# A Survey of Fuzzy Clustering Algorithms for Pattern Recognition—Part I

Andrea Baraldi and Palma Blonda, *Member, IEEE*

***Abstract*— Clustering algorithms aim at modeling fuzzy (i.e., ambiguous) unlabeled patterns efficiently. Our goal is to propose a theoretical framework where the expressive power of clustering systems can be compared on the basis of a meaningful set of common functional features. Part I of this paper reviews the following issues related to clustering approaches found in the literature: relative (probabilistic) and absolute (possibilistic) fuzzy membership functions and their relationships to the Bayes rule, batch and on-line learning, prototype editing schemes, growing and pruning networks, modular network architectures, topologically perfect mapping, ecological nets and neuro-fuzziness. From this discussion an equivalence between the concepts of fuzzy clustering and soft competitive learning in clustering algorithms is proposed as a unifying framework in the comparison of clustering systems. Moreover, a set of functional attributes is selected for use as dictionary entries in the comparison of clustering algorithms, which is the subject of Part II of this paper [1].**

***Index Terms*—Ecological net, fuzzy clustering, modular architecture, relative and absolute membership function, soft and hard competitive learning, topologically correct mapping.**

## I. Introduction

In recent years the synthesis between clustering algorithms and fuzzy set theory has led to the development of several so-called fuzzy clustering algorithms whose aim is to model fuzzy (i.e., ambiguous) unsupervised (unlabeled) patterns efficiently. The goal of this paper is to review and compare self-organization strategies of clustering algorithms that have been called fuzzy, or can be considered fuzzy, according to some definitions found in the existing literature. To the best of our knowledge, few such comparative studies have been attempted (e.g., [2]). This may be due to the difficulty of objectively comparing the great variety of clustering approaches based on a meaningful set of common functional features.

To select a meaningful set of functional features for use as dictionary entries in the comparison of clustering algorithms, Part I of this paper revises concepts such as: relative and absolute fuzzy membership functions (Section II), batch and on-line learning (Section III), prototype vector editing schemes (Section IV), growing and pruning networks (Section V-B), modular network architectures (Section V-C), topologically correct mapping (Section VI), ecological nets (Section VII) and neuro-fuzziness (Section VIII). From the existing literature we also derive our own interpretation of fuzzy clustering inductive learning, intended as a synonym of soft competitive parameter adaptation in clustering systems. This conceptual equivalence is employed as a unifying framework in the comparison of clustering algorithms.

This approach has been considered "interesting" and "quite reasonable" by some researchers [3]. However, other authors believe that since "fuzziness can be incorporated at various levels to generate a fuzzy neural network, i.e., it can be at the input, output, learning or neural levels" (see [4], where each input feature is expressed in terms of fuzzy membership values indicating a degree of belonging to each of the linguistic properties *low*, *medium*, and *high*), our "claim of calling certain networks to be fuzzy/nonfuzzy seems improper and not acceptable" [3]. In both cases, comments pertaining to the proposed terminology "fuzzy clustering algorithm" do not affect the core of this work, which is centered on the comparison of learning (self-organizing) mechanisms adopted by several clustering algorithms to model fuzzy (ambiguous) patterns.

The set of functional attributes selected for use as dictionary entries in the comparison of clustering algorithms is summarized in Section IX.

## II. Fuzzy Membership and Probability Density Functions

This section proposes a brief review of probabilistic and possibilistic fuzzy membership concepts, to be compared with Bayes' view of posterior probability and likelihood.

### A. Absolute and Relative Fuzzy Memberships

Let $X_k$ be instance $k$ of the input data set $\{X\}$, where $k = 1, \ldots, n$, such that $n$ is the total number of input instances. Let us assume that $X_k$ may belong to a generic *state* (also termed *category* or *component*) $C_i$, $i = 1, \ldots, c$, where $c$ is the total number of possible states. The extent to which $X_k$ is compatible with a vague (fuzzy) concept associated with generic state $C_i$ can be interpreted "more in terms of a *possibility (compatibility) distribution* rather than in terms of a *probability distribution*" [18, p. 58]. This legitimizes some possibility distributions, called fuzzy membership functions, that "we believe are *useful*, but might find difficult to justify on the basis of objective probabilities" [18, p. 57]. Depending on the conditions required to state that $c$ fuzzy states $C_i$, $i = 1, \ldots, c$, are a fuzzy $c$-partition of the input data set,

Manuscript received November 2, 1998; revised August 26, 1999.

A Baraldi is with ISAO-CNR, Bologna 40129, Italy (e-mail: andrea@aliseo.imga.bo.cnr.it). He is also with International Computer Science Institute, Berkeley, CA 94704 USA (e-mail: baraldi@icsi.berkeley.edu).

P. Blonda is with CNR-IESI, Bari 70126, Italy (e-mail: blonda@iesi.ba.cnr.it).

Publisher Item Identifier S 1083-4419(99)09699-5.

1083–4419/99$10.00 © 1999 IEEE

membership functions can be divided into two categories [19], [20]

1. *relative or probabilistic or constrained fuzzy membership (typicality) values* $R_{i,k}$;
2. *absolute or possibilistic fuzzy membership values* $A_{i,k}$;

where index $k$ ranges over patterns and index $i$ over concepts. Absolute and relative membership types are related by the following equation:

$$R_{i,k} = \frac{A_{i,k}}{\sum_{h=1}^{c} A_{h,k}}, \quad i = 1, \ldots, c, \quad k = 1, \ldots, n. \tag{1}$$

Relative typicality values, $R_{i,k}$, must satisfy the following three conditions [8], [18]:

1) $R_{i,k} \in [0,1]$, $i = 1, \ldots, c$, $k = 1, \ldots, n$;
2) $\sum_{i=1}^{c} R_{i,k} = 1$, $k = 1, \ldots, n$;
3) $0 < \sum_{k=1}^{n} R_{i,k} < n$, $i = 1, \ldots, c$.

Constraint 2) is an inherently probabilistic constraint [19], relating $R_{i,k}$ values to posterior probability estimates in a Bayesian framework. Because of condition 2), $R_{i,k}$ values are relative numbers dependent on the absolute membership of the pattern in all other classes, thus indirectly on the total number of classes. This also means that processing elements (PE's) exploiting a relative membership function as their activation function are context-sensitive, i.e., $R_{i,k}$ provides a tool for modeling network-wide internode communication by assuming that PE's are coupled through feed-sideways (lateral) connections [21].

Although possibilistic membership functions, $A_{i,k}$, $i = 1, \ldots, c$, $k = 1, \ldots, n$, may satisfy conditions 1) and 3) listed above (in this case, the upper bound of the membership function is one and the fuzzy set is termed normal [18]), they always differ from probabilistic memberships in condition 2), which is relaxed as follows [19]

4) $\max_i \{A_{i,k}\} > 0$, $k = 1, \ldots, n$.

Owing to condition 4), the sum of absolute memberships of a noise point in all the "good" categories need not be equal to one. Term $A_{i,k}$ is an absolute similarity value depending on fuzzy state $C_i$ exclusively, given input pattern $X_k$. In other words, $A_{i,k}$ is context-insensitive, since it is not affected by any other state. Thus, PE's exploiting an absolute membership as their activation function are independent, i.e., they feature no lateral connection.

Both probabilistic and possibilistic fuzzy clustering are affected by some well-known drawbacks. On one hand in probabilistic fuzzy clustering, owing to condition 2), noise points and outliers, featuring low possibilistic typicalities with respect to all templates (codewords), may have significantly high probabilistic membership values and may severely affect the prototype parameter estimate (e.g., [20]). On the other hand, in possibilistic fuzzy clustering, learning rates computed from absolute typicalities tend to produce coincident clusters [20], [22]. This poor behavior can be explained by the fact that cluster prototypes are uncoupled in possibilistic clustering, i.e., possibilistic clustering algorithms try to minimize an objective function by operating on each cluster independently. This leads to an increase in the number of local minima.

Different $A_{i,k}$ expressions in the existing literature and consistent with the definition provided above were found to be useful. These include the following:

$$A_{i,k} = \begin{cases} \frac{1}{1+d_{i,k}^2/\eta_i} \in (0,1] \quad [19], & (2) \\ \text{Gaussian}_{i,k} = e^{-\frac{d_{i,k}^2}{2\sigma_i^2}} \in (0,1] \\ \quad \text{(Gaussian mixtures; [23], [24])}, & (3) \\ \frac{1}{(d_{i,k}^2)^{p_i}} \in (0,\infty) \quad [25], & (4) \\ \frac{1}{(1-\text{Gaussian}_{i,k})^2} \in (1,\infty) \quad [17], & (5) \end{cases}$$

where $d_{i,k} = d(X_k, T_i)$ is assumed to be the Euclidean distance between input pattern $X_k$ and prototype (receptive field center) $T_i$ of the $i$-th category. Variables $\sigma_i$, $\eta_i$ and $p_i$ are all scale parameters belonging to range $(0,\infty)$ (see [20]).

### B. Fuzzy Memberships and Mixture Probabilities

To investigate the relationship between (objective) probability density functions and (useful) fuzzy membership functions, note that absolute membership function (3) relates probabilistic membership (1) to Gaussian mixture models, which are widely employed in the framework of optimization problems featuring a firm statistical foundation [26]–[29]. In a mixture probability model consisting of $c$ mixture components $C_i$, $i = 1, \ldots, c$, let $p(C_i)$ be the *a priori* probability that a pattern belongs to mixture component $C_i$, and $p(X_k \mid C_i)$ the conditional likelihood that the pattern is $X_k$, given that the pattern's state is $C_i$. If these statistics are known, *a posteriori* conditional probability $p(C_i \mid X_k)$ can be estimated using Bayes' rule as

$$p(C_i \mid X_k) = \frac{p(X_k \mid C_i) \cdot p(C_i)}{\sum_{h=1}^{c} p(X_k \mid C_h) \cdot p(C_h)}, \quad k = 1, \ldots, n, \quad i = 1, \ldots, c. \tag{6}$$

If $p(C_h) = 1/c, \forall h \in \{1, c\}$, i.e., all states are assumed to be equally likely, then (6) becomes

$$p(C_i \mid X_k) = \frac{p(X_k \mid C_i)}{\sum_{h=1}^{c} p(X_k \mid C_h)}, \quad k = 1, \ldots, n, \quad i = 1, \ldots, c. \tag{7}$$

The following relationships hold true:

1) $p(C_i \mid X_k), p(X_k \mid C_i)$ and $p(C_i)$ belong to range $[0,1]$;
2) $\sum_{h=1}^{c} p(C_h \mid X_k) = 1$, $k = 1, \ldots, n$, i.e., mixture components $C_i$, $i = 1, \ldots, c$, provide a complete partition of the input space;
3) $\sum_{h=1}^{c} p(C_h) = 1$.

From the comparison of (1) with (7), and of properties 1)–4) in Section II-A with properties 1)–3), in this section we can write:

*when priors are considered the same (i.e., they are ignored),* since $\{p(X_k \mid C_i)\} \subset \{A_{i,k}\}$, thus, $\{p(C_i \mid X_k)\} \subset \{R_{i,k}\}$; in other words, *when priors are ignored, (objective) likelihood and posterior probabilities are a subset of (useful) absolute and relative fuzzy membership functions, respectively.*

To summarize, the combination of (1) with constraints 1)–4) in Section II-A allows the human designer to choose any absolute membership function that, in addition to satisfying the mild condition 4) of Section II-A, is considered useful for the application at hand, although this choice may be difficult to justify on the basis of objective probabilities (see Section II-A). If the chosen absolute membership function satisfies, not only condition 4) of Section II-A, but the more severe constraint 1) of that section, then absolute membership values are equivalent to likelihood values; as a consequence, relative membership values computed with (1) can be considered posterior probability estimates in the case in which priors are ignored.

## III. On-Line versus Off-Line Model Adaptation

All inductive learning systems, i.e., learning systems progressing from special cases (e.g., training data) to general models, are based on the following concepts:

1) information extracted from a finite set of observed examples, termed *training data set*, can be used to answer questions either about unobserved samples belonging to a so-called *test set*, or about unknown properties hidden in the training data;
2) the goal of the learning process is to minimize a risk functional (theoretically computed over an infinite data set) by adapting system parameters on the basis of the finite training set [30], i.e., the learning problem is turned into an optimization problem [31].

When system parameters are learned from training data, there are two classes of learning situations, depending on how data are presented to the learner: the "batch" setting in which data are available as a block, and the "on-line" setting in which data arrive sequentially [31]. In many practical problems, when a sequential data stream can be stored for analysis as a block, or a block of data can be analyzed sequentially, the user is free to take either the batch or the on-line point of view [31].

The goal of on-line learning methods is to avoid storage of a complete data set by discarding each data point once it has been used [32]. On-line learning methods are required when

1) it is necessary to respond in real time;
2) input data set may be so huge that batch methods become impractical, because of their numerical properties (see below), or computation time, or memory requirement;
3) input data comes as a continuous stream of unlimited length which makes it totally impossible to apply batch methods [29], [31], [32].

On-line learning typically results in systems that become order-dependent during training (in line with biological complex systems [33]). Moreover, on-line systems, where parameter adaptation is example-driven [34], are more sensitive to the presence of noise as they do not average over the noise on the data, i.e., they tend to provide highly oscillatory (nonsmooth) output functions that perform poorly in terms of generalization ability. For example, clustering systems like fuzzy adaptive resource theory (ART), where a single poorly mapped pattern suffices to initiate the creation of a new unit, may be affected by overfitting, i.e., the system may fit the noise and not just the data.

Batch methods are preferred when our only interest is in a final answer, i.e., the best answer that we can obtain from a finite training data set as the exact closed form solution to a minimization problem [31]. In batch learning problems (e.g., the simple case of linear model regression, see [31] and [32, p. 92]), exact closed form solutions can lead to numerical difficulties for very large data sets. In these cases the only computationally feasible alternative is provided by iterative batch methods, such as the gradient descent of the cost function [32], [34], that sweep repeatedly through the data set. To summarize, batch learning methods are subdivided into

1) exact closed form solutions to the cost function minimization problem; these solutions are numerically and/or computationally inapplicable for very large data sets;
2) iterative batch learning algorithms, such as the gradient descent of the cost function.

In iterative batch learning algorithms, the learning rate parameter must be small and its choice is fairly critical: if the learning rate is too small the reduction in error will be very small; if it is too large instabilities (divergent oscillations) may result [32]. Analytically, when convergence of iterative batch algorithms to exact closed form solutions is analyzed, then useful hints on constraint of the learning rate value are gathered [31].

Although iterative batch learning algorithms are developed to process very large data sets, they can spend an enormous amount of computation time in processing the entire training set (e.g., in order to compute the gradient), but they may end up by taking a small learning step in the parameter space after each processing epoch. If this is the case (e.g., for gradient descent algorithms), this functional feature is clearly incompatible with the original motivation that justifies the study of such iterative batch learning schemes. In such situations, an alternative solution, called "mini-batch" [31], is to average parameter update values over subsets of the entire training data set. By taking intermediate steps in the parameter space, iterative mini-batch algorithms may converge faster than their iterative full batch counterparts. By stretching the same idea further, another alternative solution to iterative full batch algorithms is to develop their on-line (stochastic)[1] approximations [32].

On-line learning algorithms are simple and intuitive because they are based on the following heuristic [32, p. 46], [35]: the sum over the training samples, which is found in the iterative and batch solution of the cost function minimization task, is dropped by separating the contribution from the last data point to provide a sequential update formula, i.e., to allow one parameter update for every new data point presentation. Although this heuristic seems reasonable, there should be some analytical proof that the on-line procedure converges to a solution. It is known that the difference between exact batch-mode and heuristic on-line updating is arbitrarily reduced by

[1] "Stochastic" refers to the assumption that the single data point being presented to the learning system is chosen on the basis of a stochastic process [31].

the adoption of "small" learning rates [34], [36]. According to the view that on-line procedures are approximations of iterative batch algorithms, learning rate constraints capable of guaranteeing convergence of the iterative batch mode may be applied to the on-line problem under an appropriate definition of convergence [37] (for the linear regression case, see [31]). If these conditions hold, it has been observed that on-line learning systems, by requiring significantly less computation time per parameter update, can be significantly faster in converging than iterative batch algorithms [31].

In general, the learning rate $\alpha(t)$ of the on-line update rule must satisfy the three conditions applied to the coefficients of the Robbins–Monro algorithm for finding the roots of a function iteratively, which are (see [32, pp. 47, 96]):

1) $\lim_{t\to\infty} \alpha(t) = 0$;
2) $\sum_{t=1}^{\infty} \alpha(t) = \infty$;
3) $\sum_{t=1}^{\infty} \alpha^2(t) < \infty$.

For example, in an on-line update procedure, if the learning rate remains fixed, then the algorithm converges only in a stochastic sense [31], i.e., model parameters drift from their initial positions to quasistationary positions where they start to wander around in dynamic equilibrium [14]. When the learning rate decreases monotonically under Robbins–Monro conditions, e.g., $\alpha(t) = 1/t$ (see [32, p. 96], and [14]), on-line learning algorithms can be shown to converge to a point in the parameter space [31], [37]. As a brief review of batch update and on-line update techniques, refer to [14].

## IV. Prototype Vector Editing Scheme

Basically, clustering algorithms employ two reference vector generation schemes, termed clustering-by-selection and clustering-by-replacement respectively [38]. In clustering-by-selection the learning algorithm selects prototype vectors $C_i$, $i = 1, \ldots, c \leq n$, as a subset of the input data set $\{X\} = \{X_1, \ldots, X_n\}$. Input patterns selected as prototype vectors are also called support vectors [39]. Typical application fields of clustering-by-selection algorithms are perceptual grouping, hidden data-structure detection and pattern classification (when the clustering algorithm is integrated in a classification system) [38]. In clustering-by-replacement vector prototypes are a transformation of the input vectors within the pattern space. Besides perceptual grouping and hidden data-structure detection [40], clustering-by-replacement algorithms can also be applied to data requantization tasks, i.e., to detect compact data coding [5], [7], [10], [14], [16], [26], [41].

## V. Beyond Error Gradient Descent: Advanced Techniques for Learning from Data

In recent years, the neural network community has made a considerable effort in the search for learning techniques that are more effective in dealing with local minima and generalization to new examples than the traditional approaches based on simple gradient descent. These alternative approaches have to deal effectively with the *curse of dimensionality* and with the qualitative principle known as *Occam's razor*. As an example of the curse of dimensionality, consider that any function estimator increases its number of adjustable parameters with the dimensionality of input space. As a consequence, the size of training data required to compute a reliable estimate of adaptive parameters may become huge in practical problems [30], [32].

The complexity of a learning system increases with the number of independent and adjustable parameters, also termed degrees of freedom, to be adapted during the learning process. According to the qualitative principle of Occam's razor, a sound basis for generalizing beyond a given set of examples is to prefer the simplest hypothesis that fits observed data [32], [34]. This principle states that to be effective, the cost function minimized by an inductive learning system should provide a trade-off between how well the model fits the training data and *model complexity*. This also means that model complexity must be controlled by *a priori* (background) knowledge, i.e., subjective knowledge available before any evidence (e.g., empirical risk) provided by the training data is observed. Different inductive principles provide cost functions considered as different quantitative formulations of Occam's qualitative principle [30].

A rough taxonomy of advanced techniques for optimal learning, originally proposed in [35], is presented in the following.

### A. Global Optimization

Instead of local algorithms like gradient descent, one may explore techniques that guarantee global optimization while effectively facing the curse of dimensionality. Among the most significant developments in this area, support vector machines (SVM's), based on the Vapnik–Chervonenkis (VC) statistical learning theory and capable of detecting the global minimum of a cost function for classification problems, are becoming increasingly popular in finding solutions to both classification and function regression tasks [30], [39], [42].

### B. Growing Networks and Pruning

Human designers typically have the opportunity to embed task-specific prior knowledge in an inductive learning algorithm, e.g., by setting the topology and the complexity of a multilayer perceptron when backpropagation weight adaptation is applied. Pruning algorithms begin training a network expected to be large with respect to the problem at hand, and then continue by pruning nodes that do not affect the learning process significantly [34], [35]. Vice versa, growing networks start from small networks that grow gradually until convergence is reached [14], [35]. As a result, the complexity of the network is expected to be tuned to the problem at hand, i.e., generalization capability is expected to increase.

### C. Modular Architectures: Prior Structures and Experience-Based Fine-Tuning

Self-organizing neurological systems consist of highly structured, hierarchical architectures provided with feedback mechanisms [43]. In these systems, the combination of an initial architecture, produced by evolution, with experience-based additional fine-tuning prepares the whole system to function

in an entire domain by generalizing its learned behavior to instances not previously encountered [44].

In line with biological learning systems, a classical engineering paradigm consists of partitioning the solution to a problem between several modules specialized in learning a single task, i.e., modular architectures are the natural solution to most significant practical problems [33], [34]. In applied mathematics, the principle of tackling a problem by dividing it into simpler subproblems whose solutions can be combined to yield a solution to the complex problem is termed *divide and conquer* [45]. An application of this strategy can be found in [46] and [47].

In supervised learning, an interesting modular proposal that addresses the major problem of providing effective integration of the system modules is presented in [45].

Analytically, the importance of developing modular architectures has been stressed in [35], [48], where sufficient (but not necessary) conditions capable of guaranteeing local minima free cost functions are detected, such that a simple gradient descent algorithm can always reach the absolute minimum of the error surface.

Contiguous to the problem of fine-tuning modular learning systems on the basis of training experiences is the problem of prior structures, i.e., the problem of learning from tabula rasa [35]. Minsky claims that a "significant learning at significant rate presupposes some significant prior structure" [49]. In other words, important properties of the model must be "hard-wired or built-in, perhaps to be tuned later by experience, but not learned in any statistical meaningful way" [50].

Intuitively, starting from some prior learning structure, experience-based (inductive) fine-tuning of network parameters should allow a structured organization of a distributed system to emerge naturally from elementary interactions of PE's. This is tantamount to saying that competitive adaptation of distributed (localized: neuron-based, synapse-based) parameters is expected to enhance the development of structured nets consisting of specialized subsystems (modules), in line with biological neural systems.

With regard to the exploitation of distributed parameters, it can be observed that, on the one hand, most of the on-line clustering algorithms presented in the existing literature, e.g., the self-organizing map (SOM) [5], exploit a *global* (network-based) time counter, i.e., a time variable that is not specialized on a localized basis, rather than *distributed* (localized: neuron-based, synapse-based) time variables. As a consequence, at a given processing time, plasticity (i.e., the potential ability of moving a template vector toward an input pattern) is the same for every PE, in these networks. On the other hand, in recent years, several on-line Kohonen-based network models exploiting distributed time variables have been presented [12]–[15]. For example, in growing neural gas (GNG) [13], [14] and in the fully self-organizing simplified ART (FOSART) [12], [15], a connection-based time variable is equal to the number of times one lateral connection is selected at adaptation steps. In the fuzzy simplified art algorithm and in FOSART [15] a neuron-based time variable is equal to the number of epochs that PE has survived [12], [15]. To summarize, on-line clustering algorithms found in the literature depend to different degrees on distributed parameters whose competitive adaptation is consistent with the development of structured systems (i.e., specialized subsystems).

With regard to batch and iterative algorithms, such as the Fuzzy $c$-means (FCM) and fuzzy learning vector quantization (FLVQ) algorithm [7], [8], global time rather than distributed time variables are employed, i.e., at a given processing time, plasticity is the same for every PE in the net. This is justified by considering that when fuzzy clustering mechanisms are employed, all PE's acquire the same input pattern simultaneously. However, one may consider that fuzzy membership functions allow a pattern to belong to multiple categories to different degrees. In other words, in batch algorithms the plasticity of every PE is considered as being the same, even though learning "histories" of PE's may differ significantly. This becomes obvious if, for example, for every PE we define a distributed (neuron-based) time counter (equivalent to an *inverse plasticity* or *stability* variable) as the sum of the neuron learning rates at adaptation steps, such that the learning rate of each PE decreases monotonically with its local time. To summarize, in batch clustering algorithms found in the literature, exploitation of distributed rather than global parameters may deserve further investigation in the framework of developing self-organizing networks consisting of specialized subsystems.

## VI. Topologically Correct Mapping

The presentation of the competitive Hebbian rule (CHR) [49], introducing competition among lateral connections (synapses), represented a fundamental breakthrough in the evolution of fully self-organizing artificial neural network models (FSONN). In this paper, a synaptic link is defined as a lateral connection between two PE's belonging to the same neural layer, and FSONN models are defined as those distributed systems capable of

1) dynamically generating and removing PE's;
2) dynamically generating and removing lateral connections.

CHR is an exemplar-driven connection rule generating lateral connections as follows. For a given pattern $X_k$, $k \in \{1, n\}$, where $n$ is the total number of input patterns, let us consider

1) winner unit $\mathrm{PE}_{w1(X_k)}$ as the one featuring the shortest inter-pattern distance $d(X_k, T_{w1(X_k)}) \leq d(X_k, T_i)$, $i = 1, \ldots, c$, where $c$ is the total number of PE's, $T_{w1(X_k)}$ is the template pattern of processing unit $\mathrm{PE}_{w1(X_k)}$, and $d(X_k, T_i)$ is the Euclidean inter-pattern distance between $X_k$ and $T_i$;
2) second best unit, $\mathrm{PE}_{w2(X_k)}$, as the one featuring activation value $d(X_k, T_{w2(X_k)}) \leq d(X_k, T_i)$, $i = 1, \ldots, c$, $i \neq w1(X_k)$.

The exploitation of the Euclidean inter-pattern distance in competitive learning shapes neuron receptive fields as Voronoi polyhedra [51]. According to CHR, if connection between $\mathrm{PE}_{w1(X_k)}$ and $\mathrm{PE}_{w2(X_k)}$ does not exist, it is generated.

Under the hypothesis that the distribution of template vectors $T_i$, $i = 1, \ldots, c$, is dense on input manifold $X$, i.e., for each input $X_k$, $k = 1, \ldots, n$, triangle

$\triangle(X_k, T_{w1(X_k)}, T_{w2(X_k)})$ lies completely on $X$, it is proved that CHR

1) forms an output graph (lattice, network) which is the *induced Delaunay triangulation* of codewords $T_i$, $i = 1, \ldots, c$;
2) forms a topology preserving map (TPM) of $X$ in the sense proposed in [51].

To define a TPM in the sense proposed in [51], let us consider an input manifold $X \subseteq R^D$, where $D$ is the dimensionality of input space, and a graph (network) $G$, consisting of vertices (neural units) $\mathrm{PE}_i$, $i = 1, \ldots, c$, such that a category template $T_i \in X$, belonging to the pointer set $\{T\} = \{T_1, \ldots, T_c\}$, is related ("attached") to vertex $\mathrm{PE}_i$. Given codebook $\{T\}$, mapping $\phi_T$ from input manifold $X$ onto the vertices of $G$ is defined as

$$\phi_T : X \to G, \quad X_k \in X \to w1(X_k) \in G \tag{8}$$

where $X_k$ is a feature vector (input pattern), $k \in \{1, n\}$, and vertex $\mathrm{PE}_{w1(X_k)}$, also termed winner unit, is determined by

$$d(X_k, T_{w1(X_k)}) \leq d(X_k, T_i), \quad \forall i \in G \tag{9}$$

where $d(\cdot)$ is the Euclidean inter-vector distance. According to (8) and (9), a feature vector $X_k$ is mapped to vertex $\mathrm{PE}_{w1(X_k)}$, the pointer $T_{w1(X_k)}$ of which is closest to $X_k$. This is equivalent to stating that $X_k$ is mapped to vertex $\mathrm{PE}_{w1(X_k)}$ whose Voronoi polyhedron $V_{w1(X_k)}$ encloses $X_k$. The Voronoi polyhedron of a neuron $\mathrm{PE}_i$ is the receptive field centered on $T_i$, and it is identified as $V_i \subseteq R^D$. The masked Voronoi polyhedron of neuron $i$ is defined as $V_i^{(X)} = V_i \cap X \subseteq X$ [51].

Inverse mapping $\phi_T^{-1}$ from $G$ onto $X$ is defined as

$$\phi_T^{-1} : G \to X, \quad i \in G \to T_i \in X. \tag{10}$$

Two pointers $T_i$, $T_j$ are termed adjacent on the feature manifold $X$ if their masked Voronoi polyhedra are adjacent, i.e., if $V_i^{(X)} \cap V_j^{(X)} \neq 0$. Two vertices $\mathrm{PE}_i$, $\mathrm{PE}_j$ in $G$ are termed adjacent if they are connected by a lateral connection. Mapping $\phi_T$ from $X$ to $G$ is defined as neighborhood (adjacency) preserving if any pair of adjacent pointers $T_i$, $T_j$ on $X$ are assigned to vertices $\mathrm{PE}_i$, $\mathrm{PE}_j$ that are adjacent on $G$. Mapping $\phi_T^{-1}$ from $G$ to $X$ is defined as neighborhood preserving if any pair of vertices $\mathrm{PE}_i$, $\mathrm{PE}_j$ that are adjacent on $G$ are assigned to locations $T_i$, $T_j$ that are adjacent on $X$.

Thus, a TPM is defined as a mapping $\phi_T$ from $X$ to $G$ such that $\phi_T$, together with inverse mapping $\phi_T^{-1}$ from $G$ to $X$, are neighborhood (adjacency) preserving [51].

Note that exploitation of CHR in FSONN's allows generation of networks consisting of mutually disjointed (specialized and independent) maps [14], [51]. This modular organization enables the learning system to perform both *cooperative learning*, by adapting concertedly those processing units that are connected within graph $G$ (i.e., those units that belong to the same map in the graph), and competitive learning among disjointed maps to enhance specialization of the system's modules.

## VII. Artificial Cognitive Systems and Ecological Nets

To perform cognitive tasks, biological neural systems exploit

1) dishomogeneous nets, where several types of PE's are combined;
2) structured architectures, consisting of hierarchies of subnets;
3) feedback mechanisms, where feedback information is provided by the external environment to the natural system in response to the system's actions [52].

It is *the presence of this feedback interaction with the environment that characterizes all natural systems featuring cognitive capabilities* [33]. Some artificial neural systems feature none of the biological properties listed above. For example, SOM [5] and the Hopfield network [53] are homogeneous systems; they feature no structured architecture and no *supervision* or *reinforcement* by, or *feedback* from, an external environment (also termed *supervisor*). In *reinforcement learning*, the neural system is allowed to react to each training case. It is then told whether its reaction was effective or not [54]. To increase their biological plausibility, artificial neural models should employ differentiated structures provided with dishomogeneous layers, specialized subnets, hierarchies of maps, etc. In parallel, the study of artificial neural nets as stand-alone systems should evolve to become the science of *ecological nets* (econets), where neural systems as well as their external environments are modeled [52]. For example, unlike Kohonen's networks [6], an ART system employs a structured architecture to self-adjust the network dimension to problem-specific conditions. In particular, the ART *orienting subsystem* models the responses of the external environment to the learning activities of the *attentional subsystem* [10], [11], [55], [56]. Thus, an ART system belongs to the class of ecological nets.

## VIII. On Fuzzy Clustering Algorithms

A clustering algorithm performs unsupervised detection of statistical regularities in a random sequence of input patterns.

Our attention is focused on fuzzification of clustering learning schemes. In the definition presented in [9], it is stated that an artificial neural network model performs *fuzzy clustering* when it allows a pattern to belong to multiple categories to different degrees depending on the neurons' ability to recognize the input pattern. This approach is well known in the traditional field of coding techniques for data compression. Since the traditional $c$-means clustering algorithm feature a cost function (discretization error) characterized by many local minima, data compression techniques modify $c$-means algorithms by replacing their *winner-take-all* strategy (WTA), also termed crisp or hard competitive, with a "soft-max" adaptation rule [26], hereafter referred to as soft competitive learning.[2]

A WTA parameter adaptation strategy is purely competitive and allows no cooperative (soft competitive) learning. It is

[2] Note that terms soft-max and soft competitive learning rule adopted in this paper are not to be confused with the so-called soft-max function or normalized exponential employed in mixture models and mixture-of-experts [32], [45].

sensitive to initialization of templates, i.e., different initializations may lead to very different minimization results. In fact, WTA adaptations may not be able to get the system out of a poor local minimum when this lies in the proximity of the status where the system was started [14]. Unlike WTA learning, a *soft competitive learning* scheme is defined as a learning strategy that not only adjusts the winning cluster but also affects all cluster centers depending on their proximity to the input pattern [26]. In general, soft competitive learning decreases dependency on initialization and reduces the presence of dead units [14]. We observe the following.

1. The fuzzy clustering definition provided in [9] is equivalent to the definition of the soft competitive adaptation rule traditionally employed in the field of data compression [26], i.e., a clustering algorithm is termed *fuzzy clustering algorithm* iff it employs a soft competitive (noncrisp) parameter adaptation strategy.
2. As a corollary of 1), a fuzzy clustering algorithm does not necessarily exploit concepts derived from fuzzy set theory such as fuzzy set membership functions and fuzzy set operations. For example, SOM pursues soft competitive learning by means of biologically plausible update rules that employ no fuzzy set-theoretic concept. Another interesting example is the one provided by the expectation–maximization (EM) algorithm applied to optimize parameters of a Gaussian mixture [27], [32]. Although it features a firm statistic foundation and employs no fuzzy set-theoretic concept, EM applied to Gaussian mixtures can be termed fuzzy according to definition 1) presented above.

It should be noted that relative or probabilistic fuzzy membership functions are traditionally applied to clustering algorithms in the intuitive belief that "vector quantizers based on both winner and nonwinner information about the relationship of an input pattern to the prototypes will be better representatives of the overall structure of the input data than those based on local information alone" [7]. Thus, several clustering algorithms, such as FCM and FLVQ algorithms, combine local and global information in the computation of a relative fuzzy membership function [7], [8]. From a functional standpoint, connectionist models where a "useful" relative fuzzy membership function or an objective posterior probability estimate is computed (e.g., FLVQ and EM applied to Gaussian mixtures, respectively), are equivalent to distributed systems where a contextual (competitive and cooperative) effect mechanism employing feedsideways (intra-layer) connections is employed (see Section II). Note that only few clustering networks employ intra-layer connections explicitly, i.e., by means of specific data structures and parameter adaptation strategies [13]–[16].

## IX. Conclusion

From the existing literature this paper derives its own interpretation of fuzzy clustering inductive learning, intended as a synonym of soft competitive parameter adaptation in clustering systems. This conceptual equivalence is employed as a unifying framework in the comparison of clustering algorithms. Within this framework, a set of basis functional features is selected for use as dictionary entries in the comparison of clustering models. This set of features relates to the following issues.

- Fuzzy set-theoretic concepts, such as absolute and relative membership functions (see Section II).
- On-line, batch and mini-batch learning modes, which apply to global (network-based) as well as local (neuron- and connection-based) parameters in distributed learning systems (see Section III).
- Clustering-by-selection and by-replacement editing schemes (see Section IV).
- Prior structures, modular architectures, growing and pruning distributed systems (see Section V).
- Topologically correct mapping, this concept being related to that of modular architectures consisting of specialized and disjointed (mutually independent) maps, which belong to an output lattice of processing units (see Section VI).
- Ecological nets (see Section VII).
- Our own interpretation of a fuzzy clustering algorithm, intended as a clustering system exploiting soft competitive (i.e., cooperative and competitive) versus hard (crisp, purely) competitive adaptation of system parameters (see Section VIII).

## References

[1] This issue, pp. 786.
[2] A. Baraldi and F. Parmiggiani, "Fuzzy clustering: Critical analysis of the contextual mechanisms employed by three neural network models," in *SPIE Proc. Applications Fuzzy Logic Technology III*, B. Bosacchi and J. Bezdek, Eds., 1996, vol. 2761, pp. 261–270.
[3] Anonymous referee, *IEEE Trans. Neural Networks*, 1997.
[4] S. Mitra and S. K. Pal, "Self-organizing neural network as a fuzzy classifier," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 385–399, Mar. 1994.
[5] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464–1480, Sept. 1990.
[6] ——, *Self-Organizing Maps*, 2nd ed. Berlin, Germany: Springer-Verlag, 1997.
[7] J. C. Bezdek and N. R. Pal, "Two soft relative of learning vector quantization," *Neural Networks*, vol. 8, no. 5, pp. 729–743, 1995.
[8] E. C. Tsao, J. C. Bezdek, and N. R. Pal, "Fuzzy Kohonen clustering network," *Pattern Recognit.*, vol. 27, no. 5, pp. 757–764, 1994.
[9] P. K. Simpson, "Fuzzy min-max neural network—Part II: Clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 32–45, 1993.
[10] G. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759–771, 1991.
[11] G. Carpenter, S. Grossberg, N. Maukuzon, J. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 698–713, 1992.
[12] A. Baraldi and E. Alpaydın, "Simplified ART: A new class of ART algorithms," Int. Comput. Sci. Inst., Berkeley, CA, Paper TR-98-004.
[13] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds. Cambridge, MA: MIT Press, 1995, pp. 625–632.
[14] B. Fritzke, "Some competitive learning methods," Draft Doc., http://www. neuroinformatik.ruhr-uni-bochum. de/ini/VDM/research/gsn/DemoGNG, 1998.
[15] A. Baraldi and F. Parmiggiani, "A fuzzy neural network model capable of generating/removing neurons and synaptic links dynamically," in *Proc. WILF'97—II Italian Workshop Fuzzy Logic*, P. Blonda, M. Castellano, and A. Petrosino, Eds. Singapore: World Scientific, 1998, pp. 247–259.
[16] ——, "Novel neural network model combining radial basis function, competitive Hebbian learning rule, and fuzzy simplified adaptive resonance theory," in *Proc. SPIE Optical Science, Engineering, Instru-*

*mentation'97: Applications Fuzzy Logic Technology IV*, San Diego, CA, 1997, vol. 3165, pp. 98–112.
[17] ———, "Fuzzy combination of the Kohonen and ART neural network models to detect statistical regularities in a random sequence of multi-valued input patterns," in *Proc. Int. Conf. Neural Networks '97*, Houston, TX, June 1997, vol. 1, pp. 281–286.
[18] Y. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley, 1989.
[19] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, 1993.
[20] R. N. Davé and R. Krishnapuram, "Robust clustering method: A unified view," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 2, pp. 270–293, 1997.
[21] F. Ancona, S. Ridella, S., Rovetta, and R. Zunino, "On the importance of sorting in Neural Gas training of vector quantizers," in *Proc. Int. Conf. Neural Networks'97*, Houston, TX, June 1997, vol. 3, pp. 1804–1808.
[22] M. Barni, V. Cappellini, and A. Mecocci, "Comments on a possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 393–396, 1996.
[23] J. R. Williamson, "Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps," *Neural Networks*, vol. 9, no. 5, pp. 881–897, 1996.
[24] J. R. Williamson, "A constructive, incremental-learning network for mixture modeling and classification," *Neural Computat.*, vol. 9, pp. 1517–1543, 1997.
[25] J. C. Bezdek and N. R. Pal, "Generalized clustering networks and Kohonen's self-organizing scheme," *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 549–557, 1993.
[26] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, "Neural-gas network for vector quantization and its application to time-series prediction," *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 558–569, 1993.
[27] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. Ser. B*, vol. 39, pp. 1–38, 1977.
[28] E. Alpaydın, "Soft vector quantization and the EM algorithm," *Neural Networks*, vol. 11, no. 3, pp. 467–477, 1998.
[29] J. Buhmann, "Learning and data clustering," in *Handbook of Brain Theory and Neural Networks*, M. Arbib, Ed. Cambridge: MA: Bradford Books/MIT Press, 1995.
[30] V. Cherkassky and F. Mulier, *Learning From Data: Concepts, Theory, and Methods*. New York: Wiley, 1998.
[31] M. J. Jordan and C. M. Bishop, *An Introduction to Graphical Models and Machine Learning*, Draft Doc., 1998.
[32] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon, 1995.
[33] R. Serra and G. Zanarini, *Complex Systems and Cognitive Processes*. Berlin, Germany: Springer-Verlag, 1990.
[34] T. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
[35] M. Bianchini and M. Gori, "Optimal learning in artificial neural networks: A review of theoretical results," *Neurocomput.*, vol. 13, no. 5, pp. 313–346, 1996.
[36] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge MA: MIT Press, 1986.
[37] L. Bottou, "Online learning and stochastic approximations," in *Online Learning in Neural Networks*, D. Saad, Ed. Cambridge, U.K.: Cambridge Univ. Press, 1998.
[38] J. C. Bezdek, T. Reichherzer, G. S. Lim, and Y. Attikiouzel, "Multiple-prototype classifier design," *IEEE Trans. Syst., Man, Cybern. C*, vol. 28, pp. 67–79, Feb. 1998.
[39] C. Burges, "A tutorial on Support Vector Machines for pattern recognition," submitted for publication.
[40] A. Baraldi and L. Schenato, "Soft-to-hard model transition in clustering: A review," Int. Comput. Sci. Inst., Berkeley, CA, Paper ,TR-99-010.
[41] T. Hofmann and J. M. Buhmann, "Competitive learning algorithms for robust vector quantization," *IEEE Trans. Signal Processing*, vol. 46, pp. 1665–1675, June 1998.
[42] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer-Verlag, 1995.
[43] R. Llinas and P. Churchland, *The Mind–Brain Continuum*. Cambridge, MA: MIT Press, 1996.
[44] B. Happel and J. Murre, "Design and evolution of modular neural network architecture," *Neural Networks*, vol. 7, nos. 6/7, pp. 985–1004, 1994.
[45] M. J. Jordan and R. A. Jacobs, "Hierarchical mixture of experts and the EM algorithm," *Neural Computat.*, vol. 6, pp. 181–214, 1994.
[46] P. Blonda, V. la Forgia, G. Pasquariello, and G. Satalino, "Feature extraction and pattern classification of remote sensing data by a modular neural system," *Opt. Eng.*, vol. 35, no. 2, pp. 536–542, 1998.
[47] L. Bruzzone and D. F. Prieto, "A technique for the selection of kernal-function parameters in RBF neural networks for classification of remote-sensing images," *IEEE Trans. Geosci. Remote Sensing*, vol. 37, no. 2, pp. 1179–1184, 1999.
[48] M. Bianchini, P. Frasconi, and M. Gori, "Learning without local minima in radial basis function networks," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 749–756, 1995.
[49] M. L. Minsky and S. A. Papert, *Perceptrons—Expanded Edition*. Cambridge, MA: MIT Press, 1988.
[50] S. Geman, E. Bienenstock, and R. Dourstat, "Neural networks and the bias/variance dilemma," *Neural Computat.*, vol. 4, no. 1, pp. 1–58, 1992.
[51] T. M. Martinetz and K. J. Schulten, "Topology representing networks," *Neural Networks*, vol. 7, no. 3, pp. 507–522, 1994.
[52] D. Parisi, "La scienza cognitiva tra intelligenza artificiale e vita artificiale," in *Neuroscienze e Scienze dell'Artificiale: dal Neurone all'Intelligenza*, E. Biondi *et al.*, Eds. Bologna, Italy: Patron, 1991, pp. 321–341.
[53] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proc. Nat. Acad. Sciences*, 1982, vol. 74, pp. 2554–2558.
[54] T. Masters, *Signal and Image Processing with Neural Networks: A C++ Sourcebook*. New York: Wiley, 1994.
[55] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vis., Graph., Image Process.*, vol. 37, pp. 54–115, 1987.
[56] ———, "ART2: Self-organization of stable category recognition codes for analog input patterns," *Appl. Opt.*, vol. 26, no. 23, pp. 4919–4930, 1987.

**Andrea Baraldi** was born in Modena, Italy, in 1963. He graduated in electronic engineering from the University of Bologna, Italy, in 1989. His master thesis was on the development of satellite image classification algorithms.

From 1989 to 1990, he was a Research Associate at CIOC-CNR, Institute of the National Research Council (CNR), Bologna, and served in the Italian Army at the Istituto Geografico Militare, Florence, Italy, working on satellite image classifiers and GIS. As a Consultant at ESA-ESRIN, Frascati, Italy, he worked on object-oriented applications for GIS from 1991 to 1993. From 1997 to 1999, he joined the International Computer Science Institute, Berkeley, CA, with a postdoctoral fellowship in artificial intelligence. He is currently a Research Associate with ISAO-CNR, Bologna. His main interests deal with low-level vision processing, with special regard to texture analysis and neural network applications.

**Palma Blonda** (M'93) received the degree in physics from the University of Bari, Bari, Italy, in 1980.

Since 1980, her research activity has been in the area of image processing with applications to remote-sensed data and medical imaging. In 1984, she joined the Insitute For Signal and Image Processing, Italian National Research Council, Bari. Her research interests concern pattern recognition, image processing, fuzzy logic and neural networks.