or mixed classes of patterns. Obviously, it is easy to describe (and discriminate) between the class of normal signals (N) and others while discriminating between class IMI and MIX (as shown in region B) will be a difficult task (no matter what classifiers we are interested in). It is easy to discriminate between class RVH when dealing with region H however doing the same for the region E (where there is an evident mix of RVH and BVH) will be a significant classification challenge.

The series of maps for each feature (feature maps) as shown in Fig. 16 is important sources of information by helping us visualize relationships between the features. A quick visual inspec-

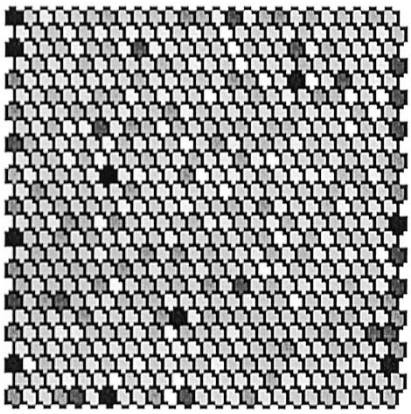


Fig. 17. Density map associated with the SOM: the brighter the color, the higher the number of the data points allocated to the corresponding neuron. The range of these numbers is from 9 (the darkest location of the map) to 0 (the brightest entries of the map).

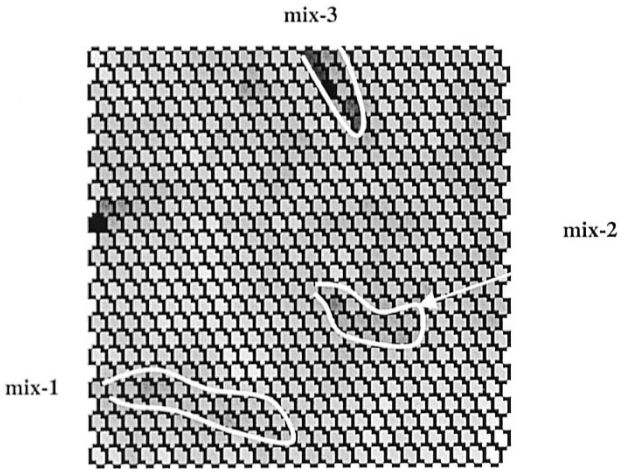


Fig. 18. Visual identification of the regions in the map of the highest diversity.

tion helps us notice that some of them are highly related (the corresponding maps are very similar). For instance

- the parameters A21 (Q amplitude in V3) and A22 (Q duration in V3) as well as partially A19 (Q duration in V1) exhibit similar behavior, showing a region with high values in the upper right corner, with a correspondence (in agreement) with the classification of region B.
- Some qualitative similarities can be seen considering the parameters A27(ST elevation in V6), A28 (ST slope in V6), and A10 (ST elevation in II).
- A qualitative similarity is shown by A14 (area under T wave in lead AVR) and A13 (ST elevation at 80 ms after J point of lead AVR).

Fig. 17 shows a density map illustrating how the ECG patterns populate the SOM In general, the data become distributed across the map quite uniformly with an exception of few entries. Nevertheless, the differences are not very substantial. The density map states that there are no any problems with the learning as there were no particularly "hyperactive" neurons during the learning process.

One may be interested in the content of the map exhibiting a significant level of diversity, Fig. 18. These regions carry a substantial level of class diversity as well as shown in Fig. 19.

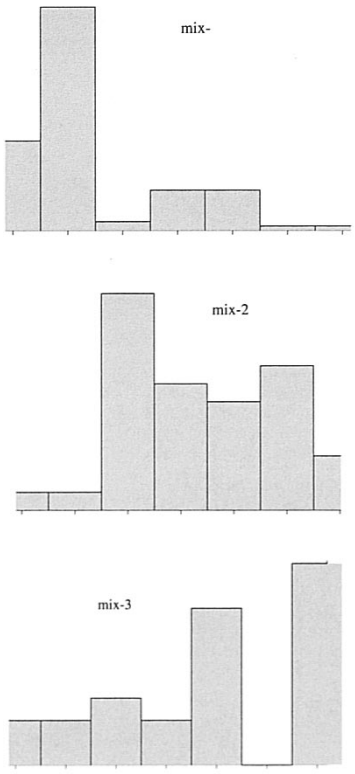


Fig. 19. Distribution of patterns in the selection regions of the map of high diversity.

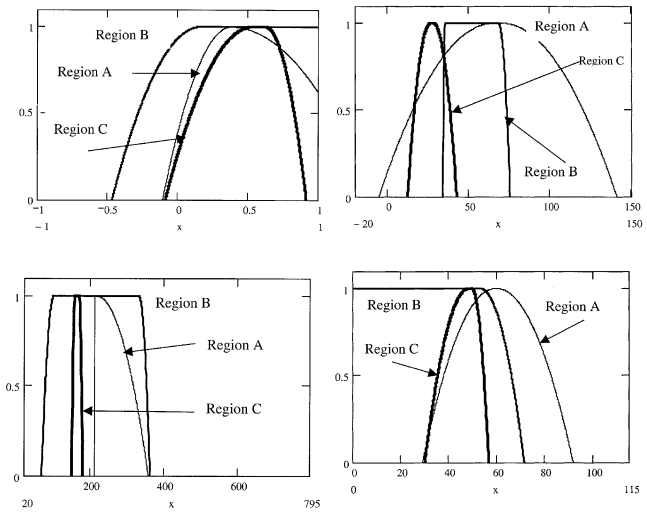
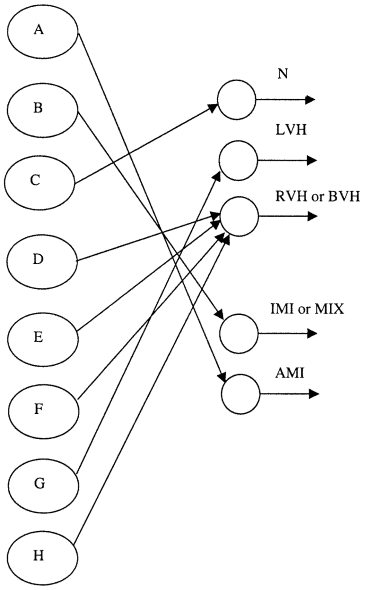


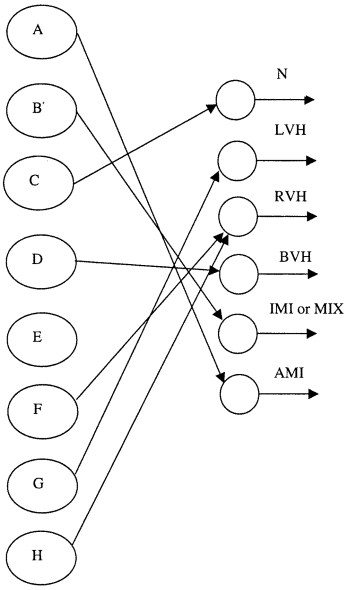
Fig. 20. Fuzzy sets formed on the basis of the fuzzy regions (clusters) defined in the self-organizing map: QRS axis in AVF in 40 ms, QRS peak-to-peak amplitude in V3, T amplitude in V3, R duration in AVL.

Again, we emphasize that the SOM-based analysis is user oriented. It is a user who makes decisions about delineating regions on the map, identify their "content" (i.e., the content in terms of classes involved) and make decisions as to expansion or contraction of the specific region. Obviously, the homogeneity map along with the data density map are used to support such decision-making problem.

Each region (cluster) comes with its own granular signature of data in the form of the fuzzy sets of the features. A few illustrative examples are included in Fig. 20. Obviously, having



(a)



(b)

Fig. 21. The use of descriptive analysis in forming a blueprint of predictive fuzzy models. (a) BVH and RVH merged into a single class, also class IMI and MIX considered together. (b) BVH and RVH treated as two separate classes.

seven classes and 39 features, it is impossible to display all of them, but even the sample shown here gives an interesting and useful insight into the meaning of the features.

VII. From Descriptive to Predictive Fuzzy Models

In light of the general taxonomy of descriptive–predictive models, one can state that the profound majority of models encountered in fuzzy modeling fall under the second category. Interestingly, the role of fuzzy sets has been diminished there in the sense that all design of the fuzzy models are immediately aimed at very detailed constructs. Hence, what is really offered by fuzzy sets, that is a global and user-friendly view at data, is not of primary concern in the predictive model.

We may envision the previous descriptive analysis to be an important step in constructing a blueprint of any further and far detailed fuzzy model, say a fuzzy classifier. In particular, we may think of the regions identified in the SOM to form a certain type of receptive fields. Then, the structure becomes evident (Fig. 21).

What is shown there pertains to some selected classes of the fuzzy neural architectures. The choice of the specific topology depends upon the classification problem we are interested in. If general classes (that is an amalgamation of RVH and BVH) are admissible, the architecture in Fig. 21(a) is a viable option. If such generalized class is not feasible, a blueprint of the model in Fig. 21(b) is a good starting point. The connections linking the receptive fields (regions in the SOM) with the output unit are initialized based on such descriptive analysis; further detailed learning is nevertheless required to achieve a full calibration of the model and increase its accuracy.

VIII. Conclusion

In this paper, we have distinguished between descriptive and predictive fuzzy models and fuzzy modeling. The focal point of these investigations is concerned with the first category of the system modeling techniques. It is shown that at descriptive fuzzy modeling is a highly designer-oriented activity with the objective to make the "internal" language of the data understood by the designer (naturally, we envision that a certain type of a visual environment is required).

The main points worth emphasizing are as follows.

- The descriptive fuzzy modeling is aimed at allowing the data "speak their own language" and translate these findings in terms (that is information granules expressed as fuzzy sets) so that they become meaningful to the designer/user. We are after the *transparency* of the constructs (and their relevancy in terms of enough experimental evidence. The noninvasive nature of this category of modeling is also apparent.

  This type of modeling is predominantly user centered. It is the designer who formulates questions and hypotheses about the possible structure in the data, validates them on a basis of the granular findings. The environment in which the designer operates needs to be highly interactive so that the crucial relationships are portrayed in the easily comprehended format and the environment facilitates any form of the "what-if" analysis. It has been shown that self-organizing maps augmented with additional visualization vehicles (such as region/cluster and density maps) are a suitable development environment for the descriptive data analysis. At this point one may be tempted to automate the process of building regions in the region (clustering) map (which could not be difficult considering that it would end up being a certain task of image processing—after all these maps are just digital images). Nevertheless, we strongly believe that there should be enough room for initiative of the designer so there are no specific constraints imposed on the way of perceiving the problem.
- The descriptive modeling is a prerequisite for predictive modeling (no matter what type of the model is sought

and what development/identification technique is considered afterwards). The descriptive modeling may help address crucial modeling questions as to the suitability of a specific detailed mode and navigate in processes of decision-making pertinent to such detailed modeling activities.

The descriptive modeling has been applied to the analysis of ECG data. It was treated as a preliminary phase of designing of the detailed classifiers by providing an invaluable insight into the nature of classes, their distribution and overlap between the classes.

REFERENCES

[1] M. R. Anderberg, *Cluster Analysis for Applications*. New York: Academic, 1973.

[2] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[3] G. Bortolan and J. L. Willems, "Diagnostic ECG classification based on neural networks," *J. Electrocardiology*, vol. 26, pp. 75–79, 1994.

[4] G. Bortolan, C. Brohet, and S. Fusaro, "Possibilities of using neural networks for diagnostic ECG classification," *J. Electrocardiology*, vol. 29, pp. 10–16, 1996.

[5] K. Cios, W. Pedrycz, and R. Swiniarski, *Data Mining Techniques*. Boston, MA: Kluwer, 1998.

[6] M. Delgado, F. Gomez-Skarmeta, and F. Martin, "A fuzzy clustering-based prototyping for fuzzy rule-based modeling," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 223–233, Apr. 1997.

[7] M. Drobics, U. Bodenhofer, W. Winiwarter, and E. P. Klement, "Data mining using synergies between self-organizing maps and inductive learning of fuzzy rules," in *Proc. of IFA-2001*, pp. 1780–1785.

[8] B. Fritzke, "Growing self-organizing networks—History, status quo, and perspectives," *Kohonen Maps*, pp. 131–144, 1999.

[9] A. F. Gomez-Skarmeta, M. Delgado, and M. A. Vila, "About the use of fuzzy clustering techniques for fuzzy model identification," *Fuzzy Sets Syst.*, vol. 106, pp. 179–188, 1999.

[10] B. Kelkar and B. Postlethwaite, "Enhancing the generality of fuzzy relational models for control," *Fuzzy Sets Syst.*, vol. 100, pp. 117–129, 1998.

[11] E. T. Kim, M. K. Park, S. H. Ji, and M. Park, "A new approach to fuzzy modeling," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 328–337, June 1997.

[12] E. Kim, H. Lee, M. Park, and M. Park, "A simple identified Sugeno-type fuzzy model via double clustering," *Inform. Sci.*, vol. 110, pp. 25–39, 1998.

[13] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*. Upper Saddle River, NJ: Prentice-Hall, 1988.

[14] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, pp. 59–69, 1982.

[15] ——, *Self-Organizing Maps*. Berlin, Germany: Springer-Verlag, 1995.

[16] T. Kohonen, S. Kaski, K. Lagus, and T. Honkela, "Very large two-level SOM for the browsing of newsgroups," in *Proc. ICANN96*, vol. 1112, 1996, pp. 269–274.

[17] Y. Lin and G. A. Cunningham, III, "A new approach to fuzzy-neural system modeling," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 190–198, Apr. 1995.

[18] E. Oja and S. Kaski, Eds., *Kohonen Maps*. Amsterdam, The Netherlands: Elsevier, 1999.

[19] W. Pedrycz, *Fuzzy Sets Engineering*. Boca Raton, FL: CRC, 1995.

[20] ——, *Computational Intelligence: An Introduction*. Boca Raton, FL: CRC, 1997.

[21] ——, "Conditional fuzzy clustering in the design of radial basis function neural networks," *IEEE Trans. Neural Networks*, vol. 9, pp. 601–612, Aug. 1998.

[22] ——, "Fuzzy set technology in data mining and knowledge discovery," *Fuzzy Sets Syst.*, vol. 3, pp. 279–290, 1998.

[23] ——, "Fuzzy equalization in the construction of fuzzy sets," Fuzzy Sets Syst., to be published.

[24] W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets: Analysis and Design*. Cambridge, MA: MIT Press, 1998.

[25] W. Pedrycz and M. Reformat, "Rule-based modeling of nonlinear relationships," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 256–269, Apr. 1997.

[26] W. Pedrycz and J. Waletzky, "Fuzzy clustering in software reusability," *Software: Pract. Exp.*, vol. 27, pp. 245–270, 1997.

[27] W. Pedrycz and A. V. Vasilakos, "Linguistic models and linguistic modeling," *IEEE Trans. Syst., Man, Cybern.*, vol. 29, pp. 745–757, Dec. 1999.

[28] G. Piatetsky-Shapiro and W. J. Frawley, Eds., *Knowledge Discovery in Databases*. Menlo Park, CA: AAAI Press, 1991.

[29] M. Russo, "FuGeNSys: A genetic neural system for fuzzy modeling," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 373–388, June 1998.

[30] E. Sanchez, "Genetic algorithms, neural networks and fuzzy logic systems," in *Proc. 2nd Int. Conf. Fuzzy Logic Neural Networks*, Japan, 1992, pp. 17–19.

[31] M. Setnes and H. Roubos, "GA-fuzzy modeling and classification: Complexity and performance," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 509–522, Oct. 2000.

[32] M. Setnes, "Supervised fuzzy clustering for rule extraction," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 416–424, Aug. 2000.

[33] R. Silipo, G. Bortolan, and C. Marchesi, "Design of hybrid architectures based on neural classifier and RBF pre-processing for ECG analysis," *Int. J. Approx. Reason.*, vol. 21, pp. 177–196, 1999.

[34] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 7–31, Feb. 1993.

[35] L. X. Wang and C. Wei, "Approximation accuracy of some neuro-fuzzy approaches," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 470–478, Aug. 2000.

[36] J. L. Willems, E. Lesaffre, and J. Pardaens, "Comparison of the classification ability of the electrocardiogram and vectorcardiogram," *Amer. J. Cardiology*, vol. 59, pp. 119–124, 1987.

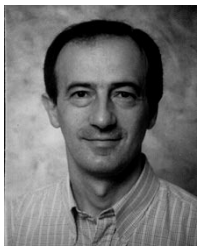
[37] L. A. Zadeh, "Fuzzy sets and information granularity," in *Advances in Fuzzy Set Theory and Applications*, M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds. Amsterdam, The Netherlands: North-Holland, 1979, pp. 3–18.

[38] ——, "Fuzzy logic = computing with words," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 103–111, Feb. 1996.

[39] ——, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 90, pp. 111–117, 1997.

[40] L. A. Zadeh and J. Kacprzyk, *Computing With Words in Information/Intelligent Systems*. Heidelberg, Germany: Physical-Verlag, 1999, vol. I and II.

[41] J. Zhang and A. J. Morris, "Process modeling and fault diagnosis using fuzzy neural networks," *Fuzzy Sets Syst.*, vol. 79, pp. 127–140, 1996.



Giovanni Bortolan (M'97) received the Doctoral degree (Laurea) from the University of Padova, Padova, Italy, in 1978.

He is Senior Researcher at the Institute of System Science and Biomedical Engineering (LADSEB), Italian National Research Council, Padova, Italy. He has published numerous papers in the areas of medical informatics and applied fuzzy sets. He is actively pursuing research in medical informatics in computerized electrocardiography, neural networks, fuzzy sets, data mining, and pattern recognition.

Witold Pedrycz (M'88–SM'94–F'99) is a Professor and Chair in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also a Canada Research Chair (CRC) in computational intelligence, and is actively pursuing research in computational intelligence, fuzzy modeling, knowledge discovery and data mining, fuzzy control including fuzzy controllers, pattern recognition, knowledge-based neural networks, relational computation, and software engineering. He has published numerous papers in this area, and is also an author of seven research monographs covering various aspects of computational intelligence and software engineering.

Dr. Pedrycz has been a Member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, and the IEEE TRANSACTIONS ON FUZZY SYSTEMS.