Neurocomputing 147 (2015) 47–59

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Automatic design of interpretable fuzzy predicate systems for clustering using self-organizing maps

Gustavo J. Meschino [a,*], Diego S. Comas [b,c], Virginia L. Ballarin [b], Adriana G. Scandurra [a], Lucía I. Passoni [a]

[a] Bioengineering Laboratory, Facultad de Ingeniería, Universidad Nacional de Mar del Plata, Juan B. Justo 4302, Mar del Plata B7608FDQ, Argentina
[b] Digital Image Processing Group, Facultad de Ingeniería, Universidad Nacional de Mar del Plata, Juan B. Justo 4302, Mar del Plata B7608FDQ, Argentina
[c] Consejo Nacional de Investigaciones Científicas y Técnicas, CONICET, Argentina

ARTICLE INFO

Article history:
Received 3 April 2013
Received in revised form
3 December 2013
Accepted 2 February 2014
Available online 21 May 2014

Keywords:
Self-organizing maps
Clustering
Fuzzy logic
Fuzzy predicates
Degree of truth

ABSTRACT

In the area of pattern recognition, clustering algorithms are a family of unsupervised classifiers designed with the aim to discover unrevealed structures in the data. While this is a never ending research topic, many methods have been developed with good theoretical and practical properties. One of such methods is based on self organizing maps (SOM), which have been successfully used for data clustering, using a two levels clustering approach. Newer on the field, clustering systems based on fuzzy logic improve the performance of traditional approaches. In this paper we combine both approaches. Most of the previous works on fuzzy clustering are based on fuzzy inference systems, but we propose the design of a new clustering system in which we use predicate fuzzy logic to perform the clustering task, being automatically designed based on data. Given a datum, degrees of truth of fuzzy predicates associated with each cluster are computed using continuous membership functions defined over data features. The predicate with the maximum degree of truth determines the cluster to be assigned. Knowledge is discovered from data, obtained using the SOM generalization aptitude and taking advantage of the well-known SOM abilities to discover natural data grouping when compared with direct clustering. In addition, the proposed approach adds linguistic interpretability when membership functions are analyzed by a field expert. We also present how this approach can be used to deal with partitioned data. Results show that clustering accuracy obtained is high and it outperforms other methods in the majority of datasets tested.

1. Introduction

Clustering aims to discover unrevealed structures in data and is a never-ending research topic. It is a major task in exploratory data mining and new approaches are constantly proposed, because the usage and interpretation of clustering depend on each particular application [1]. Clustering is currently applied in many fields, such as web and text mining, business and marketing, machine learning, pattern recognition, image analysis and segmentation, information retrieval, and bioinformatics (see e.g. [2–4]). In many complex problems, general clustering techniques are not able to adequately discover groups when directly applied on the data [5].

The self-organizing maps (SOM), introduced by Kohonen in 1982 [6], are widely used, unsupervised and nonparametric neural network, with remarkable abilities to remove noise, outliers, and deal with missing values. The SOM training process generates simultaneous clustering and projection of high-dimensional data. SOM have been successfully used in data clustering via two-level clustering approaches. The first level consists in training a SOM with a dataset. In the second level various techniques have been used, such as a second SOM [7], crisp-clustering methods [5,8,9] or fuzzy clustering techniques [10–12]. The second level links cells of the first-level SOM to form clusters. Then each datum is typically associated to the cluster assigned to its Best Matching Unit (BMU). Vectors of the codebook can be interpreted as "protoclusters," which are combined to form the actual clusters.

In the two-level clustering approach, the SOM in the first level generates a projection of the original data which makes a general clustering technique suitable in the second level. One advantage of SOM in the two-level approach is the reduction in the computational cost [5]. Even considering a small data package, some clustering algorithms become intractable. Grouping prototypes instead of grouping data directly is a solution for this problem.

* Corresponding author. Tel.: +54 223 4816600; fax: +54 223 481 0046.
E-mail addresses: gmeschin@fi.mdp.edu.ar (G.J. Meschino), diego.comas@fi.mdp.edu.ar (D.S. Comas), vballari@fi.mdp.edu.ar (V.L. Ballarin), scandu@fi.mdp.edu.ar (A.G. Scandurra), lpassoni@fi.mdp.edu.ar (L.I. Passoni).

http://dx.doi.org/10.1016/j.neucom.2014.02.059
0925-2312/

Another additional benefit is the reduction in data noise effects, since prototypes are local averages of the data and, therefore, they are less sensitive to random variations than the original data [5].

Clustering systems based on Fuzzy Logic (FL) [13–15] have been used in a wide range of clustering problems, improving the performance of traditional approaches. Based on the fuzzy set theory, FL was proposed by Zadeh [16], who stated that a complex system will be better represented by descriptive variables of linguistic types [17]. Most of the previous works are based on Fuzzy Inference Systems (FIS). The main advantages of these models are (a) they use simple IF–THEN rules to determine the conditions a datum must satisfy to belong to each cluster and (b) FIS allow modeling the data imprecision by way of membership functions. However, in FIS, aggregation and defuzzification operations must be defined, so these models do not constitute a Boolean Logic generalization. Given that defuzzification is a pragmatic combination of operators, it lacks an axiomatic link that justifies the "logic" denomination [18].

Unlike previous works, in this paper we propose the design of a new clustering system in which (a) we use predicate fuzzy logic [19] to perform the clustering task, which is a natural extension of predicate Boolean Logic and (b) the system is automatically designed (unsupervised) [20] using a two-level clustering approach that combines SOM and Fuzzy C-Means (FCM) as a second level clustering method. First, SOM are trained from the original data, considering a SOM with a number of cells much larger than the expected number of clusters. Then the SOM codebook (the set of protoclusters) is clustered. Next, the codebook clustering is analyzed and membership functions and predicates are defined. Thus we obtain a ranked clustering criteria represented as self-discovered fuzzy predicates using data information [20] which consider the behavior of the variables into the different clusters. Hereafter we will name the proposed method SOM-based Fuzzy Predicate Clustering (SFPC).

Given a datum, degrees of truth of fuzzy predicates associated to each cluster are computed using continuous membership functions defined over data features. Finally, the predicate with the maximum degree of truth determines the cluster to be assigned (the first of the ranking). From a linguistic point of view we obtain a ranking for each datum, computing the degree of truth of *"The datum D belongs to cluster k"*, being $k=1, 2, \ldots, K$, and K the amount of clusters. This ranking could also be used to determine the group to which new similar input data (not contained in the training dataset) belongs. Besides, this system allows comparing the degree of membership to the clusters and also assessing the contribution of individual features to the final decision. If required, it allows assessing how far the datum was from being assigned to other cluster (the next one in the ranking).

The proposed approach adds value to previous knowledge-based fuzzy clustering methods [21]. In this case knowledge is discovered from data, obtained using the SOM generalization aptitude and taking advantage of the well-known SOM abilities to discover natural data groupings when compared with direct clustering [5,7,22]. Also SOM provides useful information about the features in each cluster, an interesting property that was used for different applications [23]. In addition, the proposed approach adds linguistic interpretability when the clustering obtained and the knowledge discovered from the membership functions are analyzed by a field expert. This approach can also be used to deal with partitioned data as we present later in the paper, where we explain how the predicate system can join the results obtained from different partitions.

The paper is structured as follows. Section 2 discusses the main works related to SOM-clustering and fuzzy systems. Some important concepts concerning this work on SOM and fuzzy predicates systems are presented in Section 3. Section 4 details with the proposed methodology to design the fuzzy clustering system by way of a SOM–FCM scheme and the method for assessing the clustering quality is explained. In Section 5 we show the accuracy of the results on several datasets, and then we develop two examples of interpretability of membership functions and predicates. Finally we conclude by discussing the results, limitations and future research directions in Sections 6 and 7.

2. Related works

Before presenting the SFPC method proposed in this paper, in this section we discuss relevant works related to SOM-clustering and fuzzy systems applications in clustering.

2.1. SOM-clustering

In the literature, several approaches were developed in the attempt to achieve clusters of data from a trained SOM. The general approach uses a two-level process. In the first level, a SOM is trained. The second level can involve another SOM, making a hierarchical SOM [7] or some crisp-clustering method [5,8,9] which allows linking codebook cells of the first-level SOM in bigger clusters. In other works the SOM fuzzification problem was specifically studied using a FCM algorithm [11] or other fuzzy methods [10,12] as second level clustering.

Lampinen and Oja [7] proposed a clustering method using a multilayer SOM (HSOM) to achieve complex data groupings. They argue that when the abstraction level of the classification task increases, the shapes of the regions associated together become complex, requiring very large amounts of training data to form the class boundaries. In these cases, using unsupervised learning techniques can reduce the required training. They state that as the goal of SOM-learning is not only to find the most representative code vectors for the input space in mean square sense, but at the same time to realize a topological mapping from the input space to the grid of neurons; updating the internal weights of the network tends to preserve the input topographic space. Therefore, neighboring cells are related to nearby data in the original space and a grouping of data can be achieved through the SOM, where topographically nearby cells are associated with the same cluster. In particular, the HSOM algorithm is able to achieve clustering of complex data structures.

Vesanto and Alhoniemi [5] developed a two-level SOM-clustering approach that uses agglomerative clustering in the second level. A large set of prototypes, interpreted as "protoclusters," larger than the number of clusters, is achieved by the SOM in the first level. The protoclusters are combined together to form the actual clusters in the second level. Each data vector of the original dataset belongs to the same cluster as its nearest prototype. While extra abstraction levels yield higher distortion, they also effectively reduce the complexity of the reconstruction task. They indicate some advantages of using SOM in the first level that we cited in the previous section.

On the way to finding the best emergent clustering, Murtagh [8] presented a convenient framework for clustering, applying the contiguity-constrained clustering method in the second-level clustering. A SOM is trained and the prototype vectors are interpreted as clusters defined by a minimum distance criterion, with the cluster centers located on the discretized plane in such a way that proximity reflects similarity. The algorithm searches for the minimum distance between prototype vectors belonging to two different clusters and integrates them into a single group. This recursive algorithm ends when a stop condition defined by the user is reached. This condition may be the number of defined clusters.

Cottrell and Letrémy [9] stated that SOM are appropriate when the available observations have missing values. They say it can be the only possible method when data are extremely sparse. Examples are given, where this method estimates missing values with good accuracy. Then, completed data can be processed using any classical treatment.

Other authors focus on the SOM-codebook fuzzification. Sarlin and Euklund [11] applied a two-level SOM–FCM model, where FCM [24] performs clustering on the units of the SOM grid. The FCM algorithm allows each unit to have a partial membership to all identified overlapping clusters. Authors indicate that this second level fuzzy clustering on the SOM codebook enables the analysis of the membership degree in self-discovering clusters, which is not always easy to judge. In particular, fuzzy clustering enables sensible representation of the real world filled with uncertainty and imprecision. In [12], Sarlin and Eklund used a fuzzified SOM. They showed another approach computing Ward's hierarchical clustering onto a SOM data projection. A fuzzy membership degree of each node to each cluster is computed using Euclidean distances between data points and the centroids of the crisp clusters. The clustering is fuzzified by computing a function of the inverse distance between each reference vector in the codebook and each cluster center.

Another approach is presented by Sarlin [10], employing a semi-supervised version of the SOM by using class and indicator vectors to map data onto the SOM. After training a SOM, only the indicator vector is used for locating the BMU for data. Then, the class values of their BMU are used as estimates of the current state. The fuzzification is implemented on second-level state centers instead of directly on the nodes. The SOM is fuzzified by computing the inverse distance between reference vectors and each state center. An indication of financial imbalances is assessed using distances between each data vector and its BMU.

The methodology we propose in this paper (SFPC), detailed later in Section 4, to obtain fuzzy predicates for clustering systems, exploit all the abilities discussed above for the two-level clustering approach when SOM is used as first level clustering. In particular, we applied a SOM–FCM scheme, with crisp-clustering, directly on the dataset and on partitioned data. Unlike the approach used in [11], crisp-clustering is used to design the membership functions of the fuzzy predicates system. So in this work, the SOM–FCM approach is used as a tool to characterize and generalize the input space, extracting the necessary knowledge to design the clustering systems based on fuzzy predicates.

2.2. Fuzzy systems

Fuzzy systems and their applications to data clustering are highly analyzed research topics. Most current applications are based on fuzzy inference system (FIS) approaches using IF–THEN rules in both Mandami or Takagi-Sugeno-Kang systems (see e.g. [25–30]). Membership functions and rules are generated using experts' knowledge or obtaining information from data. In recent years, various algorithms have been proposed for the design of fuzzy systems from data. For example, we have applied successfully fuzzy systems based on fuzzy predicates and expert knowledge in medical-image segmentation [15]. Due to the large amount of existing literature, we only discuss some of the most recent applications and design methods for fuzzy systems that are relevant to this paper.

Mansoori [27] stated that although fuzzy clustering is superior to crisp clustering when the boundaries among the clusters are vague and ambiguous, the main limitation of both fuzzy and crisp clustering algorithms is their sensitivity to the number of potential clusters and/or their initial positions. Also the author points out that discovered knowledge is not understandable for human users. A fuzzy rule-based clustering algorithm is proposed that employs a combination between supervised and unsupervised techniques. The algorithm automatically explores the potential clusters in the data and tries to identify them with interpretable fuzzy rules. Experimental results show that clusters specified by fuzzy rules are reasonably interpretable.

In an attempt to solve the possible weak generalization for classical fuzzy system modeling methods when available data are insufficient, Deng et al. [28] presented a fuzzy system which not only makes full use of the data in the learning procedure but can effectively make leverage on the existing knowledge from the data. The method combines Mandami–Larsen models to the fuzzy systems and a reduced set density estimation technique. For validation, the authors apply this method to synthetic and real-world datasets.

Celikyilmaz and Turksen [26] showed that although traditional fuzzy models have proven to have a high capacity for approximating real-world systems they have some challenges, such as computational complexity, optimization problems and subjectivity. To solve these problems, they develop a fuzzy system modeling approach based on fuzzy functions to model systems with an improved fuzzy clustering algorithm, a new structured identification algorithm, and a nonparametric inference engine. Empirical comparisons indicate that the proposed approach yields comparable or better accuracy than fuzzy or neuro-fuzzy models based on fuzzy rule bases, as well as other soft computing methods.

Juang et al. [25] proposed a self-organizing Takagi–Sugeno-type fuzzy network with support vector learning. The antecedents of IF–THEN rules are generated via fuzzy clustering of input data, and then the support vector machine algorithm is used to tune the consequent parameters. The proposed methodology is applied to several problems, especially the skin color classification problem. For comparison, support vector machines and other fuzzy systems are applied to the same problems. The authors argue that one advantage of the method is that number of resulting rules is smaller than other existing approaches based on support vector machine.

Meschino et al. [15] presented a fuzzy system based on fuzzy predicates. Magnetic resonance brain images (MRI) are analyzed pixel-wise by fuzzy logical predicates, reproducing in a computational way the considerations that medical experts employ when they interpret these images, in order to identify the tissues that the pixels represent. Membership functions and predicates are defined preliminary by this expert knowledge and then optimized by a genetic algorithm. Compensatory Fuzzy Logic operators are used to implement the logical connectives. The aim is to determine which tissue corresponds to each pixel. The mathematical operations involved are simple and therefore processing time is short.

Fuzzy systems developed with the design method proposed in this paper (SFPC) make the most of the advantages of fuzzy predicates described in [15], but no prior experts' knowledge is required. Membership functions and fuzzy predicates are defined by using SOM–FCM schemes. The amount of predicates that “explains” the clusters is small and depends on the amount of data partitions. Besides, unlike the systems based on IF–THEN rules explained above, in the predicate scheme the degrees of truth of the predicates determine the assignation of a cluster. Due to these characteristics, the clustering obtained is linguistically interpretable and adaptable to the field from which the data comes.

3. Methods

In this section we summarize some basic concepts of SOM and fuzzy predicates that will be required in Section 4 where we detail the method proposed in this work.

3.1. Self-organizing maps

The SOM can be interpreted as a topology preserving mapping from input space onto a grid of map units [6]. As the SOM algorithm is well-known, here we only present the basic concepts required in this work.

The SOM consists of a regular grid of map cells (in this work we used a 2-D grid). Each cell is represented by a prototype vector whose dimensions equal the input vector dimensions. The set formed by all prototype vectors defines the codebook. The cell units are related to adjacent ones by a neighborhood relation. The accuracy and generalization capability of the SOM are determined by the number of map units, which typically varies from a few dozen up to several thousand. During the training phase, the SOM builds a net that folds onto the input data distribution. Data points lying near each other in the input space are mapped onto nearby map units. The codebook is changed according to the information contained in the training data. The codebook must be initialized by applying some method. There are two main approaches for this stage [31]: random initialization and data analysis based initialization.

In the remainder of this section, SOM main basics that are relevant to this work will be defined. Hereafter, data will be considered as a subset of d-uples in $[-1,1]^d$.

Definition #3.1.1. Given a datum x, the cell whose prototype vector is the nearest, according to a distance criterion, is called Best Matching Unit (BMU) of datum x [6].

By extension of Definition 3.1.1, the cell whose prototype vector is k-th nearest to an input datum is called k-th BMU for that datum, noted by b_x^k, and its prototype vector is indicated as $w_{b_x^k}$.

Definition #3.1.2. Given a cell j in a SOM with N cells, the cell nearest to $j$in the map space (grid space) is called adjacent to j [32].

By extension of Definition 3.1.2, the k-adjacent cell to j is the cell k-th nearest in the map space and this is symbolized as ad_j^k.

In order to evaluate the quality of the trained map, three kinds of error are considered in this paper: the quantization error, the topographic error and the topographic product. These errors are defined below [32,33]:

Definition #3.1.3. Quantization error E_Q is helpful to assess whether the prototype vectors of a trained SOM are really good prototypes for the training data and is computed as

$$E_Q = \frac{1}{L}\sum_{i=1}^{L} ||x_i - w_{b_{x_i}^1}||,$$

where x_i, $i=1,2,..,L$ are the training data, $w_{b_{x_i}^1}$ is the prototype vector of the BMU corresponding to datum x_i, and L is the number of train data.

Definition #3.1.4. Topographic error E_T is helpful to assess whether the data topology was preserved after training and is computed as

$$E_T = \tfrac{1}{L}\textstyle\sum_{i=1}^{L} u(x_i),$$

where $u(x_i)=1$ if the BMU for datum x_i is not adjacent to the second BMU and $u(x_i)=0$ if it is adjacent.

Definition #3.1.5. Topographic product P_T is helpful to assess whether neighborhood relations in a SOM are preserved, analyzing distances between the codebook and the data. It can be used to determine the best size map for a given data set [33]. It is defined as follows [32]:

$$P_T = \frac{1}{N(N-1)}\sum_{j=1}^{N}\sum_{k=1}^{N-1}\log(P(j,k)),$$

where $P(j,k)$ is $P(j,k) = \left(\prod_{l=1}^{k} Q_1(j,l)Q_2(j,l)\right)^{1/2k}$ and $Q_1(j,l)$ and $Q_2(j,l)$ are defined respectively as

$$Q_1(j,l) = \frac{f(w_j, w_{b_{w_j}^l})}{f(w_j, w_{ad_j^l})},$$

$$Q_2(j,l) = \frac{g(j, b_{w_j}^l)}{g(j, ad_j^l)},$$

where $f:\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a distance function in the space of data and $g:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$ is a distance function in the space of map. Deviations of P_T from zero mean that the SOM size is unsuitable for the training dataset [32]. If map vectors perform an organized projection of the training pattern according to a similarity criterion, then the errors defined in 3.1.3–3.1.5 tend to be minimized.

In a well-trained SOM, the codebook is a reduced dataset which is representative of the training dataset, with similar probabilistic density function. By running a clustering algorithm on the SOM codebook (second level clustering), we can obtain groups of prototype vectors (and hence cells), expecting that cells from the same cluster are topographically near [5,11]. This feature is used in this paper to design fuzzy systems.

In order to have in the codebook a good representation of the training dataset, parameters can be adjusted to jointly minimize SOM quantization and topographic errors and SOM's topographic product which assure conservation of data topology and mapping quality that is performed by codebook vectors on the training data [32,33]. In this paper, we propose the automatic selection and setting of the parameters of each SOM in two steps. First, for each case SOM with different combinations of quantity of cells, topology, neighborhood function and training type are constructed. Map sizes correspond to 1, 2, 3 and 4 times the quantity of cells obtained with the heuristic formula [34]: $\#cells = 5\sqrt{L}$ where L represents the quantity of training data. Then we select the SOM that jointly minimize error measures E_Q, E_T and P_T. We use a multi-objective minimization of these measures, performing an exhaustive search and varying the map size.

3.2. Fuzzy systems

In the traditional approach, a predicate is understood to be a Boolean-valued function $P: X \to \{\text{true}, \text{false}\}$, called the predicate on X. However, predicates have many different interpretations in mathematics and logic, and their precise definition varies from theory to theory. In this work we consider predicates as synonymous of propositions. In traditional logic and philosophy the term “proposition” refers to the content of a meaningful declarative sentence. This includes possessing the property of being either true or false.

Predicates can be considered to have a degree of truth associated, which can take the values 0 (“false”) or 1 (“true”). This approach allows generalizing this definition to consider fuzzy predicates, as they are presented in this section, where some basic definitions regarding fuzzy predicates logic are given, in order to unify the notation.

Definition #3.2.1. A fuzzy predicate p is a linguistic expression (a proposition) with degree of truth $\nu(p)$ in a $[0,1]$ interval. It applies the “principle of gradualism” which states that a proposition may be both true and false, possessing some degree of truth (or falsehood) assigned [16].

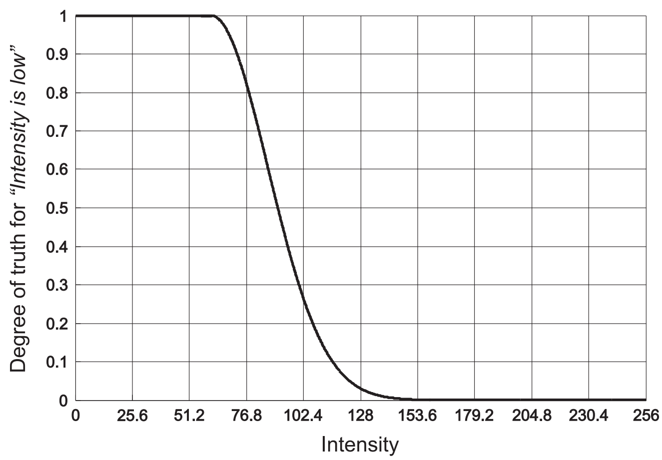


Fig. 1. Example of a membership function for quantifying the degree of truth of the fuzzy predicate *"Intensity is low"*. For low values of the intensity, the degree of truth of the predicate takes high values. As the intensity value increases, the degree of truth of the predicate decreases.

Definition #3.2.2. A simple fuzzy predicate *sp* is a fuzzy predicate whose degree of truth $\nu(sp)$ can be obtained by some of the next alternatives:

- The application of a membership function associated with a fuzzy term, to a quantitative variable. For example, let us consider the variable "intensity" varying in the range $[0, 255]$ and the fuzzy term "low". The degree of truth of the predicate "*Intensity is low*" can be quantified by means of the membership function given in Fig. 1, considering the value of the variable "intensity".
- The association is of discrete values in the $[0, 1]$ interval to language labels (generally adjectives) of a variable. For example, let us consider the predicate "*The customer is satisfied*". Its degree of truth could be quantized choosing "*Totally*", "*Quite satisfied*", "*Somewhat satisfied*" or "*Unsatisfied*", related to the degree of truths 1.0, 0.7, 0.3, and 0.0 respectively.
- The determination of real values in the $[0, 1]$ interval is directly by an individual (or an expert). It is applied in cases where no variable can be used for quantization. For example: "*The general aspect is good*", "*The material is soft*", "*The service is good*", etc.

Definition #3.2.3. A compound predicate *cp* is a fuzzy predicate obtained by combination of simple fuzzy predicates or other compound fuzzy predicates, joined by logical connectives and operators (and, or, not, implication, double-implication).

For example, the compound predicate *c*: "The pressure is high and the intensity is intermediate or low" can be expressed as $c \equiv p \wedge (q \vee r)$ where *p*: "The pressure is high"; *q*: "The intensity is intermediate" and *r*: "The intensity is low".

Definition #3.2.4. Compound predicates can be represented as a tree structure, with its nodes associated by logical connectives and the successive branches related to lower hierarchical level predicates (simple or compound).

The degree of truth of a compound predicate can be computed by considering the degrees of truth of the simple predicates involved. For this purpose it is necessary to define logic systems where the operations of conjunction, disjunction, order and negation are functions defined over a set of degrees of truth for predicates, into the real interval [0,1], such that when the degrees of truth are restricted to $\{0, 1\}$, these operations become classic Boolean predicates.

In the present work, based on previous successful results, we choose compensatory logic operations: Geometric Mean Based Compensatory Logic (GMCL) and Arithmetic Mean Based Compensatory Logic (AMCL) [35]. We also compare the results with the standard triangular norms (Max–Min) [19]. In the remainder of this section, GMCL and AMCL operators, and standard triangular norms (Max–Min) will be defined.

Definition #3.2.5. The conjunction and the disjunction of the GMCL, respectively noted by $C : [0, 1]^n \rightarrow [0, 1]$ and $D : [0, 1]^n \rightarrow [0, 1]$, are defined as [35]

$$C(\mu_1, \mu_2, \ldots, \mu_n) = (\mu_1 \mu_2 \ldots \mu_n)^{1/n},$$

$$D(\mu_1, \mu_2, \ldots, \mu_n) = 1 - [(1-\mu_1)(1-\mu_2)\ldots(1-\mu_n)]^{1/n}.$$

Definition #3.2.6. The conjunction and the disjunction of the AMCL, respectively noted by $C : [0, 1]^n \rightarrow [0, 1]$ and $D : [0, 1]^n \rightarrow [0, 1]$, are defined as [35]

$$C(\mu_1, \mu_2, \ldots, \mu_n) = \left[\min(\mu_1, \mu_2, \ldots, \mu_n) \frac{1}{n} \sum_{i=1}^{n} \mu_i \right]^{1/2},$$

$$D(\mu_1, \mu_2, \ldots, \mu_n) = 1 - \left[\min(1-\mu_1, 1-\mu_2, \ldots, 1-\mu_n) \frac{1}{n} \sum_{i=1}^{n} (1-\mu_i) \right]^{1/2}.$$

Definition #3.2.7. The standard triangular norms are formed by a fuzzy conjunction and a fuzzy disjunction, respectively noted by $C : [0, 1]^n \rightarrow [0, 1]$ and $D : [0, 1]^n \rightarrow [0, 1]$, that are defined as [19]

$$C(\mu_1, \mu_2, \ldots, \mu_n) = \min(\mu_1, \mu_2, \ldots, \mu_n),$$

$$D(\mu_1, \mu_2, \ldots, \mu_n) = \max(\mu_1, \mu_2, \ldots, \mu_n).$$

The fuzzy complement operation is a function of $N : [0, 1] \rightarrow [0, 1]$, which in all these logics computed as $N(\mu) = 1 - \mu$ where μ is the degree of truth of a fuzzy predicate.

Compensatory operators (GMCL and AMCL) are sensitive to the whole set of operands. This is evident since they are based on geometric and arithmetic means. In contrast, in the widely used operations Max–Min only one of the operands dominates the result, ignoring the values of the remaining operands.

Considering compensatory operators, the value of the conjunction and disjunction can be influenced by, and therefore "compensated" for, the value of any of the degrees of truth considered in the operation. An increase or decrease in the degree of truth of the conjunction or disjunction as a result of changes in the degree of truth of one component can be compensated by an increase or decrease, respectively, of the degree of truth of other component. This feature makes compensatory logic especially suited for selection problems; yet it is also convenient for ranking, appraising, and classificatory purposes [36].

4. Proposed method

In this section we present in detail the SOM-based Fuzzy Predicate Clustering (SFPC) method and we propose different options for its configuration and setting parameters. Also we comment on the procedure used for the assessment of the performance of the clustering obtained.

As introduced in Section 1, we propose a SOM-based method to design fuzzy predicate clustering systems. These clustering systems allow:

- To assign a cluster for every datum included into the dataset.
- To give a linguistic interpretation of the clustering obtained. Each cluster can be explained by one or more fuzzy predicates, so that an expert user is able to give linguistic meaning to membership functions that were discovered automatically.

- To assign a new datum not included in the original dataset to one of the clusters discovered, knowing what the fuzzy predicate was that caused the assignment.

Each cluster is described by a fuzzy predicate (or more than one, as it will be detailed in Section 4.2) as "*The datum belongs to cluster i*", whose degree of truth will be computed by simple predicates based on the features of the dataset.

Let us explore this proposal first considering only one SOM, and then we will be able to present some different options to extract more information from the dataset by taking data subsets or considering partitioned data.

4.1. General method: one SOM, K predicates

In this section we detail the basics for using SOM as a tool for automatically building the membership functions that will be used to quantify the degree of truth of the simple predicates. In this case we obtain as many predicates as clusters we want. Given one datum, by computing the degree of truth of each predicate we will be able to find the one with the maximum, and in consequence assign the datum to the cluster defined by this predicate.

The general steps are the following:

Step 1: Train a SOM with different combinations of numbers of cells, topology, and neighborhood functions and select the SOM that jointly minimizes the error measures E_Q, E_T andP_T (as described at the end of Section 3.1).

Step 2: Find K cluster centers (centroids) $\{C_i\}_{i=1,2,\ldots,K}$ over the SOM codebook by some classical simple clustering algorithm. In this work we applied FCM with crisp clustering [24], operating with Euclidian distances, but other algorithms and distances could be used.

Step 3: Create membership functions for the simple predicates from the clusters obtained in the previous stage.

We propose Gaussian functions. However, other shapes could also be considered, but this case is not covered in the current work. Centers of the Gaussian functions for each feature are the cluster centers (also called centroids) found after codebook clustering, and the widths of these Gaussian functions are computed as the standard deviations of prototype vectors in each cluster.

So, in this stage we define Gaussian membership functions for each feature i and for each cluster k. Centers of these Gaussian functions are the cluster centers obtained in the previous step, namely $\{c_{ik}\}_{\substack{i=1,2,\ldots,d\\k=1,2,\ldots,K}}$.

Then we compute standard deviations of prototype vectors of the codebook contained in each cluster, for each feature $\{\sigma_{ik}\}_{\substack{i=1,2,\ldots,d\\k=1,2,\ldots,K}}$. Considering that MF created indicate the correspondence between feature values and clusters, the standard deviation controls the width (sigma) of the Gaussian functions. The standard deviation acts as a parameter controlling how the degree of truth of the MF decreases when values of a feature for a specific datum move away from the cluster center. We show a typical membership function in Fig. 2, for a generic cluster and a generic feature.

Given a dataset where each datum is a feature vector, a d-uple $(f_1, f_2, \ldots, f_d)$, we will have K Gaussian membership functions for each feature f_i, $i=1,2,\ldots,d$. Let us call them $\{mf_{ik}\}_{\substack{i=1,2,\ldots,d\\k=1,2,\ldots,K}}$.

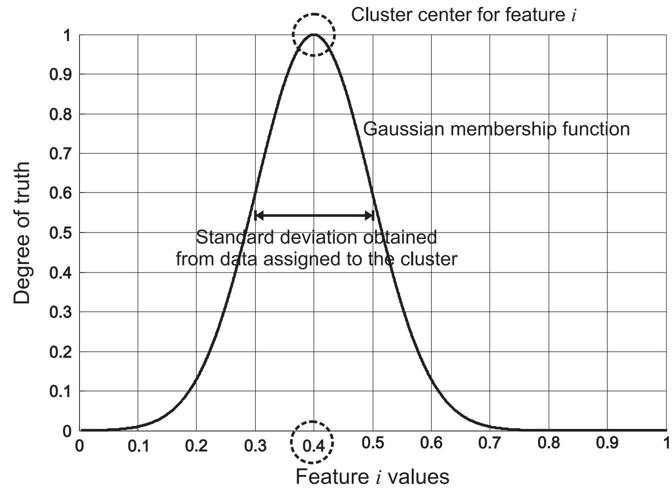


Fig. 2. Example of a Gaussian membership function for a generic cluster, and a generic feature *i*. The center of the Gaussian is the centroid of the cluster obtained from the SOM codebook and the deviation of the Gaussian is the standard deviation of the data that were assigned to this cluster.

Step 4: Now we can create one fuzzy predicate for each cluster (K compound predicates) by logically operating with the degrees of truth:

$$p_k(D) \equiv mf_{1k}(D) \wedge mf_{2k}(D) \wedge \ldots, \wedge \ mf_{dk}(D); \quad k=1,2,\ldots,K,$$

where $p_k(D)$ can be linguistically read as "*The datum D belongs to cluster k*" and $mf_{ik}(D)$ can be linguistically interpreted as *"Feature i of the datum D is near the prototypes belonging to cluster k*". The nearer the value of feature i of the datum D to the center of the Gaussian function c_{ik}, the higher the degree of truth of $mf_{ik}(D)$. As $mf_{ik}(D)$ are higher, $p_k(D)$ should also be higher, reflecting the fact that if the datum D is near the cluster center k, then the datum D belongs to cluster k.

A datum D will be assigned to the cluster whose predicate has the highest degree of truth. In addition, if no predicates have degree of truth higher than a determined threshold, D could be labeled as an "outlier".

4.2. Dataset partitioning

In addition, in this section we propose an extension of the method to be able to deal with partitions of the dataset. Essentially, it consists of a two-stage design, wherein the basic stage performs steps 1–3, as mentioned in Section 4.1, while in the upper stage only step 4 is implemented, but in this case we have different options to form the predicates.

This scheme would allow its application in geographically distributed computer processors; hence it enables the processing of large volumes of data asynchronously with low bandwidth needs for the transmission of intermediate results, only $\{c_{ik}\}_{\substack{i=1,2,\ldots,d\\k=1,2,\ldots,K}}$ and $\{\sigma_{ik}\}_{\substack{i=1,2,\ldots,d\\k=1,2,\ldots,K}}$ must be transmitted to the upper stage. Even a parallel implementation, none distributed geographically, based on dataset partitioning would aim to reduce the overall process time.

To explore the design performance, M different nodes at the basic stage are proposed. Therefore M is defined by the specifications of the system: the amount of geographically sparse nodes or the minimum amount of data in each partition that enables an adequate performance of the SOM. To simulate its behavior we propose a random dataset partition into M disjoint subsets, in order to train M different SOM. Despite this partition being random, it would be desirable that each partition is balanced; i.e. there is approximately the same number of data for each cluster. Since in a real problem we cannot know what the cluster of each datum is, at least we should be able to assume that data represent the original population; i.e. there are data corresponding to different cases or situations that we want to represent with clustering.

At the upper stage, for each k, a unique cluster must be defined using the M centroids. To that end, considering a centroid, the $M-1$ nearest centroids are added (by considering Euclidian distances among them) as belonging to the same cluster. As a result, M clusters belonging to the same final cluster on data will have the same cluster index $\{c_{ik}\}_{\substack{i=1,2,\dots,d\\ k=1,2,\dots,K}}$.

Next, we propose, three different options to create fuzzy predicates to represent the clusters, described as follows.

4.2.1. Option 1: clustering ensemble (K predicates from each SOM, one decision by SOM)

In this option we consider each SOM and its predicate set as an independent clustering system, i.e. we apply to each subset what was presented in the previous section. We obtain K predicates for each SOM (labeled $\{p_{ij}\}_{\substack{i=1,2\dots,M\\ j=1,2\dots,K}}$). Given a datum, the degrees of truth of the K predicates for each SOM are obtained, and a cluster assignment is conducted. The final assigned cluster will be chosen by voting, as is typical in a clustering ensemble approach [37].

4.2.2. Option 2: M independent fuzzy predicates for each cluster

In this case we take all predicates generated in the same way as the previous option. As before, we obtain M predicates for each cluster. But in this option, given a datum, degrees of truth of K by M predicates are obtained. Cluster assignment is done simply by taking the one represented for the predicate whose degree of truth is maximum, no matter which partition provided the predicate.

4.2.3. Option 3: a unique compound fuzzy predicate for each cluster

In this option we explore exhaustively the advantages of using predicate logic. We define a unique predicate for each cluster using the "or" connective to consider simple predicates (obtained from partitions) for each feature. We obtain K predicates with the following general expression:

$$\begin{aligned} p_k(D) \equiv & [mf_{1k1}(D) \vee mf_{1k2}(D) \vee \dots \vee mf_{1kM}(D)] \\ & \wedge [mf_{2k1}(D) \vee mf_{2k2}(D) \vee \dots \vee mf_{2kM}(D)] \wedge \dots \\ & \wedge [mf_{dk1}(D) \vee mf_{dk2}(D) \vee \dots \vee mf_{dkM}(D)];\ k=1,2,\dots,K. \end{aligned}$$

Given a datum, the degrees of truth of the K compound predicates are obtained. Like previous cases, cluster assignment is done by taking the cluster represented for the predicate whose degree of truth is the maximum. This option gives the highest interpretability for the predicates, though they could be really complex when dealing with high-dimensional data (big quantity of features).

4.3. Algorithm configuration: parameters

Results may be different depending on the way the predicates are defined, and, of course, the option chosen to arrange and evaluate the predicate set as given in the previous section. In addition, there are some other parameters that should be chosen for each particular problem:

- Number of clusters K: defined by the problem to be solved.
- Cluster algorithm used to make the second-level clustering: should be a simple and low computational cost method, like FCM or K-means.
- Type of distances used in the second-level clustering algorithm and during SOM training: typically Euclidian distance is used, but other distance definitions could be considered.
- Number of data partitions M: it should be chosen according to the number of data, taking into account that each SOM should be able to capture the characteristics of data, so each partition should not contain only a few training data. In the case of sparse data nodes it would be determined by the number of nodes, as presented in the previous section.
- Type of logical system for computing logical operations: compensatory fuzzy logics (GMCL or AMCL) or Max–Min logic are compared in this work, but other logical systems could be used (for further information on others fuzzy operators [38] can be consulted).

4.4. Method validation: assessing the clustering quality

In order to assess the quality of the clustering obtained, several measures could be considered. The general quality can be assessed by considering the so called "internal measures" like Inter-cluster Density, and Intra-cluster Variance. These indexes seek compact and separated clusters. This type of measures is needed in cases we do not have any previous labeled data, as it is often the case, and these measures play a fundamental role to set the parameters of the algorithm (summarized in Section 4.3).

Other measures need data previously labeled as the gold standard, known as "external measures". Gold standards are ideally produced by human experts or based on previous experimentation, normally labeling data at the same moment they are obtained [39].

For this work we chose the assessment of the Accuracy measure, which is an external measure. It requires labeled data, and an analysis of majority labels in each discovered cluster is needed as well. Once each real class is assigned to one and only one cluster, this measure is simply computed as the ratio between the quantity of data assigned correctly to the quantity of available data (it can also be computed as the ratio between the sum of the main diagonal of the confusion matrix to the sum of its elements) [40].

To be independent of the algorithm initialization (mainly the random data partitions), each configuration is run 10 times and values reported are Accuracy averages.

5. Results

In this section we analyze the numerical results for Accuracy from the proposed method. We make a detailed analysis and comparisons against other methods, including a modification of the proposed method which does not use a two-level clustering scheme, by replacing the SOM–FCM scheme with a simple basic clustering algorithm over the original data. We also present graphs to help these comparisons. We show that the proposed method outperforms or at least achieves the results of the others.

Moreover, we show an example of how an expert can give a linguistic interpretation to the automatically discovered membership, and how the predicates can be read.

5.1. Accuracy of the proposed method

We tested the method using the following datasets:

- Iris data (3 classes, 4 features, 150 data) [20].
- Wine data (3 classes, 13 features, 198 data) [41].
- MRI data, 12,000 pixels randomly selected from simulated magnetic resonance images (4 classes, 3 features, 12,000 data)

[42]. These data were taken without any noise or distortion, and they come from computer simulations.

- MRI data, 200 pixels randomly selected from the previous dataset (4 classes, 3 features, 200 data) [42].
- Wisconsin breast cancer data (32 features – 3 selected, 2 classes, 569 data) [43].
- Pima Indians diabetes data (8 features – 3 selected, 2 classes, 768 data) [43].
- Moon data (2 features, 2 classes, 1000 data). This is a synthetic dataset whose characteristics will be presented in the next section.

For the Wisconsin breast cancer and Pima Indians diabetes datasets, we took the features chosen in [43] in order to facilitate results comparison.

We compared clustering accuracy values obtained by means of the proposed clustering algorithm and the following ones (an acronym is defined for each one):

- **SFPC**: SOM-based Fuzzy Predicate Clustering (the proposed method) is applied to the dataset. SOM sizes are determined automatically according to the quantity of training data and their dimension (as described in Section 4.1 and discussed in Section 3.1). The number of partitions (M, if $M>1$) is determined taking into account the quantity of data available in each particular dataset (therefore M is different for each dataset). The larger the quantity of data, the larger could be M.
- Data-based Fuzzy Predicate Clustering (**DFPC)**: here the two-level SOM–FCM based scheme is replaced by a single clustering algorithm (we used FCM, but it could be another one) which is applied directly on the partitions of the dataset (or to the whole dataset if $M=1$). In this case, information for membership parameters is taken directly from data clustering instead of from the SOM–FCM clustering. These results were included in order to add further comparisons and discussions.
- **SOM–FCM**: a SOM is trained with the dataset and FCM with crisp clustering is applied to the codebook (configuring a basic two-level clustering scheme).
- ***K*-means**: the K-means algorithm is applied to the dataset, considering random initial centroids (K data are randomly chosen as initial cluster centers). K centroids are obtained and data are assigned to the cluster when the Euclidian distance to its centroid is the minimum.
- Expectation-Maximization **(EM)** [44]: it is a method used to fit a Gaussian mixture model to a dataset. Probability density functions characterizing the dataset are obtained. Then probability of belonging to each cluster is determined, assigning the most likely.
- **FCM** [24]**:** FCM algorithm is applied to the dataset considering random initial centroids. So K centroids are obtained and data are assigned to the cluster with the largest membership value.

Clustering accuracies obtained are shown in Fig. 3, where the accuracy is represented by vertical bars. We show different graphs for each dataset. The height of bars indicates the accuracy as shown in the vertical axis. Three main groups of bars are shown. From left to right, the first main group corresponds to the DFPC method, the second group corresponds to the proposed method (SFPC) and the last one includes other traditional clustering algorithms.

In Fig. 3, for DFPC and SFPC methods:

- accuracy for $M=1$ case is presented, when no partition of data is done;
- accuracies for other values of M are also included, values chosen according to the number of data in each dataset;
- once M is set, results for the different configuration options for the fuzzy predicates (Option 1, Option 2 and Option 3) are also included; and

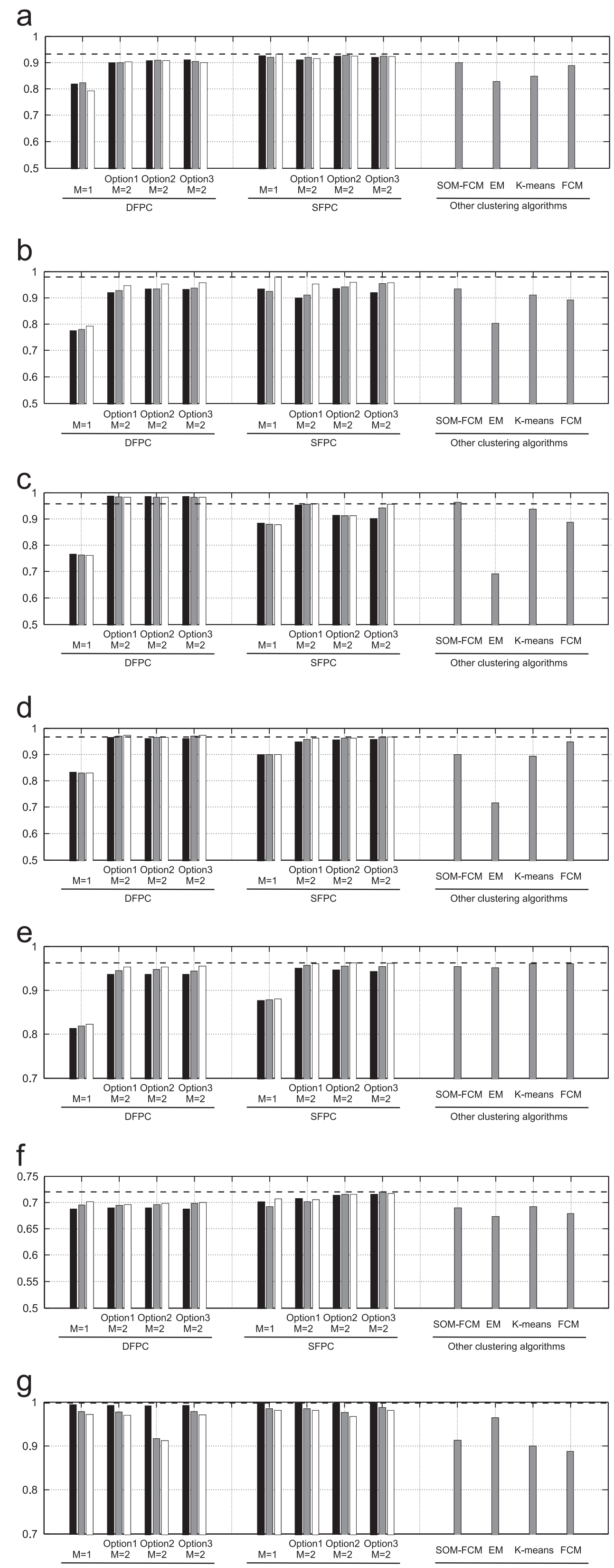


Fig. 3. Clustering accuracy for test datasets: bars indicate the different logic operator systems (black: Max–Min; gray: AMCL; white: GMCL). All SOM sizes are automatically set and the number of data partitions (M) is indicated for each bar group. The algorithm option is indicated as Option 1, Option 2, and Option 3 as presented in Section 4.2. Horizontal dotted line indicates the best accuracy value obtained for SFPC: (a) Iris, (b) Wine, (c) MRI1, (d) MRI2, (e) Wisconsin breast cancer, (f) Pima Indians diabetes, and (g) Moon.

- results are represented in each case in groups of three bars corresponding to different logic systems (black: Max–Min; gray: AMCL; white: GMCL).

For the last main group, the results for known methods are shown: two-level SOM-clustering (SOM–FCM), Expectation-Maximization (EM), *K*-means and Fuzzy C-Means (FCM). In these cases, only one bar is needed. Finally, a horizontal dotted line is included in Fig. 3, which indicates the best accuracy value obtained for SFPC.

In the following paragraphs, a detailed result analysis is done. When comparing two different results in terms of accuracy, we refer to a percentage difference of the accuracies, i.e. difference between acc_1 and acc_2 is given as

$$difference\% = \frac{acc_1 - acc_2}{acc_2} \times 100\%.$$

Results for Iris dataset (Fig. 3a) suggest that SFPC outperforms the other clustering methods. It improves 3.2% of the SOM-Clustering accuracy (the best for simple clustering methods in this dataset) when GMCL logic system is applied. In this case SFPC was better than DFPC, particularly for $M=1$ (no data partition). So for this dataset the best results are obtained for SFPC taking $M=1$ and GMCL.

Considering the Wine dataset (Fig. 3b), SFPC again outperforms the other clustering methods, improving 4.5% the SOM-Clustering accuracy (also the best one in this dataset) when the GMCL logic system is applied. SFPC is better than DFPC for $M=1$ (no data partition) and slightly better for $M=2$ (Options 2 or 3). For this dataset the best results are obtained for SFPC taking $M=1$, and GMCL.

Let us consider the MRI dataset. Since it was published for research purposes many methods have been presented, using it as test data. We used these data in this work by taking a large number of samples (12,000, MRI1 dataset) and only a few samples (200, MRI2 dataset). Using less data, the accuracy decreases as expected, but the proposed method still shows a better performance than the simple clustering methods, as can be seen in Fig. 3d, which evidences the method's good robustness and generalization abilities.

For the MRI1 dataset (Fig. 3c), we observe that DFPC obtains the best results, surprisingly for a Max–Min logic system. In this case, we choose $M=10$ because this dataset contains a large quantity of data. The predicates approach remains being the best one, but in this particular case SFPC could not achieve the performance of DFPC or SOM-Clustering.

For the MRI2 dataset (Fig. 3d), the best performance is again achieved for DFPC, in this case for the GMCL logic system. The fact it possesses less information (less quantity of data) demonstrates the generalization abilities of the proposed approach. In this case, we choose $M=3$ because the dataset contains a lot less data than MRI1. Again the predicates approach is the best one, but again SFPC could not achieve the performance of DFPC, but anyway it outperforms the SOM-Clustering performance by 7.4%.

Considering the Wisconsin breast cancer dataset (Fig. 3e), the best performance is obtained by SFPC using the GMCL logic system, $M=3$, Options 2 or 3. It outperforms DFPC only by 0.9% and SOM-Clustering by 0.8%. The Wisconsin breast cancer dataset was included in previous relevant works [43], obtaining an accuracy of 0.977 using supervised methods, in particular the K-Nearest Neighbors (KNN) which considered 15 data from each class as training samples. Using the proposed method we obtained an accuracy of 0.962, highly satisfactory considering that our method does not require any previous label information.

Considering the Pima Indians diabetes dataset (Fig. 3f), the best performance is obtained by SFPC using the AMCL logic system, $M=4$, Option 3. It outperforms all settings of DFPC by more than 3.1% and SOM-Clustering by 3.0%. For the Pima Indians diabetes dataset, in the same work mentioned for the previously considered dataset [43], the authors reported an accuracy of 0.777 using the ANFIS supervised method. We obtained an accuracy of 0.720 by applying the unsupervised proposed method.

Considering the Moon dataset (Fig. 3g), the best performance is obtained by SFPC using the Max–Min logic system. This is a synthetic dataset. It always outperforms all settings of DFPC, and improves SOM-Clustering by 8.5%.

Observing Fig. 3, we note that in the cases tested the clustering of the SOM codebook (SOM–FCM scheme) gives high accuracy measures. This is a hint to consider that SOM captures the clustering information in the data and it is able to generalize the feature space, even when the quantity of data is not very large. This is observed in the SOM–FCM bars in the graphs. It even outperforms the other methods in the MRI1 dataset. This dataset has a large quantity of data. In addition it is a relatively simple problem, since data are from simulations without any noise or distortion.

In most cases, the compensatory logic systems (AMCL and GMCL) gave better results than Max–Mix, except for the MRI1, MRI2 and Moon datasets. These datasets come from computer simulations.

Accuracy for DFPC for $M=1$ (no data partition) was always worse than SFPC. It allows us to conclude that partitioning datasets gives better results and at the same time, and that using SOM for obtaining the membership parameters is a good approach. When changing from $M=1$ (taking only one SOM, the whole dataset) to $M>1$ (partitioning the dataset) we can see improvements in the MRI1, Wisconsin breast cancer and Pima Indians diabetes datasets. So it seems better to make partitions than to take the whole dataset.

Applying clustering in a two-level approach (SOM–FCM) is much more robust along the iterations compared to clustering applied directly to the dataset, which is evidenced when a variance analysis over the different options in the proposed method (SFPC) is conducted. Variances are not shown to make figures clearer.

Choosing Option 3 gives slightly better results in several cases, but there are minor differences between the options in these datasets. However the approach using only one composed predicate explaining each cluster (Option 3) is a good choice for simple linguistic interpretation. In all datasets analyzed SFPC works well, outperforming simpler methods. In cases where it was not the best case, the predicates approach still was the best method (DFPC). Combining the results for different datasets, the reliable preliminary choice for clustering a new dataset could be using SFPC with $M>1$ and GMCL, Option 3.

5.2. Interpretability of membership functions

The proposed method allows a deeper analysis of the predicates in order to make them linguistically interpretable. Taking the general predicate: $p_k(D) \equiv mf_{1k}(D) \wedge mf_{2k}(D) \wedge \ldots, \wedge mf_{dk}(D); k = 1, 2, \ldots, K$, a linguistic expression can be built: *"The datum D belongs to cluster k"* is equivalent of saying *"Feature 1 is* mf_{1k}*and Feature 2 is* mf_{2k}*and … and Feature D is* mf_{dk}*"*.

An expert can make further interpretations to give mf_{ik} meanings applicable to the field from which the dataset comes.

5.2.1. Example 1: moon dataset

As an application example we considered the Moon dataset, which is visualized in Fig. 4. It is a synthetic dataset helpful to show the methodology for analysis of the membership functions obtained. It was chosen in order to give a simple case for linguistic interpretation, but at the same time it is a case where simple clustering techniques present lower clustering accuracy.

After applying the method proposed in this paper ($M=3$), we obtained the membership functions seen in Fig. 5 for feature 1 (a) and feature 2 (b).

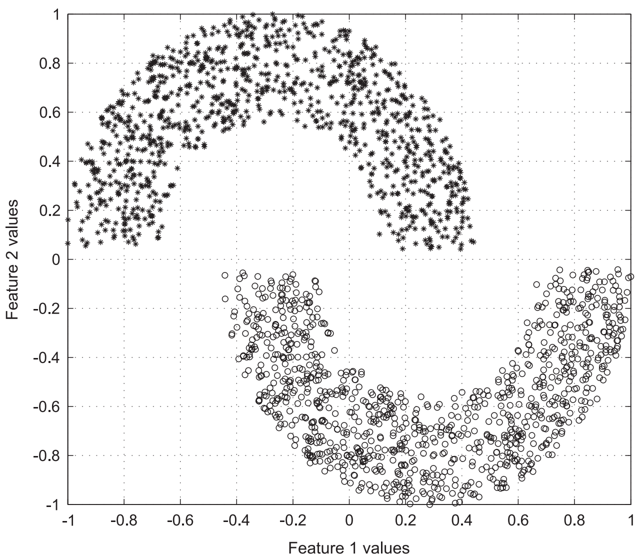


Fig. 4. The double moon dataset (here called "Moon"), 2 features, 2 classes.

By analyzing Fig. 5a we can see two well-defined groups of membership functions, one of them centered at -0.3 and another centered on $+0.3$. We can associate the first group of membership functions with the name "*Negative near* -0.3" and the second one with the name "*Positive near* $+0.3$."

We can carry out the same analysis for Fig. 5b, and since the membership functions drop in negative and positive values, we can repeat the same denominations: "*Negative near* -0.6" and "*Positive near* $+0.6$."

Then the predicates for class decision are

- For Class 1: "*The datum D belongs to Class 1*" is equivalent to saying "*Feature 1 is Negative near* -0.3 *and Feature 2 is Positive near* $+0.6$".
- For Class 2: "*The datum D belongs to Class 2*" is equivalent to saying "*Feature 1 is Positive near* $+0.3$ *and Feature 2 is Negative near* -0.6".

In real cases, the predicates would look linguistically interpretable and they could give some useful information for dataset analysis.

5.2.2. Example 2: MRI2 dataset

We also include the following analysis for the MRI2 dataset, which is nearer to a real case because its features and classes can be named as they are in real Magnetic Resonance Imaging (MRI) data. By processing this dataset we try to identify 4 classes; it was

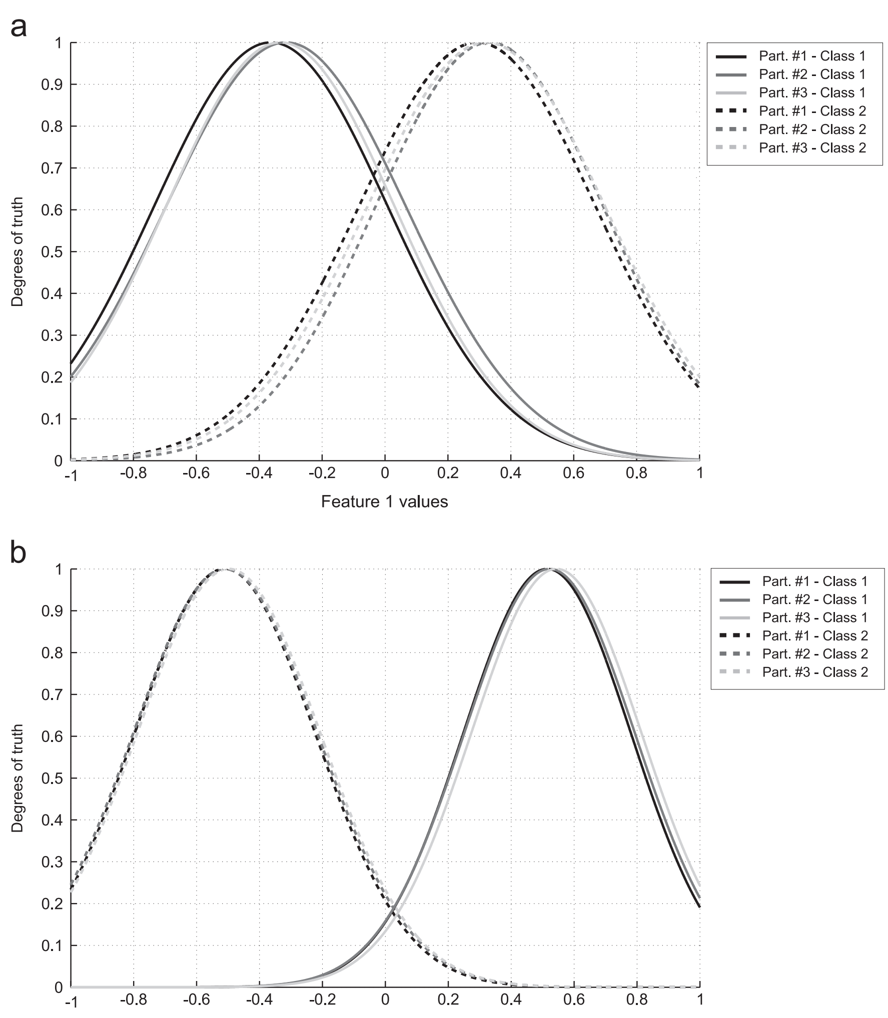


Fig. 5. Membership functions obtained after processing the Moon dataset. Different traces indicate different classes: solid line: Class 1; dotted line: Class 2. (a) Membership functions for feature 1. (b) Membership functions for feature 2.

presented and analyzed using other methods [15]. As it comes from brain MRI data, each datum represents a pixel in the image and the classes represent different types of brain tissue or areas in the images: gray matter, white matter, cerebrospinal fluid and background. In the same way, features represent different types of images obtained, named “T1 Intensity,” “T2 Intensity,” and “PD Intensity.” Considering the background as an easy-to-find class we only will analyze the other classes. We show the membership functions obtained by the SFPC method ($M=3$) in Fig. 6.

We note in Fig. 6a that there are several Gaussian curves grouped together on the left, identified for cerebrospinal fluid (each one corresponding to different partitions of the dataset and therefore different SOM analyses, shown with a solid line). All of them are placed near the lowest “T1 intensities” (-1 to -0.6 in the normalized scale), so they could be associated to low intensities in the pixels with these intensities values, which could be interpreted and named as *“Dark”*. By observing Fig. 6b in the same way, we can see the curves for the same class (again the solid

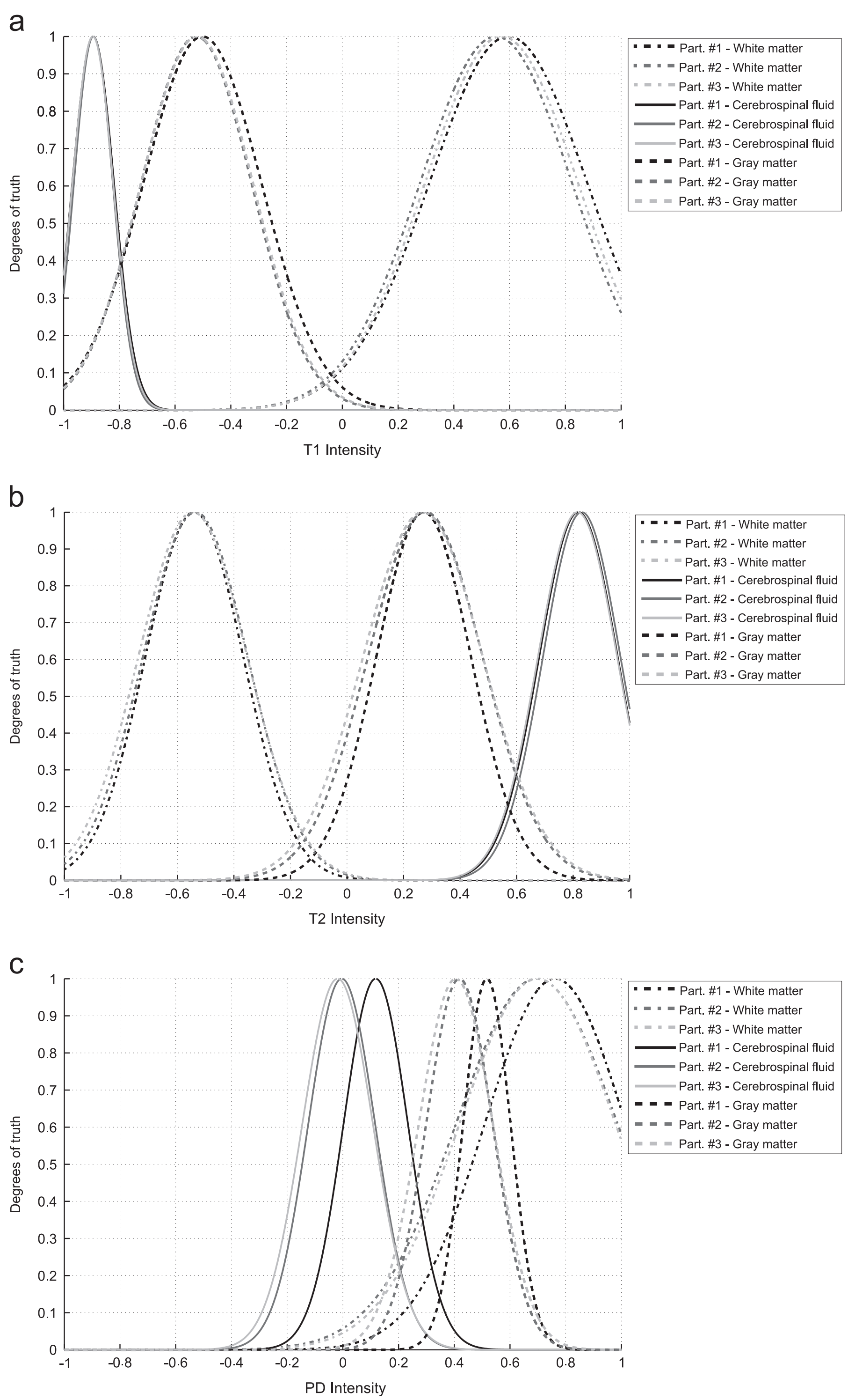


Fig. 6. Membership functions obtained by processing MRI2 dataset. Different traces indicate different brain tissues (classes): white matter, cerebrospinal fluid and gray matter. (a) Membership functions for feature “T1 Intensity”. (b) Membership functions for feature “T2 Intensity”. (c) Membership functions for feature “PD Intensity”.

line) are at the right of the graph, leading us to the opposite concept which could be *"Bright"*, as they represent the highest intensities, from 0.4 to 1 in the normalized scale. Finally we can note in Fig. 6c that the corresponding curves are in the middle of the axis, which could be translated as *"Gray"* (mean intensities).

The previous predicate can now be linguistically read, using these interpretations: "*The pixel D belongs to cerebrospinal fluid*" is equivalent to saying "*T1 intensity is Dark and T2 intensity is Bright and PD intensity is Gray*".

Following similar steps, we can also add the following expressions: "*The pixel D belongs to gray matter*" is equivalent to saying "*T1 intensity is Dark-Gray and T2 intensity is Medium-Gray and PD intensity is Light-Gray*", and finally: "*The pixel D belongs to white matter*" is equivalent to saying "*T1 intensity is Very-Bright and T2 intensity is Dark and PD intensity is Very-Bright*".

The linguistic predicates are built by replacing the generic "Feature *i*" denominations by the feature names and replacing the generic expression mf_{ik}, $i=1,2,\ldots,d; k=1,2,\ldots,K$ by interpretation of the curves in the data field. The logic structure is preserved. In this way, any of the options given for the presented method are likely to be translated to more or less complex linguistic expressions.

6. Discussion

In the steps 1 and 2 of the proposed method, the necessary knowledge to define the fuzzy predicates is extracted. The SOM codebook is not a replica of training data but it is a generalized representation of them, which makes it a good representation of the information contained in the data. We not only consider the quantification error but a combination of several quality measures. This is relevant at this stage, because it avoids overtraining and it assures the codebook is a generalization of the training data. We can see SOM as statistical seekers of the data space, considering different partial populations (one for each partition).

After training SOM, the codebook, and hence the cells, are partitioned by some simple clustering method (in a two-level clustering scheme). This allows the analysis of fuzzy ranges that features take in each cluster and how they relate to each other, which is conducted by defining membership functions and the predicate basis. As a result, we obtain structured knowledge from data besides the clustering partition.

The main advantage of this clustering methodology is the reduction of the computational cost within the stage of the identification of clusters; given that it is carried out onto the reduced data dimension of SOM projection. An interesting advantage of second-level clustering is the ability to reduce noise. The SOM codebooks act like a low pass filter, smoothing the data's high frequency variations.

Another point to note is the ability to discover clusters not linearly separable using the emergent concept applied to the U-Matrix on SOM with many neurons. Several clustering algorithms exploit the emergent properties of such SOM that outperform other popular clustering algorithms when the clustering of complex datasets is addressed.

The inclusion of the fuzzy logic approach allows taking advantage of the benefits seen in previous works, typically the robustness against data noise and the interpretability of the results. Additionally, using predicates instead of rules makes the system a natural extension of the Boolean logic, and in contrast to what occurs in FIS, no inference, aggregation, and defuzzification stages are required.

SOM used in SOM–FCM are obtained in the same way as those for SFPC. In particular, the accuracy for SFPC with $M=1$ is always higher than SOM–FCM. This highlights the contribution of the fuzzy predicates to the system's performance. Besides, the search criterion for the SOM is shown when the SOM–FCM accuracy is compared to that of K-means, EM, and FCM.

The proposed fuzzy predicate system can be configured in several ways, by changing the way predicates are built, the number of partitions of data, among other configurations options. So given a dataset, various different instances of the method can be tested in order to obtain appropriate results to discover knowledge for the field of the data.

Results are encouraging in terms of clustering accuracy when compared to other techniques. It is important to note that the accuracy is higher than that achieved with two-level SOM-clustering alone, which allows concluding that fuzzy predicates are the fundamental component of the system. Also, the clustering interpretation via fuzzy predicates constitutes an important contribution of this work.

7. Conclusion

We proposed a system using fuzzy predicates, created by way of two-level clustering approach using SOM to tackle ranked clustering problems. SOM are applied in order to preliminarily process the data, which allow constructing membership functions reinterpreted as simple fuzzy predicates involving data features. This feature enables its use in a sparse network, where SOM are performed at the node stage and fuzzy predicates are computed at the upper one. Then we offered several alternatives to arrange these simple predicates to obtain compound predicates, which are used to determine a belonging ranking for the discovered clusters.

Even when there are parameters and options that must be set before running the process, the proposed system is self-designed and it possesses the advantages of other fuzzy systems based on expert knowledge.

After the clustering process, interpretation of the results is an optative and later process, though it constitutes an important and relevant output of the system. Knowledge is extracted from data, the system itself acting like an expert who discovers information contained in the dataset.

Results show that clustering accuracy obtained is high and it outperforms other methods in the majority of the datasets tested. The linguistic interpretability of the predicates allows explaining the characteristics of the clusters discovered, and it constitutes a significant contribution of this paper.

The approach using only one compound predicate explaining each cluster is the best choice for linguistic interpretation. That is why we are proceeding in this way, searching for other integration schemes. Instead of the use of the "or" operation to aggregate the "opinion" of different SOM, this approach will be extended for the use of Type-2 Fuzzy Logic, which is proposed as immediate future work. By reinterpreting the membership obtained, it is expected that Type-2 membership functions will give a novel, compact and simpler interpretation for the fuzzy predicates.

References

[1] V. Estivill-Castro, Why so many clustering algorithms: a position paper, SIGKDD Exploration Newsletter 4 (2002) 65–75.

[2] R.B. Dubey, M. Hanmandlu, S.K. Gupta, S.K. Gupta, The brain MR image segmentation techniques and use of diagnostic packages, Acad. Radiol. 17 (2010) 658–671.

[3] F. Zhao, Fuzzy clustering algorithms with self-tuning non-local spatial information for image segmentation, Neurocomputing 106 (2013) 115–125.

[4] C.-C. Hsu, S.-H. Lin, W.-S. Tai, Apply extended self-organizing map to cluster and classify mixed-type data, Neurocomputing 74 (2011) 3832–3842.

[5] J. Vesanto, E. Alhoniemi, Clustering of the self-organizing map, IEEE Trans. Neural Netw. 11 (2000) 586–600.

[6] T. Kohonen, Self organized formation of topological correct feature maps, Biol. Cybern. 43 (1982) 59–96.

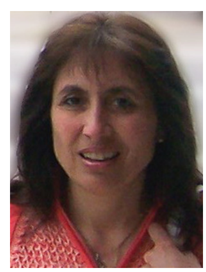
[7] J. Lampinen, E. Oja, Clustering properties of hierarchical self-organizing maps, J. Math. Imaging Vis. 2 (1992) 261–272.

[8] F. Murtagh, Interpreting the Kohonen self-organizing feature map using contiguity-constrained clustering, Pattern Recognit. Lett. 16 (1995) 399–408.

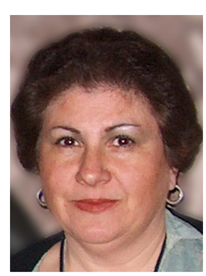
[9] M. Cottrell, P. Letrémy, Missing values: processing with the Kohonen algorithm, Applied Stochastic Models and Data Analysis, Brest, France (2005) 489–496.
[10] P. Sarlin, Exploiting the self-organizing financial stability map, Eng. Appl. Artif. Intell. 26 (2013) 1532–1539.
[11] P. Sarlin, T. Eklund, Fuzzy clustering of the self-organizing map: some applications on financial time series, in: Proceedings of the 8th International Workshop on Self-Organizing Maps, Espoo, Finland, 2011.
[12] P. Sarlin, T. Eklund, Financial performance analysis of European banks using a fuzzified self-organizing map, in: Proceedings of the 15th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES'11), Kaiserslautern, Germany, 2011, pp. 186–195.
[13] R. Thawonmas, S. Abe, Function approximation based on fuzzy rules extracted from partitioned numerical data, IEEE Trans. Syst. Man Cybern. B 29 (1999) 525–534.
[14] H. Kahramanli, N. Allahverdi, Rule extraction from trained adaptive neural networks using artificial immune systems, Expert Syst. Appl. 36 (2009) 1513–1522.
[15] G.J. Meschino, R.E. Andrade, V.L. Ballarin, A framework for tissue discrimination in magnetic resonance brain images based on predicates analysis and compensatory fuzzy logic, Int. J, Intell. Comput. Med. Sci. Image Process. 2 (2008) 209–224.
[16] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[17] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Inf. Sci. 8 (1975) 199–249.
[18] R.E. Andrade, G.M. Téllez, E.F. González, J. Marx-Gómez, M.I. Lecich, Compensatory logic: a fuzzy normative model for decision making, Investig. Oper. 27 (2006) 188–197.
[19] H. Dubois, D. Prade, Fuzzy Sets and Systems, Theory and Applications, Academic Press, New York, 1980.
[20] G.J. Meschino, D.S. Comas, V.L. Ballarin, A.G. Scandurra, L.I. Passoni, Using SOM as a tool for automated design of clustering systems based on fuzzy predicates, in: P. Estévez, J. Príncipe, P. Zegers (Eds.), Advances in Self-Organizing Maps, Springer-Verlag, Berlin Heidelberg, 2013, pp. 85–94.
[21] W. Pedrycz, Fuzzy clustering with a knowledge-based guidance, Pattern Recognit. Lett. 25 (2004) 469–480.
[22] M.Y. Kiang, Extending the Kohonen self-organizing map networks for clustering analysis, Comput. Stat. Data Anal. 38 (2001) 161–180.
[23] J.T. Laaksonen, J. Markus Koskela, E. Oja, Class distributions on SOM surfaces for feature extraction and object retrieval, Neural Netw. 17 (2004) 1121–1133.
[24] E.H. Ruspini, A new approach to clustering, Inf. Control 15 (1969) 22–32.
[25] C.-F. Juang, S.-H. Chiu, S.-W. Chang, A self-organizing TS-type fuzzy network with support vector learning and its application to classification problems, IEEE Trans. Fuzzy Syst. 15 (2007) 998–1008.
[26] A. Celikyilmaz, I.B. Turksen, Enhanced fuzzy system models with improved fuzzy clustering algorithm, IEEE Trans. Fuzzy Syst. 16 (2008) 779–794.
[27] E.G. Mansoori, FRBC: a fuzzy rule-based clustering algorithm, IEEE Trans. Fuzzy Syst. 19 (2011) 960–971.
[28] Z. Deng, Y. Jiang, F.-L. Chung, H. Ishibuchi, S. Wang, Knowledge-leverage-based fuzzy system and its modeling, IEEE Trans. Fuzzy Syst. 21 (2013) 597–609.
[29] J. Guevara, R.H. Jr., S. Canu, Kernel functions in Takagi-Sugeno-Kang fuzzy system with nonsingleton fuzzy input, in: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Hyderabad, India, 2013.
[30] M. Sinaee, E.G. Mansoori, Fuzzy rule based clustering for gene expression data, in: Proceedings of the 4th International Conference on Intelligent Systems, Modelling and Simulation, Bangkok, 2013, pp. 7–11.
[31] M. Attik, L. Bougrain, F. Alexandre, Self-organizing map initialization, in: W. Duch, J. Kacprzyk, E. Oja, S. Zadrożny (Eds.), Artificial Neural Networks: Biological Inspirations – ICANN 2005, Springer, Berlin Heidelberg, 2005, pp. 357–362.
[32] H.-U. Bauer, K.R. Pawelzik, Quantifying the neighborhood preservation of self-organizing feature maps, IEEE Trans. Neural Netw. 3 (1992) 570–579.
[33] G. Polzlbauer, Survey and comparison of quality measures for self-organizing maps in: Proceedings of the Fifth Workshop on Data Analysis (WDA'04), 2004, pp. 67–82.
[34] J. Vesanto, Data Exploration Process Based on the Self-Organizing Map (Ph.D. thesis), Helsinki University of Technology, Finland, 2002.
[35] A. Bouchet, J.I. Pastore, R.E. Andrade, M. Brun, V. Ballarin, Arithmetic mean based compensatory fuzzy logic, Int. J. Comput. Intell. Appl. 10 (2011) 231–243.
[36] R.E. Andrade, J.M. Gómez, G.M. Téllez, E.F. González, Compensatory logic: a fuzzy approach to decision making, in: Proceedings of 4th International Symposium on Engineering of Intelligent Systems (EIS'2004), Madeira, Portugal, 2004.
[37] B. Verma, A. Rahman, Cluster-oriented ensemble classifier: impact of multicluster characterization on ensemble classifier learning, IEEE Trans. Knowl. Data Eng. 24 (2012) 605–618.
[38] I. Batyrshin, O. Kaynak, I. Rudas, Fuzzy modeling based on generalized conjunction operations, IEEE Trans. Fuzzy Syst. 10 (2002) 678–683.
[39] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, J. Intell. Inf. Syst. 17 (2001) 107–145.
[40] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, California, USA, 2008.
[41] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, Modeling wine preferences by data mining from physicochemical properties, Decis. Support Syst. 47 (2009) 547–553.
[42] R.K. Kwan, A.C. Evans, G.B. Pike, MRI simulation-based evaluation of image-processing and classification methods, IEEE Trans. Med. Imaging 18 (1999) 1085–1097.
[43] S.N. Ghazavi, T.W. Liao, Medical data mining by fuzzy modeling with selected features, Artif. Intell. Med. 43 (2008) 195–206.
[44] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag, New York, 2006.

Gustavo J. Meschino received Ph.D. from the Universidad Nacional de Mar del Plata, Argentina (UNMdP) in 2008. He has been working for this University as a researcher since 1998. He is a Professor of Computational Intelligence and Electronic Circuits. Permanently he acts as a director of undergraduate and graduate students works based on Computational Intelligence applications. His current research interest includes Computational Intelligence applications for optimization, prediction, classification and clustering, and Image Processing, especially in the Bioengineering area.

Diego S. Comas received Electronic Engineering degree by Facultad de Ingeniería from the Universidad Nacional de Mar del Plata (UNMdP), Mar del Plata, Argentina, in 2010, and he is currently coursing his Ph.D. studies at the same University, developing his thesis on Type-2 Fuzzy Logic applied to medical image segmentation, supported by a CONICET Scholarship. He is an Assistant Professor of Probability, Stochastic Processes and Image Processing. His current research interests are Fuzzy Logic and digital image processing.



Virginia L. Ballarin received Ph.D. in Biological Sciences – Bioengineering from the University of Tucumán, Argentina, in 2003 and M.S. and B.S. degrees in Electronics Engineering from the University of Mar del Plata (UNMdP) in 1998 and 1994 respectively. She is a Professor of Stochastic Processes and Digital Image Processing. She is the leader of the Digital Image Processing Lab with research emphasis in mathematical morphology, pattern recognition and medical image processing. She is the Vice President of Argentinean Society of Bioengineering. She is the Secretary of the WIE IEEE chapter and member of the IEEE Engineering in Medicine and Biology Society.



Adriana G. Scandurra received Ph.D. from the Universidad Nacional de Mar del Plata (UNMdP) in 2010. She is an assistant professor at the Electronic Department of the Facultad de Ingeniería at the UNMdP. She is also a researcher since 1999, acting as a co-director on Bioengineering projects. Currently, she is a member of Computational Intelligence Group of the Laboratory of Bioengineering. Her current research interests include Fuzzy Systems, Evolutionary Computation and Data Modeling.

Lucía I. Passoni received Ph.D. from the Universidad Nacional de Mar del Plata (UNMdP) in 2005. She is a full-time professor at the Electronic Department of the Facultad de Ingeniería at the UNMdP. She is also a researcher since 1994. Currently, she is the leader of Computational Intelligence Group of the Laboratory of Bioengineering (UNMdP). Her current research interests include Neural Networks, Fuzzy Systems, Evolutionary Computation and Knowledge Management.