Semantic Driven Fuzzy Clustering for Human-Centric Information Processing Applications

Paulo Fazendeiro[1,2] and José Valente de Oliveira[2]

[1] University of Beira Interior, Department of Informatics, Av. Marques d'Ávila e Bolama, 6200-001 Covilha, Portugal
fazendeiro@ubi.pt

[2] University of Algarve, UAlg Informatics Lab, Campus de Gambelas, 8000-062 Faro, Portugal
jvo@ualg.pt

Abstract. This chapter presents an overview of fuzzy clustering techniques aiming at human-centric information processing applications and introduces the accuracy-interpretability tradeoff into the conceptualization of the clustering process. Nowadays it is a matter of common agreement that the cornerstone notion of information granulation is fundamental for a successful outcome of exploratory data analysis and modeling in fields like science, engineering, economics, medicine and many others. There is no doubt that fuzzy clustering is an excellent medium to obtain such information granules. For a matter of self-containment the chapter starts by presenting the fundamentals of fuzzy clustering along with some variants and extensions. In the second part of the chapter, the fuzzy clustering approach is highlighted as a valuable human-centric interface: the roadmap from data to information granules is displayed along with a discussion on some mechanisms to implement user relevance feedback. In the last part of the chapter a semantic driven evolutionary fuzzy clustering algorithm is analyzed, as a particular instance of a class of unsupervised clustering algorithms which embraces constraints usually applied in supervised learning. The results show that these more general constraints while tuning the equilibrium between accuracy and interpretability concomitantly help to unveil the structural information of the data.

1 Introduction

From the strictly conceptual point of view the human-centric development paradigm and the human-centric information processing pursuits share the final goal of making the synthesized system highly adaptable to the needs of the human user and capable of presenting a natural interface which fosters the user-system interaction. In this sense fuzzy clustering techniques undoubtedly have an important place in the human-centric armamentarium. The contribution of fuzzy clustering techniques to the human-centric paradigm effort can be roughly enumerated along three different application vectors. First of all, from the end-user perspective, the intuitive nature of the fuzzy sets offers a privileged mean of communication of the exploratory data analysis findings in a user-friendly way.

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 119–151.
springerlink.com

Sections 3.1 and 3.2 present a detailed discussion of this aspect. The second facet, perhaps more interesting from the system designer's point of view but also very important to the user interaction and feedback to the system, has to do with the seamless integration of human knowledge as a support or guiding mechanism of the clustering activity. In Sect. 3.3 we pinpoint some interesting trends in this regard. Finally a third aspect, less obvious and more related to the behavior of the system being developed, concerns the incorporation of human-defined semantic interpretability constraints into the clustering process and taking advantage of these as a mean to accomplish a transparent system with good accuracy. In Sect. 4 we present such constraints along with their generalizations to the clustering framework and report some results which point to the merit of considering them from inception.

The chapter is organized into three main parts. Section 2 presents an overview of fuzzy clustering techniques emphasizing the competitive advantage over crisp clustering, pointing problems, shortcomings and possible solutions. Section 3 discusses the development of information granules and mechanisms of user customization. Whereas these two parts highlight the role of the fuzzy clustering algorithmic framework as support to the human-centric paradigm Sect. 4 sustains that designing the clustering algorithm in a human-centric way can be a valuable asset. Following this perspective, simple human-centric semantic constructs that are commonly used in supervised learning as a way to balance interpretability-accuracy equilibrium are successfully transposed to unsupervised fuzzy clustering, resulting in clustering techniques which provide interesting results from the end-user point of view.

2 Overview of Fuzzy Clustering

Generally speaking clustering is the process of searching for a finite and discrete set of data structures (categories or clusters) within a finite, otherwise unlabeled, usually multi-variate data set. In the literature it is common to find that the goal of clustering is the partition of the data set into groups so that data in one group are similar to each other and are as different as possible from data in other groups, cf. [42, 75]. Two complementary facets are enclosed in this unsupervised learning task: the elicitation of a model of the overall structure of the data and the pursuit for a manageable representation of a collection of objects into homogeneous groups.

The notion of similarity between elements (patterns) of the data set is a concept of paramount importance, with direct implications on the clustering endeavor. Usually the similarity is defined at the expense of some appropriate distance function. In cluster analysis common choices for distance functions include the Hamming (city block) distance inducing diamond shaped clusters, the Euclidean distance inducing (hyper) spherical clusters and the Tchebyshev distance inducing (hyper) box shaped clusters. As a matter of fact these examples are members of the Minkowski family of distances, or L_p norms, defined as:

$$D(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} . \qquad (1)$$

Distance can be used to measure the similarity between two data points or between a data point and a prototype of the cluster. The prototype is a mathematical object, usually a point in the feature space (e.g. the center of the cluster) or even a geometric subspace or function, acting as a representative of the cluster while trying to capture the structure (distribution) of its associated data.

Traditionally the clustering algorithms are categorized in two main types: hierarchical and objective function based partitional clustering. Every new cluster determined by a hierarchical algorithm is based on the set of previously established clusters. The distance between individual points has to be generalized to the distance between subsets (linkage metric) in order to merge (or split) clusters instead of individual points. The type of the used linkage metric significantly affects hierarchical algorithms, since each cluster may contain many data points and present different geometrical shapes, sizes and densities. The distance is computed for every pair of points with one point in the first set and another point in the second set. Due to the pairwise combinatorial nature of the process the hierarchical approach tends to be computationally inefficient with the growth of the number of data elements. This approach is very sensitive to anomalous data points (noise and outliers) and is unable to handle overlapping clusters. A reason for this is that bad decisions made at an early stage of the algorithm will be propagated and amplified up to the end since the intermediate clusters are not revisited for further improvement (the points can not move from one cluster to another).

The second major category of clustering algorithms attempts to directly decompose the data set into a collection of disjoint clusters. This partition is builded during an iterative optimization process repeated until its associated cost function reaches a minimum (global or local). The cost function, also designed performance index or objective function, is a mathematical criterion expressing some desired features (emphasizing local or global structure of the data) of the resulting partition.

Combining some heuristics with an adequate formulation of the objective function it is possible to design an optimization process which is able to determine at least suboptimal partitions. One such formulation, for that matter the most used in practice, is the sum-of-squared-error distances or minimum variance criterion representing each of C clusters by their mean (the so-called centroid $\mathbf{v}_i \in \mathbb{R}^n$, $i = 1, \ldots, C$ of its points):

$$Q = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij} D_{ji}^2(\mathbf{x}_j, \mathbf{v}_i) \qquad (2)$$

where $\mathbf{X}= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ denotes a set of feature vectors (or patterns) in the $\mathbb{R}^n$ space. $D_{ji}(\mathbf{x}_j, \mathbf{v}_i)$ is a measure of the distance from $\mathbf{x}_j$ to the ith cluster prototype. The elements, $u_{ij} \in \{0, 1\}$, $i = 1, \ldots, C$ $j = 1, \ldots, N$ form a matrix designated as the *partition matrix* which maps the patterns to a cluster. If

$u_{ij} = 1$ the pattern j belongs to cluster i, otherwise if $u_{ij} = 0$ the pattern j is not accounted as a member of cluster i. This formulation is appealing because it still favors sets of well separated clusters with small intra-cluster distances whereas replaces all the pair-wise distances computation by a single cluster representative. Thus the computation of the objective function becomes linear in N and it is now feasible the application of an iterative optimization process aiming at gradual improvements of the builded clusters.

The c-Means algorithm (also referred in the literature as k-Means or hard c-Means) [26, 54] is the best known squared error-based example of such a process. For a given initialization of the C centroids the heuristic approach consists of two-step major iterations that follow from the first-order optimality conditions of (2): first reassign all the points to their nearest cluster, thus updating the partition matrix, and then recompute the centroids, $\mathbf{v}_j$ (its coordinates are the arithmetic mean, separately for each dimension, over all the points in the cluster), of the newly assembled groups. This iterative procedure continues until a stopping criterion is achieved (usually until no reassignments happen). In spite of its simplicity and speed this algorithm has some major drawbacks. It is much dependent on the initial centroids assignment (frequently in practice it is run for a number of times with different random assignments and the best resulting partition is taken), does not ensure that the result has a global minimum of variance, is very sensitive to outliers and lacks scalability.

Another not so obvious disadvantage is related to the binary nature of the elements of the partition matrix and consequently of the induced partitions. This kind of partition matrix is based on classical set theory, requiring that an object either does or does not belong to a cluster. The partitioning of the data into a specified number of mutually exclusive subsets is usually referred as *hard clustering*. In many situations this is not an adequate representation.

Consider for instance a borderline point located in the boundary between two clusters or otherwise an outlier datum located at nearly the same distance from the centers of two clusters. In these frequent situations the point is almost as typical of one cluster as it is of the other, thus a more natural partition would be one which allowed the objects to belong to several clusters simultaneously (with different degrees of membership.) This is precisely the central concept behind *fuzzy clustering* methods with foundations in the fuzzy sets theory [77]. A *fuzzy set* is characterized by a *membership function* that maps each point of the universe $\mathbf{X}$ to a number in the interval $[0, 1]$ (1 represents full degree of inclusion and 0 non-membership at all).

We can say that the relaxation of the constraint imposed on the partition matrix to $u_{ij} \in [0, 1]$ is more realistic and able to provide a richer insight of the data structure, especially when in presence of ambiguous data or clusters without sharp boundaries. Indeed, the fuzzy logic approach to clustering differs from the conventional set theory approach mainly because a generic datum may belong to more than one cluster with a different degree of membership (usually a value between 0, non-membership, and 1, full degree of inclusion). Hence the data points near the core of a given cluster exhibit a higher degree of membership than those lying farther away (near its border). Within this framework it is possible to capture the uncertainty, vagueness and flexibility inherent to the data set and to the concepts being formed. In the remaining of this section we review some different algorithmic approaches that allows the construction of fuzzy partitions, i.e., algorithms which represent a cluster as a fuzzy set.

2.1 The Fuzzy C-Means Clustering Algorithm

Fuzzy clustering was introduced as early as 1969 by Ruspini [68]. Fuzzy C-Means (FCM) is a simple and widely used clustering algorithm. The algorithm results from an optimization problem that consists in the minimization, with respect to **V**, the set of prototypes, and **U**, the fuzzy membership matrix, of the following index (objective function) [6]:

$$Q_{FCM} = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i) \tag{3}$$

where $m > 1$ is the so-called fuzziness parameter ($m = 2$ is a common choice) that controls the influence of membership grades or in other words how much clusters may overlap, cf. [44], and D stands for a norm distance in $\mathbb{R}^n$, under the following conditions on the partition matrix elements:

$$u_{ij} \in [0,1] \text{ for all } i = 1, \ldots, C \text{ and } j = 1, \ldots, N \tag{4}$$

$$\sum_{i=1}^{C} u_{ij} = 1 \text{ for all } j = 1, \ldots, N \tag{5}$$

$$\sum_{j=1}^{N} u_{ij} > 0 \text{ for all } i = 1, \ldots, C \tag{6}$$

Condition (5) induces a fuzzy partition in the strict sense and assures that every datum has a similar global weight on the data set. Constraint (6) guarantees that none of the C clusters is empty, thus implying a cluster partition with no less than C clusters. Notice the similarity between (2) and (3). As a matter of fact they are coincident apart from a fixed transformation (the introduction of the fuzzifier, m) introduced as a mean to prevent that under condition (5) the same minimum as the one obtained by the crisp standard formulation was reproduced.

For this constrained nonlinear optimization problem there is no obvious analytical solution. Therefore the most popular and effective method to minimize the constrained objective function consists in resorting to a technique known as alternating optimization. This means that one set of parameters is kept fixed while the other is being optimized, and next they exchange roles. The prototype **V** and membership **U** update equations are obtained from the necessary conditions of a minimum:

$$\frac{\partial Q_{FCM}}{\partial \mathbf{V}} = 0 \ \ \text{(assuming } \mathbf{U} \text{ to be constant);} \tag{7}$$

$$\frac{\partial Q_{FCM}}{\partial \mathbf{U}} = 0 \ \ \text{(assuming } \mathbf{V} \text{ to be constant).} \tag{8}$$

Additionally the consideration of (5) in the original objective function (3) by means of Lagrange multipliers converts the constrained problem into its constrained-free version. Some straightforward computations lead to the update formula of the partition matrix:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{D_{ji}(\mathbf{x}_j,\mathbf{v}_i)}{D_{jk}(\mathbf{x}_j,\mathbf{v}_k)}\right)^{\frac{2}{(m-1)}}}. \tag{9}$$

This formula does not depend on the chosen distance function, however the determination of the prototypes is more complicated since many distance norms do not lead to a closed-type expression. A common practical choice is to use the Euclidean distance or L_2 norm (for a generalization to L_p, $p > 0$, the interested reader is referred to [38]) leading to the following prototype update equation:

$$\mathrm{v}_i = \frac{\sum_{j=1}^{N} u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^{N} u_{ij}^m}. \tag{10}$$

The alternate optimization of **U** and **V** proceed iteratively until no significant change of the objective function is registered. It has been proven that the generated sequence of solutions, for fixed $m > 1$ always converge to local minima or saddle points of (3) [9]. Informally, what the resulting algorithm will do is to search for the clusters that minimize the sum of the intra-cluster distances. In general the performance of fuzzy algorithms, when compared with the corresponding hard partitioning ones, is superior and they are less prone to be trapped in local minima [6]. However, like its hard counterpart the FCM algorithm shares the problem of high sensitivity to noise and outliers, something that is common to the generality of the least-squares approaches and that can drastically distort the optimal solution or facilitate the creation of additional local minima. Next we discuss an alternative formulation, specifically designed to tackle this problem.

2.2 The Possibilistic C-Means Clustering Algorithm

The influence of noise points can be reduced if the memberships associated with them are small in all clusters. However, as can be seen from the probabilistic-like constraint (5), the memberships generated by the FCM are relative numbers expressing the concept of sharing of each pattern between clusters rather than the concept of typicality of a given pattern to a given cluster. This means that noise points and outliers will also have significantly high membership values. A more general form of fuzzy partition, the *possibilistic* partition, can be obtained by relaxing the constraint (5) in order to address this problem.

In this case the assignment of low membership in each cluster to noise points depends on giving up of the normalization condition (5), leading to possibilistic instead of fuzzy partitions. To avoid the trivial solution (i.e. a matrix with null elements) Krishnapuram and Keller [47] added to (3) a punishment term for low memberships resulting in the augmented possibilistic c-means (PCM) objective function:

$$Q_{PCM} = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i) + \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1 - u_{ij})^{m} \tag{11}$$

where the distance parameters $\eta_i > 0$ $(i = 1, \ldots, C)$ are specified by the user. Notice that the second term expresses the desire to have strong assignments of data to clusters. Due to the nature of the membership constraint, we call *possibilistic clustering algorithm* (PCM) a fuzzy clustering algorithm which minimizes (11) under the constraint (6). The partition matrix update equations, as before for the FCM case, are obtained by setting the derivative of the objective function equal to zero while holding the prototype parameters fixed:

$$u_{ij} = \frac{1}{1 + \left(\frac{D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i)}{\eta_i} \right)^{\frac{1}{(m-1)}}}. \tag{12}$$

This update expression clearly emphasizes the typicality interpretation of the membership function. Unlike the FCM formulation, the degree of membership of one point to a cluster depends exclusively of its distance to the center of that cluster. For the same cluster, closer points obtain higher membership than the ones farther away from it. Moreover (12) shows that η_i determines the distance of the "definite" assignment ($u_{ij} > 0.5$) of a point to a cluster (simply considering $m = 2$ and substituting η_i by $D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i)$ results in $u_{ij} = 0.5$). So it is useful to choose each η_i separately, according to the individual geometrical features of each cluster. Unfortunately these are not always available so Krishnapuram and Keller recommend several methods to determine η_i [47,48]. Using the fuzzy intra cluster distance a sound probabilistic estimation of these weight factors can be obtained:

$$\eta_i = \frac{\sum_{j=1}^{N} u_{ij}^{m} D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i)}{\sum_{j=1}^{N} u_{ij}^{m}}. \tag{13}$$

The update formula for the prototypes is the same as the one used in the FCM method since the second term in (11) simply vanishes when computing the derivative of the objective function with respect to the prototype parameters. If we take a closer look at (12) we see that the membership degree of a pattern to a cluster depends only on the distance of the pattern to that cluster, but not on its distance to other clusters. So it happens that in some situations this algorithm can originate coincident clusters (converging to the same local optimal point), thus disregarding clusters with lower density or less points, or even presents stability problems due to sensitivity to initialization [48]. Thus to overcome these drawbacks of the possibilistic approach it is common practice to initialize PCM with a prior run of the probabilistic FCM.

2.3 Other Approaches to Fuzzy Clustering

The literature on fuzzy clustering is remarkably rich, cf. [72], and in a broad sense it reflects the attempts made to surpass the problems and limitations of the FCM and PCM algorithms. In the two former sections we reviewed FCM and PCM and their prototypes's update equations assuming the Euclidean distance as the standard metric. However when combined with a squared error-based objective function this distance induces hyper-spherical clusters. To overcome this geometrical constraint imposed by clustering algorithms based on a fixed distance metric several algorithms using adaptive distance measures have been proposed. Two of the most well known are the Gustafson-Kessel algorithm [32] which replaces the Euclidean distance by the Mahalanobis distance (an interesting generalization of the Euclidean distance) with a specific covariance matrix for each cluster and the unsupervised Gath-Geva algorithm [30] where the distance is based on the fuzzification of the maximum likelihood estimation method. Both of these algorithms are well fitted to find ellipsoidal clusters with varying size and orientation (there are also axis-parallel variants of these algorithms and to some extent they can also be used to detect lines).

In the field of image processing and recognition the geometry of the fuzzy clusters is a key aspect for image analysis tasks. Both FCM and PCM use point prototypes. If we are interested in finding particular cluster shapes, algorithms based on hyper-planar or functional prototypes, or prototypes defined by functions, are a good choice. The distance is no longer defined between two patterns (i.e. a datum and a prototype), instead it is measured between a pattern and a more complex geometric construct. This class of algorithms includes the fuzzy c-varieties [7] for the detection of linear manifolds (lines, planes or hyper-planes), fuzzy c-elliptotypes [8] for objects located in the interior of ellipses, fuzzy shell clustering for the recognition of object boundaries (e.g. fuzzy c-shells [16] in the detection of circles, hyper-quadric shells [45], fuzzy c-rectangular shells [40]) and fuzzy regression models [36]. The interested reader may follow a comprehensive explanation of these branch of methods in [41].

In addition to PCM other methods have been proposed in order to improve the robustness of the FCM algorithm to noisy data points and outliers while maintaining the constraint (5) (thus circumventing the problem of cluster coincidence of the PCM approach). For instance the technique presented in [59] and [14] consists in the introduction of an additional *noise cluster* aiming at grouping the points with low probability of belonging to the remaining clusters. This probability depends on the mean value of the squared distances between patterns and the prototypes of the normal clusters. Latter on, this technique was extended in order to accommodate different noise probabilities per cluster [15].

The great majority of the algorithms presented hitherto result from alternating the optimization of the membership functions and prototype locations in an iterative process. Therefore the clustering model constrains (and is constrained to) the particular shapes of the membership functions and the positions of the prototypes to those determined by the updating equations derived from the objective function. However, the user might be interested in the use of a certain type of membership function with more adequate shapes to the problem in question or in certain cluster prototypes satisfying some application-specific needs. The *alternating cluster estimation* (ACE) framework [65] is able to provide, when required, this extra flexibility. In applications such as extraction of fuzzy rules from data, where each fuzzy set should have a clear semantic meaning (for instance associated to linguistic labels like "high" temperature or "about 80" degrees), a convex fuzzy set with limited support may be more preferable than the non-convex membership functions generated by FCM or PCM. Notwithstanding that ACE embodies FCM and PCM as particular instances of the framework, the requirement that the updating equations for the membership function and the prototypes should result from the necessary conditions for local extrema is now rejected and the user is free to choose the pair of updating equations which is better fitted for the problem at hand. At first sight this generalization may seem to be lacking mathematical soundness however it has proven its usefulness in practical examples.

In many practical applications the data sets can be heavily contaminated by noise points which promote the proliferation of local minima. In these cases, the probability of the alternate optimization getting stuck at local optimal values is far from being negligible. To obviate this problem, stochastic algorithms have been used in cluster analysis, many of them inspired on biological paradigms such as the natural evolution of species or swarm-based behavior. Examples of such approaches to fuzzy clustering include the use of genetic algorithms [18,19,35,43,51,73], evolutionary programming [69], evolutionary strategies [76], ant colony optimization [66] and particle swarm optimization [67]. Notwithstanding that these attempts do not guarantee optimal solutions, demand the definition of a set of problem-specific parameters (e.g. population size) and are very computationally time-consuming they can undoubtedly contribute to avoid local extrema and reduce the sensitivity to initialization.

2.4 Determination of the Number of Fuzzy Partitions

In the great generality of the partitional algorithms the number of clusters C is the parameter having greater influence on the resulting partition. The chosen clustering algorithm searches for C clusters, regardless of whether they are really present in the data or not. So when there is no prior knowledge about the structure of the data a natural question arises: what is the right number of clusters for a particular data set? This question is known in the literature as the cluster validity problem and distinct validity measures have been proposed in order to find an answer, cf. [27, 34, 42, 55, 58, 75]. However, in spite of a greater practical adhesion to some of them, due to the subjective and application-dependent character of the problem there is no consensus on their capability to provide a definitive answer to the foregoing question. For partitional fuzzy clustering it is advisable that the validity indices account both for the data set (e.g. their variance) and the resulting membership degrees. An example of such class of validity indices, exhibiting good behavior when matched against a set of other indices [60], is the Xie-Beni index [74], also known as the compactness and separation index, computed as the ratio of the compactness of the fuzzy partition of a data set to its separation:

$$XB = \frac{\sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i)}{N \min_{i \neq j} D_{ij}^{2}(\mathbf{v}_i, \mathbf{v}_j)}. \tag{14}$$

The interested reader is referred to [33] for further examples and properties of hard/fuzzy validation indices. The effectiveness of a particular choice of C is verified *a posteriori* by *cluster validity analysis*, performed by running the clustering algorithm for different values of C, several times with different initializations. However, since different validity measures may produce conflicting results (even runs with different initializations may introduce some distortion for the same measure) it is advisable that they should be used only as guidelines to find a plausible range for the correct number of clusters.

The cluster validity problem was also tackled by unsupervised techniques with no *a priori* assumption on the number of clusters. Many of these approaches (e.g. [28,29,53]) take advantage of the fact that (3) is minimized when the number of clusters is equal to the cardinality of the data set (when prototypes and data coincide) by adding to the cost function (3) a regularization term which is minimized when all the patterns are assigned to one cluster. These algorithms start with a large number of clusters which is progressively reduced until convergence. Regretfully, in practice the problem of cluster validity is replaced by the determination in advance of another user supplied parameter with major influence in the clustering outcome and dictating which clusters are discarded.

An interesting blending between fuzzy partitional clustering techniques and hierarchical algorithms was presented in [31]. The objective is to exploit the advantages of hierarchical clustering while overcoming its disadvantages in dealing with overlap between clusters. At every new recursive agglomerative step the proposed algorithm adaptively determines the number of clusters in each bifurcation by means of a weighted version of the unsupervised optimal fuzzy clustering algorithm [30]. The final outcome of the clustering is the fuzzy partition with the best validity index value. Needless to say, the algorithm presents sensitivity to the adopted validity index.

Unsupervised stochastic techniques have also been applied to cluster validity analysis. In [56] a genetic fuzzy clustering algorithm is used for the classification of satellite images into different land cover regions. The objective function is replaced directly by a validity index (in this case the Xie-Beni index) and a variable chromosome length (depending on the number of clusters represented by each individual) allows the simultaneous evolution of solutions with a different number of clusters. The outcome is the best (in the Xie-Beni sense) of the evaluated fuzzy partitions.

3 The Role of Fuzzy Clustering in the Human-Centric Paradigm

The concept of *linguistic variable* [79, 80] plays a pivotal role in the formation of fuzzy information granules. Informally, a linguistic variable is a granulated variable whose granular values are words or phrases represented by fuzzy sets (altogether with their connectives, modifiers and negation). These linguistic characterizations are, usually, less specific than the numeric ones, but in compensation are safer. Thus the linguistic variable can be viewed as a way to accomplish (lossy) compression of information. Moreover the linguistic variable provides a descriptive mean for complex or poorly understood systems and, more important, offers a bridge between linguistics and computation, cf. [81]. As Zadeh [83] pointed out, *the fuzziness of granules, their attributes and their values is a central characteristic of the ways in which human concepts are formed, organized and manipulated.* This observation supports what seems to be one of the most human-centric approaches to discover structure in data: fuzzy clustering.

As previously referred, the contribution of fuzzy clustering techniques to the human-centric paradigm effort can be described across three main lines: (i) user-friendly communication of the results, (ii) seamless integration of human knowledge and (iii) incorporation of human-defined semantic interpretability constraints in order to accomplish a transparent system with good accuracy. The purpose of this section is to present a detailed discussion on the two first aspects. The incorporation of human-defined semantic constraints into the clustering endeavor is addressed in Sect. 4.

3.1 Information Granulation

Information granules are simultaneously a mean and an objective. Due to the limited capability of human mind and sensory organs to deal with complex information its decomposition into manageable chunks of information is essential. The aggregation of similar or nearby objects into information granules (class abstraction) as well as the encapsulation of functional commonalities are fundamental skills for a successful approach to the great majority of problems that we face everyday. This granulation may be crisp or fuzzy.

Crisp granules are derived with the apparatus of the classical set theory and are common components in various methods of information analysis, e.g. decision trees, interval analysis or rough set theory. Fuzzy granules found their inspiration in the human capability to reason in an uncertain and imprecise environment and are supported by the theory of fuzzy information granulation (TFIG) [82], a part of the fuzzy sets and fuzzy logic framework. Furthermore the fuzzy logic approach relies on the notion of (fuzzy) set, opposite to the member of a classical set, to represent uncertain and imprecise knowledge. This last facet is the point of departure to the model identification with different levels of descriptive precision and granularity, viz. (fuzzy) granulation, cf. [78, 81]. In this setting, typically, an information granule is a fuzzy set and the process of information granulation

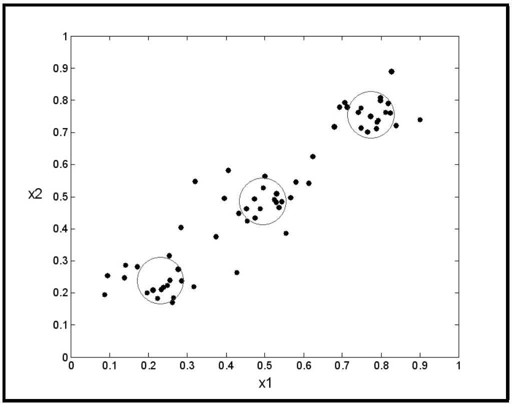

Fig. 1. Simple data set in $\mathbb{R}^2$ and clustering results of the FCM algorithm. Dots represent data points and unfilled circles represent the clusters' centers.

consists in describing a crisp or fuzzy object as a collection of fuzzy granules (or eventually as relationships between them).

In Sect. 2 we reviewed the standard fuzzy c-means algorithm (FCM), its assets and common alternatives to overcome its shortcomings. Next with the help of a visually appealing example the path leading from raw data to information granules is briefly explained. To facilitate the visualization we consider a synthetic data set defined in $\mathbb{R}^2$ as depicted in Fig. 1.

It is composed of three visually separable clusters resulting from a normal distribution of twenty elements around 3 distinct points. Suppose that the clusters' centers, marked as unfilled circles in Fig. 1, were found by an adequate fuzzy clustering method (in this case FCM). The purpose here is to describe those clusters invoking simple fuzzy granules. Let's assume that the clustering algorithm has produced the partition matrix where each data point is characterized by a set of membership values, one per each cluster: the closer the point is to the cluster's center, the higher the membership value of that point. This relation can be perceived in Fig. 2 where only the maximum value of membership for each data point is shown (in the Z axis).

Each one of the resulting clusters may be conceived as a multidimensional granule, however to be clearly understandable and subject to human communication it has to be expressed in terms of simpler qualitative attributes defined for each feature.

To accomplish this, first the dimensionality of this fuzzy relation is reduced by a simple operation of projection to the corresponding coordinate spaces. For every 2-dimensional granule G defined on $\mathbf{X_1} \times \mathbf{X_2}$ there are two projections $G_{proj\mathbf{X_1}}$ and $G_{proj\mathbf{X_2}}$ with the following membership functions (for discrete sets sup is replaced by max):

$$G_{proj\mathbf{X_1}}(a) = \sup_{y \in \mathbf{X_2}} G(a, y), \quad a \in \mathbf{X_1}; \tag{15}$$

$$G_{proj\mathbf{X_2}}(b) = \sup_{x \in \mathbf{X_1}} G(x, b), \quad b \in \mathbf{X_2}. \tag{16}$$

Computing the correspondent projections, each cluster induces a one-dimensional *discrete* fuzzy set per feature. Figure 3 depicts the projection into one of the coordinate spaces (notice that for ease of visualization the individual fuzzy sets are depicted as piecewise linear functions when, in fact, they are composed of discrete elements).

To extend this fuzzy set to the whole one-dimensional domain an adequate enveloping fuzzy set (convex completion) or a suitable parameterized fuzzy set approximation is usually necessary. Obviously this approximation implies some

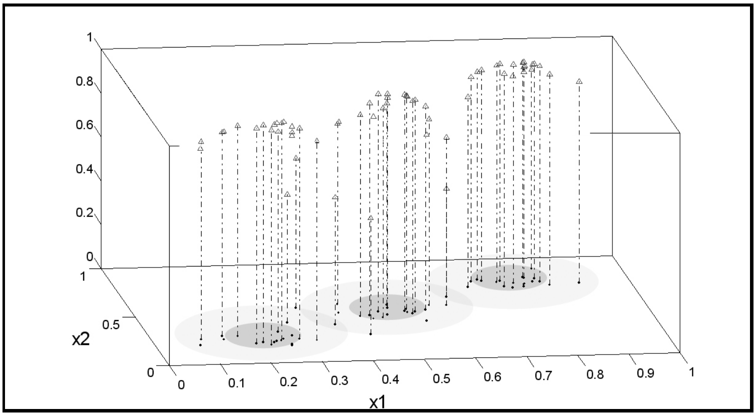

Fig. 2. Clustering results of the FCM algorithm depicting the maximum membership value of each data point

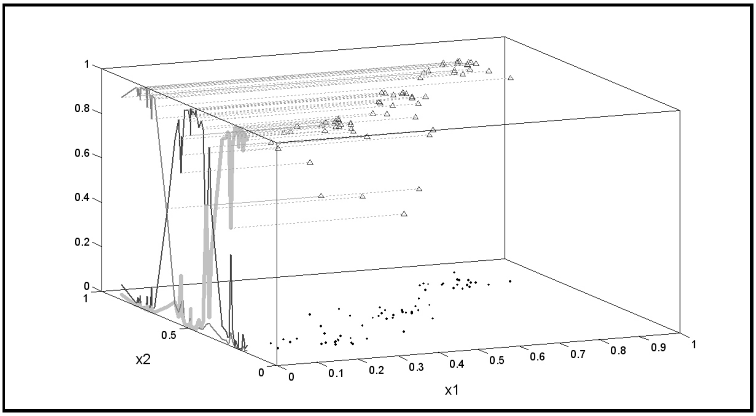

Fig. 3. Dimensionality reduction by projection to the coordinate spaces. The figure depicts the projection to $\mathbf{X_2}$.

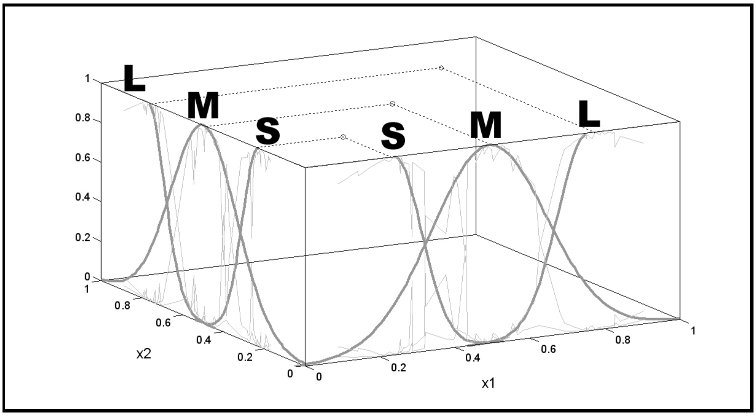

Fig. 4. Synthesis of interpretable Information Granules. A functional approximation was performed followed by a semantic conversion conveying meaning to each one-dimensional fuzzy set.

loss of information. In the given example each one-dimensional fuzzy set was approximated by a Gaussian membership function distributed around the prototype's projection, see Fig. 4, and altogether they form a fuzzy partition across each single domain. Finally a last step must be performed if one wishes to describe each multidimensional granule in an human-friendly way: if possible each one-dimensional fuzzy set must be associated to a linguistic value with a clear semantic meaning (in Fig. 4, S stands for Small, M for Medium and L stands for Large).

The multidimensional granule is thus defined as a combination of one-dimensional fuzzy sets encoding linguistic labels relevant for the problem at hand. For each cluster the one-dimensional fuzzy sets where its prototype's projection attains the maximum value are chosen as the clusters' representatives. Hence each multidimensional cluster may be expressed as cartesian products of simpler granules. Referring back to Fig. 4 the overall data set may be entirely described in this concise form:

$$S_1 \times S_2 + M_1 \times M_2 + L_1 \times L_2 \tag{17}$$

where + represents disjunction and X_1 and X_2 play the role of linguistic variables assuming the values small (S), medium (M) and large (L), necessarily with different concretization in each domain.

3.2 Elicitation of the Information Granules

Fuzzy clusters are information granules represented by fuzzy sets, or more generally by fuzzy relations in some multi-dimensional data space. However, as was emphasized above, in order to take full advantage of their expressive power they should be able to be described as propositions in a natural language.

In opposition to our previous oversimplified example (Figs. 1, 2, 3 and 4) there are many situations posing several difficulties to the adequate elicitation of a *semantic mapping* between data space and feature space. Just to give an example, consider a data set with 4 well separable clusters in $\mathbb{R}^2$ and centers in the vicinity of the vertices of a square with sides parallel to the coordinate axes. In this case the correspondent projections into the one-dimensional spaces, would result in two pairs of very close fuzzy sets per feature and consequently almost undiscernible between them. The approaches to develop semantically sound information granules as a result of the fuzzy clustering process range from purely *prescriptive* methods to purely *descriptive* techniques, cf. [63]. In the prescriptive characterization of the fuzzy sets the meaningful granules are expressed intuitively by an observer in such a way that they capture the essence of the problem. The descriptive design involves the detailed computation of the membership functions based on the available numeric data.

The work presented in [12] is a good example of this latter approach complemented with some semantic concerns. The overall technique can be summarized in three steps. First, a cluster analysis is performed on the data set. The clustering algorithm (e.g. FCM) induces C information granules and this number of clusters has a major effect on the information granularity. In the second step the prototypes are projected into each dimension, being their projections further clustered in order to obtain a pre-specified number of clusters, i.e., one-dimensional granules. The final step consists in quantifying the resulting one-dimensional prototypes as fuzzy sets in the feature space by means of Gaussian membership functions with a desired level of overlap. The second step of this double-clustering technique is not computationally demanding (the number of prototypes is much lesser than the number of data elements) and promotes the fusion of projections, which otherwise would result in undiscernible data sets, into one single representative thus permitting the representation of granules via highly comprehensible fuzzy propositions.

The prescriptive approach can be illustrated by the interesting technique of context clustering [61] (see also [39]). In essence the algorithm results from an extension to the FCM algorithm replacing the standard normalization constraint (5) by a conditioning constraint dictated by the context under which the clustering is being performed. The context is specified by the user and can assume the form of an information granule (linguistic term) defined in a peculiar feature or a logical combination between granules in the same feature space or even a composite context resulting from the Cartesian product of fuzzy sets defined in different feature spaces. Informally, we can say that the (fuzzy) linguistic context acts as a data window focusing the clustering effort on particular subsets of the data or regions of interest, thus enabling a deeper insight on the internal structure of those information granules.

The technique reported in [63] tries to present a balanced tradeoff between the prescriptive and descriptive approaches. The descriptive component is represented by the clustering algorithm (the experiments report to the standard FCM) performed in the multi-dimensional data space. Given two different runs of the clustering algorithm, searching for a different number of clusters, the resulting granules present necessarily a different granularity level. The distinct granularity of the resulting information granules (the mixture of coarser and finer granules) can be turned into an advantage. The prescriptive component task is to conciliate different granular representations by means of specific operations of generalization (information granules combined *or*-wise) and specialization (information granules refined *and*-wise) of the fuzzy relations. The logic operators (s-norms and t-norms) are defined in advance then, if we intend to decrease the granularity of the finer result, the algorithm finds the coarser granule and the respective generalization (selected amongst the possible pairwise generalizations of the finer granular set) with optimal similarity (based on the overall difference between membership values of each datum in the original set and in the given generalization). On the other hand, when we intend to increase the granularity of the collection of information granules a similar process is performed, viz., the optimal replacing of a granule by the pair of granules forming its closest specialization.

The representation of information granules via multidimensional hyper-boxes with sides parallel to the coordinates greatly simplifies their transparent expression as decomposable relations of classical sets in the corresponding feature spaces. In [3] the standard FCM was modified through a gradient-based technique in order to accommodate the Tchebyshev distance. This distance induces a hyper-box shaped geometry of the clusters, however due to the interaction between clusters there exists a deformation of the hyper-boxes which need to be reconstructed in an approximate manner. When compared with FCM with Euclidean distance, the algorithm equipped with Tchebyshev distance exhibited less sensitivity to the size of the data groupings, being able to identify smaller clusters. The algorithm produces a description of the data consisting of hyper-boxes (whose sizes depend on a given threshold) which encompass the core of the data and a residual portion of the data described by the standard FCM membership expression. Another interesting approach to hyper-box granulation combined with fuzzy clustering was presented in [2]. The proposed measure of information density (the ratio between cardinality and specificity of a set) is maximized in a recursive manner departing from the numeric data which is progressively mixed with the produced granular data. As a result of this granulation process the data is compressed while the number of information granules in the high data density areas is reduced. Next the information granules are clustered using the FCM algorithm combined with a parametric method of representation of the hyper-boxes. This results in a collection of cluster prototypes interpretable in the original data space as hyper-boxes altogether with a fuzzy partition matrix representing the membership of data into clusters. Due to the reduction of the number of information granules in high density areas the FCM problem of under-representing smaller groups of data is thus obviated. Moreover, the hyper-box representation of the prototypes has direct transposition as fuzzy decomposable relations in the feature space enabling a transparent interpretation of the information granules.

Independently of the followed approach (descriptive, prescriptive or both) one should be aware that the elicitation of information granules in a human comprehensible way is dependent of the characteristics of the application at hand and on the judicious decisions of the data analyst.

3.3 Enhancements to Accommodate Human Knowledge

Notwithstanding that in a great number of applications there is no labeling information about the data at hand, or otherwise the labeling of the data set is a fastidious task requiring a lot of expert time due to its high cardinality, or even it is error prone and potentially ambiguous due to the nature of the problem being tackled; there are situations where the clustering endeavor can and should be guided, at least in a confined way, by the inclusion of additional information about the data structure and the inherent characteristics of the problem. Automatic text classification of extensive corpora, categorization of Web sources, recognition of handwritten text characters or image segmentation are just some examples of applications where usually the data analyst is confronted with a small subset of labeled data. Better than blindly attack the problem as an unsupervised clustering pursuit the available information should be properly taken into account.

In cases like the ones mentioned we can say that we are faced with a semi-supervised clustering and as was emphasized in [5] the labeled data can be used quite successfully not only to define the number of clusters but also by using the cluster centers as a way to affect the cluster assignment for unlabeled data. A very straightforward formulation that allows us to play with the balance between our confidence in the reliability of the available labeled data and the automated data exploration was presented in [64]. Simply stated the partially supervised clustering algorithm lays on an additive objective function aiming at structure finding, minimizing the fuzzy within cluster variance as in the standard FCM, and accounting for the data already categorized by minimizing the misclassification error:

$$Q_{semi} = Q_{FCM} + \alpha \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij}^{l*} - u_{ij}^{l})^m D_{ji}^2(\mathbf{x}_j, \mathbf{v}_i) \tag{18}$$

Here l alludes to the subset of classified patterns and u_{ij}^{l*} stands for the partition matrix containing membership grades previously assigned to the selected patterns, possibly by a domain expert. The optimization of u_{ij}^{l} intends to make it close to the information already available. The nonnegative regularization factor α defines the balance between the supervised and unsupervised learning. Higher the value of α, higher our beliefs in the labeling decisions already made and consequently our willingness to overlook some level of structural optimization. As previously mentioned, being an optimization scheme that relies on alternate optimization, it can be trapped by local optima and is also very sensitive to initialization. Latter on this problem was ameliorated for this specific case, by using evolutionary optimization techniques [50] obviously at the expense of a great deal of computational cost.

When one does not have the knowledge about how many data classes are there but is still able to provide some indications on the resemblance or proximity of some data points, this information can be captured for instance with the help of another interesting approach presented in [52]. The underlying principle of the proximity fuzzy C-means (P-FCM) is the guided collaboration between data processing and knowledge processing through the consideration of proximity constraints. These are expressed as a symmetric and reflexive mapping of pairs of patterns to the unit interval (its value is maximal when two patterns are coincident). The relation between the fuzzy partition matrix produced by the clustering algorithm and the proximity mapping is set up by the following expression:

$$\hat{p}[k_1, k_2] = \sum_{i=1}^{C} min(u_{ik_1}, u_{ik_2}) \tag{19}$$

This expression is used to build the deduced symmetric proximity matrix. The algorithm consists of two main phases that are realized in interleaved manner. The first phase has a data driven nature and consists in applying the standard FCM to the patterns. The second concerns the accommodation of the proximity-based hints and involves some gradient oriented learning. In order to guide the gradient search procedure the objective function for the second phase penalizes the differences between the available proximity levels and the corresponding ones from the deduced proximity matrix. Since in fuzzy relational clustering, cf. [37,46], the data is described by specifying pairwise similarities or dissimilarities between objects, at first sight it seems that there is some resemblance between this algorithm and relational clustering pursuits. However it is worthwhile to note that in this case one is provided with object data describing the objects through feature measurements and using this technique, or for that matter other with similar inspiration, it is possible to conciliate strictly structural algorithmic information with the available relational information (if there is any).

4 Deploying Semantic Constraints for Data Clustering

The need for semantically valid fuzzy systems (classifiers, models, controllers) is a matter of common agreement nowadays. A set of semantic properties, necessary to ensure an interpretable fuzzy system during system design, have been outlined and analyzed [70, 62, 71]. Based on these properties, several different constraints have been derived and applied to several real-world situations, including [21,22,24,25]. For a more complete and recent survey of the literature on these matters the reader is referred to [10,11]. The great majority of the reported work on semantic concerns can be classified as belonging to the supervised learning or optimization schemes. Typically, the semantic conditions become part of the learning rule of a neural network, or part of the fitness function of a clustering algorithm, or becomes an optimization goal of its own in multi-objective optimization schemes. In this section we elaborate on a set of semantic constraints easily justifiable at the pure human-centric conceptual level, describing them in a more sound formal framework and demonstrating that they generalize a set of constraints commonly employed in partitional fuzzy clustering. Later the evolutionary semantic driven (ESD) fuzzy clustering algorithm [23] is contextualized as an illustrative example of employing such human-defined constraints where those are used as the defining characteristic which enables the correct identification of the number of clusters present in the data set.

4.1 Translation of Semantic Constraints to Clustering

The ultimate objective of clustering is the description of the inherent structure of the data in a comprehensible way. Fuzzy clustering algorithms apport a valuable surplus when aiming at such goal since the identification of regions of interest of a data set can be transposed to propositions on meaningful linguistic labels, thus facilitating the empirical semantic validation of the model.

This translation is highly dependent of the semantic soundness of the fuzzy sets in the distinct feature spaces. In this respect the set of semantic properties postulated in [71], in the context of fuzzy modeling, can be adopted as useful guidelines. These properties emerged as a mean to clarify the meaning of a linguistic term (a fuzzy set) when matched against other linguistic terms in the same universe of discourse. The proposed set of properties includes: a moderate number of membership functions, coverage, normality, natural zero positioning and distinguishability.

Three of these properties seem to have an inherent interest for the clustering endeavor:

1. A moderate number of membership functions since although this number is clearly application dependent, if one intends to describe the structure of the data in a human-friendly way there are strong reasons for imposing an upper bound on the number of clusters (in the limit, when the number of membership functions approaches the cardinality of the data, a fuzzy system becomes a numeric system). This constraint makes sense not only in the feature space, where the typical number of items efficiently handled at the short-term memory (7±2) [57] can be adopted as the upper limit of linguistic terms, but also in the data space since a high number of clusters result in information granules with a high granularity level.
2. Coverage which states that membership functions should cover the entire universe of discourse, so that every datum may have a linguistic representation.
3. Distinguishability since this property is clearly related with cluster separation (membership functions should be distinct enough from each other).

It is worth noting that these properties can be ensured using a variety of constrains, hereafter referred as interpretability constraints.

Consider the sigma-count operator, $M_p(\mathcal{L}_x)$, defined as follows:

$$M_p(\mathcal{L}_x) = \sqrt[p]{u_1^p + \ldots + u_C^p} \qquad (20)$$

where $\mathcal{L}_x$ is a fuzzy set representing a real-valued pattern x from the data set, u_i $(i = 1, \ldots, C)$ is the membership degree of x in the i-th cluster, p being a positive integer.

Therefore, in the clustering context coverage can be formalized in the following way:

$$\forall_{x \in \mathbf{X}} M_p(\mathcal{L}_x(x)) > 0 \tag{21}$$

Alternatively, one can ensure coverage using the concept of optimal interfaces [70]. Let $\mathbf{X} = [a, b] \in \mathbb{R}^m$. The ordered pair $(\mathcal{L}_x, \mathcal{N}_x)$ is said a pair of optimal interfaces *iff*

$$\forall_{x \in \mathbf{X}} \mathcal{N}_x(\mathcal{L}_x(x)) = x \tag{22}$$

where $\mathcal{L}_x : \mathbf{X} \rightarrow [0, 1]^n$ is the mapping provided by the input interface of the variable x and $\mathcal{N}_x : [0, 1]^n \rightarrow \mathbf{X}$ with $\mathcal{N}_x([0, \ldots, 0]) = \emptyset$ is the mapping provided by the output interface associated with the variable x.

Analogously distinguishability can also be enunciated with the help of the sigma-count measure:

$$\forall_{x \in \mathbf{X}} M_p(\mathcal{L}_x(x)) \leq 1 \tag{23}$$

The rationale for this constraint is straightforward: if we have two clusters "very close" to each other eventually there will be points in between the clusters' centers with high membership in both clusters. If the clusters are far apart, then there should not be such cases. Next we show that these constraints generalize the common constraint (5).

Proposition 1. *A partition matrix* $\mathbf{U}$ *represented by a class of membership degrees satisfying the constraint (23) and the optimal interface definition*

$$\forall_{j \in \{1, \ldots, N\}} \mathcal{N}(\mathcal{L}(x_j)) = x_j \tag{24}$$

generalizes the constraint (5).

Proof. Observe that $\mathcal{L}(x_j) \triangleq [u_{1j}\ u_{2j}\ \ldots\ u_{Cj}]'$. Consider by absurdity that $\exists_{s \in \{1, \ldots, N\}} \forall_{i \in \{1, \ldots, C\}} : u_{is} = 0$, or equivalently $\exists_{s \in \{1, \ldots, N\}} : \mathcal{L}(x_s) = \emptyset$. However, by definition, the output mapping $\mathcal{N}$ is undefined for the empty set, thus $\mathcal{N}(\mathcal{L}(x_s)) \neq x_s$, which contradicts (24).

Therefore $\forall_{j \in \{1, \ldots, N\}} \exists_{i \in \{1, \ldots, C\}} : u_{ij} > 0$, implying that

$$\sum_{i=1}^{C} u_{ij} > 0 \text{ for all } j = 1, \ldots, N. \tag{25}$$

If $\mathbf{U}$ satisfies the distinguishability constraint then

$$\forall_{j \in \{1, \ldots, N\}} M_p(\mathbf{u}_j) \leq 1 \tag{26}$$

where $\mathbf{u}_j \triangleq [u_{1j}\ u_{2j}\ \ldots\ u_{Cj}]'$, or in an equivalent way:

$$M_p(\mathbf{u}_j) = \left(\sum_{i=1}^{C} u_{ij}^p \right)^{1/p} \leq 1,\ j = 1, \ldots, N \tag{27}$$

Obviously for $p = 1$, $M_1(\mathbf{u}_j) = \sum_{i=1}^{C} u_{ij} \leq 1$, that together with (25), i.e., $0 < \sum_{i=1}^{C} u_{ij} \leq 1$ generalizes the common constraint (5). □

Notice that $p = 1$ determines the strongest constraint whereas $p = \infty$ describes a loose constraint. Actually it is straightforward to verify that [70]

$$M_1(\mathbf{u}_j) \geq M_2(\mathbf{u}_j) \geq \ldots M_\infty(\mathbf{u}_j) = H(\mathbf{u}_j) \tag{28}$$

where $H(\mathbf{u}_j)$ denotes the height of the fuzzy set $\mathbf{u}_j$, i.e., its maximum membership degree.

With this formulation it is clear that:

1. $\forall_{j \in \{1,\ldots,N\}}$ there is always some non-null membership degree in some cluster.
2. Given a cluster i, $i \in \{1, \ldots, C\}$, there is no guarantee that it has elements, so it may happen that $\sum_{j=1}^{N} u_{ij} = 0$.

In a first analysis, this can be obviated if a penalty term on the number of clusters is included on the cost functional or if we allow that the optimization process may also determine the number of clusters. In this work we followed the second approach as described in the following sections.

4.2 Evolutive Semantic Driven Fuzzy Clustering

Evolutionary Algorithms (EAs) are adaptive robust methods widely applicable to search, optimization and learning problems [13, 17]. EAs require a limited amount of knowledge about the problem being solved. Relative evaluation of the candidate solutions is enough and no derivatives of cost functions are required. The evolution of the potential solutions over successive generations comprises different phases. Generally speaking, the first phase involves the quantitative evaluation of each individual in the population. This value determines the probability that an individual has to be selected and to carry its genetic material for the next phase. In the second phase, the selected individuals (potential solutions) are given the chance to mate and exchange genetic material with other individuals by means of a crossover operator. The result of this reproduction phase is a new offspring population, which replaces (or sometimes compete with) the previous population. Some of these newly born individuals were possibly prone to some mutations. This process continues until a stop criterion has been met.

Chromosome Representation

Evolutionary algorithms, being a general optimization strategy, can be adapted to objective function based fuzzy clustering. Obviously, it is necessary to find an adequate representation for the parameters to be optimized, viz. the prototypes and the membership degrees of the partition matrix. However the simultaneous optimization of the C prototypes and the $C \times N$ membership degrees seems, to say the least, unpractical. Thus we restricted the complexity of the optimization task by optimizing only the prototypes and computing the corresponding membership degrees using the updating expression of the PCM model (12) which, as was previously said, does not impose a strict fuzzy partition since the degree of membership of each point depends exclusively of its distance towards the prototype.

Fitness Assignment

In the presented evolutionary semantic driven fuzzy clustering algorithm the quantitative assessment of each potential solution is based on the sum-of-squares criterion (3) but introduces two extra terms in order to ensure the above mentioned properties of coverage and distinguishability:

$$\begin{aligned} Q_{cov} &= \textstyle\sum_x \|x - x^*\|^2 \\ Q_{dist} &= \textstyle\sum_x [(M_p(\mathbf{u}_j) - 1)^2 step(M_p(\mathbf{u}_j) - 1)] \end{aligned} \tag{29}$$

where *step* represents the unit step function (equals 1 if its argument is greater than zero and equals 0 otherwise). The index, Q_{cov}, based on the concept of optimal interfaces [71], seeks an adequate coverage level. To compute $x^* = \mathcal{N}_x(\mathcal{L}_x(x))$ the internal representation $\mathcal{L}_x$ was specified in agreement with (12) and the output interface $\mathcal{N}_x$ was given by the center of gravity defuzzification method:

$$x_j^* = \frac{\sum_{i=1}^{C} u_{ij}\mathbf{v}_i}{\sum_{i=1}^{C} u_{ij}} \tag{30}$$

The index Q_{dist} is intended to keep the clusters prototypes apart from each other. This implies that the points with a sigma-count above the unity become penalized.

A very interesting side effect of these indices, is that in a variable number of clusters optimization environment Q_{dist} attains its minimum for the single cluster case, $C = 1$, whereas Q_{cov} attains its minimum when the number of clusters is equal to the number of patterns, $C = N$. To fully comprehend the importance of this side effect, one should recall that in a variable number of clusters optimization scheme, Q_{FCM} (3) also attains its minimum at $C = N$. Therefore, when both three criteria are considered, a balance is obtained for some C in between 1 and N. In the reported experiments we will demonstrate that this tradeoff can be used in our advantage as a mean of determination of the number of clusters.

The quantitative assessment of each individual in the population is given by the following cost functional:

$$Q_{ESD} = Q_{FCM} + Q_{cov} + Q_{dist} \tag{31}$$

Thus the variable length chromosome, exhibiting a set of real coded cluster prototypes which result in low intra-cluster distance, combined with an adequate coverage level (and accurate positioning) altogether with clearly distinct clusters obtains an higher fitness value than other lacking any of these three desired features.

Since EAs are designed for maximization problems, the current minimization problem was converted into a maximization one using the following transformation:

$$fitness(x) = \frac{1}{1 + Q_{ESD}(x)} \tag{32}$$

We would like to stress that in the expression (31) a set of variable weights could be considered in order to give different importance to each of the three terms, however such study is beyond the scope of this work. In the reported experiments we treated each objective in the same way: for each chromosome its value was calculated, then the whole range of values across the entire population of chromosomes is translated to the unit interval and only after this normalization the three objectives were summed up as in (31).

4.3 Numerical Examples

A collection of synthetic data sets presenting distinct challenges for the clustering process was used to demonstrate the viability of our approach. In order to simplify the visualization in these examples the patterns are distributed in $\mathbb{R}^2$. Moreover we present the results obtained from two *de facto* benchmark data sets from the clustering literature: the Ruspini data set [68] and the Iris Plants data set [1]. The presented results refer to mean values of ten independent runs per data set.

The parameters of the EA were kept constant in the presented experiments. A stopping criterion of 1000 iterations was used. The population was composed of 200 chromosomes, each one representing a set of real-coded cluster prototypes, as depicted in Fig. 5.

The maximum number of prototypes per chromosome was dependent on the number of patterns and fixed on the common heuristic approximation for the maximum number of clusters in a data set, viz. $\sqrt{N}$. Each cluster prototype had

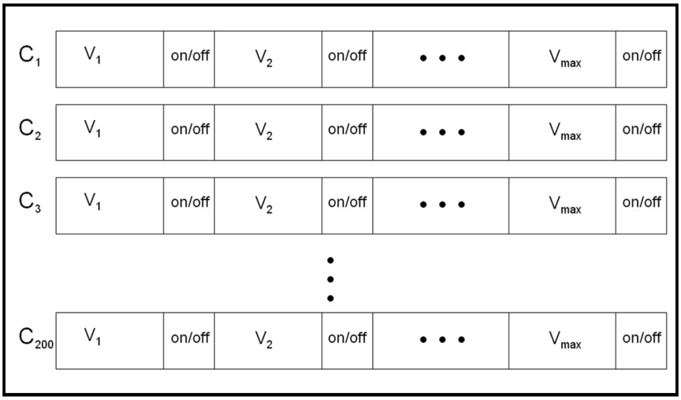

Fig. 5. Graphical representation of the population

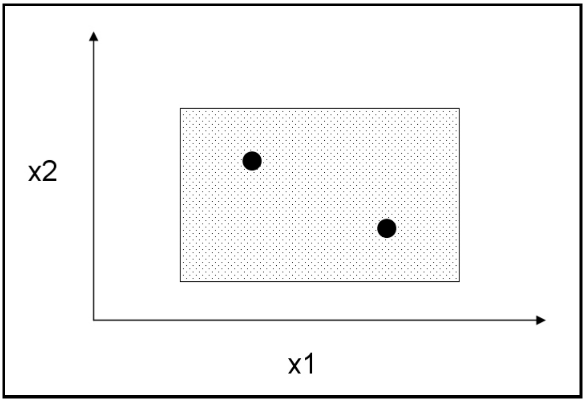

Fig. 6. The BLX-α crossover operator. The big dots represent the parents, while the small dots indicate possible children.

a bit field indicating whether or not it was active, allowing a variable number of clusters representation. During crossover this bit field was exchanged between parents – as we will see the crossover of the prototypes was more elaborated.

The selection operator applied in the experiments was stochastic sampling with replacement also known as the roulette wheel selection method. This sampling method selects parents according to a spin of a weighted roulette wheel. The high-fit individuals will have more area assigned to them on the wheel and hence, a higher probability of being selected to the mating pool where they are combined with other individuals by means of a crossover operator.

The blend crossover operator, BLX-α, is specifically designed for real-valued chromosomes [20] and was used to combine the genes encoding the prototypes. The resulting offsprings are distributed across a hyper-box encompassing the two parents. The parameter α extends the bounds of the hyper-box, hence to the children is given the possibility to explore new search space inside of an extended range given by their parents, see Fig. 6.

Each real-valued component, x_i, of the new offspring is randomly chosen (with an uniform distribution) from the interval $[x_{imin} - I.\alpha,\ x_{imax} + I.\alpha]$, where $x_{imin} = min(x_i^A, x_i^B)$, $x_{imax} = max(x_i^A, x_i^B)$ and $I = x_{imax} - x_{imin}$ with A and B denoting the two progenitors. In fact BLX-α is designed to promote diversity, greater with the increase of α, counteracting the decrease in variance that results from the application of the selection operator hence preventing premature convergence of the population. In this work the BLX-α crossover operator was applied with 0.9 probability and the parameter α was set to 0.5 – a common choice in the literature.

The probability of the uniform mutation operator was set to 0.05. In order to prevent that good solutions disappear from the population an elitist approach was used. For each $c \in \{2, \ldots, \sqrt{N}\}$ the best chromosome with c active prototypes was preserved for the next generation. The remaining parameters of the

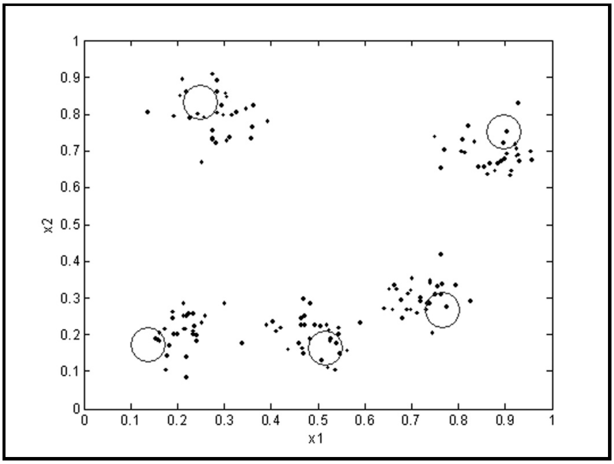

Fig. 7. Data set with 5 clusters each one comprising 30 patterns. The unfilled circles mark the prototypes.

algorithm were defined as follows: $p = 1$ in (29); $m = 2$ in (3), (12) and (13). In the presented experiments we used the Euclidean distance as the measure of dissimilarity.

Example 1

In this example the data set is composed of five visually separable clusters resulting from a normal distribution of thirty elements around 5 distinct points (Table 1 presents the details of the distribution).

Figure 7 presents the data set and the means of the cluster prototypes (unfilled circles) for ten distinct runs.

It is interesting to refer that the algorithm was able to find the number of focal elements used to generate the data set in every run. Also the positioning of the centers of the clusters reveals a noticeable feature of the algorithm: the distinguishability index is contributing to maintain the areas of influence of the clusters apart from each other.

Table 1. Details of the normal distribution of the data set of Example 1 (5 clusters)

Cluster	#Points	Mean	Standard Deviation
1	30	(0.3; 0.8)	(0.05; 0.05)
2	30	(0.2; 0.2)	(0.05; 0.05)
3	30	(0.5; 0.2)	(0.05; 0.05)
4	30	(0.7; 0.3)	(0.05; 0.05)
5	30	(0.9; 0.7)	(0.05; 0.05)

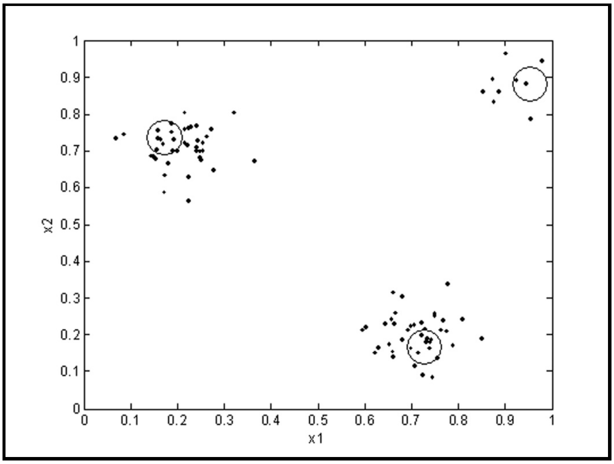

Fig. 8. Data set with 2 clusters each one comprising 40 patterns and an additional cluster with 10 patterns. The unfilled circles mark the prototypes.

Example 2

The data set presented here is formed by three visually separable clusters. Two of them are composed of forty patterns whereas the third one is composed of only ten elements. Table 2 presents the details of the distribution.

Table 2. Details of the normal distribution of the data set of Example 2 (3 clusters)

Cluster	#Points	Mean	Standard Deviation
1	40	(0.2; 0.7)	(0.05; 0.05)
2	40	(0.7; 0.2)	(0.05; 0.05)
3	10	(0.9; 0.9)	(0.05; 0.05)

Figure 8 presents the data set and the means of the cluster prototypes for the ten distinct runs. The algorithm was able to recognize the correct number of clusters.

In the next experiment the cluster was further reduced to five points. Even for this more demanding case, the algorithm was able to recognize the data structure, see Fig. 9.

Next, the third "cluster" was further shrunk to only two points, see Fig. 10. In this case the algorithm negotiated these as outliers.

In order to infer the robustness of the algorithm to noise, we kept the 2 main clusters and injected 50 noise points uniformly distributed in the unit square,

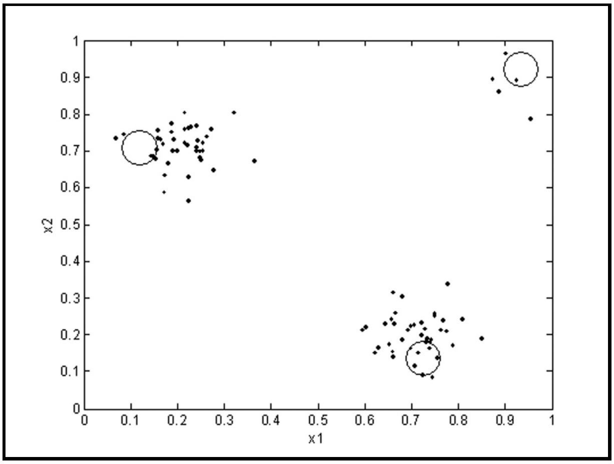

Fig. 9. Data set with 2 clusters each one comprising 40 patterns and an additional cluster with 5 patterns. The unfilled circles mark the prototypes.

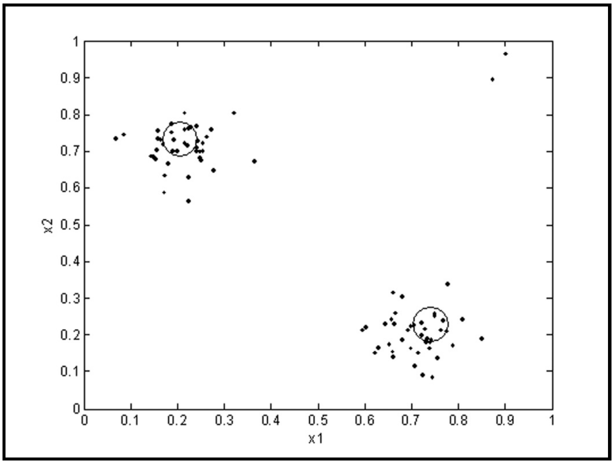

Fig. 10. Data set with 2 clusters each one comprising 40 patterns and two outliers. The unfilled circles mark the prototypes.

Fig. 11. Once again the algorithm was able to correctly identify the two clusters and find the optimal number of clusters.

Ruspini Data Set

The Ruspini data set [68] has become a benchmark to assessing the performance of clustering algorithms. Figure 12 shows the Ruspini data set and clusters found by the proposed algorithm.

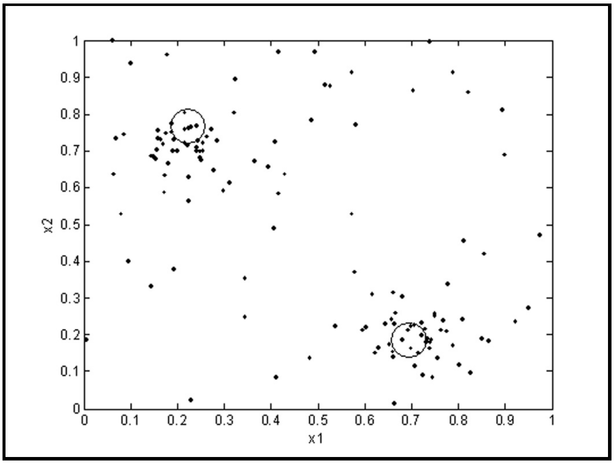

Fig. 11. Data set with 2 clusters each one comprising 40 patterns plus 50 noisy patterns. The unfilled circles mark the prototypes.

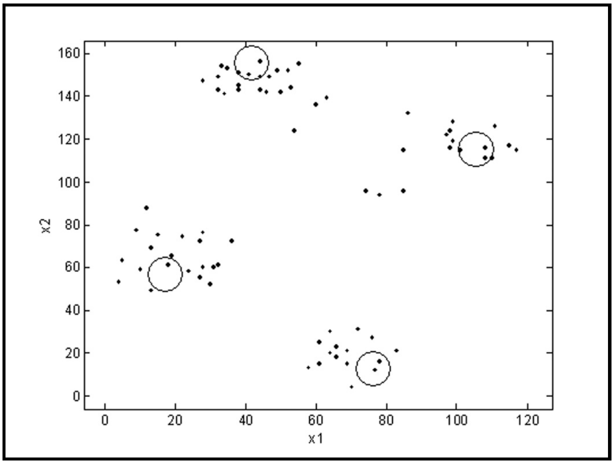

Fig. 12. Ruspini data set. The unfilled circles mark the prototypes.

Although the data has a much different scale than the unit square considered in the previous examples no additional adjustments were made to the algorithm. However this does not mean that when faced with huge data sets with large dimensionality these are not due. As a matter of fact for these cases a scale invariant distance function should be used and the relative weight of the distinguishability constraint should have to be analyzed. It is interesting to note that the algorithm was able to find the optimal number of clusters.

Iris Plants

The Iris Plants database is one of the best known databases in the pattern recognition literature. The data set contains 3 classes referring to 3 types of iris plant physically labeled as Setosa, Versicolour and Virginica. There are 50 instances per class, which are described by 4 attributes: sepal length, sepal width, petal length and petal width. The first class is linearly separable from the others, but the other two classes are known to overlap each other in their numeric representations.

In this case the results of the algorithm were not conclusive. In exactly 50% of the runs the suggested number of clusters was 3 whereas in the remaining ones it produced only two clusters. This is a curious result since, although Iris database contains three physical labeled classes, from a geometrical point of view it is probably composed of only two distinct clusters. For the runs which identify 3 clusters the overall mean value of correct classifications was 92.75 ± 1.58. The error rate results exclusively from patterns misclassified in the two overlapping classes.

As a concluding remark it is worthwhile to stress that the translation of the human-defined semantic constraints referring to coverage and distinguishability of linguistic terms into the clustering framework is encouraging since in the presented experiments the proposed evolutionary algorithm, with the inclusion of the semantic constraints, was able to find the centers of the clusters and, more important, to determine the correct number of clusters without appealing to the computation of any kind of validity measures.

5 Conclusions

The fuzzy clustering framework provides a collection of tools well suited to discover structural information among challenging data sets with overlapping chunks of data and vaguely defined boundaries between clusters. Moreover, due to the wealthy of tools available and seamless integration of linguistic meaning it positions itself as an excellent mechanism to support the construction of information granules. These features assume a level of paramount importance not only for the data analyst, but also for the end-user who definitely needs a user-friendly mean to interact with the system. This interaction subsumes two facets: the human-like intuitive presentation of the clustering findings and consequent validation of these, as well as the transmission of additional knowledge that can be used to improve the clustering results. Along the first two parts of this chapter it was made clear that the fuzzy clustering framework is perfectly able to provide answers to each one of these aspects.

The leitmotif of the last part of the chapter was the discussion of the fuzzy clustering from a diametrically different point of view: *"Can the clustering process be conceptualized in a more human-oriented way?"* In this regard it was shown that the adoption of a set of semantic constraints aiming at the enhancement of the human perception of the system can also be applied to fuzzy clustering algorithms.

References

1. Anderson, E.: The Irises of the Gaspe peninsula. Bull. Amer. Iris Soc. 59, 2–5 (1935)
2. Bargiela, A., Pedrycz, W.: Recursive Information Granulation: Aggregation and Interpretation Issues. IEEE Trans. on Systems, Man, and Cybernetics 33(1), 96–112 (2003)
3. Bargiela, A., Pedrycz, W.: A Model of Granular Data: a Design Problem with the Tchebyshev FCM. Soft Comput. 9, 155–163 (2005)
4. Bargiela, A., Pedrycz, W.: The roots of granular computing. In: Proc. Int. Conference on Granular Computing, pp. 806–809 (2006)
5. Bensaid, A., et al.: Partially supervised clustering for image segmentation. Pattern Recognition 29, 859–871 (1996)
6. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
7. Bezdek, J., Coray, C., Gunderson, R., Watson, J.: Detection and characterization of cluster substructure, I linear structure: Fuzzy c-lines. J. Appl. Math. 40(2), 339–357 (1981)
8. Bezdek, J., Coray, C., Gunderson, R., Watson, J.: Detection and characterization of cluster substructure, II Fuzzy c-varieties and convex combinations thereof. J. Appl. Math. 40(2), 358–372 (1981)
9. Bezdek, J., Hathaway, R., Sabin, M., Tucker, W.: Convergence theory for fuzzy c-means: Counterexamples and repairs. IEEE Trans. Syst., Man and Cybern. 17, 873–877 (1987)
10. Casillas, J., Cordón, O., Herrera, F., Magdalena, L. (eds.): Accuracy Improvements in Linguistic Fuzzy Modeling. Physica-Verlag (2003)
11. Casillas, J., Cordón, O., Herrera, F., Magdalena, L. (eds.): Interpretability Issues in Fuzzy Modeling. Physica-Verlag (2003)
12. Castellano, G., Fanelli, A., Mencar, C.: Generation of interpretable fuzzy granules by a double-clustering technique. Archives of Control Sciences - Special Issue on Granular Computing 12(4), 397–410 (2002)
13. Coello, C., Van Veldhuizen, D., Lamont, G.: Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic Algorithms and Evolutionary Computation Series, vol. 5. Springer, Heidelberg (2002)
14. Dave, R.: Characterization and detection of noise in clustering. Pattern Recognition Letters 12, 657–664 (1991)
15. Dave, R., Sen, S.: On generalising the noise clustering algorithms. In: Proc. of the 7th IFSA World Congress, IFSA 1997, pp. 205–210 (1997)
16. Dave, R.: Fuzzy shell clustering and application to circle detection in digital images. Int. J. General Syst. 16, 343–355 (1990)
17. Davis, L.: Handbook of genetic algorithms. Van Nostrand Reinold (1991)
18. Egan, M.: Locating clusters in noisy data: a genetic fuzzy c-means clustering algorithm. In: Proceedings of the Conference of the North American Fuzzy Information Processing Society, pp. 178–182. IEEE, Los Alamitos (1998)
19. Egan, M., Krishnamoorthy, M., Rajan, K.: Comparative Study of a Genetic Fuzzy c-means Algorithm and a Validity Guided Fuzzy c-means Algorithm for Locating Clusters in Noisy Data. In: Proc. International Conference on Evolutionary Computation, pp. 440–445 (1998)
20. Eshelman, L., Schaffer, J.: Real-coded genetic algorithms and interval-schemata. In: Whitley, L.D. (ed.) Foundations of Genetic Algorithms 2, pp. 187–202. Morgan Kaufmann, San Mateo (1993)

21. Fazendeiro, P., Valente de Oliveira, J.: A Survey of Fuzzy Control Strategies for Neuromuscular Blockade Using Continuous Infusion of Atracurium. In: Proc. FUZZ-IEEE 2002, Honolulu, USA, pp. 547–552 (2002)
22. Fazendeiro, P., Valente de Oliveira, J.: A Working Hypothesis on the Semantics/Accuracy Synergy. In: Proc. Joint EUSFLAT-LFA, Barcelona, Spain, pp. 266–271 (2005)
23. Fazendeiro, P., Valente de Oliveira, J.: A semantic driven evolutive fuzzy clustering algorithm. In: Proc. of FUZZ-IEEE, London, UK, July 2007, pp. 1977–1982 (2007)
24. Fazendeiro, P., Valente de Oliveira, J., Pedrycz, W.: Multi-objective evolutionary optimization of accuracy and interpretability for neuromuscular blockade control. In: Proc. of IFSA 2005 - The International Fuzzy Systems Association World Congress, Beijing, China, pp. 908–913 (2005)
25. Fazendeiro, P., Valente de Oliveira, J., Pedrycz, W.: A Multi-Objective Design of a Patient and Anaesthetist-Friendly Neuromuscular Blockade Controller. IEEE Trans. on BioMedical Engineering 54(9), 1667–1678 (2007)
26. Forgy, E.: Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. Biometrics 21, 768–780 (1965)
27. Fralley, C., Raftery, A.: How many Clusters? Which Clustering Method? Answers via Model-Based Cluster Analysis. The Computer Journal 41(8), 578–588 (1998)
28. Frigui, H., Krishnapuram, R.: Clustering by competitive agglomeration. Pattern Recognition 30(7), 1109–1119 (1997)
29. Frigui, H., Krishnapuram, R.: A robust competitive clustering algorithm with applications in computer vision. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(5), 450–465 (1999)
30. Gath, I., Geva, A.: Unsupervised Optimal Fuzzy Clustering. IEEE Trans. Pattern Analysis and Machine Intelligence 11, 773–781 (1989)
31. Geva, A.: Hierarchical Unsupervised Fuzzy Clustering. IEEE Trans. Fuzzy Systems 7(6), 723–733 (1999)
32. Gustafson, D., Kessel, W.: Fuzzy clustering with a fuzzy covariance matrix. In: Proc. IEEE Conf. Decision Contr., San Diego, CA, pp. 761–766 (1979)
33. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On Clustering Validation Techniques. Journal of Intelligent Information Systems 17(2/3), 107–145 (2001)
34. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Clustering algorithms and validity measures. In: Proc. 13th International Conference on Scientific and Statistical Database Management, pp. 3–22 (2001)
35. Hall, L., Ozyurt, B., Bezdek, J.: Clustering with a genetically optimized approach. IEEE Trans. on Evolutionary Computation 3(2), 103–112 (1999)
36. Hathaway, R., Bezdek, J.: Switching regression models and fuzzy clustering. IEEE Trans. Fuzzy Syst. 1, 195–204 (1993)
37. Hathaway, R., Bezdek, J.: Clustering incomplete relational data using the non-Euclidean relational fuzzy c-means algorithm. Pattern Recognition Letters 23, 151–160 (2002)
38. Hathaway, R., Bezdek, J., Hu, Y.: Generalized Fuzzy c-Means Clustering Strategies Using L_p Norm Distances. IEEE Transactions on Fuzzy Systems 5, 576–582 (2000)
39. Hirota, K., Pedrycz, W.: Fuzzy computing for data mining. Proceedings of the IEEE 87(9), 1575–1600 (1999)
40. F. Hoppner, "Fuzzy shell clustering algorithms in image processing– Fuzzy c-rectangular and two rectangular shells," IEEE Trans. Fuzzy Syst., 5, 599–613 (1997)
41. Hoppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis. John Wiley, Chichester (1999)

42. Jain, A., Murty, M., Flynn, P.: Data clustering: A review. ACM Comput. Surv. 31(3), 264–323 (1999)
43. Klawonn, F., Keller, A.: Fuzzy clustering with evolutionary algorithms. International Journal of Intelligent Systems 13(10/11), 975–991 (1998)
44. Klawonn, F., Höppner, F.: What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier. In: Berthold, M.R., Lenz, H.-J., Bradley, E., Kruse, R., Borgelt, C. (eds.) IDA 2003. LNCS, vol. 2810, pp. 254–264. Springer, Heidelberg (2003)
45. Krisnapuram, R., Frigui, H., Nasroui, O.: Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation. IEEE Trans. Fuzzy Syst. 3(1), 29–60 (1995)
46. Krishnapuram, R., Joshi, A., Yi, L.: A Fuzzy Relative of the k-Medoids Algorithm with Application to Web Document and Snippet Clustering. In: Proc. IEEE International Fuzzy Systems Conference, pp. 1281–1286 (1999)
47. Krishnapuram, R., Keller, J.: A possibilistic approach to clustering. IEEE Trans. on Fuzzy Systems 1, 98–110 (1993)
48. Krishnapuram, R., Keller, J.: The possibilistic C-Means algorithm: insights and recommendations. IEEE Trans. on Fuzzy Systems 4, 385–393 (1996)
49. Liu, G.: Introduction to Combinatorial Mathematics. Mc-Graw-Hill, New York (1968)
50. Liu, H., Huang, S.-T.: Evolutionary semi-supervised fuzzy clustering. Pattern Recognition Letters 24, 3105–3113 (2003)
51. Liu, J., Xie, W.: A genetics-based approach to fuzzy clustering. In: Proc. of the IEEE International Conference on Fuzzy Systems, vol. 4, pp. 2233–2240 (1995)
52. Loia, V., Pedrycz, W., Senatore, S.: P-FCM: a proximity-based fuzzy clustering for user-centered web applications. International Journal of Approximate Reasoning 34, 121–144 (2003)
53. Lorette, A., Descombes, X., Zerubia, J.: Fully unsupervised fuzzy clustering with entropy criterion. Proceedings 15th International Conference on Pattern Recognition 3, 986–989 (2000)
54. MacQueen, J.: Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. 1, 281–297 (1967)
55. Maulik, U., Bandyopadhyay, S.: Performance evaluation of Some Clustering Algorithms and Validity Indices. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(12), 1650–1654 (2002)
56. Maulik, U., Bandyopadhyay, S.: Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. IEEE Trans. on Geoscience and Remote Sensing 41(5), 1075–1081 (2003)
57. Miller, G.: The magic number seven, plus or minus two: Some limits on our capacity for processing information. Psychol. Rev. 63, 81–97 (1956)
58. Milligan, G., Cooper, C.: An examination of procedures for Determining the Number of Clusters in a Data Set. Psychometrika 50(2), 159–179 (1985)
59. Ohashi, Y.: Fuzzy Clustering and robust estimation. In: Proc. of 9th Meeting SAS User Group Int., Florida (1984)
60. Pal, N., Bezdek, J.: On cluster validity for the fuzzy c-means model. IEEE Transactions on Fuzzy Systems 3(3), 370–379 (1995)
61. Pedrycz, W.: Conditional fuzzy clustering in the design of radial basis function neural network. IEEE Trans. Neural Networks 9, 601–612 (1998)
62. Pedrycz, W., Valente de Oliveira, J.: An Algorithmic Framework for Development and Optimization of Fuzzy Models. Special issue of Fuzzy Sets and Systems on Fuzzy Modelling 80(1), 37–55 (1996)

63. Pedrycz, W., Vukovich, G.: Abstraction and Specialization of Information Granules. IEEE Trans. on Systems, Man and Cybernetics, Part B 31(1), 106–111 (2001)
64. Pedrycz, W., Waletzky, J.: Fuzzy Clustering with Partial Supervision. IEEE Trans. on Systems, Man and Cybernetics, Part B 27(5), 787–795 (1997)
65. Runkler, T., Bezdek, J.: Alternating cluster estimation: A new tool for clustering and function approximation. IEEE Trans. Fuzzy Syst. 7, 377–393 (1999)
66. Runkler, T.: Ant colony optimization of clustering models. International Journal of Intelligent Systems 20(12), 1233–1261 (2005)
67. Runkler, T., Katz, C.: Fuzzy Clustering by Particle Swarm Optimization. In: Proc. IEEE International Conference on Fuzzy Systems, pp. 601–608 (2006)
68. Ruspini, E.: A New Approach to Clustering. Information Control 15, 22–32 (1969)
69. Sarkar, M.: Evolutionary programming-based fuzzy clustering. In: Proc. of the Fifth Annual Conference on Evolutionary Programming, pp. 247–254. MIT Press, Cambridge (1996)
70. Valente de Oliveira, J.: A design methodology for fuzzy system interfaces. IEEE Trans. on Fuzzy Systems 3(4), 404–414 (1995)
71. Valente de Oliveira, J.: Semantic Constraints for Membership Function Optimization. IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Man 29(1), 128–138 (1999)
72. Valente de Oliveira, J., Pedrycz, W. (eds.): Advances in Fuzzy Clustering and Its Applications. Wiley, Chichester (2007)
73. Wei, C.-H., Fahn, C.-S.: A distributed approach to fuzzy clustering by genetic algorithms. In: Proceedings of the Asian Fuzzy Systems Symposium, pp. 350–357 (1996)
74. Xie, X., Beni, G.: A Validity Measure for Fuzzy Clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence 13, 841–847 (1991)
75. Xu, R., Wunsch II, D.: Survey of Clustering Algorithms. IEEE Trans. on Neural Networks 16(3), 645–678 (2005)
76. Yuan, B., Klir, G., Swan-Stone, J.: Evolutionary fuzzy c-means clustering algorithm. In: Proc. of IEEE International Conference on Fuzzy Systems, vol. 4, pp. 2221–2226 (1995)
77. Zadeh, L.: Fuzzy sets. Inform. Contr. 8, 338–353 (1965)
78. Zadeh, L.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. on Systems, Man, and Cybernetics SMC-3(1), 28–44 (1973)
79. Zadeh, L.: The concept of linguistic variable and its application to approximate reasoning. Information Sciences 8, 199–249 (part I) (1975); 301–357 (part II)
80. Zadeh, L.: The concept of linguistic variable and its application to approximate reasoning. Information Sciences 9, 43–80 (1976) (part III)
81. Zadeh, L.: Fuzzy logic = computing with words. IEEE Trans. on Fuzzy Systems 4(2), 103–111 (1996)
82. Zadeh, L.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 90, 111–127 (1997)
83. Zadeh, L.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft Computing 2(1), 23–25 (1998)