Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2015, Article ID 265135, 13 pages
http://dx.doi.org/10.1155/2015/265135

Research Article

Fuzzy Clustering Using the Convex Hull as Geometrical Model

Luca Liparulo, Andrea Proietti, and Massimo Panella

Department of Information Engineering, Electronics and Telecommunications (DIET), University of Rome "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy

Correspondence should be addressed to Massimo Panella; massimo.panella@uniroma1.it

Received 22 March 2015; Accepted 3 April 2015

Academic Editor: Katsuhiro Honda

A new approach to fuzzy clustering is proposed in this paper. It aims to relax some constraints imposed by known algorithms using a generalized geometrical model for clusters that is based on the convex hull computation. A method is also proposed in order to determine suitable membership functions and hence to represent fuzzy clusters based on the adopted geometrical model. The convex hull is not only used at the end of clustering analysis for the geometric data interpretation but also used during the fuzzy data partitioning within an online sequential procedure in order to calculate the membership function. Consequently, a pure fuzzy clustering algorithm is obtained where clusters are fitted to the data distribution by means of the fuzzy membership of patterns to each cluster. The numerical results reported in the paper show the validity and the efficacy of the proposed approach with respect to other well-known clustering algorithms.

1. Introduction

Clustering algorithms always represented an efficient and important method for analysing either small or big amounts of data, namely, for dividing groups of objects into clusters by using some measures of similarity or dissimilarity on the basis of a suited number of features representing data [1–3]. The applications of clustering span in every field of science and technology, especially in machine learning, computer science, statistics, engineering, physics, mathematics, medicine, and so on. Early in the twentieth century, a huge number of algorithms and the related variants have been proposed in the literature, each being adapted to the specific field of application [4–16].

Clustering techniques deal with unsupervised learning as they are used when it is not possible to define data labels a priori. They utilize several metrics for the determination of similar objects (patterns) belonging to the same group (a cluster) that, in turn, are different from patterns of other clusters [17]. Clearly, the shape of a cluster is influenced by the chosen metric, such as Euclidean, Manhattan, Chebyshev, or Mahalanobis distance; in fact, two patterns can be "close" (or "similar") using one metric and "far" (or "dissimilar") by using another one.

Similar considerations are also valid when clusters are considered as fuzzy sets [18]. In this case, the patterns are assigned to several clusters in a nonexclusive way by determining the degree of fuzzy membership of every pattern to the present clusters. However, the geometrical constraints imposed by the membership function (MF) may represent a remarkable obstacle for the clustering analysis. In this regard, most algorithms tend to create spherical, ellipsoidal, or polygonal fuzzy clusters having a simple geometry that is computationally affordable but possibly unfit to the actual distribution of data.

Different taxonomies hold for both fuzzy and crisp algorithms; the most considered aspects are as follows:

(i) K-clustering or free-clustering techniques, according to the a priori determination of the number (K) of clusters;

(ii) partitional or hierarchical (agglomerative/divisive) procedures for cluster generation, where the dataset is partitioned directly into a set of disjoint clusters or else the solution depends on the previous or successive ones in a hierarchical sequence;

(iii) sequential (online) or batch (iterative) algorithms, through which either clusters are updated sequentially at any presentation of a new pattern or they are updated iteratively considering a given set of data. As discussed successively, there may exist hybrid cases when a dataset is used to determine clusters sequentially but several times, as, for example, in a learning process by epochs or in tuning procedures of parameters;

(iv) model-based, distribution-based, or density-based clusters, when clusters are associated with geometric models defined in the data space or they are associated with suitable statistic distributions or density functions;

(v) point-to-centroid or point-to-boundary based metrics, where the distances of patterns from clusters are computed considering a single prototype (i.e., a point or centroid) representing each cluster or distances are scaled according to the actual extension of clusters in the data space, independently of the use of model-based, distribution-based, or density-based clusters.

Nowadays, there are no clustering algorithms whose performance is universally recognized to be satisfactory for all problems. A trade-off is often necessary among computational complexity, model fitting, and explanatory tools of the clusters' structure, depending on the nature of data under analysis and the specific field of application. Iterative algorithms perform clustering until a stopping rule is verified; they tend to be more accurate than sequential algorithms that, in turn, are faster but depend on the pattern presentation order. In this regard, well-known online clustering methods recently proposed in the literature are the recursive fuzzy c-means [19], recursive Gustafson-Kessel clustering [20], recursive subtractive clustering (eTS) method [21], evolving clustering method (ECM) [22], dynamic evolving neural-fuzzy inference system (DENFIS) method [23], and so on.

Furthermore, K-clustering techniques have a great limitation, since they are useful only for those problems when the number of clusters may be known in advance [24–26]. Actually, there is a huge amount of literature that focuses on the problem of "cluster validity," that is, how to determine the optimal value of K for a given dataset [27–29]. These methods are able to evaluate whether a final clustering result is better than another one by means of suited criteria as, for instance, the compactness and separability of clusters. Therefore, they usually work by defining an index and then by finding the minimum (or maximum) of the values associated with each clustering solution.

The underlying idea of this paper is to propose a new approach to fuzzy clustering, with the aim of relaxing some constraints imposed by known algorithms and using a new method for the computation of MFs. The starting point is Simpson's idea of the well-known "Fuzzy Min-Max" clustering algorithm [30]: we propose a free-clustering, partitional, online algorithm using model-based clusters whose shape is determined in a new way by the convex hull computation. Our contribution comes from the awareness that Simpson's method is very efficient but it has an important constraint, the shape of clusters, given that it creates hyperboxes parallel to the coordinate axes of the data reference frame only. This constraint will be removed by using the convex hull computation of clusters and, necessarily, an original methodology in order to define a metric associated with the MFs.

The use of unconstrained clusters in the analysis of large datasets allows us to assort patterns in extremely compact clusters [31, 32]. Nevertheless, we will show that the use of fuzzy logic combined with a more flexible geometry of clusters yields robust results with respect to the uncertainty of data by means of computationally efficient procedures [33–35]. Anyway, the approach herein proposed, essentially applied to the class of online algorithms and model-based clusters fitted by convex geometrical polytopes, can be generalized also to a larger choice of algorithms, even in the case of hierarchical procedures, iterative algorithms, and nonconvex models of clusters.

The paper is organized as follows. In Section 2, we introduce and discuss well-known techniques for convex hull computation with regard to their application in the field of pattern recognition, in particular for data clustering; an overview of the most relevant works presented in the literature is reported in Section 3. The new fuzzy clustering algorithm proposed in the paper is illustrated in detail in Section 4, where the use of convex hull is demonstrated by means of simple toy tests. Successively, the way by which MFs are determined in order to represent fuzzy clusters based on the adopted geometrical models is clearly explained in Section 5, while the performance of the proposed algorithm and its comparison with other popular clustering algorithms, considering different datasets, are reported in Section 6. Finally, our conclusions and discussions are drawn in Section 7.

2. Convex Hull Computation and Data Clustering

In this paper, we propose a novel and generalized fuzzy clustering algorithm, which is useful to analyse data for online and real-time applications [36]. The shape of clusters is generalized by using less regular structures seemingly more complex but more computationally affordable and also it is able to fit better the local distribution of data, with less sparse geometrical structures as well as using more flexible and dynamic clustering rules. In this regard, we propose the use of the convex hull for the determination of irregular convex polytopes.

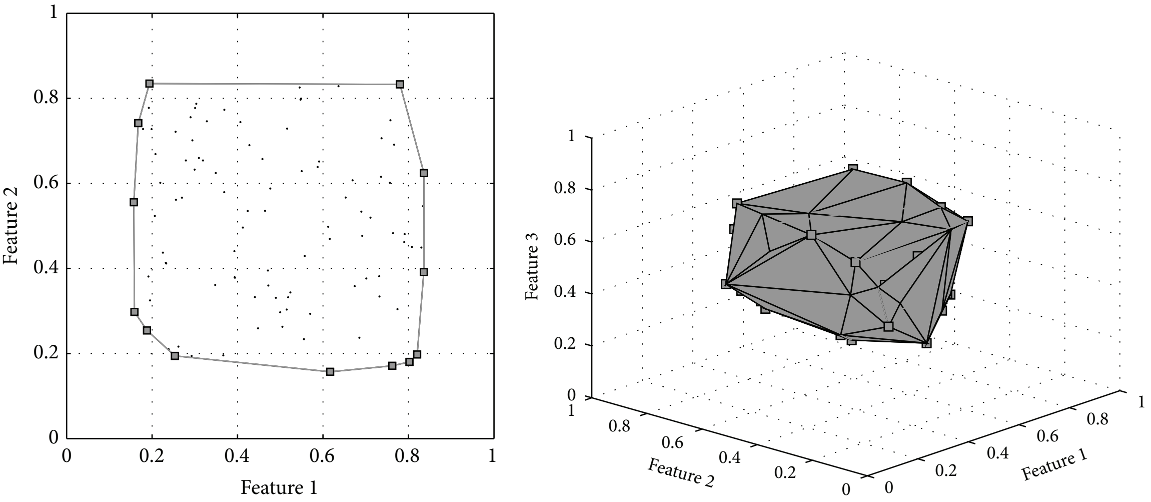
The convex hull of a set of points is the smallest convex set that contains these points, as illustrated in Figure 1 for 2D and 3D datasets. We can represent an N-dimensional convex hull by a set of L points in $\mathbb{R}^N$ called "vertices" or, equivalently, by $(N - 1)$-dimensional faces called "facets." Each facet is characterized by the following:

(i) a set of vertices;

(ii) a set of neighboring facets;

(iii) a hyperplane equation.

The $(N - 2)$-dimensional faces are the "ridges" of the convex hull; each ridge is the intersection of the vertices of two neighboring facets. The relationship between the number of vertices and facets of convex polytopes for $N \geq 3$ is not trivial; for this reason the convex hull determination



(a) 2D dataset

(b) 3D dataset

Figure 1: Some examples of convex hull for 2D and 3D datasets; the Quickhull algorithm is applied to determine the convex hulls.

is also referred to as the "vertex enumeration" or "facet enumeration" problem.

There are many methods for the convex hull evaluation [37–42]. In this paper, we propose the "Quickhull" algorithm [43], which is able to compute the convex hull in 2D, 3D, and higher dimensions. The Quickhull realizes an efficient implementation of the convex hull algorithm by combining a 2D procedure with the N-D Beneath-Beyond algorithm [44]. Precisely, Quickhull algorithm uses a simplification of the Beneath-Beyond theorem to determine efficiently the visible facets for a given set of points.

3. Related Works

Several works can be found in the literature that focus on the use of the convex hull within clustering. For example, an interesting method is the one proposed in [45], which is a two-level fuzzy clustering adaptive method able to expand or merge "flexible" convex polytopes that makes use of convex hull for modeling clusters. Several experimental results are given to show the validity of this method but there are several drawbacks, in particular the dependence of the MF on four parameters, the use of a training set to initialize the clustering algorithm, and the high computational cost for a pattern inclusion within a convex set, calculated by means of a pure geometrical method.

An intelligent fuzzy convex hull based clustering approach to address the problem stating number of clusters and parameter adjustment is proposed in [46], where the authors fused the concept of convex hull with fuzziness parameter. The proposed algorithm tries to capture the basic idea of clustering and to provide an optimal set of clusters such that the overlapping of cluster points can be easily identified by defining the boundary around the points. The numerical results seem to be interesting but the authors considered just one dataset, so the proposed algorithm lacks other comparisons by using more data.

A three-phase K-means convex hull triangulation approach is shown in [47], which is able to detect clusters with both complex and nonconvex shapes. The authors show optimal results just by comparing their numerical simulations with the spectral clustering using the Nyström method. The convex hull is applied only at the end of the first clustering phase in which a standard K-means algorithm can determine the correct initialization.

A novel pattern recognition method called "NFPC" has been introduced for training a neurofuzzy classifier by the identification of convex subsets of patterns in the data space [48]. The performed tests ensured the accuracy of the proposed method with respect to other different classifiers. Although, strictly speaking, this is not a clustering algorithm, it uses a convex-set initialization and it incorporates the fuzziness into the decision surfaces to further improve the classification performance.

A fast and reliable distance measure between two convex clusters has been also proposed by using Support Vector Machines (SVM) and a merging algorithm that groups convex clusters obtained from clustering [49]. In addition, a new semisupervised clustering method based on convex hull has been proposed [50]; in its learning stage, by using the patterns whose class is known (a class in this case is also representing a cluster), the method builds the initial convex hull as the boundary for each class. Successively, in the classification stage, the class of any pattern is determined considering the convex hull having the vertex at minimum distance from the pattern.

In [51], the authors propose a dynamic convex hull based clustering algorithm dealing with data appearing sequentially and where clusters are modified by using a combination of vertices of the convex hull containing the dataset. The developed algorithm is assessed at first on some empirical data and then it is applied for the monitoring of a complex system to illustrate its efficiency in real-time applications.

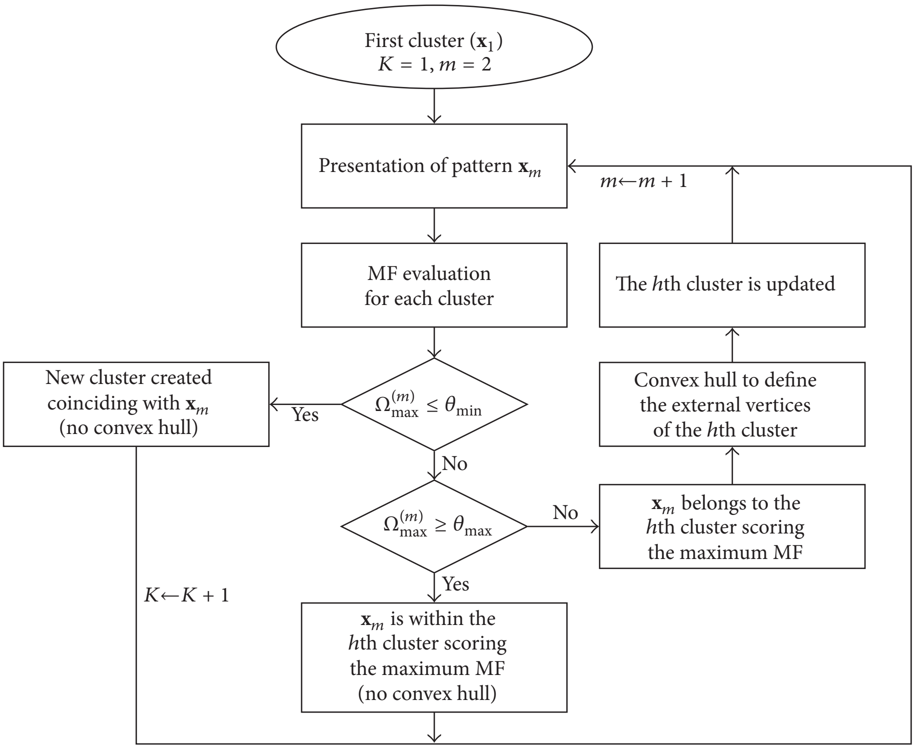

First cluster ($\mathbf{x}_1$)
$K = 1, m = 2$
Presentation of pattern $\mathbf{x}_m$
$m \leftarrow m + 1$
MF evaluation for each cluster
The hth cluster is updated
New cluster created coinciding with $\mathbf{x}_m$ (no convex hull)
Yes
$\Omega_{\max}^{(m)} \le \theta_{\min}$
Convex hull to define the external vertices of the hth cluster
No
$\Omega_{\max}^{(m)} \ge \theta_{\max}$
No
$\mathbf{x}_m$ belongs to the hth cluster scoring the maximum MF
Yes
$K \leftarrow K + 1$
$\mathbf{x}_m$ is within the hth cluster scoring the maximum MF (no convex hull)

Figure 2: Flowchart of the proposed CH algorithm; the evaluation of MFs is illustrated in Section 5 considering the two alternative options (9) or (11).

With respect to early clustering approaches that make use of convex hull computations, we propose an algorithm where the convex hull is not only used at the end of clustering for the geometric data interpretation but also used during the fuzzy data partitioning. Furthermore, we will show that this procedure involves a fuzzy set determination through the convex hull and hence a suited method is proposed for associating an MF to each convex hull-shaped cluster. We propose a brand new approach that makes use of kernel-based membership functions to model clusters, where the convex hull is adopted for the point-to-boundary metric evaluation only. In other words, a pure fuzzy clustering algorithm is obtained, where clusters are fitted to the data distribution also considering the fuzzy membership of patterns to each cluster.

4. The Proposed Algorithm for Fuzzy Clustering

We illustrate in the following the proposed method by considering two fundamental aspects: the illustration of the new clustering algorithm and a discussion of some computational remarks.

Generally, all the algorithms manipulating heterogeneous data originating from different sources need a preprocessing step for data normalization. It is used in order to accommodate every feature of the data space in the range between 0 and 1, so that any metric defined in the data space can be managed with absolute reference values.

Let M be the number of patterns of the dataset $\mathscr{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ and let N be the number of data features; that is, each pattern of the dataset is represented by N-tuple of real numbers:

$$\mathbf{x}_m = \left[x_{m1} \quad x_{m2} \quad \cdots \quad x_{mN}\right], \qquad m = 1, \ldots, M. \tag{1}$$

When the data features have a different nature, either physical or semantic, patterns can be normalized column by column:

$$x_{mj} \longleftarrow \frac{x_{mj} - b_j}{a_j - b_j}, \qquad m = 1, \ldots, M, \; j = 1, \ldots, N, \tag{2}$$

where $a_j = \max\{x_{mj}\}$ and $b_j = \min\{x_{mj}\}$ for $m = 1, \ldots, M$. Alternatively, when some data homogeneity there exists, an affine normalization is usually preferred:

$$x_{mj} \longleftarrow \frac{x_{mj} - b}{a - b}, \qquad m = 1, \ldots, M, \; j = 1, \ldots, N, \tag{3}$$

where $a = \max\{x_{mj}\}$ and $b = \min\{x_{mj}\}$ for $m = 1, \ldots, M$ and $j = 1, \ldots, N$. In the present case we adopted the column-by-column normalization since the used datasets present patterns with heterogeneous features.

The proposed method will be denoted in the following as convex hull (CH) clustering algorithm. Its basic operations are summarized by the flowchart shown in Figure 2. As previously stated, it is a free-clustering sequential algorithm and so the number of clusters is not fixed in advance and it may change during the pattern presentation process.

Let K be the number of clusters currently identified during the operation of the algorithm; the following steps can summarize the detailed operations of the CH algorithm.

(i) The algorithm is initialized considering the first pattern $\mathbf{x}_1$, which is identified as the first cluster. In other words, the first cluster will coincide with the first pattern of the dataset and K is set to 1.

(ii) Successively, the algorithm iterates for each pattern $\mathbf{x}_m$, $m = 2, \ldots, M$, of the dataset. Let $\mathbf{\Omega}(\mathbf{x}_m)$ be the array of MF values of the mth pattern with respect to the K clusters currently determined; that is,

$$\mathbf{\Omega}(\mathbf{x}_m) = [\mu_1(\mathbf{x}_m) \;\; \mu_2(\mathbf{x}_m) \;\; \cdots \;\; \mu_K(\mathbf{x}_m)], \tag{4}$$

where each MF is obtained using the procedures illustrated in Section 5 and considering the convex hull representing each cluster.

Let $\Omega_{\max}^{(m)}$ be the maximum value in $\mathbf{\Omega}(\mathbf{x}_m)$ scored in correspondence with the hth cluster:

$$\Omega_{\max}^{(m)} = \mu_h(\mathbf{x}_m), \qquad h = \arg\max_{r=1,\ldots,K} \{\mu_r(\mathbf{x}_m)\}. \tag{5}$$

Let $\theta_{\min}$ and $\theta_{\max}$ be two parameters that will be successively discussed; three different conditions are successively evaluated in a conditional branch as follows.

(1) $\Omega_{\max}^{(m)} \leq \theta_{\min}$: the algorithm recognizes that no clusters meet the membership criteria for that pattern and hence a new cluster is created that coincides with the current pattern $\mathbf{x}_m$ and no convex hull computation is performed; thereafter $K \leftarrow K + 1$.

(2) $\Omega_{\max}^{(m)} < \theta_{\max}$: the algorithm assigns $\mathbf{x}_m$ to the hth cluster scoring the maximum MF; based on the former condition, we are sure that some MFs in $\mathbf{\Omega}(\mathbf{x}_m)$ will be greater than $\theta_{\min}$. Therefore, the algorithm reestimates the convex hull relevant to the hth cluster and it stores the set of points constituting the new convex hull that represents that cluster in a suited array. The value of K does not change.

(3) $\Omega_{\max}^{(m)} \geq \theta_{\max}$: the algorithm assigns $\mathbf{x}_m$ to the hth cluster scoring the maximum MF. Unlike the previous case, the algorithm does not perform the convex hull, since it is supposed that $\mathbf{x}_m$ is within the boundaries of the hth cluster because of its high MF value. This choice implies a great saving in terms of computational cost and avoids the computation of the convex hull in situations where it might not be necessary.

In addition, this choice aims at ruling the case $\Omega_{\max}^{(m)}$ very close to 1, in which the current pattern likely belongs to the mth cluster with a high degree of membership, either within the cluster vertices or not. In this situation, the algorithm assumes that the pattern has a sufficiently high membership value and so it can be assigned to the cluster without requiring the clusters boundary update. For this reason, it also avoids the fact that clusters may become too large, causing inaccuracies or errors in the overall clustering results. Also in this case the value of K does not change.

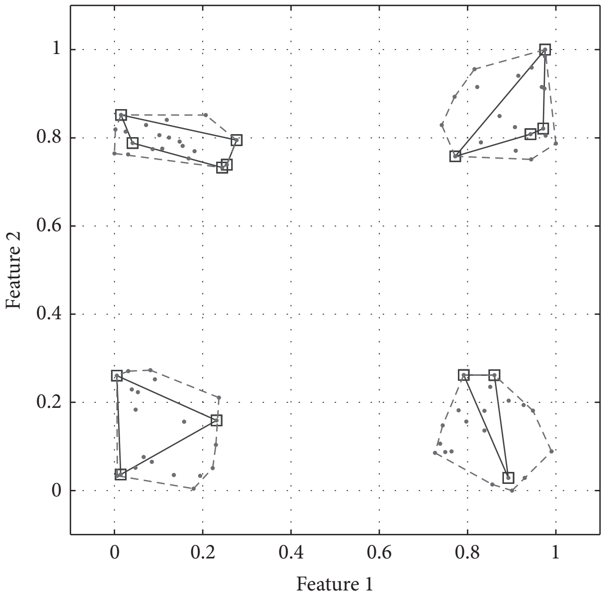


Figure 3: An example of CH algorithm result in a synthetic 2D case: continuous lines represent the convex hull really computed by the CH algorithm in order to build the MF of each fuzzy cluster; dashed lines represent the virtual convex hull that bounds each cluster.

We used a method to set the value of both $\theta_{\min}$ and $\theta_{\max}$ in advance on the basis of the dataset under analysis. Therefore, this procedure can be considered as being integrated within the proposed algorithm with no further optimizations necessary in this regard (alternative methods can be investigated and adopted in future works). Let $\boldsymbol{\sigma}$ be the vector of the standard deviations of the patterns of the dataset evaluated along each column (a feature):

$$\boldsymbol{\sigma} = [\sigma_1 \;\; \sigma_2 \;\; \cdots \;\; \sigma_N]. \tag{6}$$

The values adopted in the following will be

$$\theta_{\min} = \min_{j=1,\ldots,N} \{\sigma_j\}, \qquad \theta_{\max} = 2 \cdot \max_{j=1,\ldots,N} \{\sigma_j\}. \tag{7}$$

An example of the algorithm output is shown in Figure 3 considering a simple 2D dataset consisting of $M = 80$ patterns that can be equally partitioned into $K = 4$ clusters. The real output of the algorithm is represented by the solid lines, which show the convex hull used to build the MF of each fuzzy cluster as explained successively. Obviously, the subdivision of patterns among the various clusters can be represented by the dashed lines, which show the plain convex hull of each cluster though not really computed by the CH algorithm. In fact, we outline the fact that not all the points of a cluster may be used to calculate the convex hull and the related MF of a cluster, thanks to the last condition previously explained that allowed assigning a pattern to a cluster without updating the convex hull.

5. MF Evaluation of Convex Hull-Shaped Clusters

When clusters in a dataset have particular geometries or, more in general, when one wants to disengage from specific geometrical structures (such as hypercubes, hyperspheres, and regular polytopes), it is useful and appropriate to rely on more flexible and, at the same time, computationally affordable MFs [52]. Since the actual structure of clusters is possibly irregular, the MF should be based on a point-to-boundary distance of patterns from clusters rather than on a point-to-centroid based metric. The shape of the resulting MF will follow in this manner the particular form on which a cluster is structured.

In this paper, the convex hull is adopted as a geometrical model for fuzzy clusters. As previously illustrated, each convex hull is represented by a number L of vertices corresponding to some patterns belonging to that cluster. Each convex hull will be associated with an MF whose trend is inversely related to the distance of a pattern from its boundaries; looking at Figure 3 for the 2D case, the distances are considered from the polygon bounded by L straight sides between the vertices of the convex hull. To achieve this goal for any dimension, that is, for any number of features of the dataset, the basic idea successively illustrated is to use a mixture of $L + 1$ kernel functions centered at each vertex of the convex hull and at the cluster's centroid as well.

Considering the computational cost, the convex hull evaluation, in particular the Quickhull algorithm, has $O(f_n)$ complexity, where f_n is the maximum number of facets of a convex hull with n vertices and n is the number of processed points [53]. In case of datasets with a high number of dimensions, the convex hull approach suffers from a relatively high computational speed; in such cases, the computational cost can be overcome by using parallel algorithms implemented on multicore processors [54] and GPU [55], which also minimize the impact of irregular data. This choice can improve the performance in the case of simple 2D [56, 57] and 3D [58, 59] datasets, but also in the case of generic N-dimensional data [60, 61]. Finally, the use of convex hull exhibits a good flexibility that combines computational performance with good spatial representation, since the convex hull is generally more compact, in terms of spatial occupation, when compared to the volume of hyperboxes.

In the following, we illustrate two possible procedures for the MF computation of convex hull-shaped clusters, using either Gaussian or cone-shaped kernel functions.

5.1. Gaussian-Based Kernels. This method exploits the superposition of an appropriate number of univariate (isotropic) Gaussian kernels to associate the MF with the convex hull. Let $L \times N$ be a matrix $\mathbf{V}$, where N is the number of features of the data space and L is the number of vertices of the convex hull representing the cluster:

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_L \end{bmatrix} = \begin{bmatrix} v_{11} & \cdots & v_{1N} \\ \vdots & & \vdots \\ v_{L1} & \cdots & v_{LN} \end{bmatrix}. \tag{8}$$

Let $\mathbf{x}$ be the pattern whose MF to the cluster must be computed. By using the Gaussian method, the MF takes the form:

$$\mu^{(\text{gauss})}(\mathbf{x}) = \exp\left\{-\frac{\gamma^2}{2\delta} d_2^2(\mathbf{x}, \mathbf{c})\right\} + \sum_{i=1}^{L} \exp\left\{-\frac{\gamma^2}{2\delta} d_2^2(\mathbf{x}, \mathbf{v}_i)\right\}, \tag{9}$$

where $\mathbf{c} = [c_1, c_2, \ldots, c_N]$ is the cluster centroid, $d_2^2(\mathbf{x}, \mathbf{c})$ is the squared point-to-centroid Euclidean distance, $d_2^2(\mathbf{x}, \mathbf{v}_i)$ is the squared point-to-ith-vertex Euclidean distance, and δ is the maximum distance that can occur between two patterns. This value is not an external parameter, which requires an initial set-up; rather it depends on the number of features according to the following expression:

$$\delta = \sqrt{N}. \tag{10}$$

The variance of each Gaussian kernel is set to a fixed value equal to $1/\gamma^2$. As shown in Figure 4, the value of γ determines how quickly the function decreases and hence it is related to the fuzziness of the resulting MF; the higher the value of γ is, the faster the function goes to zero.

The proposed CH algorithm where the MFs in (4) are computed by using (9) will be denoted in the following as convex hull with Gaussian-based kernels (CH-GBK).

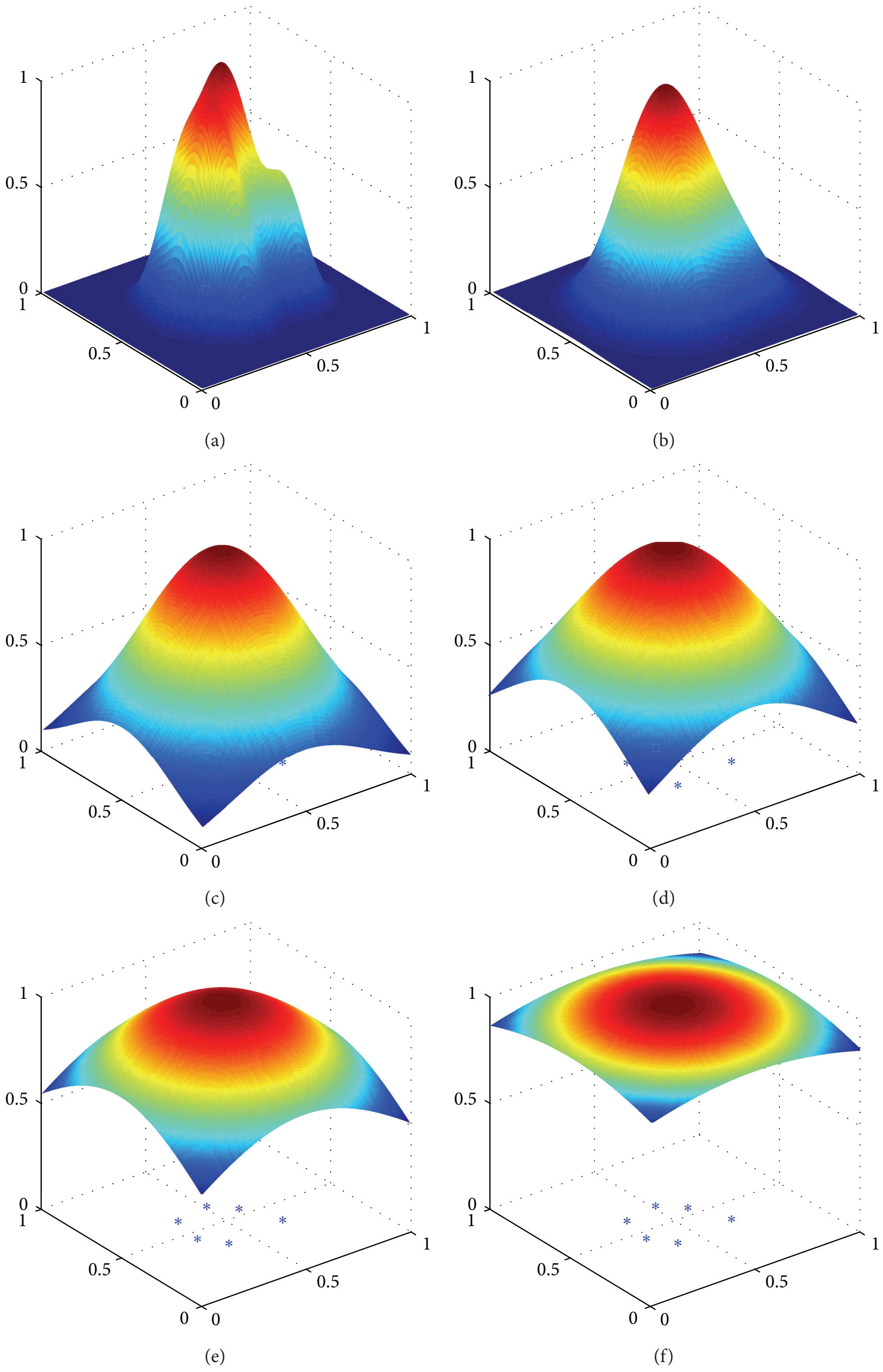
5.2. Cone-Based Kernels. This method uses cone-shaped kernel functions to build the MF of the convex hull according to the following expression:

$$\mu^{(\text{cone})}(\mathbf{x}) = \max\left[0, 1 - \frac{\gamma}{\delta} d_2(\mathbf{x}, \mathbf{c})\right] + \sum_{i=1}^{L} \max\left[0, 1 - \frac{\gamma}{\delta} d_2(\mathbf{x}, \mathbf{v}_i)\right], \tag{11}$$

where $d_2(\mathbf{x}, \mathbf{c})$ is the point-to-centroid Euclidean distance and $d_2(\mathbf{x}, \mathbf{v}_i)$ is the point-to-ith-vertex Euclidean distance.

The proposed CH algorithm where the MFs in (4) are computed by using (11) will be denoted in the following as convex hull with cone-based kernels (CH-CBK).

In either Gaussian-based or cone-based MFs, the γ parameter defines how quickly the function tends to zero: the higher the value of γ is, the faster the function goes to zero, as shown, for example, in Figure 5, by the graphical representation of the cone-based MF using different values of



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 4: 2D Gaussian MFs with different values of γ: (a) $\gamma = 15$; (b) $\gamma = 10$; (c) $\gamma = 4$; (d) $\gamma = 3$; (e) $\gamma = 2$; (f) $\gamma = 1$. The patterns determining the convex hull are plotted as asterisks in the data plane.

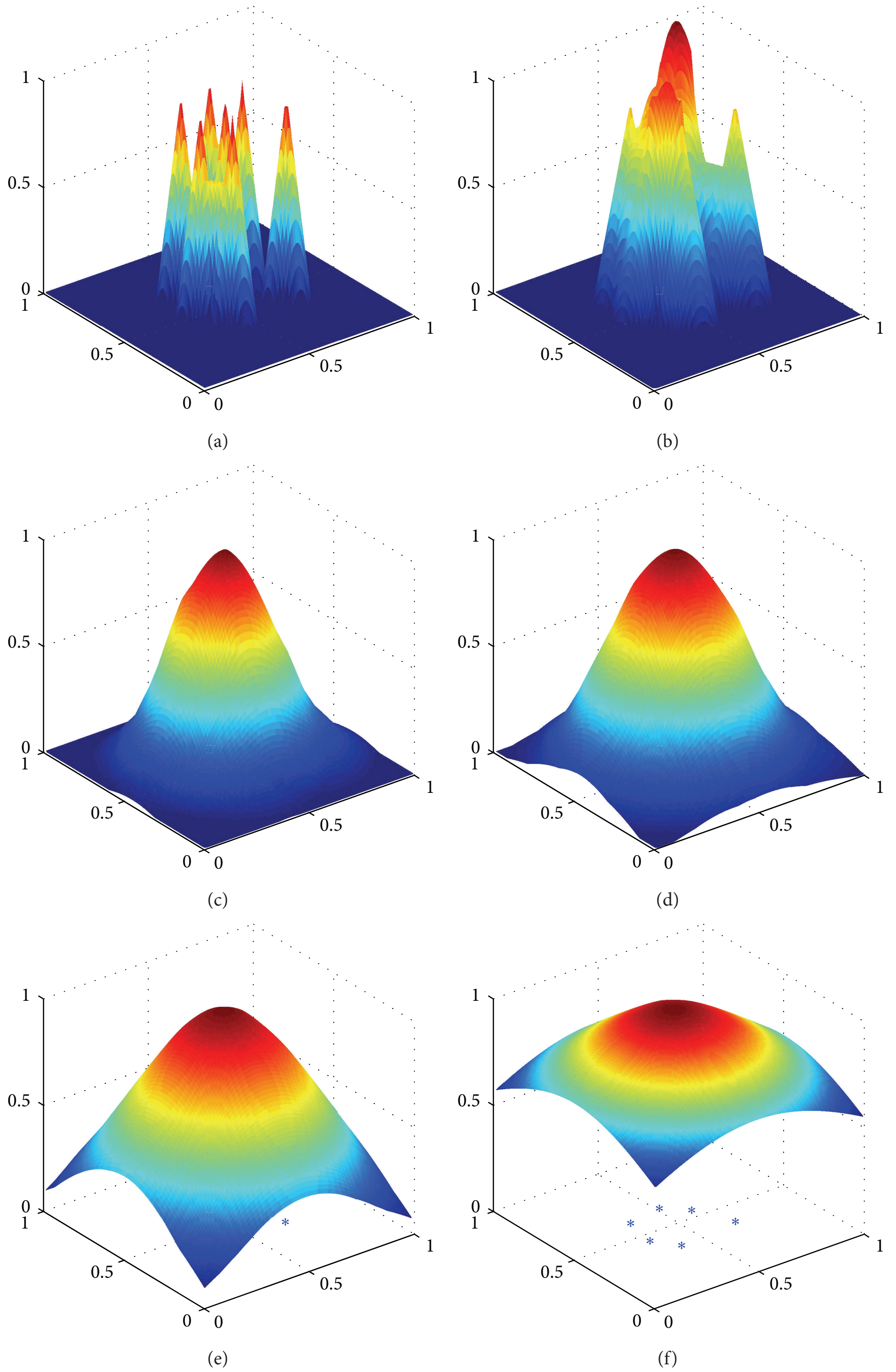


FIGURE 5: 2D cone-based MFs with different values of γ: (a) $\gamma = 15$; (b) $\gamma = 10$; (c) $\gamma = 4$; (d) $\gamma = 3$; (e) $\gamma = 2$; (f) $\gamma = 1$. The patterns determining the convex hull are plotted as asterisks in the data plane.

γ. For this reason, the γ parameter determines the fuzziness of the related MF.

Cone-shaped MFs have some similarities and some differences with respect to the Gaussian ones. Differently from Gaussian kernels, this method uses functions that effectively reach the zero value of the MF. As the Gaussian one, the cone-shaped method uses the superposition of $L + 1$ functions: L functions are placed at the vertices of the convex hull and one is placed on its centroid. The latter is useful in both cases to assign the right relevance to the cluster centroid and to fill the gap that may exist in the convex hull around the cluster centroid, since the other functions are placed at the vertices of the convex hull.

We will adopt isotropic Gaussian kernels having the same variance for each vertex and cone-shaped kernels having a hyperspherical (isotropic) section with the same radius for each vertex. Regarding this aspect, some preliminary tests were carried out for both Gaussian and cone-based kernels to ascertain the necessity of computing more parameters for each MF [62]. These tests proved that isotropic functions, with a predetermined width used for each vertex, are able to obtain good results in terms of performance and efficiency.

The two proposed options for MF evaluation, that is, Gaussian-based and cone-based, are considered in this paper since they can perform differently in terms of efficiency or accuracy. In fact, in [62] we performed specific tests in order to compare the performance of such methods. The results of these tests proved that cone method is slightly faster than the Gaussian one, which in turn is more accurate in many practical cases. This is also confirmed by the following experimental results.

6. Experimental Results

The performances of the proposed CH algorithm are validated through the analysis of several clustering benchmarks. We present some experimental results that are representative of a general behaviour. Several datasets having a different number of features and clusters are considered: Hepta, [63], Iris, User Knowledge Modeling (UKM), and Seed [64].

Although some datasets, like Iris and UKM, should be properly used for classification benchmarks, they are often adopted also for clustering. Generally, patterns of the same class may be grouped in several clusters of a dataset (i.e., different regions of the data space) or else the same cluster may contain patterns of different classes. In case of Iris, for example, many clustering algorithms are able to identify only two clusters given that patterns of two classes are overlapping in the input space of the considered dataset.

We compared our approach with several clustering algorithms characterized by different taxonomies: K-means (partitional-batch-crisp) [65], FCM (partitional-batch-fuzzy) [66], Min-Max (partitional-sequential-fuzzy), and Clusterdata (hierarchical-batch-crisp) in the MATLAB (ver. R2013a) environment. Any clustering algorithm has a dependence on one or more critical parameters that affect its overall performance. For this reason, a sound comparison between different algorithms should take into account also the bootstrap procedures to set up the related parameters. For instance, the number of clusters in K-means and FCM (i.e., K and c, resp.) should be determined by using a cluster validity index as the ones mentioned in Section 1. In the same way, clusters and the related number must be selected within the hierarchy generated by Clusterdata. FCM, Min-Max, and the proposed CH algorithm need a suited choice of the fuzzification parameter for MFs. Min-Max and CH algorithms adopt a sequential procedure and so the performance should be averaged over different permutations of the pattern presentation order (unless differently justified by the specific nature of the dataset). Similarly, the performance should be averaged over different centroid initializations for K-means and FCM.

In addition, the determination of optimal parameters is critical for online algorithms given that different operative frameworks may be considered as follows:

(i) A training/tuning set is used to find the optimal values of parameters, possibly using cluster validity procedures, and then the algorithm is used for online clustering of different test sets (hopefully generated by the same random process as the one of the training/tuning set).

(ii) The same dataset is firstly used to find the optimal parameters and successively it is clustered by using such parameters. In this case, the online algorithm is inserted into a totally batch, iterative procedure; therefore its use is justified only by more accurate performances or faster computational times with respect to iterative clustering algorithms.

(iii) The parameters are fixed in advance by relying on some a priori hypotheses; successively the algorithm is used for pure online clustering. In this case, the values of parameters may be adjusted adaptively, for instance, if the specific application requires a data analysis by epochs or it is a big data problem where errors due to the initial guess marginally affect the overall clustering performance.

In the following, we consider the measure of the error rate in terms of pattern assignment for known benchmarks. Precisely, these numerical tests focus on the number of errors obtained by the CH algorithm compared with the aforementioned clustering algorithms. A study on cluster validity procedures is out of the scope of this paper; consequently, in order to have a broad context for the analysis and consider algorithms of different nature, in the following, we will assume that all the parameters of these algorithms are ideally obtained through a suitable procedure, to generate the right number of clusters. In fact, although clustering is an unsupervised learning problem, we are using some reference datasets for benchmarking and hence we know the right number of clusters and the true label of each pattern.

Consequently, the value of K is suitably fixed in advance for K-means as well as the value of c for FCM, using in this case a default fuzzification parameter $m = 2$ for its well-known MF. Moreover, since the result of these algorithms depends on a centroids initialization, we will take as whole

Table 1: Mean error rate (%) of pattern assignments averaged over 100 different runs of the algorithms.

Algorithm	Hepta (3D)	Iris (4D)	UKM (5D)	Seed (7D)
CH-GBK	0.00	16.53	52.16	21.84
CH-CBK	0.00	16.74	51.11	22.07
K-means	23.69	18.23	52.62	10.95
FCM	0.20	10.67	57.45	10.00
Min-Max	3.33	24.26	67.11	32.05
Clusterdata	0.00	34.00	75.19	66.67

Table 2: Best error rate (%) of pattern assignments obtained over 100 different runs of the algorithms.

Algorithm	Hepta (3D)	Iris (4D)	UKM (5D)	Seed (7D)
CH-GBK	0.00	4.67	40.20	9.05
CH-CBK	0.00	6.67	35.48	8.10
K-means	0.00	11.33	44.17	10.95
FCM	0.00	10.67	49.63	10.00
Min-Max	0.00	6.00	45.16	12.38
Clusterdata	0.00	34.00	75.19	66.67

result the average over 100 different initializations. For Clusterdata algorithm, we choose in the generated hierarchy the solution containing the right number of clusters.

For instance, the γ parameter controls the fuzziness of MFs in the original Simpson's algorithm and a threshold is also necessary to compare the MF values and control the hyperbox expansion process. For Min-Max and CH algorithms, we randomly change the presentation order of patterns for 100 times; for each sorting we consider the value of γ that yields the right number of clusters and then we take as whole result the average over the 100 different sortings. For Min-Max, also the maximum size θ of a hyperbox is considered [67], which is another critical parameter: it is varied (for each sorting of patterns) in the range from 0.1 to 0.9 with 0.1 steps, in order to obtain the best choice of both γ and θ.

The results in terms of mean error rate of pattern assignment, which is the percentage of patterns with respect to the cardinality of the dataset not correctly assigned to the right cluster, possibly averaged over different runs of that algorithm, are shown in Table 1. For each dataset, both the CH-GBK and the CH-CBK methods are able to achieve a performance comparable with the FCM algorithm and even a better performance than K-means, Min-Max, and Clusterdata algorithms. We remark that the same performance is obtained with respect to FCM although CH is an online free-clustering algorithm that is faster than the iterative FCM, since it analyses the data only once and it does not suffer from the initialization guess of centroids, while being robust against the pattern presentation order.

Another result can be shown in Table 2, which illustrates the minimum (e.g., the best) error rate (%) obtained after 100 different runs of the above introduced algorithms for each of the four datasets. We outline the fact that both CH-GKB and CH-CKB algorithms are able to obtain the best performance in terms of patterns assignment.

As discussed in the paper, the performance of every algorithm depends on at least one initialization parameter. A more interesting test has been carried out for comparing the performance of the proposed approach versus the popular Min-Max algorithm that, as discussed above, is online/sequential algorithm as well and it also depends on the order of pattern presentation. In Table 3 we report the results obtained after running CH-GKB, CH-CKB, and Min-Max algorithms for 100 runs and counting the times in which an algorithm is able to obtain the minimum error rate for each data sorting. Both CH-GKB and CH-CKB achieve the best performance in terms of number of runs in which they are able to obtain the minimum error rate (we remark that the sum in each column may be greater than 100% because CH and Min-Max can obtain the same error rate that is the same best result for a given run).

7. Conclusions

In this paper, we propose a new fuzzy clustering algorithm with two different variations both based on a new metric to calculate the MF of fuzzy sets representing the clusters. The algorithm is intended to eliminate as much as possible the dependence of clustering results on the use of simple and predetermined geometrical models for clusters. We solve this problem through the computation of a suited convex hull representing the cluster. We also reduced the dependence on critical thresholds and parameters, which often leads to wrong computations of the number of clusters and wrong data partitions.

The experimental results show that the proposed CH approach is able to achieve a comparable performance with respect to other well-known clustering algorithms, introducing some desirable features thanks to the use of a sequential and free-clustering approach whose computational complexity is controlled. The only critical parameter to be optimized is the fuzziness γ of the adopted MFs. The experimental tests confirm that our algorithm is quite stable in a wide range of γ, considering any heuristic procedure to find the optimal value of this parameter.

Table 3: Number of runs, over 100 different trials, for which the best performance is obtained.

Algorithm	Hepta (3D)	Iris (4D)	UKM (5D)	Seed (7D)
CH-GBK	100	86	85	98
CH-CBK	100	86	85	98
Min-Max	81	14	11	3

In the future, the CH algorithm could be considered also for iterative and hierarchical clustering procedures and it could be enhanced by using suited techniques to generalize the shape of clusters to nonconvex structures as well. In fact, the proposed approach for the MF computation can be applied independently of the use of the convex hull, for example, considering a set of vertices of a concave polytope family, which could be more suited to fit semicircle, curvilinear, and other irregular structures, although they may appear in peculiar datasets only.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *Journal of Cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.

[2] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2001.

[3] R. B. Zadeh and S. Ben-David, "A uniqueness theorem for clustering," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09)*, pp. 639–646, June 2009.

[4] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[5] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[6] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ, USA, 1988.

[7] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data," *Data Mining and Knowledge Discovery*, vol. 11, no. 1, pp. 5–33, 2005.

[8] R. Parisi, A. Cirillo, M. Panella, and A. Uncini, "Source localization in reverberant environments by consistent peak selection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 1, pp. I37–I40, Honolulu, Hawaii, USA, April 2007.

[9] M. Panella and G. Martinelli, "Binary neuro-fuzzy classifiers trained by nonlinear quantum circuits," in *Applications of Fuzzy Sets Theory*, G. P. F. Masulli and S. Mitra, Eds., vol. 4578 of *Lecture Notes in Computer Science*, pp. 237–244, Springer, Heidelberg, Germany, 2007.

[10] R. H. Sheikh, M. M. Raghuwanshi, and A. N. Jaiswal, "Genetic algorithm based clustering: a survey," in *Proceedings of the 1st International Conference on Emerging Trends in Engineering and Technology (ICETET '08)*, pp. 314–319, July 2008.

[11] G. Kootstra, J. Ypma, and B. de Boer, "Active exploration and keypoint clustering for object recognition," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '08)*, pp. 1005–1010, May 2008.

[12] M. Panella and G. Martinelli, "Neurofuzzy networks with nonlinear quantum learning," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 3, pp. 698–710, 2009.

[13] M. Panella, F. Barcellona, and R. L. D'Ecclesia, "Forecasting energy commodity prices using neural networks," *Advances in Decision Sciences*, vol. 2012, Article ID 289810, 26 pages, 2012.

[14] K. Simiński, "Neuro-fuzzy system with weighted attributes," *Soft Computing*, vol. 18, no. 2, pp. 285–297, 2014.

[15] P. Gajdoš and V. Snášel, "A new FCA algorithm enabling analyzing of complex and dynamic data sets," *Soft Computing*, vol. 18, no. 4, pp. 683–694, 2014.

[16] A. Proietti, M. Panella, F. Leccese, and E. Svezia, "Dust detection and analysis in museum environment based on pattern recognition," *Measurement*, vol. 66, pp. 62–72, 2015.

[17] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, Elsevier, Burlington, Mass, USA, 4th edition, 2009.

[18] L. A. Zadeh, "Fuzzy sets and their application to pattern classification and clustering analysis," in *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*, G. J. Klir and B. Yuan, Eds., pp. 355–393, World Scientific Publishing, River Edge, NJ, USA, 1996.

[19] D. Dovžan and I. Škrjanc, "Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes," *ISA Transactions*, vol. 50, no. 2, pp. 159–169, 2011.

[20] D. Dovzan and I. Skrjanc, "Recursive clustering based on a Gustafson-Kessel algorithm," *Evolving Systems*, vol. 2, no. 1, pp. 15–24, 2011.

[21] P. P. Angelov and D. P. Filev, "An approach to online identification of takagi-sugeno fuzzy models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 484–498, 2004.

[22] M.-H. Masson and T. Denux, "ECM: an evidential version of the fuzzy c-means algorithm," *Pattern Recognition*, vol. 41, no. 4, pp. 1384–1397, 2008.

[23] N. K. Kasabov and Q. Song, "DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, 2002.

[24] R. J. Hathaway and J. C. Bezdek, "Recent convergence results for the fuzzy *c*-means clustering algorithms," *Journal of Classification*, vol. 5, no. 2, pp. 237–247, 1988.

[25] M. Panella, "A hierarchical procedure for the synthesis of ANFIS networks," *Advances in Fuzzy Systems*, vol. 2012, Article ID 491237, 12 pages, 2012.

[26] M. Panella, L. Liparulo, and A. Proietti, "A higher-order fuzzy neural network for modeling financial time series," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '14)*, pp. 3066–3073, Beijing, China, July 2014.

[27] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 370–379, 1995.

[28] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, 2002.

[29] W. Wang and Y. Zhang, "On fuzzy cluster validity indices," *Fuzzy Sets and Systems*, vol. 158, no. 19, pp. 2095–2117, 2007.

[30] P. K. Simpson, "Fuzzy min-max neural networks. Part 2. Clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, pp. 32–45, 1993.

[31] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224–227, 1978.

[32] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 28, no. 3, pp. 301–315, 1998.

[33] W. Pedrycz and G. Vulkovich, "Fuzzy clustering with supervision," *Pattern Recognition*, vol. 37, no. 7, pp. 1339–1349, 2004.

[34] G. Beliakov and M. King, "Fuzzy c-means density based clustering using data induced metric," in *Artificial Intelligence and Applications*, pp. 234–239, 2005.

[35] J.-P. Mei and L. Chen, "Fuzzy clustering with weighted medoids for relational data," *Pattern Recognition*, vol. 43, no. 5, pp. 1964–1974, 2010.

[36] M. Maisto, M. Panella, L. Liparulo, and A. Proietti, "An accurate algorithm for the identification of fingertips using an RGB-D camera," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 2, pp. 272–283, 2013.

[37] J.-D. Boissonnat and M. Teillaud, "On the randomized construction of the delaunay tree," *Theoretical Computer Science*, vol. 112, no. 2, pp. 339–354, 1993.

[38] K. Q. Brown, "Voronoi diagrams from convex hulls," *Information Processing Letters*, vol. 9, no. 5, pp. 223–228, 1979.

[39] F. P. Preparata and S. J. Hong, "Convex hulls of finite sets of points in two and three dimensions," *Communications of the ACM*, vol. 20, no. 2, pp. 87–93, 1977.

[40] D. R. Chand and S. S. Kapur, "An algorithm for convex polytopes," *Journal of the Association for Computing Machinery*, vol. 17, pp. 78–86, 1970.

[41] T. H. Cormen, C. E. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Mass, USA, 3rd edition, 2009.

[42] J. Sklansky, "Measuring concavity on a rectangular mosaic," *IEEE Transactions on Computers C*, vol. 21, no. 12, pp. 1355–1364, 1972.

[43] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996.

[44] F. P. Preparata and M. I. Shamos, *Computational Geometry—An Introduction*, Springer, New York, NY, USA, 1985.

[45] I. H. Suh, J.-H. Kim, and F. C.-H. Rhee, "Convex-set-based fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 3, pp. 271–285, 1999.

[46] R. Keshari and A. Sinha, "An intelligent fuzzy convex hull based clustering approach," *International Journal of Computer Applications*, vol. 100, no. 8, pp. 38–41, 2014.

[47] M. B. Abubaker and H. M. Hamad, "K-means-based convex hull triangulation clustering algorithm," *Research Notes in Information Science*, vol. 9, no. 1, pp. 19–29, 2012.

[48] W. M. Grohman and A. P. Dhawan, "Fuzzy convex set-based pattern classification for analysis of mammographic microcalcifications," *Pattern Recognition*, vol. 34, no. 7, pp. 1469–1482, 2001.

[49] F. C.-H. Rhee and B.-I. Choi, "A convex cluster merging algorithm using support vector machines," in *Proceedings of the IEEE International conference on Fuzzy Systems*, vol. 2, pp. 892–895, May 2003.

[50] W. Guiliang, D. Xiangqian, L. Bo, and X. Peiyong, "A new semi-supervised clustering mehtod based on Convex Hull," in *Proceedings of the 4th International Conference on Intelligent Computation Technology and Automation (ICICTA '11)*, vol. 2, pp. 1036–1038, March 2011.

[51] F. Theljani, K. Laabidi, S. Zidi, and M. Ksouri, "Convex hull based clustering algorithm," *International Journal of Artificial Intelligence*, vol. 10, article S13, 2013.

[52] L. Liparulo, A. Proietti, and M. Panella, "Improved online fuzzy clustering based on unconstrained kernels," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '15)*, IEEE, Istanbul, Turkey, August 2015.

[53] V. Klee, "Convex polytopes and linear programming," in *Proceedings of the IBM Scientific Computing Symposium*, pp. 123–158, January 1966.

[54] M. Nakagawa, D. Man, Y. Ito, and K. Nakano, "A simple parallel convex hulls algorithm for sorted points and the performance evaluation on the multicore processors," in *Proceedings of the International Conference on Parallel and Distributed Computing, Applications and Technologies*, pp. 506–511, December 2009.

[55] S. Srungarapu, D. P. Reddy, K. Kothapalli, and P. J. Narayanan, "Fast two dimensional convex hull on the GPU," in *Proceedings of the 25th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA '11)*, pp. 7–12, March 2011.

[56] R. Miller and Q. F. Stout, "Efficient parallel convex hull algorithms," *IEEE Transactions on Computers*, vol. 37, no. 12, pp. 1605–1618, 1988.

[57] J. Zhou, X. Deng, and P. Dymond, "A 2-D parallel convex hull algorithm with optimal communication phases," in *Proceedings of the 11th International Parallel Processing Symposium (IPPS '97)*, pp. 596–602, April 1997.

[58] T.-T. Cao, A. Nanjappa, M. Gao, and T.-S. Tan, "A gpu accelerated algorithm for 3D Delaunay triangulation," in *Proceedings of the Symposium on Interactive 3D Graphics*, pp. 47–54, March 2014.

[59] A. Stein, E. Geva, and J. El-Sana, "CudaHull: fast parallel 3D convex hull on the GPU," *Computers and Graphics*, vol. 36, no. 4, pp. 265–271, 2012.

[60] Y. Leung, J.-S. Zhang, and Z.-B. Xu, "Neural networks for convex hull computation," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 601–611, 1997.

[61] N. Amato, M. Goodrich, and E. Ramos, "Parallel algorithms for higher-dimensional convex hulls," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 683–694, Santa Fe, NM, USA, 1994.

[62] L. Liparulo, A. Proietti, and M. Panella, "Fuzzy membership functions based on point-to-polygon distance evaluation," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '13)*, pp. 1–8, Hyderabad, India, July 2013.

[63] A. Ultsch, "Clustering wih SOM: U^*C," in *Proceedings of the 5th Workshop on Self-Organizing Maps*, vol. 2, pp. 75–82, September 2005.

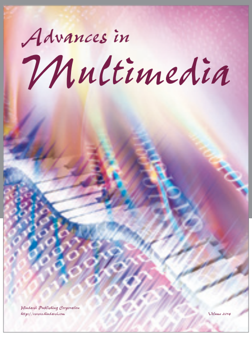
[64] K. Bache and M. Lichman, *UCI Machine Learning Repository*, 2013.

[65] J. B. MacQueen, "Some methods for classification and analysis of multi-variate observations," in *Proceedings of the 5th Berkeley*

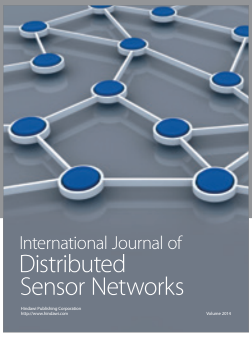
Symposium on Mathematical Statistics and Probability, L. M. L. Cam and J. Neyman, Eds., vol. 1, pp. 281–297, University of California Press, Berkeley, Calif, USA, 1967.

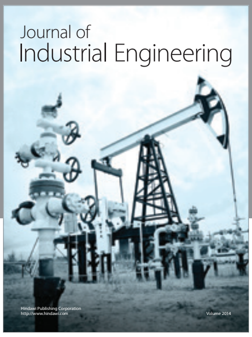
[66] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

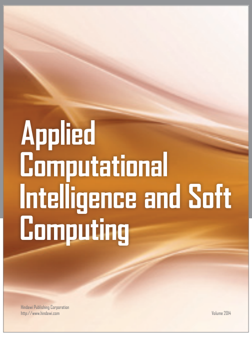
[67] P. K. Simpson, "Fuzzy min-max neural networks—I: classification," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 776–786, 1992.

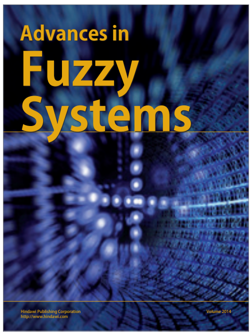

Advances in
Multimedia

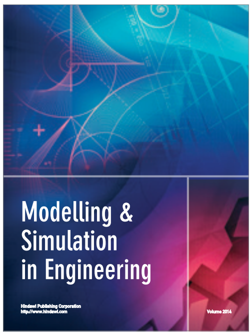
The Scientific World Journal

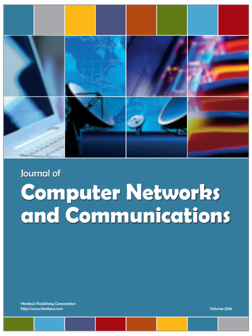

International Journal of
Distributed Sensor Networks

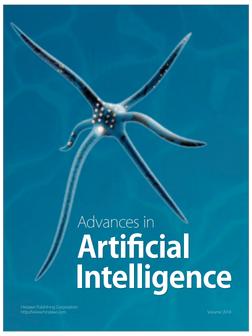

Journal of
Industrial Engineering

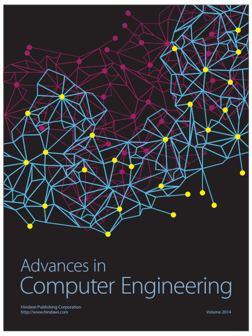

Applied Computational Intelligence and Soft Computing


Advances in
Fuzzy Systems

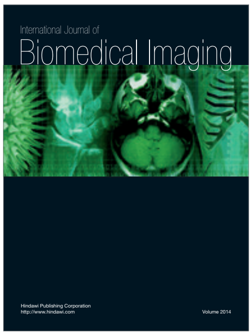

Modelling & Simulation in Engineering

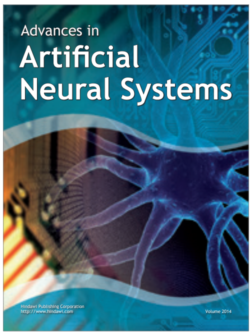

Journal of
Computer Networks and Communications

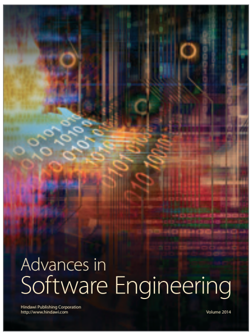

Advances in
Artificial Intelligence


Advances in
Computer Engineering

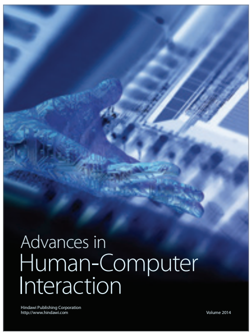
International Journal of
Computer Games Technology

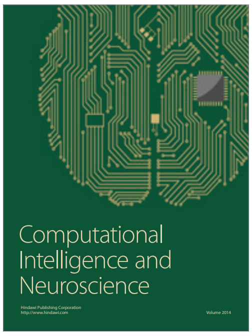

International Journal of
Biomedical Imaging


Advances in
Artificial Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer Interaction


Computational Intelligence and Neuroscience

International Journal of
Reconfigurable Computing

Journal of
Electrical and Computer Engineering