Gastroenterology Dataset Clustering Using Possibilistic Kohonen Maps

Anas Dahabiah, John Puentes, and Basel Solaiman
TELECOM Bretagne, Département Image et Traitement de l'Information, Brest
FRANCE
anas.dahabiah@telecom-bretagne.eu
http://www.telecom-bretagne.eu

Abstract: - Kohonen maps are an efficient mechanism in signal processing and data mining applications. However, all the existing versions and approaches of this special type of neural networks are still incapable to efficiently handle within a simple, fast, and unified framework, the imperfection of the patterns' information elements on the one hand like the uncertainty, the missing data, etc., and the heterogeneity of their measuring scale (qualitative, quantitative, ordinal, etc.) on the other hand. Therefore, we propose in this paper a possibilistic Kohonen network essentially based on two fuzzy measures: the possibility and the necessity degrees, to deal with all these aspects together in a robust way. Concrete examples and medical applications will also be given to clarify and to easily explain the proposed algorithm.

Key-Words: - Self-Organizing Maps (SOMs), Fuzzy Logic (Possibility Theory), Imperfect information, Gastroenterology Dataset, Similarity.

1 Introduction

Kohonen networks (known as Self-Organizing Maps "SOMs") are an effective mechanism in signal processing. They can convert a complex high-dimensional input signal into a simpler low-dimensional discrete map [1] [2]. Ritter [3] has shown that SOMs represent a nonlinear generalization of principal components analysis (another dimension-reduction technique). Thus, they are nicely appropriate for cluster analysis, image and sound processing, and many other applications [1] [4]. The SOM training algorithm resembles also to vector quantization (VQ) algorithms, such as K-means. The important distinction is that in addition to the best-matching weight vector, its topological neighbors on the map are updated too.

Actually, Kohonen networks are a special type of the neural networks based on the competitive learning which is based on similarity estimation. All the previous works and applications of the SOMs suppose generally that the value of each input is precise, certain, and given in order to calculate the similarity and to estimate the new weights of the network, while in reality, a remarkable amount of incomplete and imperfect values may be presented to the input of the artificial neural networks. For this reason, we will propose in this paper an approach fundamentally based on possibility theory to estimate the similarity and the weights, taking into account the imperfection of the data sets. Our paper is organized as follow: section 2 briefly presents the SOMs, the algorithm, and the limitations. Section 3 contains the basic principles of possibility theory and the proposed approach to overcome the limitations. Then, a concrete example is presented in section 4 and a real medical application is illustrated in section 5 to clarify and to simply explain our approach. Our remarks and perspectives are discussed in section 6.

2 Self-Organizing Maps

Kohonen networks [1] were introduced in 1982 by the Finnish researcher Tuevo Kohonen as a special type of neural networks to reduce the dimensionality of the input signals. They have been call self-organizing maps thanks to their ability to elucidate or reproduce some fundamental organizational property of the input data without benefit of supervised training procedures. Like neural networks, SOMs are feedforward and fully connected. Feedforward networks don't allow looping or cycling. "Fully connected" means that every node in a given layer is connected to every node in the next layer, and unconnected to any node in the same layer. Each connection between nodes has a weight associated with it, which is assigned randomly to a value between zero and one at initialization. Adjusting these weights represents the key for the learning mechanism. Input variable values need to be normalized or standardized so that certain variables don't overwhelm others in the learning algorithm. Unlike most neural networks, SOMs have no hidden layer. Data from the input layer is passed along directly to the output layer. The output layer is represented in the form of a lattice whose shape is usually rectangular (see figure 1) or hexagonal.

For a given object (record, instance, stimuli, feature vector, etc.), a particular field value (attribute, variable, observation, feature, sample, example, etc.) is forwarded from a particular input node to every node in the output layer. The values of all the fields, together with the weights assigned to each connection, determine the values of a scoring function (such as Euclidean

distance) for each output node. The output node with the best outcome from the scoring function would then be designed as the winning node, or the Best Matching Unit (BMU). This node becomes the center of neighborhood of excited neurons whose weights are adjusted so as to further improve the score function. In other words, these nodes will participate in the adaption (learning) process.

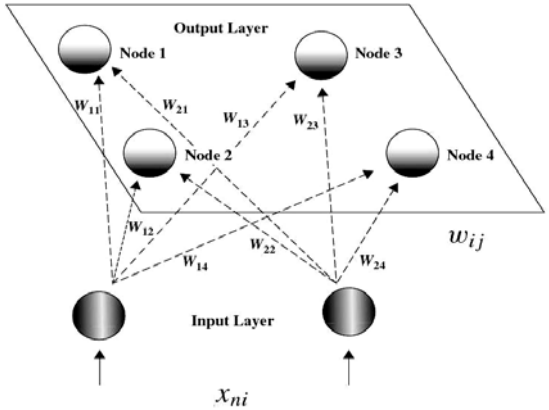

Fig. 1 Self-organizing map topology

2.1 Kohonen Algorithm

For each input vector from the data set

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_n \\ \cdot \\ \cdot \\ X_s \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & .. & x_{1i} & .. & x_{1N} \\ x_{21} & x_{22} & .. & x_{2i} & .. & x_{2N} \\ .... & .... & .... & .... & .... & .... \\ x_{n1} & x_{n2} & .. & x_{ni} & .. & x_{nN} \\ .... & .... & .... & .... & .... & .... \\ x_{s1} & x_{s2} & .. & x_{si} & .. & x_{sN} \end{bmatrix}$$

where X_n represents an object in the data set like a patient record in a medical database for example and x_{ni} represents an attribute of X_n (like the hemoglobin or the age of the patient), do:

I- Competition:

For each output node j, calculate the similarity (or the dissimilarity) between the two vectors W_j and X_n using the scoring function (the Euclidean distance given by equation 1 for example):

$$D(w_j, X_n) = \sqrt{\sum_i (w_{ij} - x_{ni})^2} \quad (1)$$

and find the wining node J that maximizes the Similarity (or minimizes the dissimilarity) over all the output nodes.

II- Cooperation:

Identify all output nodes j within the neighborhood of J defined by the neighborhood size R. For these nodes, do the following for all input record fields:

A. Select the nodes in the neighborhood of the winning node that will participate in the learning phase. The weights of these nodes are adjusted so as to further improve the score function. In other words, these nodes will have an increased chance of winning the competition once again, for a similar set of field values.

B. Adjust the weights:

$$w_{ij,new} = w_{ij,current} + \eta(x_{ni} - w_{ij,current})$$

C. Adjust the learning rate and neighborhood size, as needed.

D. Stop when the termination criteria are met.

2.1.1 Prior Works' Limitation

In the beginning, the conventional SOM training as proposed by Kohonen (the previous paragraph) were only capable to process crisp quantitative (numeric) information elements since both determining the BMU from the map units and updating BMU's topological neighbors are based on numeric distance function, typically the Euclidean.

Later, the necessity to handle both the qualitative and the quantitative data under a unified framework were imposed due to the huge number of heterogeneous data encountered in the large databases. To fulfill this need, many techniques and approaches have been proposed [5]:

The first approach supposes that the qualitative variables must be always presented to Kohonen nets by using as many neurons as the number of the values that the variable can take. In this case, only one of the neurons will be turned on according to the value of the variable. All the other neurons will be turned off. This technique is called one-of-n encoding. The only exception to this rule is if the qualitative variable is binary (taking one of only two possible values), then one neuron can be used. It is turned on for one value, and off for the other. Figure 2 [5] depicts an illustrative example of this transformation. In this example it is supposed that

the qualitative variable Favorite-Drink can take four possible categories {Coke, Pepsi, Mocca, Nescafe}.

Name	Favorite_Drink	Amount
Jane	Coke	7
Mary	Pepsi	7
Tom	Mocca	7
Helen	Nescafe	7

⇨

Name	Coke	Pepsi	Mocca	Nescafe	Amount
Jane	1	0	0	0	7
Mary	0	1	0	0	7
Tom	0	0	1	0	7
Helen	0	0	0	1	7

Fig. 2 Qualitative attribute "Favorite-Drink" is transformed to four binary attributes according to its domain

For many network models, it is theoretically possible to encode qualitative variables by assigning few values to the same neuron. For instance, if a qualitative attribute takes three possible values, we might code the first as fully off, the second as half on, and the last one as fully on. Though many nets are capable of reacting to inputs coded in this way, the learning is usually slowed down considerably when this is done.

This approach has the following five main drawbacks: 1) It is unable to determine the similarity information among the qualitative values. For example, the transformed relation does not show that Coke is more similar to Pepsi than Mocca. 2) When the domain of a qualitative attribute is large, transforming it to a set of binary attributes increases the dimensionality of the relation, resulting in wasting storage space and increasing training time. 3) It is hard to maintain the new schema. When the domain of an attribute changes, the transformed relation schema needs to be changed too. For instance, if "juice" is added to the domain of Favorite-Drink, an additional attribute "juice" needs to be included in the transformed relation schema. 4) New binary attributes are unable to reflect the semantics of the original attribute. For example, after the transformation, the four binary attributes cannot express the meaning of Favorite-Drink. 5) When the number of the categories is large, the data vectors are all similar to each other. For example, suppose that we have eight categories. The activation vector for a case belonging to the third category would be (0, 0, 1, 0, 0, 0, 0, 0), while another case's activation vector might be (0, 0, 0, 0, 0, 0, 1, 0). The coordinates in six of the eight dimensions are exactly the same for both cases. Now suppose that we are trying to teach a neural network to respond with such vectors as outputs. If it simply responds to each case by turning off all its outputs, only one of them will be wrong. Accordingly, this will produce a relatively small mean square error.

To overcome the aforementioned shortcomings, and to consider the heterogeneous data simultaneously, the generalized SOMs (denoted as GSOMs) have been proposed [5]. These maps adapt a general distance representation structure, called distance hierarchy to facilitate the distance computation. This hierarchy has been proved to be general distance representation mechanism for both qualitative and quantitative values. It is composed of nodes and links; where higher-level nodes represent more general concepts while lower-level nodes represent more specific concepts. An example of such hierarchy is schematized in figure 3. All the attributes of the training dataset and their corresponding components of the GSOM units are both associated with a distance hierarchy. To compute the distance between a training pattern and a GSOM unit, the attribute values of the pattern and the corresponding components of the GSOM unit should be mapped to their associated distance hierarchies, according to the method well-illustrated in [5]. Then, the distance is computed by aggregating the distances between the mapping points in their hierarchies.

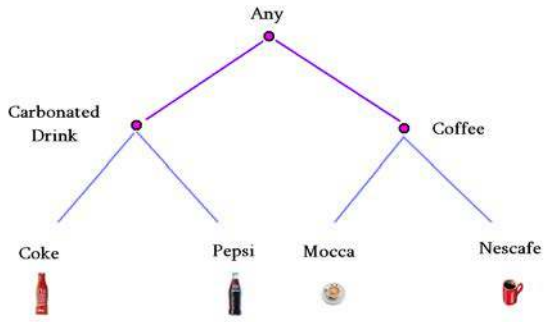

Fig. 3 An example of a hierarchical structure

In spite of the fact that GSOM are able to easily measure the distance between the qualitative as well as the quantitative variable in an efficient manner as proved in [5], and by overcoming all the drawbacks of one-of-n encoding technique, their uses, however, is limited to the crisp values assigned in a certain and precise manner. i.e. this extended version of Kohonen maps stands incapable to deal with the different types of information imperfection (missing data, imprecision, probabilistic uncertainty, etc.).

Actually, the imperfection of the information elements in database objects has almost been neglected and ignored, and the majority of the proposed algorithms and methods assume that in the worst case the variables can be cleaned and prepared in order to get a perfect training set. This optimistic point of view cannot be applied to a great deal of the everyday databases for two main reasons. On the one hand, there are always some attributes that cannot be estimated exactly (precisely) because of the measuring instrument tolerance, or the expert uncertainty and doubt, so they can be given as a vague or as imprecise values modeled by possibility

distributions. On the other hand, it is common to find missing values of the attributes in a data set. Deleting the fields or the attributes that contain such values might decrease the size of the learning base to a great deal plaguing the learning process. Estimating the values of the missing data before the learning process might be complicated, long, and uncertain.

In reality, only few attempts and efforts have carried out to seriously consider the ill-defined variables, like the fuzzy-neuro approach proposed in [6]. This approach supposes that the input feature values can be described in terms of some combination of membership values for the linguistic properties that characterize each of them. For instance, instead of presenting the vector $X_i = [x_{i1} \quad x_{i2} \quad ... \quad x_S]$ as the input to the SOM, the membership degrees of all its components to the characteristic properties of each of them are calculated. For instance, if we suppose that each variable is described via the same three properties "low", "medium" and "high" as shown in the example depicted in figure 4, then the input of the SOM will be

$$X_i = [\mu_{low(x_{i1})}(X_i), \ \mu_{medium(x_{i1})}(X_i), \ \mu_{high(x_{i1})}(X_i), \ldots \\ \ldots, \ \mu_{low(x_{iS})}(X_i), \ \mu_{medium(x_{iS})}(X_i), \ \mu_{high(x_{iS})}(X_i)] \quad (2)$$

As we might notice from this simple example, the number of the input neurons required will notably be increased. Instead of one input neuron for each variable, *n* input units is needed, where *n* is the number of the describing properties of the considered variable. As the training time and the storage space are strongly impacted by the number of the neurons, this technique could be considerably be slowed in some applications in data mining, and we must look for another strategy to better prepare the data at the inputs of the SOM.

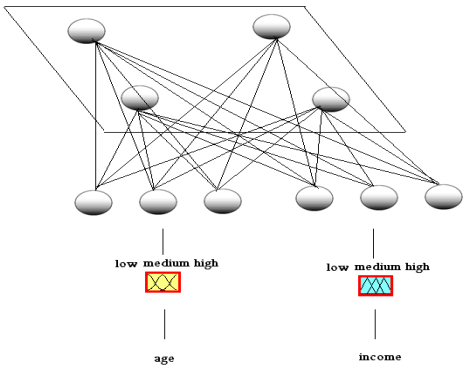

Fig. 4 A Kohonen map used to handle the ambiguity of information elements

In fact, data preparation can make the difference between a SOM that trains in few days and performs quite well, versus another that works in few minutes and performs excellently, since the training time can often be expensive, especially when the SOM cannot achieve a good learning. Consequently, there is an imposed need to find an approach that considers the heterogeneous and the imperfection of information elements, providing robust, simple, low-cost, and fast solutions. This issue is extremely fundamental in all soft-computation methods.

2.2 SOM Information Elements Visualization

To visualize the cluster shape and structure of a data cloud, and to achieve an efficient exploratory data analyses, several techniques have been proposed in the literature. These techniques are usually based on vector projection, using physical coordinates, color coding, etc. However, the most commonly used strategy to visualize the clusters on the SOM is distance matrices. In this technique, the distances between each unit *i* and the units in its neighborhood *R* are calculated:

$$D_i = \left\{ \left\| W_i - W_j \right\| \setminus j \in R, j \neq i \right\}$$

The distances, or for example the median of these distances [7], for each map unit are typically visualized using color, although other techniques are also possible [8]. The unified distance matrix (U-distance-matrix) [9] visualizes all distances between each map unit and its neighbors. This is possible due to the regular structure of the map grid. The cluster borders can be identified as "mountains" of high distances separating the "valleys" of low distances that represent the clusters themselves. It is also possible to use a unified similarity matrix. In this case, the significations are inversed. Figure 5 presents examples of both spatial and color projections as well as the U-matrix. We can see that it is hard to see local details in the dense areas using the PCA, except in the interactive visualization environments where the user can zoom in on the interesting details. Contrary to PCA-projection, the map grid has equal amount of space for each map unit, and hence, map units even in the dense areas can be seen clearly.

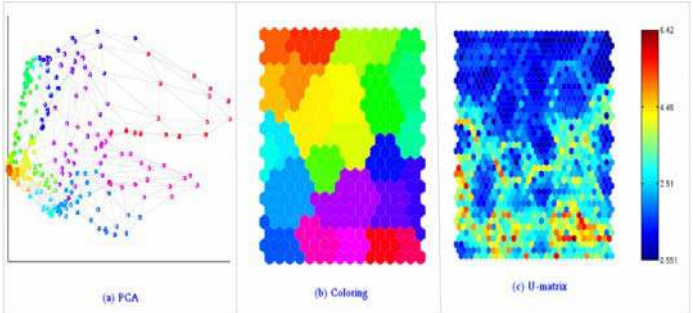

Fig. 5 SOM cluster visualization models using the PCA-projection, the coloring, and the U-distance-matrix

3 Possibility theory

Possibility theory [10-15] provides a method to formalize subjective uncertainties of events, that is to say a means of assessing to what extent the occurrence (the realization) of an event is possible and to what extent we are certain of its occurrence, without having however the possibility to measure the exact probability of this realization because we don't know an analogous event to be referred to, or because the uncertainty is the consequence of observation instrument reliability absence. Let's attribute to each event defined on the universe of discourse Ω (in other words to each element belonging to $\rho(\Omega)$) a coefficient ranging between 0 and 1 assessing to which degree the occurrence of an event is possible, where the value "1" means that the event is completely possible, while the value "0" means that the event is impossible. To define this coefficient, we introduce the possibility measure Π which is a function defined over $\rho(\Omega)$, taking values in $[0,\ 1]$, such that:

Axiom 1: $\Pi(\phi) = 0$ (3)

Axiom 2: $\Pi(\Omega) = 1$ (4)

Axiom3: $\forall A_1, A_2, ... \in \rho(\Omega)$

$\Pi(\cup_{i=1,2,..} A_i) = \sup_{i=1,2,...} \Pi(A_i)$ (5)

where sup indicates the supremum of the concerned values.

We can say that the possibility measure is totally defined, if we can attribute a possibility coefficient to all the singletons of Ω. Consequently, the possibility distribution function π defined on Ω, whose values are included in $[0,1]$, such that $\sup_{x\in\chi} \pi(x) = 1$ must be defined. As a result the function Π can be defined form the function π by:

$\forall A \in \rho(\Omega)\ \ \Pi(A) = \sup_{x\in A} \pi(x)$ (6)

Reciprocally, π can be defined form Π by:

$\forall x \in \Omega\ \ \pi(x) = \Pi(\{x\})$ (7)

We should also mention here that the characteristic function of a subset from Ω can be considered as a possibility distribution π defined on Ω. To calculate the possibility degree of the couple (x, y) given that $x \in \Omega_1$ and $y \in \Omega_2$ where Ω_1, Ω_2 are two non-interactive universes of discourse, the conjoint possibility distribution defined on the Cartesian product $\Omega_1 \times \Omega_2$ should be calculated from:

$\forall x \in \Omega_1\ \forall y \in \Omega_2\ \ \pi(x, y) = \min(\pi_\chi(x), \pi_\gamma(y))$ (8)

In fact, the possibility measure is not sufficient to describe the incertitude of the realization of an event, because sometimes the realization of both the event *A* and its complement A^C could be completely possible simultaneously ($\Pi(A) = 1$ and $\Pi(A^C) = 1$ at the same time). This means that in this particular case it is impossible to take a decision concerning the realization of *A* depending on the estimated possibility measure. For this reason, another function, defined on $\rho(\Omega)$, whose values are included in $[0,\ 1]$ and which is called the necessity measure (denoted *N*) is defined as follows:

Axiom 1: $N(\phi) = 0$ (9)

Axiom 2: $N(\Omega) = 1$ (10)

Axiom 3: $\forall A_1 \in \rho(\Omega) \quad \forall A_2 \in \rho(\Omega)$

$N(\cap_{i=1,2,....} A_i) = \inf_{i=1,2,...} N(A_i)$ (11) where *inf* stands for infimum.

3.1 Possibility-Based Similarity estimation

Suppose that we have two objects O_j and O_k containing "S" attributes (O_j represents the weight vector of Kohonen network, and O_k represents the input vector for example):

$O_j = [x_{1j} \quad x_{2j} \quad . \quad . \quad x_{ij} \quad . \quad . \quad x_{Sj}]$

$O_k = [x_{1k} \quad x_{2k} \quad . \quad . \quad x_{ik} \quad . \quad . \quad x_{Sk}]$.

Each attribute could take a precise or an imprecise value modeled by its possibility distribution, and this value can be quantitative (numeric), qualitative (nominal), or ordinal. The values of some attributes could be unassigned (missing value). Besides, each attribute is associated with a "tolerance function" [11] defined by an expert as a formula or as a table permitting to describe mathematically to which degree we consider that two values of this attribute are similar. An example of tolerance function is the function that we call "close to". Such a function can be defined by the following formula:

$\mu_a(a_x, a_y) = 1 - \frac{|a_x - a_y|}{\Delta}$ if $|a_x - a_y| \le \Delta$ (12)

$\mu_a(a_x, a_y) = 0$ Otherwise

Where Δ is a variable that influences the slope of the function and consequently the notion of "close to". The tolerance function can be also:

- The function of tolerance "True/false": two values of an attribute are similar if they are identical (similarity equals to 1). If the values are different, the similarity is null, this type of functions is used especially when dealing with nominal variables having independent categories. In the case of ordinal variables we must use the function "close to".

- The "ad hoc" tolerance functions which are defined by the experts to reflect their point of view about the similarities between the attributes.

In our approach the similarity between the two objects O_j and O_k can be estimated by means of two measures: the possibility degree of similarity between O_j and O_k that tells us to which degree it is possible that these vectors are similar, and the necessity degree of similarity of these vectors that tells us to which degree we are certain of their similarity. The probability of the similarity between O_j and O_k exists between the necessity degree that represents the lower limit and the possibility degree that represents the upper limit. To calculate the possibility and the necessity degrees of resemblance, we must calculate the local possibility and necessity degrees between their corresponding attributes and aggregate them by taking their average, for example in order to take a decision concerning the total similarity. The local possibility and necessity degrees of similarity between x_{ij} given by its possibility distribution $\pi_{X_j, x_{ij}}(x_{ij}, y)$ and x_{ik} given by its possibility distribution $\pi_{X_k, x_{ik}}(x, x_{ik})$ for all $i \in \{1,2,...,S\}$ are calculated according to the following relations:

Supposing that D is the definition domain of the considered attribute x_i ($U = D \times D$) and that μ is the tolerance function associated to this attribute, the conjoint possibility distribution π_D is calculated as:

$$\pi_D(x_{ij}, x_{ik}) = \min(\pi_{X_j, x_{ij}}(x), \pi_{X_k, x_{ik}}(y)) \quad (13)$$

In this case, the local possibility degree of similarity π_i can be calculated as:

$$\pi_i(x_{ij}, x_{ik}) = \sup_{u \in U}[\min(\mu(u), \pi_D(u))] \quad (14)$$

The local necessity degree of similarity N_i can be calculated as:

$$N_i(x_{ij}, x_{ik}) = \inf_{u \in U}[\max(\mu(u), 1 - \pi_D(u))] \quad (15)$$

We consider that if the value of an attribute is given in one object and is unassigned in the other (the case of missing values), it is completely possible that these values are similar $\pi_i = 1$ but we are entirely uncertain $N_i = 0$. The total possibility (necessity) degree of a certain node j is the average of all the local possibility (necessity) degrees connected to this node.

3.2 Possibilistic Approach

In order to overcome the drawbacks and the limitations of the conventional auto-organizing maps discussed in section 2.1.1, we propose the following simple modifications:

1)- Concerning the similarity between the output units and the input vectors, we can **either** calculate the similarities between all the input vectors using the possibilistic similarity that we have previously proposed (section 3.1) to deal with the heterogeneity and the imperfections of the information elements at the same time [13-15], and then to introduce this possibilistic similarity matrix to the inputs of the SOMs, **or** we model the similarity between each output unit and the input vector, using the necessity and the average necessity and possibility degrees of the resemblance between each variable of the vector and the corresponding weight. In this last case, the winning neuron is the one that has the greatest necessity degree. When two or more neurons have equal necessity degrees, their possibility degrees are compared. It is recommended to introduce the possibilistic inter-variable similarity matrix to the input of the SOMs (the first solution) when the number of the training vector is small, since the number of the input neurons is equal to their number. However, for a significant number of the learning vectors, it is more judicious to apply the second solution to accelerate the learning phase.

2)- As the weights are given as a vector of precise values and the input attributes could modeled or transformed to possibility attributions like $\{\pi(x_1)/x_1, \pi(x_2)/x_2, ..., \pi(x_i)/x_i, ...\}$, we suggest to use the following equation to compute the new weight:

$$w_{new} = w_{current} + \frac{\eta}{\sum \pi(x_i)} \sum [\pi(x_i) \times (x_i - w_{current})]$$

Thanks to the robust tools of possibility theory, these simple and logical modifications can easily handle the aforementioned challenging points assuring the generality of the approach in a simple and fast way as we will show via a concrete example in the following paragraph.

4 Illustrative Example

Consider the following simple example. Suppose that we have a dataset with two attributes, "age" and "income", which have already been normalized (see table 1), and the knowledge about these attributes might be imprecise modeled by a possibility distribution. Suppose that we would like to use a 2 × 2 Kohonen map to represent hidden clusters in the data set. Therefore, we would have the topology shown in Figure 1. This type of data set cannot be solved using the traditional scoring functions that suppose that the compared values have to be crisp. Besides, we cannot manage to apply the weight adjustment equation with such imprecise attributes, given as possibility distributions.

1	x_{11} is about 0.8 figure 6-a	$x_{12} = 0.8$	Older person with high income
2	$x_{21} = 0.8$	$x_{22} = 0.1$	Older person with low income
3	x_{31} is about 0.2 figure 6-b	x_{32} is somehow high figure 6-c	Younger person with high income
4	$x_{41} = 0.1$	$x_{42} = 0.1$	Younger person with low income
5	x_{51} is given by its possibility distribution π_{51} figure 6-d	$x_{52} = 0.8$	Older person with high income

Table 1 The dataset of our example

With such a small network, we set the neighborhood size to zero ($R = 0$), so that only the winning node will be awarded the opportunity to adjust its weight. Also, we set the learning rate η to 0.5. Finally, assume that the weights have been randomly initialized as follows:

$w_{11} = 0.90 \quad w_{21} = 0.80 \quad w_{12} = 0.90 \quad w_{22} = 0.20$

$w_{13} = 0.10 \quad w_{23} = 0.80 \quad w_{14} = 0.10 \quad w_{24} = 0.20$

For the first input vector, we perform the following competition, cooperation, and adaptation sequence:

A. Competition: We compute the necessity and the possibility degrees of similarity between this input vector and the weight vector for each of the four output nodes (see table 2):

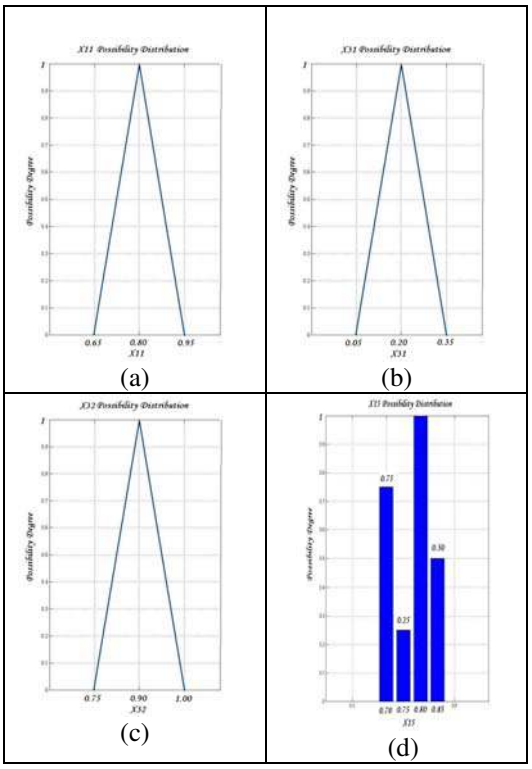

Fig. 6 Possibility distributions of the imprecise values

Node j	$N(w_{1j},x_{j1}), P(w_{1j},x_{j1})$	$N(w_{2j},x_{j2}), P(w_{2j},x_{j2})$	$N(W_j,X_1)$	$P(W_j,X_1)$
1	0.2,0.6	1,1	0.60	0.80
2	0.2,0.6	0,0	0.10	0.30
3	0,0	1,1	0.50	0.50
4	0,0	0,0	0	0

Table 2. The necessity and the possibility degrees (the first vector)

The winning node for this first input record is therefore "node 1", since it maximizes the similarity (modeled by the possibility and the necessity degrees) between the input vector for this record, and the weight vector, over all nodes. Node 1 won the competition for the first record because its weights are more similar to the field values for this record than the other nodes' weights. For this reason, we may expect node 1 to exhibit an affinity for records of older persons with high-income. In other words, we may expect node 1 to represent a cluster of older, high-income persons.

B. Cooperation: In this simple example we have set the neighborhood size R = 0 therefore, only the winning

node, “node 1”, will be able to adjust its weights.

C. Adaptation: For the winning node, “node 1”, the weights are adjusted as follows:

$w_{11,new} = 0.85$, $w_{21,new} = 0.8$

For the next input vector $X_2 = [0.8,0.1]$, see table 3:

Node j	$N(w_{1j},x_{j1}), P(w_{1j},x_{j1})$	$N(w_{2j},x_{j2}), P(w_{2j},x_{j2})$	$N(W_j,X_2)$	$P(W_j,X_2)$
1	0.333,0.333	0,0	0.1665	0.1665
2	0.333,0.333	0.333,0.333	0.333	0.333
3	0,0	0,0	0	0
4	0,0	0.333,0.333	0.1665	0.1665

Table 3. The necessity and the possibility degrees (the second vector)

Node 2 won the competition because its weights (0.9, 0.2) are more similar to the field values for this record than the other nodes’ weights. As a result:
$w_{12,new} = 0.85$, $w_{22,new} = 0.15$.

For the third input vector, see table 4:

Node j	$N(w_{1j},x_{j1}), P(w_{1j},x_{j1})$	$N(w_{2j},x_{j2}), P(w_{2j},x_{j2})$	$N(W_j,X_3)$	$P(W_j,X_3)$
1	0,0	0.2,0.6	0.1	0.3
2	0,0	0,0	0	0
3	0.2,0.6	0.2,0.6	0.2	0.6
4	0.2,0.6	0,0	0.1	0.6

Table 4. the necessity and the possibility degrees (the third vector)

Figure 7 shows us the main steps of possibility and necessity degree calculation.

Node 2 wins the competition, and as a result: $w_{13,new} = 0.15$, $w_{23,new} = 0.85$.

For the vector $X_4 = [0.1,0.1]$, see table 5.

Node j	$N(w_{1j},x_{j1}), P(w_{1j},x_{j1})$	$N(w_{2j},x_{j2}), P(w_{2j},x_{j2})$	$N(W_j,X_4)$	$P(W_j,X_4)$
1	0,0	0,0	0	0
2	0,0	0.333,0.333	0.1665	0.1665
3	1,1	0,0	0.5	0.5
4	1,1	0.333,0.333	0.667	0.667

Table 5. The necessity and the possibility degrees (the fourth vector)

Node 3 wins the competition, and as a result: $w_{14,new} = 0.1$, $w_{24,new} = 0.15$.

Finally, for X_5, see table 6:

Node j	$N(w_{1j},x_{j1}), P(w_{1j},x_{j1})$	$N(w_{2j},x_{j2}), P(w_{2j},x_{j2})$	$N(W_j,X_5)$	$P(W_j,X_5)$
1	0.333,0.50	1,1	0.667	0.75
2	0.333,0.333	0,0	0.1665	0.1665
3	0,0	1,1	0.50	0.50
4	0,0	0,0	0	0

Table 6. the necessity and the possibility degrees (the last vector)

Node 1 wins the competition, so given that $w_{11,current} = 0.85$ and x_{11} is given as possibility distribution $\left\{\frac{0.75}{0.70}\ \frac{0.25}{0.75}\ \frac{1}{0.80}\ \frac{0.50}{0.85}\right\}$ (figure 2-d), the new weight $w_{11,new}$ is calculated as following:

$$w_{11,new} = 0.85 + \frac{0.5}{(0.75+0.25+1+0.50)}[0.75(0.70-0.85)+0.25(0.75-0.85)+(0.80-0.85)+0] = 0.81$$

Given that $w_{21} = 0.80$ and $x_{12} = 0.80$ the new weight $w_{21,new}$ won’t change.

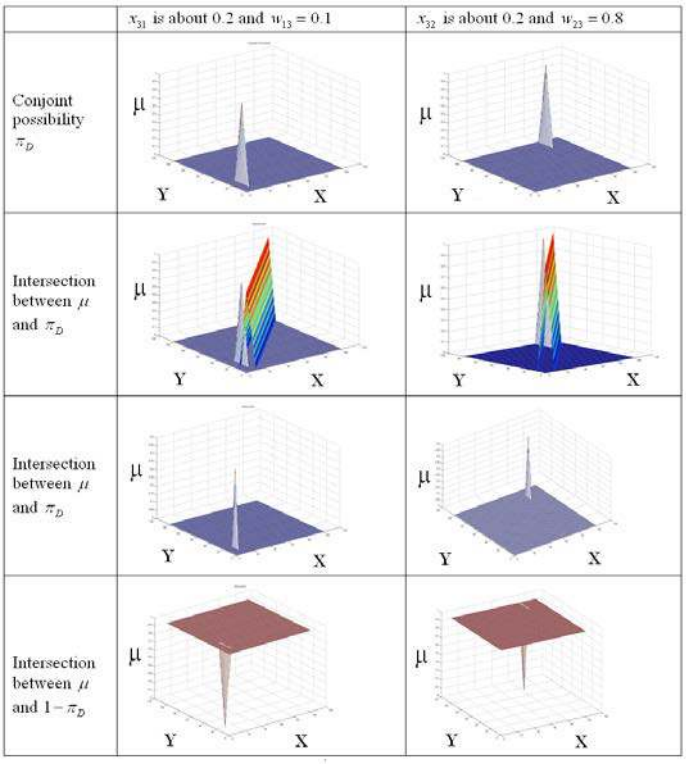

Fig. 7 Local possibility and necessity calculation. X represents the first fuzzy proposition concerning the value of the attribute in the first object. Y represents the second fuzzy proposition concerning the value of the same attribute in the second object. μ represents the possibility or the necessity degree.

Notice that even if the value of an attribute is given as a probability distribution, it can easily be transformed to possibility distribution using a suitable transformation [16], and the estimation of the similarity will then be simple and straightforward unlike the prior conventional methods that stand paralyzed in front of such cases.

5 Medical Application

The proposed method will be applied in the following to the medical gastroenterology database of the hospital "Morvan" in Brest, France. This database [17] is briefly described in the following subsection, and the results will be presented in section 6.2.

5.1 Medical Dataset Description

The possibility-based similarity modeling exploited by this clustering algorithm was tested on a digestive endoscope atlas of documented endoscopic lesions descriptions, and scene information of the upper gastrointestinal tract, esophagus, stomach, and duodenum [17]. Database images attributes description characterize observed anomalies or lesions identified by an expert, according to a well defined and exhaustive description structure:

- Object location: anatomic (longitudinal), position in the organ (axial), and distance from the teeth.
- Repeated objects: number of identical objects and spatial organization.
- Object aspect: shape and edge, dominant color and color regularity, relief and regularity of relief, sizes (major, minor axes and thickness), axes ratio and major axis orientation, height, motility, effect of insufflations, and consistency.
- Relation with adjacent organ: color contrast, texture contrast, and consequences on the lumen.

These descriptors represent 24 features, summing 145 distinct values for simple objects. Attributes are either semantically or numerically coded, with the help of an adapted interface by an expert physician. Complex objects are defined when two or more simple objects are related on the same visual scene. Each one has its own attributes and a spatial relation (closeness and order, rated as: into, in contact, in contact and upward, around, and around and upward) links them, resulting in 33 features and 206 different values. Depending on the secondary objects types, other relationships may appear like relative sizes and consistency, combined with the absence of some features in some objects, uncertain and incomplete descriptions.

Endoscopic diagnosis relies on the analysis of associations between these elementary lesions and the medical context, which includes sex, age, clinical antecedents, consultation circumstances, and symptoms. Complementary exams may be necessary to determine whether the initial diagnosis is confirmed or refused. These exams include histological examination of biopsy specimens, coloration of the digestive mucosa, and morphological or functional evaluations. Once processed, all these elements and diagnostic decisions are then recorded on a medical report. Among the documented lesions and pathologies we find: dilated lumen, stenosis, extrinsic compression, web, ring, hiatal hernia, undigested food, liquid blood, blood clot, z-line, spot, circular Barrett's, moniliasis, simple erosion, ulcer, and Petechial mucosa.

Lesions and diagnosis are intended to be independently described under this scheme, even though practical experience shows that endoscopic findings may point towards a particular diagnosis, whereas other diagnosis alternatives including the same lesions could also be specified. For this reason, the digestive endoscope atlas also defines specialized findings, which are classes that describe more in detail the generic diagnosis. Validated by medical experts, the atlas consists of 89 endoscopic findings, 126 endoscopic diagnoses, and 118 specialized findings, a priori descriptions.

To illustrate our results in a clear and simple manner, a subset of cases (*CB*) belonging to the main image database were processed. This facilitates understanding the studied possibilistic clustering approaches, the results graphic representation, and emphasizes the general character of the proposed approach. Defined as $CB = \{p_1, p_2, \ldots, p_{18}\}$, the subset of cases contains 18 described images, presented in figure 8. *CB* is structured in the following manner according to the ground-truth provided by the specialist: $P_1 = \{p_1, p_2\}$ corresponds to the "Dilated Lumen" pathology; $P_2 = \{p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}\}$ conforms with the description of "Esophagus Stenosis"; $P_3 = \{p_{11}, p_{12}, p_{13}, p_{14}\}$ is a set of images that represent the "Extrinsic Compression" pathology; $P_4 = \{p_{15}\}$ describes the "Web Shape" pathology; $P_5 = \{p_{16}, p_{17}, p_{18}\}$ is a set of images on which the "Ring Shape" pathology is visible.

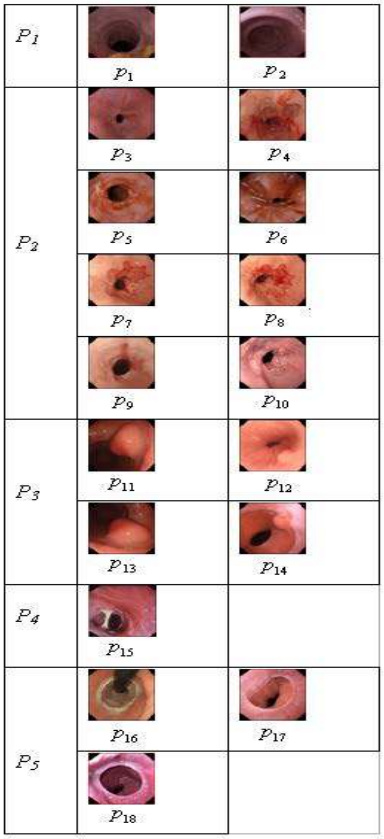

Figure 8. The medical dataset.

5.2 Experimental Results

In the beginning, the proposed method has been applied to the dataset *CB* described in the precedent section using the batch training in which the dataset is presented entirely in each epoch. Firstly, we assume that we have a 5×4 map and then a 15×15 output units. The visualization models briefly illustrated in section 2.2 will be used to display the experimental outcome.

Figure 9 schematizes the unified similarity matrix using the color codes depicted in the vertical color bar at the upper left corner of figure 9. The similarity between each object and the output unites in the map are also coded with colors according to their associated bars in this figure. As it is clearly shown, the objects having the same pathology belong to the same or neighbor units. Looking closely to the coloring maps of these objects that reassemble to fingerprints, we can notice that colored maps of all the objects having the same class are approximately the same, i.e. the similar objects have similar fingerprints that differ from one pathology to another. Figure 10 depicts another type of results representation using bar charts in each output unit that present its similarity to all the objects. Again, it can be remarked that units having objects of the same class are more similar than the other ones. The visual presentation of these information elements using the principal component analysis in a 3-dimensional coordinate space is also plotted in figure 11. Another example is given in figures 12 and 13 in which we apply the same steps to a 15×15 unit map. The same remarks and observations can be deduced. The fingerprints of each object are clearer in this example.

At last, to show the generality and the validity of our approach in all the types of training, all the aforementioned steps are applied to a 5×4 unit map using the sequential mode of training (figures 14 and 15). It is clear that we get similar analysis and conclusions from the observation of the depicted plots.

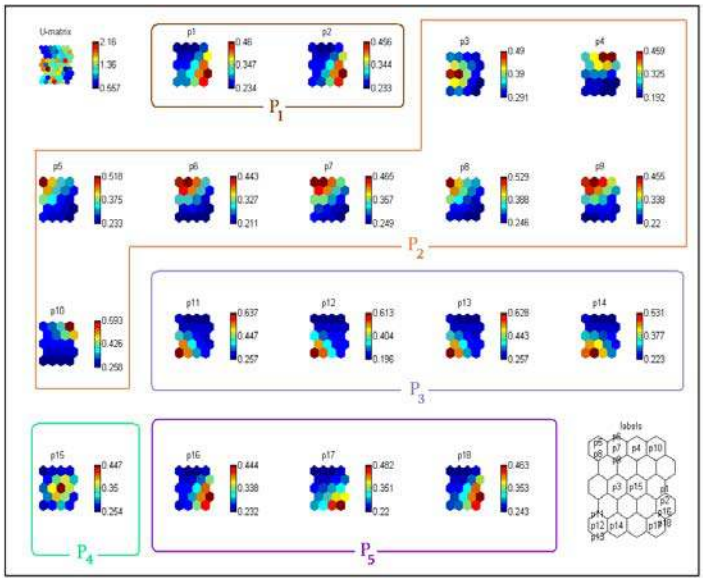

Fig. 9 Visualization of the SOM of the gastroenterology dataset using a 5×4 unit map (batch training). U-matrix on the top left, then component planes, and map unit labels on bottom right

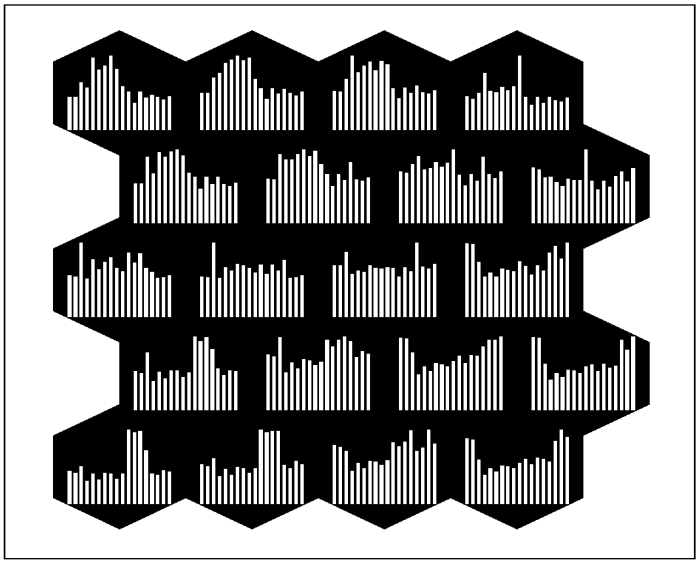
Fig. 10 The bar charts in each output unit show its similarity to each object of the 5×4 unit map (batch training)

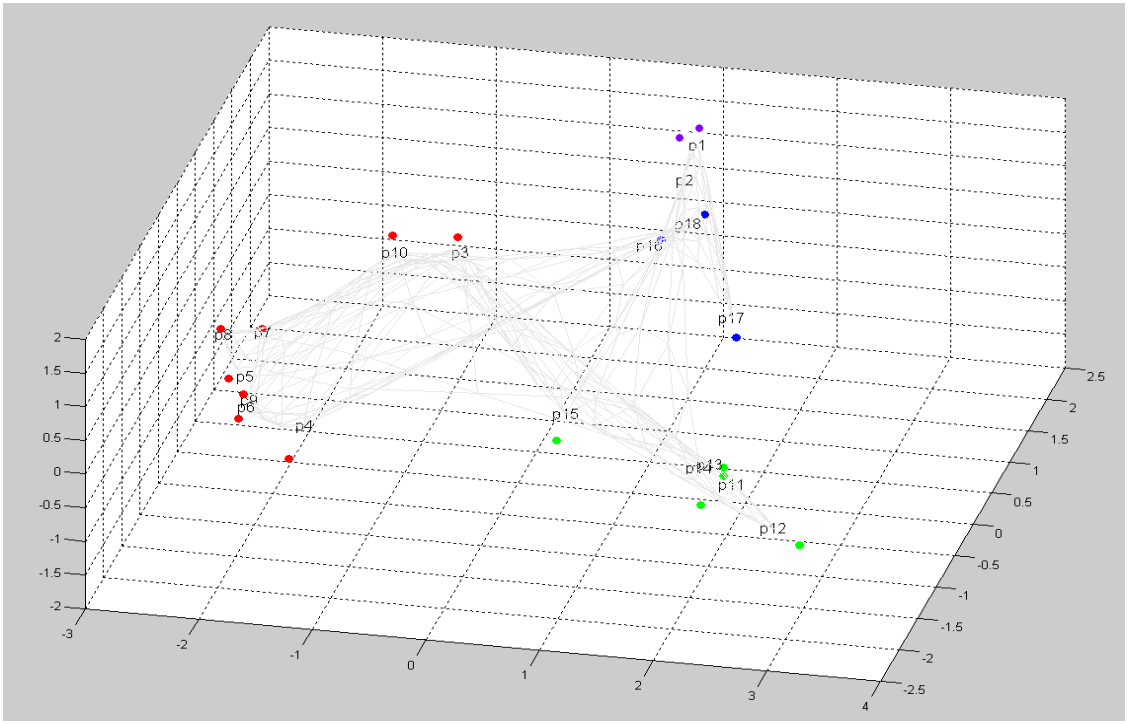

Fig. 11 Projection of the gastroenterology dataset to the sunspace spanned by its three eigenvectors with greatest eigenvalues.
The different pathologies have been plotted using different colors.
Neighboring map units are connected with lines. Labels associated with map units are also shown

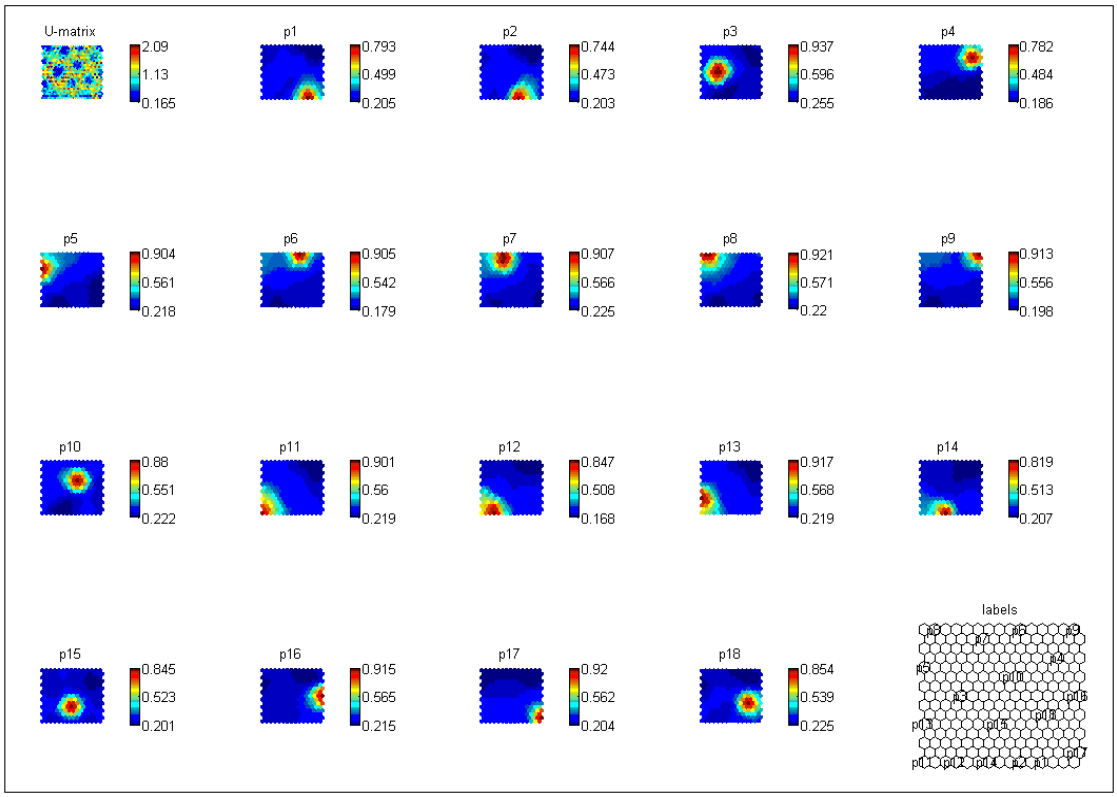

Fig. 12 Visualization of the SOM of the gastroenterology dataset using 15×15 unit map (batch training).
U-matrix on the top left, then component planes, and map unit labels on bottom right

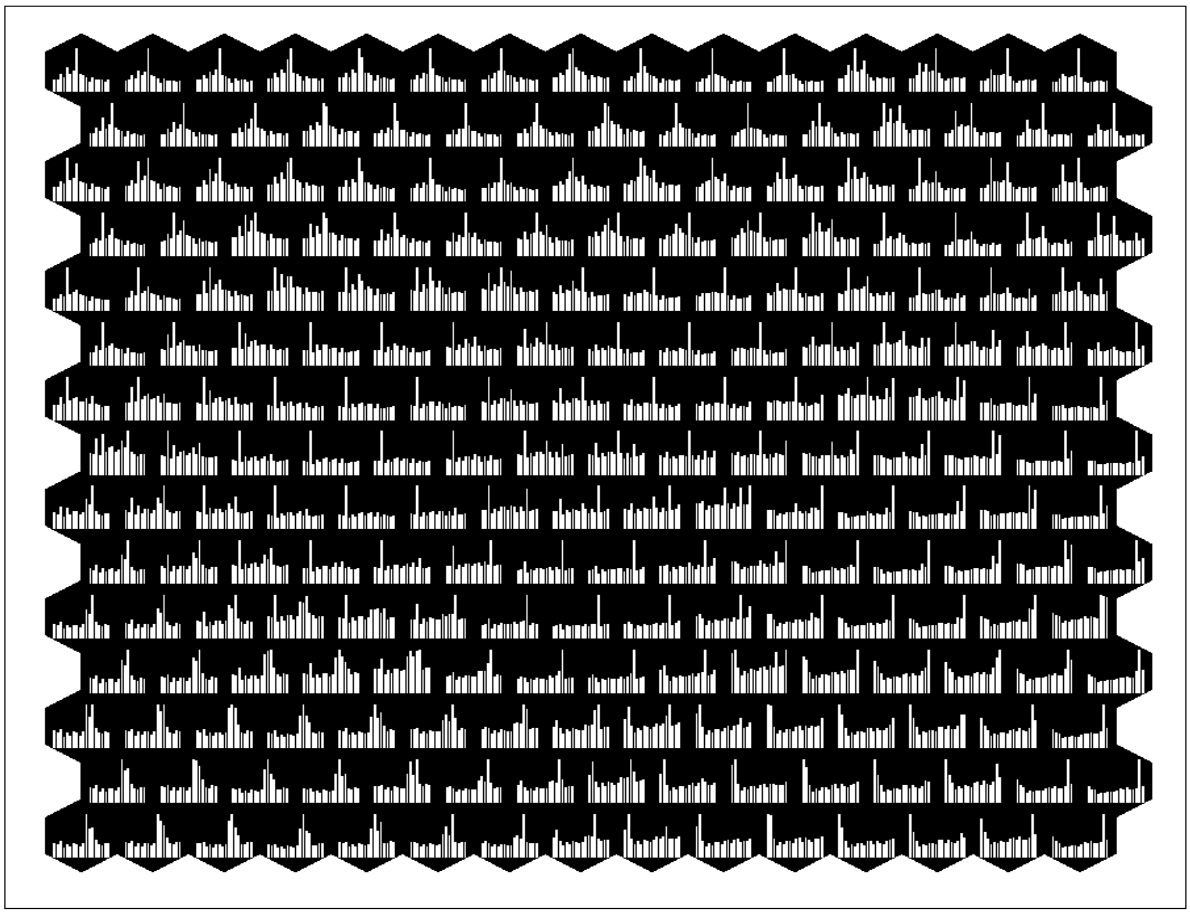

Fig. 13 The bar charts in each output unit of a 15×15 unit map show its similarity to each object (batch training)

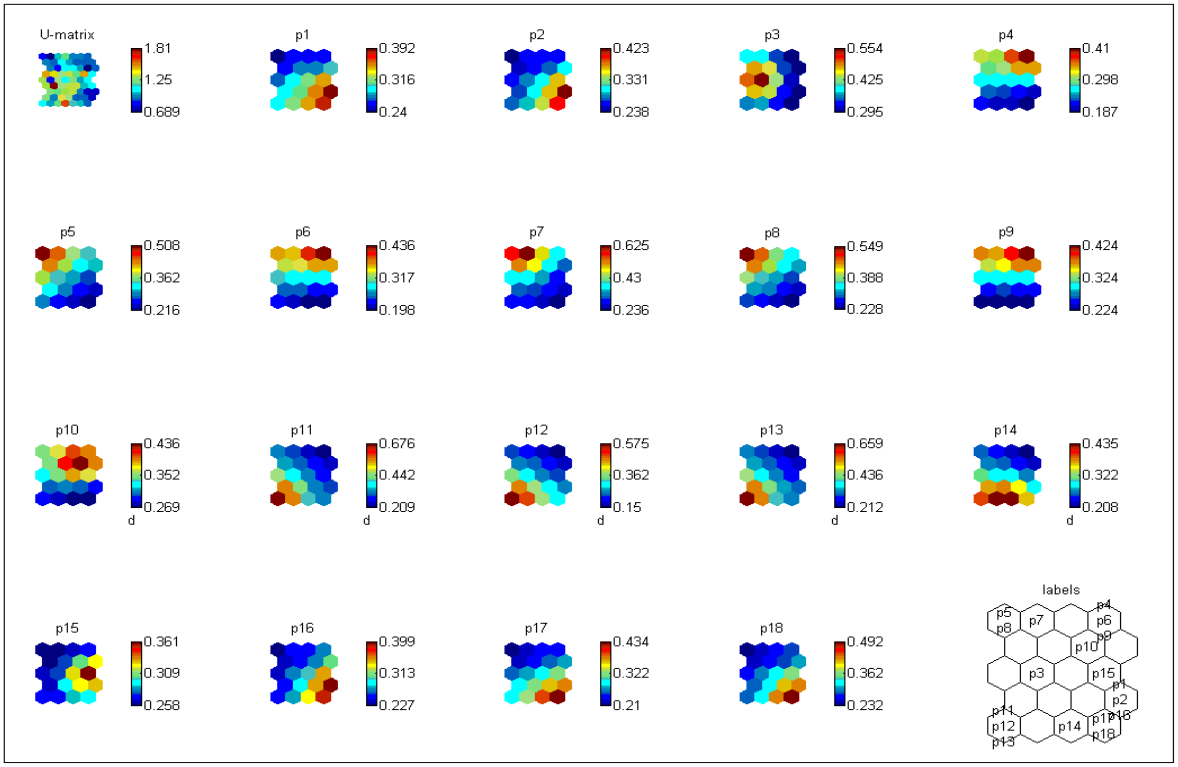

Fig. 14 Visualization of the SOM of the gastroenterology dataset using 5×4 unit map (sequential training)

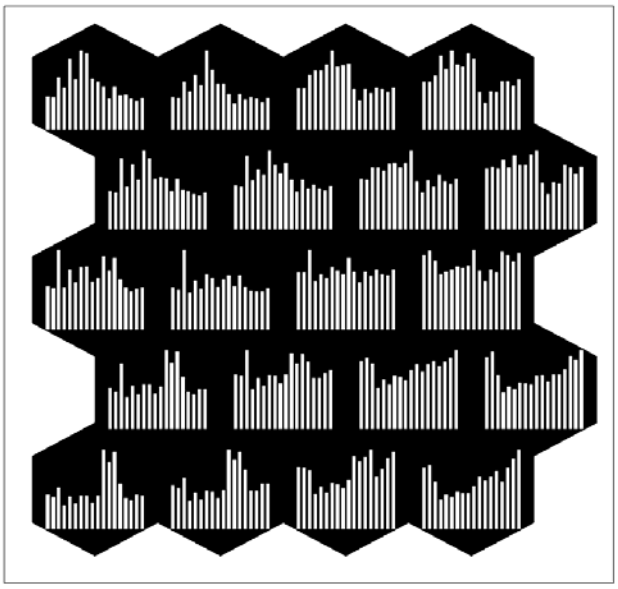

Fig. 15 The bar charts in each output unit of a 5×4 unit map show its similarity to each object (sequential training)

6 Discussion and perspectives

Kohonen networks are a type of the unsupervised artificial neural networks that are trained to produce and to visualize low-dimensional views of high-dimensional data, akin to multidimensional scaling (MDS). These networks have been widely used in different signal processing and data mining applications thanks to their simplicity and performance. Nevertheless, calculating the similarity between the input vector and the weight vector using the scoring function limits their use especially when dealing with imperfect and heterogeneous data. For instance, the traditional networks cannot solve the case presented in the example given in the section 4 for instance.

To overcome this drawback and to ameliorate the performance of these networks, we modeled the similarity by two fuzzy measures: the possibility and the necessity degrees in order to insure a wide use of Kohonen networks taking into account the imperfection of the variables (uncertain, imprecise or missing data). In fact, these degrees could estimate the similarity between the qualitative, quantitative and the ordinal variables. Thus, they can be used without any modification in any other domain of signal and image processing (image retrieval for example). They can also be used in all the data mining techniques that demand the estimation of similarity (the k-nearest neighbor for instance). In addition to their generality and their similarity to human reasoning, these measures insure a fast computation time which is very important in neural network applications since they are based fundamentally on basic mathematical operators (max, min, etc.) and because we don't need additional preprocessing phases to prepare the data and to estimate the missing values or to deal with the imprecise observations. In fact, this last operation (data cleaning and data preparation) could be very cost and complicated, and could reduce scientifically the size of the training set especially when we have missing values in many attributes of the records.

The proposed approach has been applied to a gastroenterology dataset, and the classes of the objects have correctly been assigned. It has been shown that the objects of the same class flock in neighbor output units. This may enable the doctors to study the similarity between the objects themselves and between the classes of the dataset as well, in order to build knowledge databases. Then doctors can study the characteristics and the properties of the features of the similar objects in order to get new potential unknown rules as future work using the techniques of rule extraction algorithms in data mining. Briefly, these characteristics of Kohonen networks and the possibility measures can open new directions for future researches and can solve practical problems.

Acknowledgement

The authors are very grateful to C. Le Guillou and J-M. Cauvin from CHU Brest University Hospital, for the access to the gastroenterology database

References:

[1] T. Kohonen, *Self-organizing maps*, third edition, Springer-Verlag New York, inc. 2001.

[2] D. Larose, *Discovering knowledge in data: an introduction to data mining*, Wiley, pp. 163-179, 2004.

[3] H. Ritter, *Self-organizing feature maps: Kohonen maps.* Arbib, ed., The handbook of brain theory and neural networks, pp. 846-851, MIT press, Cambridge, 1995.

[4] H. Wang, *Classification and Clustering for Knowledge Discovering*, Springer, Studies in Computational Intelligence 4, 2005.

[5] C.C. Hsu, *Generalizing Self-Organizing Map for Categorical Data*, IEEE Transactions on Neural Networks, vol. 17 (2), pp. 294-304, 2006.

[6] S. Mitra, *Self-Organizing Neural Network as a Fuzzy Classifier*, IEEE Transactions on Systems, Man, and Cybernetics, vol. 24 (3), pp. 384-399, 1994.

[7] M. A. Kraaijveld, J. Mao, and A. K. Jain. *A nonlinear projection method based on kohonen's topology preserving maps*. IEEE Transactions on Neural Networks, vol. 6(3), pp. 548–559, 1995.

[8] D. Merkl and A. Rauber. *Alternative ways for cluster visualization in selforganizing maps*. In Proceedings of the Workshop on Self-Organizing Map, pp. 106–111, 1997.

[9] A. Ultsch and H. P. Siemon. *Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis*. In Proceedings of International Neural Network Conference, p.p. 305–308, 1990.

[10] B. Bouchon-Meunier, *Uncertainty Management in Medical Applications*, Chapter 1 in Nonlinear Biomedical Signal Processing, Akay, M., (ed.), IEEE Press, 2000.

[11] H. Rakoto, J. Hermosillo, M. Ruet, *Integration of experience based decision support in industrial processes*. IEEE Proceeding of SMC'02, vol. 7, pp. 6-12, 2002.

[12] A. Dahabiah, J. Puentes, B. Solaiman, *Digestive Casebase Mining Based on Possibility Theory and Linear Unidimensional Scaling*. 8th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, pp. 218-223, 2009

[13] A. Dahabiah, J. Puentes, B. Solaiman, *Possibilistic Evidential Clustering*. 8th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, pp. 212-217, 2009

[14] A. Dahabiah, J. Puentes, B. Solaiman, *Digestive database evidential clustering based on possibility theory*. WSEAS transactions on biology and biomedicine, vol. 5 (9), pp. 239-248, 2008

[15] A. Dahabiah, J. Puentes, B. Solaiman, *Possibilistic Pattern Recognition in a Digestive Database for Mining Imperfect Data*. *WSEAS Transactions on Systems,* vol. 8 (2), pp. 229-240, 2009

[16] M. H. Masson, T. Denoeux, *Inferring a Possibility Distribution from Empirical Data*, Fuzzy Sets and Systems, vol. 157 (3), pp. 319-340, 2006

[17] C. Le Guillou, J.M. Cauvin, *From Endoscopic Imaging and Knowledge to Semantic Formal Images*, Springer Lecture Notes in Computer Science, vol. 4370, pp. 189-201, 2007