mathematics

MDPI

Article

Fuzzy Clustering Methods with Rényi Relative Entropy and Cluster Size

Javier Bonilla [1,2,*], Daniel Vélez [1], Javier Montero [1] and J. Tinguaro Rodríguez [1]

[1] Department of Statistics and Operations Research, Universidad Complutense de Madrid, 28040 Madrid, Spain; danielvelezserrano@mat.ucm.es (D.V.); monty@mat.ucm.es (J.M.); jtrodrig@mat.ucm.es (J.T.R.)
[2] Comisión Nacional del Mercado de Valores, 28006 Madrid, Spain
* Correspondence: javierlb@ucm.es

Abstract: In the last two decades, information entropy measures have been relevantly applied in fuzzy clustering problems in order to regularize solutions by avoiding the formation of partitions with excessively overlapping clusters. Following this idea, relative entropy or divergence measures have been similarly applied, particularly to enable that kind of entropy-based regularization to also take into account, as well as interact with, cluster size variables. Particularly, since Rényi divergence generalizes several other divergence measures, its application in fuzzy clustering seems promising for devising more general and potentially more effective methods. However, previous works making use of either Rényi entropy or divergence in fuzzy clustering, respectively, have not considered cluster sizes (thus applying regularization in terms of entropy, not divergence) or employed divergence without a regularization purpose. Then, the main contribution of this work is the introduction of a new regularization term based on Rényi relative entropy between membership degrees and observation ratios per cluster to penalize overlapping solutions in fuzzy clustering analysis. Specifically, such Rényi divergence-based term is added to the variance-based Fuzzy C-means objective function when allowing cluster sizes. This then leads to the development of two new fuzzy clustering methods exhibiting Rényi divergence-based regularization, the second one extending the first by considering a Gaussian kernel metric instead of the Euclidean distance. Iterative expressions for these methods are derived through the explicit application of Lagrange multipliers. An interesting feature of these expressions is that the proposed methods seem to take advantage of a greater amount of information in the updating steps for membership degrees and observations ratios per cluster. Finally, an extensive computational study is presented showing the feasibility and comparatively good performance of the proposed methods.

Keywords: fuzzy clustering; entropy; relative entropy; Rényi entropy; differential evolution algorithm; Gaussian kernel

Citation: Bonilla, J.; Vélez, D.; Montero, J.; Rodríguez, J.T. Fuzzy Clustering Methods with Rényi Relative Entropy and Cluster Size. *Mathematics* **2021**, *9*, 1423. https://doi.org/10.3390/math9121423

Academic Editor: Hsien-Chung Wu

Received: 14 May 2021
Accepted: 15 June 2021
Published: 18 June 2021

1. Introduction

Clustering analysis [1–4], as a part of the multivariate analysis of data, is a set of unsupervised methods whose objective is to group objects or observations taking into account two main requirements: (1) objects in a group or cluster should be similar or homogeneous; and (2) objects from different groups should be different or heterogeneous.

Historically, clustering methods have been divided into two classes attending to the output they provide: A first class is that of hierarchical methods [5,6], which proceed by either agglomerating or splitting previous clusters, respectively, ending or starting the clustering process with all objects grouped in a single cluster. Thus, the main feature of hierarchical methods is that they provide a collection of interrelated partitions of the data, each partition or set of clusters being composed of a different number of clusters between 1 and N, where N is the number of objects or observations being clustered. This leaves the user with the task of choosing a single partition exhibiting an adequate number of clusters

for the problem being considered. The second class of clustering methods is that of non-hierarchical procedures, which provide a single partition with a number of clusters either determined by the algorithm or prespecified by the user as an input to the method. Several general clustering strategies have been proposed within this non-hierarchical class, such as prototype-based [7–12], density-based [13,14], graph-based [15–18], or grid-based [19,20] methods. Prototype-based methods typically assume the number K of clusters is provided by the user, randomly construct an initial representative or prototype for each of the K clusters, and proceed by modifying these prototypes in order to optimize an objective function, which usually measures intra-cluster variance in terms of a metric such as the Euclidean distance.

The most extended instance of prototype-based method is the K-means algorithm, first proposed by Forgy [7] and further developed by MacQueen [8]. As for most prototype-based methods, the basis of the K-means algorithm is a two-step iterative procedure, by which each object is assigned to the cluster associated with the closest prototype (i.e., from a geometric point of view, clusters can be seen as the Voronai cells of the given prototypes, see [21,22]), and prototypes are then recalculated from the observations that belong to each corresponding cluster. The K-means specific prototypes are referred to as cluster centroids and are obtained as the mean vectors of observations. The method can thus be seen as a heuristic procedure trying to minimize an objective function given by the sum of squared Euclidean distances between the observations and the respective centroids. As shown in [23], the K-means algorithm has a superlineal convergence, and it can be described as a gradient descent algorithm.

Several works (e.g., [24–27]) discuss the relevance of the K-means algorithm and its properties, problems, and modifications. Particularly, its main positive features are as follows [25,27,28]:

- Its clustering strategy is not difficult to understand for non-experts, and it can be easily implemented with a low computational cost in a modular fashion that allows modifying specific parts of the method.
- It presents a linear complexity, in contrast with hierarchical clustering methods, which present at least quadratic complexity.
- It is invariant under different dataset orderings, and it always converges at quadratic rate.

On the other hand, the K-means algorithm also presents some known drawbacks:

- It assumes some a priori knowledge of the data, since the parameter K that determines the number of clusters to build has to be specified before the algorithm starts to work.
- It converges to local minima of the error function, not necessarily finding the global minimum [29], and thus several initializations may be needed in order to obtain a good solution.
- The method of selection of the initial seeds influences the convergence to local minima of the error function, as well as the final clusters. Several initialization methods have been proposed (see [25] for a review on this topic), such us random selection of initial points [8] or the K-means++ method [30], which selects the first seed randomly and then chooses the remaining seeds with a probability inversely proportional to distance.
- Although it is the most extended variant, the usage of the Euclidean metric in the objective function of the method presents two relevant problems: sensitivity to outliers and a certain bias towards producing hyperspherical clusters. Consequently, the method finds difficulties in the presence of ellipsoidal clusters and/or noisy observations. Usage of other non-Euclidean metrics can alleviate these handicaps [24].

The introduction of the notions and tools of fuzzy set theory [31] in cluster analysis was aimed at relaxing the constraints imposing that each observation has to completely belong to exactly one cluster. These constraints lead to consider all observations belonging to a cluster as equally associated to it, independently of their distance to the corresponding centroid or the possibility of lying in a boundary region between two or more clusters. However, it may seem natural to assume that observations which lie further from a cluster

centroid have a weaker association to that cluster than nearer observations [24]. Similarly, detecting those observations that could be feasibly associated to more than one cluster can be a highly relevant issue in different applications. In this sense, the fuzzy approach enables observations to be partially associated to more than one cluster by modeling such association through [0,1]-valued membership degrees, rather than by a binary, 0 or 1 index. These ideas led to the proposal of the Fuzzy *C*-means method (FCM, initially introduced in [32] and further developed in [33]), which extended the *K*-means method by allowing partial or fuzzy membership of observations into clusters and triggered research in the field of fuzzy clustering.

Later, the idea of applying a regularization approach to some ill-defined problems in fuzzy clustering led to the consideration of entropy measures as a kind of penalty function, since a large entropy value indicates a high level of randomness in the assignation of observations to clusters or similarly a high level of overlapping between clusters (see [34–37] for an ongoing discussion of the notion of overlap in fuzzy classification). This idea is first implemented in [38] by modifying the FCM to minimize Shannon's entropy [39]. Similarly, entropy measures are also applied to deal with the difficulties found by the FCM on non-balanced cluster size datasets, where misclassifications arise as observations from large clusters are classified in small clusters or big clusters are divided into two clusters, as shown for instance in [21]. Indeed, by introducing new variables to represent the ratios of observations per cluster, these problems can be addressed through the use of relative entropy measures or divergences between the membership functions and such ratios [21,40,41], for example Kullback–Leiber [42] and Tsallis [43] divergences, respectively, applied in the methods proposed in [44,45].

The aim of this paper is then to follow and extend the ideas of those previously mentioned works by studying the application of Rényi divergence [46] in the FCM objective function as a penalization term with regularization functionality. In this sense, a relevant feature of Rényi's entropy and divergence is that, due to their parametric formulation, they are known to generalize other entropy and divergence measures; particularly, they, respectively, extend Shannon's entropy and Kullback–Leiber divergence. Furthermore, the kind of generalization provided by Rényi entropy and divergence is different from that provided by Tsallis entropy and divergence. This makes their application in the context of fuzzy clustering interesting, as more general clustering methods can be devised from them. Moreover, as discussed in Section 3, the iterative method derived from the proposed application of Rényi divergence seems to make a more adaptive usage of the available information than previous divergence-based methods, such as the previously referenced one [45] based on Tsallis divergence.

Indeed, as far as we know, two previous proposals apply either Rényi entropy or divergence to define new clustering methods. The first one is found in [47], where Rényi divergence is taken as a dissimilarity metric that is used to substitute the usual variance-based term of the FCM objective function, therefore without a regularization aim. The second proposal [48] does however employ Rényi entropy as a regularization term in the context of fuzzy clustering of time data arrays, but it does not consider observations ratios per cluster, and thus it does not apply Rényi divergence as well.

Therefore, the main contribution of this work consists of the proposal of a fuzzy clustering method using Rényi divergence between membership degrees and observations ratios per cluster as a regularization term, thus adding it to the usual variance term of FCM objective function when considering cluster sizes. Looking for some more flexibility in the formation of clusters, a second method is then proposed that extends the first one by introducing a Gaussian kernel metric. Moreover, an extensive computational study is also carried out to analyze the performance of the proposed methods in relation with that of several standard fuzzy clustering methods and other proposals making use of divergence-based regularization terms.

This paper is organized as follows. Section 2 introduces the notation to be employed and recalls some preliminaries on fuzzy clustering, entropy and divergence measures, and

kernel metrics. It also reviews some fuzzy clustering methods employing divergence-based regularization. Section 3 presents the proposed method and its kernel-based extension. Section 4 describes the computational study analyzing the performance of the proposed methods and comparing it with that of a set of reference methods. Finally, some conclusions are exposed in Section 5.

2. Preliminaries

This section is devoted to recalling some basics about fuzzy clustering, entropy measures, and kernel functions needed for the understanding of the proposed methods. In this way, Section 2.1 introduces notations and reviews some of the most extended prototype-based clustering methods. Section 2.2 recalls general notions on entropy and divergence measures and describes some clustering methods making use of divergence measures. Finally, Section 2.3 provides the needed basics on kernel metrics.

2.1. Some Basics on Clustering Analysis

Let X be a set of N observations or data points with n characteristics or variables at each observation. We assume X is structured as a $N \times n$ data matrix, such that its ith row $X_i = (x_{i1}, \ldots, x_{in})^T \in \mathbb{R}^n$ denotes the ith observation, $i = 1, \ldots, N$. It is also possible to assume that observations are normalized in each variable (e.g., by the min-max method) in order to avoid scale differences, so we can consider the data to be contained in a unit hypercube of dimension n.

The centroid-based clustering methods we deal with are iterative processes that classify or partition the data X into K clusters $\{C_{kt}\}_{k=1}^{K}$ in several steps $t = 1, 2, \ldots$, with $t \in \mathbb{T} \subset \mathbb{N}$, where $\mathbb{T}$ is the set of iterations or steps. At each iteration t, the partition of X in the K clusters is expressed through a $N \times K$ matrix $M_t = [\mu_{ikt}]_{(N,K)}$, such that μ_{ikt} denotes the membership degree of observation X_i into cluster C_{kt}. Notation M is used to represent a generic $N \times K$ partition matrix. In a crisp setting, only allowing for either null or total association between observations and clusters, it is $\mu_{ikt} \in \{0, 1\}$, while in a fuzzy setting it is $\mu_{ikt} \in [0, 1]$ in order to allow partial association degrees. In either case, the constraints

$$\sum_{k=1}^{K} \mu_{ikt} = 1 \text{ for any } i = 1, \ldots, N \text{ and } t \in \mathbb{T} \tag{1}$$

(Ruspini condition [49]), and

$$0 < \sum_{i=1}^{N} \mu_{ikt} < N \text{ for any } k = 1, \ldots, K \text{ and } t \in \mathbb{T}, \tag{2}$$

are imposed to guarantee the formation of data partitions with non-empty clusters. Let us remark that fuzzy methods with $\mu_{ikt} \in [0, 1]$ do not actually need to impose Equation (2), since in practice the iterative expressions for the membership degrees obtained by only using Equation (1) as constraint already guarantee that $\mu_{ikt} > 0$ for all k and i.

Similarly, the centroid $V_{kt} = (v_{kt1}, \ldots, v_{ktn})^T$ of the kth cluster at iteration t is obtained for each method through a certain weighted average of the observations (using membership degrees to cluster C_{kt} as weights), and the $K \times n$ matrix with the n coordinates of all K centroids at a given iteration t is denoted by V_t. We also employ notation V to denote a generic centroid $K \times n$ matrix. Initial centroids or seeds $\{V_{k0}\}_{k=1}^{K}$ are needed to provide the starting point for the iterative process, and many initialization methods are available to this aim [25].

Iterative prototype-based methods can be seen as heuristic procedures intended to minimize a certain objective function $J(M, V)$, which is usually set to represent a kind of within-cluster variance [8] through squared distances between observations and centroids. Thus, although the aim of any clustering procedure is to find the global minimum of such objective function, iterative heuristic methods can only guarantee convergence to a local minimum [29], which is usually dependent on the selected initial centroids or seeds. Only the Euclidean distance or metric $d(X_i, X_j) = \left[(X_i - X_j)^T (X_i - X_j)\right]^{1/2}$ for

any $X_i,\ X_j \in \mathbb{R}^n$ is employed in this paper (although we also implicitly consider other metrics through the usage of kernel functions, see Section 2.3). For simplicity, the distance between an observation X_i and a centroid V_{kt} is denoted by $d_{ikt} = d(X_i, V_{kt})$ and its square by $d^2_{ikt} = d(X_i, V_{kt})^2$.

The iterative process of the methods discussed in this paper typically consider a number of steps at each iteration, which at least include: (i) updating the centroids from the membership degrees obtained in the previous iteration; and (ii) updating the membership degree of each observation to each cluster taking into account the distance from the observation to the corresponding (updated) cluster centroid. In this regard, without loss of generality (it is a matter of convention where an iteration starts), we always assume that, at each iteration, centroids are updated first, and thus that initial degrees μ_{ik0} for all i and k have to be also obtained from the initial seeds $\{V_{k0}\}_{k=1}^K$.

Different stopping criteria can be used in order to detect convergence of the iterative process or to avoid it extending for too long: a maximum number of iterations t_{max} can be specified, and a tolerance parameter $\varepsilon > 0$ can be used so that the method stops when the difference between centroids of iterations t and $t+1$ is below it, $\max_k \left| V_{k(t+1)} - V_{kt} \right| < \varepsilon$, or similarly when the difference between membership functions of iterations t and $t+1$ is small enough, $\max_{i,k} \left| \mu_{ik(t+1)} - \mu_{ikt} \right| < \varepsilon$.

Finally, as mentioned above, some difficulties may arise when applying the FCM method on non-balanced datasets, i.e., when clusters present quite different sizes. Such difficulties can be addressed by expanding the optimization problem through the introduction of a set of new variables $\phi_{kt} \in [0,1]$, $k = 1, \ldots, K$, called observations ratios per cluster [21,40] as the constraint

$$\sum_{k=1}^K \phi_{kt} = 1 \text{ for any } t \in \mathbb{T} \tag{3}$$

is also imposed. These variables allow weighting the contribution of each cluster to the objective function as inversely proportional to the cluster's size, facilitating in practice the formation of clusters with different sizes. Notation Φ is used to represent a generic vector $(\phi_1, \ldots, \phi_K)$ of observations ratios. In those methods making use of observations ratios, we assume that these are updated at the end of each iteration, from updated centroids and membership degrees. This entails that initial ratios ϕ_{k0}, $k = 1, \ldots, K$, have to be somehow obtained from the initial centroids and membership degrees. Moreover, the iterative formulae for membership degrees and observations ratios provided by some methods show an interdependence between both kinds of variables. As a consequence, specific expressions to initialize both membership degrees and observations ratios for $t = 0$ have to be provided in these cases. In this sense, for the computational experiments described in Section 4, we applied Equations (44) and (45) to this aim for all methods requiring such specific initialization.

2.1.1. K-Means

In terms of the notation just introduced and assuming a fixed number of cluster K, the K-means method consists in a two-step iterative process that tries to find the partition matrix M and the centroids V that solve the optimization problem

$$\min_{M,V} J(M,V) = \min_{M,V} \sum_{k=1}^K \sum_{i=1}^N \mu_{ikt} d^2_{ikt}, \quad \forall t \in \mathbb{T} \tag{4}$$

subject to $\mu_{ikt} \in \{0,1\}\ \forall i,k,t$, as well as to Equations (1) and (2).

Given a set of initial seeds $\{V_{k0}\}_{k=1}^K$, as exposed above, we assume that initial membership degrees $\mu_{ik0} \in \{0,1\}$ are obtained by assigning each observation to the cluster with the nearest centroid, in a crisp way, i.e., $\mu_{ik0} = 1$ when $d(X_i, V_{k0}) = \min_{l=1,\ldots,K} d(X_i, V_{l0})$ and

$\mu_{ik0} = 0$ otherwise. Then, at each iteration $t > 0$, the method first updates the centroids through the formula

$$V_{kt} = \frac{1}{N_{k(t-1)}} \sum_{i=1}^{N} \mu_{ik(t-1)} X_i \tag{5}$$

where $N_{k(t-1)} = \sum_{i=1,\ldots,N} \mu_{ik(t-1)}$ denotes the number of observations belonging to cluster $C_{k(t-1)}$, i.e., its cluster size. Next, observations' membership degrees are updated, again assigning each observation to the cluster with the nearest (updated) centroid, i.e., $\mu_{ik(t-1)}$ when $d(X_i, V_{kt}) = \min_{l=1,\ldots,K} d(X_i, V_{lt})$ and $\mu_{ikt} = 0$ otherwise. These two steps are repeated until a stopping criterion is met.

2.1.2. Fuzzy C-Means

This method was proposed by Dunn [32] as a generalization of the *K*-means allowing to consider partial association degrees between observations and clusters. Later, Bezdek [33] improved and generalized Dunn's proposal by introducing the fuzzifier parameter $m > 1$, which allows controlling the fuzziness of the clusters. Therefore, although apparently similar to the *K*-means, the standard FCM algorithm allows the elements μ_{ikt} of the partition matrices M_t to be continuous degrees in the unit interval [0,1] instead of binary indexes in {0,1}, and thus tries to solve the optimization problem

$$\min_{M,V} J_m(M, V) = \min_{M,V} \sum_{i=1}^{N} \sum_{k=1}^{K} \mu_{ikt}^m d_{ikt}^2, \ \forall\, t \in \mathbb{T} \tag{6}$$

subject to $\mu_{ikt} \in [0, 1]\ \forall i, k, t$ as well as to Equations (1) and (2). Departing from a set of initial seeds $\{V_{k0}\}_{k=1}^{K}$, at each iteration, it is possible to apply the method of Lagrange multipliers in order to find a local minimum of Equation (6) verifying the imposed constraints. This leads to update the centroids following the expression

$$V_{kt} = \frac{\sum_{i=1}^{N} \mu_{ik(t-1)}^m X_i}{\sum_{i=1}^{N} \mu_{ik(t-1)}^m} \ \ \forall t > 0 \tag{7}$$

as well as to obtain the membership degrees as

$$\mu_{ikt} = \left[d_{ikt}^2\right]^{\frac{-1}{(m-1)}} / \sum_{g=1}^{K} \left[d_{igt}^2\right]^{\frac{-1}{(m-1)}} \ \forall t \geq 0 \tag{8}$$

The limit behavior of the FCM method in terms of the fuzzifier parameter m is studied for instance in [50]: when m tends to infinity, the centroids converge to the global average of data, and, when m tends to 1, the method tends to be equivalent to the K-means. In addition, in [50], different results on the convergence of the FCM are proved, which guarantee its convergence to a local or global minimum or to a saddle point of Equation (6). Similar results for alternative FCM methods and generalizations are demonstrated in [51,52]. In [53], it is shown that Equation (6) is convex only when M or V is fixed. The FCM has basically the same advantages and disadvantages discussed in Section 1 for the *K*-means (see also [24] for a detailed comparison between *K*-means and FCM).

2.1.3. Fuzzy C-Means with Cluster Observations Ratios

A modification of the FCM method is proposed in [21,40] with the idea of adding cluster size variables Φ, facilitating the formation of clusters of different sizes. The optimization problem the modified method tries to solve is

$$\min_{M,\Phi,V} J_m(M, \Phi, V) = \min_{M,\Phi,V} \sum_{i=1}^{N} \sum_{k=1}^{K} \phi_{kt}^{1-m} \mu_{ikt}^m d_{ikt}^2, \forall t \in \mathbb{T} \tag{9}$$

subject to $\mu_{ikt} \in [0,1]\ \forall i,k,t$ as well as to Equations (1)–(3). The application of Lagrange multipliers to Equation (9) leads as before to Equation (7) for centroid updating, as well as to obtain new membership degrees by following the expression

$$\mu_{ikt} = \frac{\left(\phi_{k(t-1)}^{1-m} d_{ikt}^2\right)^{\frac{-1}{m-1}}}{\sum_{g=1}^{K}\left(\phi_{g(t-1)}^{1-m} d_{igt}^2\right)^{\frac{-1}{m-1}}}\ \forall t > 0 \tag{10}$$

Finally, the ratios of observations per cluster are to be computed as

$$\phi_{kt} = \frac{\left(\sum_{i=1}^{N} \mu_{ikt}^m d_{ikt}^2\right)^{\frac{1}{m}}}{\sum_{g=1}^{K}\left(\sum_{i=1}^{N} \mu_{igt}^m d_{igt}^2\right)^{\frac{1}{m}}}\ \forall t > 0 \tag{11}$$

Notice that, due to the interdependence between membership degrees and observations ratios, an initialization method has to be provided to obtain these when $t = 0$. As mentioned above, Equations (44) and (45) are applied to this aim for this and the following methods.

2.2. Entropy and Relative Entropy in Cluster Analysis

This section aims to recall the definitions of some entropy and relative entropy measures and describes a couple of fuzzy clustering methods that employ divergence measures with a regularization functionality.

2.2.1. Entropy Measures

The notion of information entropy was a cornerstone in the proposal of a general information and communication theory by Shannon in the mid-20th century [39]. Given a random variable $\mathcal{A}$, the information entropy of $\mathcal{A}$ can be understood as the mathematical expectation of the information content of $\mathcal{A}$, thus measuring the average amount of information (or, equivalently, uncertainty) associated with the realization of the random variable. When applied to fuzzy distributions verifying probabilistic constraints, such as membership degrees to clusters for which the Ruspini condition [49] holds, entropy can be understood as a measure of the fuzziness of such distribution [45]. In this context, entropy is maximum when overlap between the different classes or clusters is present. This has motivated the application of entropy measures in fuzzy clustering as a penalization term aiming to regularize the solution of clustering problems by avoiding too overlapping partitions. Furthermore, entropy measures have also found a relevant application in recent developments in non-fuzzy clustering ensembles [54,55]. Next, we recall the definitions of the entropy measures proposed by Shannon, Tsallis, and Rényi.

Definition 1 [39]**.** *Let $\mathcal{A}$ be a discrete random variable with finite support $\chi = \{x_1, \ldots, x_s\}$ and probability mass function $P(\mathcal{A})$. Then, the Shannon entropy of $\mathcal{A}$ is defined as*

$$\mathcal{H}_S(\mathcal{A}) = -\sum_{x_i \in \chi} P(x_i) \ln P(x_i) \tag{12}$$

Definition 2 [43]**.** *Let $\mathcal{A}$ be a discrete random variable with finite support $\chi = \{x_1, \ldots, x_s\}$ and probability mass function $P(\mathcal{A})$. Then, Tsallis entropy of $\mathcal{A}$ is defined as*

$$\mathcal{H}_T(\mathcal{A}) = \left(1 - \sum_{x_i \in \chi} P(x_i)^q\right) / (q-1) \tag{13}$$

where the parameter $q \in \mathbb{R}$ is interpreted as the degree of non-extensiveness or pseudo-additivity.

Tsallis entropy was introduced in clustering analysis for non-extensive statistics in the context of physical statistics and thermodynamics [43]. Notice that Equation (13) generalizes Equation (12), since, when $q \to 1$, Equation (12) is obtained.

Definition 3 [46]. *Let $\mathcal{A}$ be a discrete random variable with finite support $\chi = \{x_1, \ldots, x_s\}$ and probability mass function $P(\mathcal{A})$. Then, Rényi entropy of $\mathcal{A}$ is defined as*

$$\mathcal{H}_R(\mathcal{A}) = \ln\left(\sum_{x_i \in \chi} P(x_i)^{\alpha}\right) / (1 - \alpha) \tag{14}$$

where the parameter $\alpha > 0$ indicates the order of the entropy measure.

Notice that Equation (14) generalizes several other entropy measures, such as Shannon entropy (when $\alpha \to 1$, see [46]), Harley entropy (when $\alpha \to 0$, see [56]), minimum entropy (when $\alpha \to \infty$, see again [46]), and collision entropy (when $\alpha = 2$, see [57]). Notice also that Equation (14) is a generalized mean in the Naguno–Kolmogorov sense [46].

2.2.2. Relative Entropy

Following the authors of [58,59], given two discrete random variables $\mathcal{A}$ and $\mathcal{B}$ with the same finite support $\chi = \{x_1, \ldots, x_s\}$ and probability mass functions $P(\mathcal{A})$ and $Q(\mathcal{B})$, the relative entropy in the context of statistics would be understood as the loss of information that would result from using $\mathcal{B}$ instead of $\mathcal{A}$. If this loss of information were null or small, the value of the divergence $\mathcal{D}(\mathcal{A} \parallel \mathcal{B})$ would be equal to zero or close to zero, while, if the difference is large, the divergence would take a large positive value. In addition, the divergence is always positive and convex, although, from the metric point of view, it is not always a symmetrical measure, that is $\mathcal{D}(\mathcal{A} \parallel \mathcal{B}) \neq \mathcal{D}(\mathcal{B} \parallel \mathcal{A})$.

Next, we recall the definitions of Kullback–Leiber, Tsallis, and Rényi relative entropy measures.

Definition 4 [42]. *In the above conditions, the Kullback–Leiber relative entropy is given by*

$$\mathcal{D}_{KL}(\mathcal{A} \parallel \mathcal{B}) = \sum_{x_i \in \chi} P(x_i) \ln\left(\frac{P(x_i)}{Q(x_i)}\right) \tag{15}$$

Definition 5 [43]. *In the above conditions, the Tsallis relative entropy is given by*

$$\mathcal{D}_T(\mathcal{A} \parallel \mathcal{B}) = \left(\sum_{x_i \in \chi} P(x_i) \left[\left(\frac{P(x_i)}{Q(x_i)}\right)^{q-1} - 1\right]\right) / (q - 1) \tag{16}$$

where the parameter $q \in \mathbb{R}$ has a similar interpretation as in Definition 2.

Definition 6 [46]. *In the above conditions, the Rényi relative entropy is given by*

$$\mathcal{D}_R(\mathcal{A} \parallel \mathcal{B}) = \frac{1}{(\alpha - 1)} \ln\left(\sum_{x_i \in \chi} Q(x_i)^{1-\alpha} P(x_i)^{\alpha}\right) \tag{17}$$

where the parameter $\alpha > 0$ provides the order of the divergence measure as in Definition 3.

In [60], it is shown that Equation (17) generalizes Equation (15) when $\alpha \to 1$. In addition, when $\alpha = 1/2$, the Bhattacharyya's distance [61] divided by 2 is obtained.

2.2.3. Fuzzy C-Means with Kullback–Leiber Relative Entropy and Cluster Size

The application of Kullback–Leiber relative entropy in fuzzy clustering is proposed in [41], using Equation (15) as a regularization term to penalize solutions with a high level of overlapping between clusters. In that work, Equation (15) is added to the FCM objective

function exposed in Section 2.1.2 (without the fuzzifier parameter m), substituting the probability distribution $P(\mathcal{A})$ in Equation (15) for the fuzzy membership degrees in the partition matrix M, as well as $Q(\mathcal{B})$ for the observations ratios per cluster Φ. A parameter $\zeta > 0$ is introduced to weight the influence of the regularization term. In this way, the method proposed in [41] aims to solve the optimization problem

$$\min_{M,\Phi,V} J_\zeta(M,\Phi,V) = \min_{M,\Phi,V} \sum_{i=1}^{N}\sum_{k=1}^{K} \mu_{ikt} d_{ikt}^2 + \zeta\left(\sum_{i=1}^{N}\sum_{k=1}^{K} \mu_{ikt}\, ln\left(\frac{\mu_{ikt}}{\phi_{kt}}\right)\right), \forall t \in \mathbb{T} \tag{18}$$

subject to $\mu_{ikt} \in [0,1]\ \forall i,k,t$ as well as to Equations (1)–(3).

Minimizing Equation (18) by the method of Lagrange multipliers again leads to updating centroids through Equation (7), as well as computing membership degrees by following the expression

$$\mu_{ikt} = \phi_{k(t-1)}\ exp\left(-d_{ik(t-1)}^2/\zeta\right) / \sum_{g=1}^{K} \phi_{g(t-1)}\ exp\left(-d_{ig(t-1)}^2/\zeta\right)\ \forall t > 0 \tag{19}$$

Finally, the obtained formula for the ratios of observations per cluster is

$$\phi_{kt} = \sum_{i=1}^{N} \mu_{ikt}/N\ \forall t > 0 \tag{20}$$

2.2.4. Fuzzy C-Means with Tsallis Relative Entropy and Cluster Size

A first proposal for the application of Tsallis entropy in fuzzy clustering is given in [62], using Equation (13) without considering cluster size variables. Later, Tsallis relative entropy, Equation (16), is applied in [45] as a penalty function into a regularization term, which leads to a method that attempts to solve the optimization problem

$$\min_{M,\Phi,V} J_{m,\zeta}(M,\Phi,V) = \min_{M,\Phi,V} \sum_{i=1}^{N}\sum_{k=1}^{K} \phi_{kt}^{1-m}\ \mu_{ikt}^m d_{ikt}^2 + \frac{\zeta}{(m-1)}\left(\sum_{i=1}^{N}\sum_{k=1}^{K} \phi_{kt}^{1-m}\mu_{ikt}^m - \sum_{i=1}^{N}\sum_{k=1}^{K} \mu_{ikt}\right),\ \forall t \in \mathbb{T} \tag{21}$$

subject to $\mu_{ikt} \in [0,1]\ \forall i,k,t$ as well as to Equations (1)–(3). Notice that this objective function builds on the objective function of the FCM with cluster observations ratios exposed in Section 2.1.3, therefore considering the fuzzifier parameter m in the intra-cluster variance term along with the regularization parameter $\zeta > 0$ weighting the Tsallis relative entropy penalization term, in which the parameter m plays the role of the non-extensiveness parameter q.

Using the Lagrange multipliers method on Equation (21) and its constraints again leads to Equation (7) for centroid updating and calculating the partition matrix elements at each iteration as

$$\mu_{ikt} = \left(\phi_{k(t-1)}^{1-m}\left(d_{ikt}^2 + \frac{\zeta}{(m-1)}\right)\right)^{\frac{-1}{m-1}} / \sum_{g=1}^{K}\left(\phi_{g(t-1)}^{1-m}\left(d_{igt}^2 + \frac{\zeta}{(m-1)}\right)\right)^{\frac{-1}{m-1}}\ \forall t > 0 \tag{22}$$

while the obtained expression for the ratio of observations per cluster is

$$\phi_{kt} = \left(\sum_{i=1}^{N} \mu_{ikt}^m\left(d_{ikt}^2 + \frac{\zeta}{(1-m)}\right)\right)^{\frac{1}{m}} / \sum_{g=1}^{K}\left(\sum_{i=1}^{N} \mu_{igt}^m\left(d_{igt}^2 + \frac{\zeta}{(1-m)}\right)\right)^{\frac{1}{m}}\ \forall t > 0 \tag{23}$$

2.3. Kernel Metrics

When the observations X present separability issues in the metric space $\mathbb{R}^n$ in which they are contained, a fruitful strategy may be that of mapping them into a greater dimension space $\mathbb{R}^{n'}$, with $n' > n$, through an adequate transformation $\varphi : \mathbb{R}^n \to \mathbb{R}^{n'}$ such that the data become separable in this new space. This can allow the formation of more separated

clusters than in the original space, potentially leading to obtain a better partition of the observations X. This idea, which has been successfully implemented in diverse data analysis proposals, e.g., support vector machines [63] or clustering analysis [64], is typically referred to as the *kernel trick*, since it is possible to apply it through the so-called kernel functions without explicitly providing the transformation φ. The reason behind this is that, when only distances between mapped observations need to be obtained, these can be calculated through a kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ such that $K(X_i,\ X_j) = \langle \varphi(X_i), \varphi(X_j) \rangle$ for any $X_i,\ X_j \in \mathbb{R}^n$, where $\langle \cdot, \cdot \rangle$ denotes the inner product of the metric space $\mathbb{R}^{n'}$ [65].

In this way, squared distances between mapped observations and centroids in $\mathbb{R}^{n'}$ can be easily obtained through a kernel function, since it holds that

$$d_{ik}^2 = d(X_i, V_k)^2 = \|\varphi(X_i) - \varphi(V_k)\|^2 = \tag{24}$$

$$= \langle \varphi(X_i), \varphi(X_i) \rangle + \langle \varphi(V_k), \varphi(V_k) \rangle - 2\langle \varphi(X_i), \varphi(V_k) \rangle = \tag{25}$$

$$= K(X_i,\ X_i) + K(V_k,\ V_k) - 2K(X_i,\ V_k) \tag{26}$$

In this work, we apply the Gaussian or radial basis kernel given by the function

$$K(X_i,\ V_k) = exp\left(-\gamma \|X_i - V_k\|^2\right) \tag{27}$$

where the parameter $\gamma > 0$ controls the proximity between the transformed observations. Indeed, since substituting Equation (27) into Equation (26) leads to

$$d_{ik}^2 = 2\left(1 - exp\left(-\gamma \|X_i - V_k\|^2\right)\right) \tag{28}$$

it is easy to see that, when γ increases, d_{ik}^2 approximates to 2 more quickly, so each observation tends to become isolated; however, when γ tends to zero, all observations tend to be part of the same neighborhood. Furthermore, as for instance shown in [64], for a Gaussian kernel centroid, updating can be performed by simply introducing the kernel function in the expression for centroid calculation, in such a way that Equation (7) is just translated as

$$V_{kt} = \frac{\sum_{i=1}^{N} \mu_{ikt}^m K\left(X_i,\ V_{k(t-1)}\right)\ X_i}{\sum_{i=1}^{N} \mu_{ikt}^m K\left(X_i,\ V_{k(t-1)}\right)} \tag{29}$$

A Gaussian kernel version of the FCM method was first proposed by the authors of [64], although an earlier version is introduced in [66] using an exponential metric without explicit reference to the usage of kernel functions.

3. Fuzzy Clustering with Rényi Relative Entropy

This section presents the main proposal of this work, consisting in the application of Rényi relative entropy between membership degrees and observations ratios per cluster as a regularization term in the context of fuzzy clustering. Thus, our proposal situates and builds on the line of previous works studying the usage of relative entropy measures for improving the performance of fuzzy clustering methods, and particularly that of the FCM method, as reviewed in the last section.

The specific motivation for our proposal is to study whether the implementation of Rényi relative entropy as a regularization term of membership degrees along with cluster sizes can lead to obtain better iterative formulas to find optimal partitions than those provided by other methods, such as the FCM itself or its extensions based on the Kullback–Leiber or Tsallis relative entropy measures.

Let us point out that, to the extent of our knowledge, there are only two previous proposals making use of Rényi entropy or relative entropy in the context of fuzzy clustering:

in [47], Equation (17) is used as a dissimilarity metric without a regularization functionality, while, in [48], Equation (14) is applied with a regularization aim in the context of the application of FCM to time data arrays, but without taking into account observations ratios per cluster. Furthermore, neither the work in [47] nor that in [48] makes use of kernel metrics.

Therefore, in this work, we propose a first method using Equation (17) as a penalization function in a regularization term taking into account both membership degrees and observations ratios per cluster, which is expected to provide better results than just using a regularization term based on Equation (14) without cluster sizes, as in [48]. Moreover, our proposal adds Rényi relative entropy regularization to the FCM objective function already taking into account cluster sizes, contrary to using an objective function based only in Equation (17) without considering intra-cluster variance, as in [47]. The second proposed method extends the first one by introducing a Gaussian kernel metric, thus enabling some more flexibility in the formation of clusters than that provided by the first method.

To describe our proposal, this section is divided into two subsections, one for each of the proposed methods. In these subsections, we first present the proposed objective function and constraints of the corresponding method, and then expose as a theorem the expressions obtained for the iterative formulas through the application of Lagrange multipliers on the objective function. Next, the proof of the theorem is given. Finally, the steps of the algorithm associated to each method are detailed.

3.1. Fuzzy C-Means with Rényi Relative Entropy and Cluster Size

As mentioned, the objective function of the first proposed method follows the idea of Equations (18) and (21), in this case adding a regularization term based on Rényi relative entropy to the usual FCM objective function when considering observations ratios per cluster. Therefore, this method seeks to solve the optimization problem

$$\min_{M,\Phi,V} J_{m,\zeta}(M,\Phi,V) = \min_{M,\Phi,V} \sum_{i=1}^{N}\sum_{k=1}^{K} \phi_{kt}^{1-m}\mu_{ikt}^{m}d_{ikt}^{2} + \frac{\zeta}{(m-1)}\ln\left(\sum_{i=1}^{N}\sum_{k=1}^{K}\phi_{kt}^{1-m}\mu_{ikt}^{m}\right) \tag{30}$$

subject to Equations (1) and (3).

We do not explicitly impose Equation (2) avoiding empty clusters since, as can be seen in Equation (31) below, the obtained iterative expression for membership degrees trivially guarantees that these are always greater than 0. Besides, let us remark that the objective function in Equation (30) adds a penalization function based on Rényi relative entropy, Equation (17), to the objective function of the FCM when considering observations ratios per cluster, Equation (9). As in Equations (18) and (21), the parameter $\zeta > 0$ is introduced to weight the influence of this regularization term. Notice also that, similarly to Equation (21), the fuzzifier parameter $m > 1$ plays the role of the order parameter α in the Rényi-based term.

Next, the Lagrange multipliers method is applied to the previous optimization problem with objective function Equation (30) in order to obtain iterative expressions for the membership degrees μ_{ikt} and the observations ratios per cluster ϕ_{kt}, as well as for updating the centroids V_{kt}. It is important to stress that, since the constraints in Equations (1) and (3) are orthogonal for any $i = 1,\ldots,N, t \in \mathbb{T}$, the resolution of the Lagrange multipliers method for several constraints can be handled separately for each constraint. Moreover, as the centroids V_{kt} are unconstrained, the Lagrangian function defined for their optimization is equal to Equation (30).

Theorem 1. *The application of the Lagrange multipliers method to the objective function Equation (30) constrained by Equations (1) and (3) provides the following solutions* $\forall t > 0$.

For updating the centroids Equation (7):

$$V_{kt} = \left(\sum_{i=1}^{N} \mu_{ik(t-1)}^{m} \, X_i\right) / \left(\sum_{i=1}^{N} \mu_{ik(t-1)}^{m}\right), \; \forall k = 1, \ldots, K$$

For membership degrees:

$$\mu_{ikt} = \frac{\left(\phi_{k(t-1)}^{1-m}\left(d_{ikt}^{2} + \frac{\zeta\left(\sum_{i=1}^{N}\sum_{k=1}^{K}\phi_{k(t-1)}^{1-m}\mu_{ik(t-1)}^{m}\right)^{-1}}{(m-1)}\right)\right)^{\frac{-1}{m-1}}}{\sum_{g=1}^{K}\left(\phi_{g(t-1)}^{1-m}\left(d_{igt}^{2} + \frac{\zeta\left(\sum_{i=1}^{N}\sum_{g=1}^{K}\phi_{g(t-1)}^{1-m}\mu_{ig(t-1)}^{m}\right)^{-1}}{(m-1)}\right)\right)^{\frac{-1}{m-1}}}, \; \forall i = 1, \ldots, N, k = 1, \ldots, K \tag{31}$$

For observations ratios per clusters:

$$\phi_{kt} = \frac{\left(\sum_{i=1}^{N} \mu_{ikt}^{m} d_{ikt}^{2} + \frac{\zeta\left(\sum_{i=1}^{N}\mu_{ikt}^{m}\right)}{(m-1)\left(\sum_{i=1}^{N}\sum_{g=1}^{K}\phi_{g(t-1)}^{1-m}\mu_{igt}^{m}\right)}\right)^{\frac{1}{m}}}{\sum_{g=1}^{K}\left(\sum_{i=1}^{N} \mu_{igt}^{m} d_{igt}^{2} + \frac{\zeta\left(\sum_{i=1}^{N}\mu_{igt}^{m}\right)}{(m-1)\left(\sum_{i=1}^{N}\sum_{g=1}^{K}\phi_{g(t-1)}^{1-m}\mu_{igt}^{m}\right)}\right)^{\frac{1}{m}}}, \; \forall k = 1, \ldots, K \tag{32}$$

Proof. First, in order to obtain the centroids updating formula, and taking into account that $d_{ikt}^{2} = \left[(X_i - V_{kt})^{T}(X_i - V_{kt})\right]$ and the centroids are unconstrained, the derivative of Equation (30) with respect to V_{kt} is equaled to zero, leading to

$$\sum_{i=1}^{N} \phi_{kt}^{1-m} \mu_{ikt}^{m} \left[-2(X_i - V_{kt})\right] = 0, \; \forall k = 1, \ldots, K \tag{33}$$

Equation (7) easily follows by solving for V_{kt} in Equation (33) and replacing each unknown μ_{ikt} with the corresponding known $\mu_{ik(t-1)}$ from the previous iteration. Equation (7) provides a local minimum as the second derivative of Equation (30) is positive.

Then, addressing the optimization of the membership degrees μ_{ikt}, we build the Lagrangian function associated to Equation (30) restricted to the conditions imposed by Equation (1):

$$L_{m,\zeta}(M, \Phi, V) = \sum_{i=1}^{N}\sum_{k=1}^{K} \phi_{kt}^{1-m} \, \mu_{ikt}^{m} d_{ikt}^{2} + \frac{\zeta}{(m-1)} \ln\left(\sum_{i=1}^{N}\sum_{k=1}^{K} \phi_{kt}^{1-m} \mu_{ikt}^{m}\right) - \sum_{i=1}^{N} \lambda_{it}\left(\sum_{k=1}^{K} \mu_{ikt} - 1\right) \tag{34}$$

By taking the derivative of Equation (34) with respect to μ_{ikt} and equaling to zero, the following is obtained:

$$\mu_{ikt} = \left(\frac{\lambda_{it} \, \phi_{kt}^{m-1}/m}{d_{ikt}^{2} + \frac{\zeta\left(\sum_{i=1}^{N}\sum_{g=1}^{K}\phi_{gt}^{1-m}\mu_{igt}^{m}\right)^{-1}}{(m-1)}}\right)^{\frac{1}{m-1}} \quad \forall i = 1, \ldots, N, k = 1, \ldots, K \tag{35}$$

Since at iteration t both μ_{ikt} and ϕ_{kt} in the right-side of Equation (35) have to be considered unknown, we, respectively, approximate them by $\mu_{ik(t-1)}$ and $\phi_{k(t-1)}$. This leads to

$$\mu_{ikt} = \left(\frac{\lambda_{it}\, \phi_{k(t-1)}^{m-1}/m}{d_{ikt}^2 + \frac{\zeta \left(\sum_{i=1}^{N} \sum_{g=1}^{K} \phi_{g(t-1)}^{1-m} \mu_{ig(t-1)}^{m} \right)^{-1}}{(m-1)}} \right)^{\frac{1}{m-1}} \quad \forall i = 1, \ldots, N, k = 1, \ldots, K \tag{36}$$

Now, for each $i = 1, \ldots, N$, we impose that the μ_{ikt} as given in Equation (36), $k = 1, \ldots, K$, have to fulfill the corresponding constraint in Equation (1), that is

$$1 = \sum_{k=1}^{K} \left(\frac{\lambda_{it}\, \phi_{k(t-1)}^{m-1}/m}{d_{ikt}^2 + \frac{\zeta \left(\sum_{i=1}^{N} \sum_{g=1}^{K} \phi_{g(t-1)}^{1-m} \mu_{ig(t-1)}^{m} \right)^{-1}}{(m-1)}} \right)^{\frac{1}{m-1}} \quad \forall i = 1, \ldots, N \tag{37}$$

Solving for λ_{it} in Equation (37), we get

$$\lambda_{it} = m / \left[\sum_{k=1}^{K} \left(\phi_{k(t-1)}^{1-m} d_{ikt}^2 + \frac{\phi_{k(t-1)}^{1-m} \zeta}{(m-1) \left(\sum_{i=1}^{N} \sum_{g=1}^{K} \phi_{g(t-1)}^{1-m} \mu_{ig(t-1)}^{m} \right)} \right)^{\frac{-1}{m-1}} \right]^{m-1} \quad \forall i = 1, \ldots, N \tag{38}$$

Then, Equation (31) is obtained by replacing λ_{it} in Equation (36) with Equation (38). It is straightforward to check that Equation (36) is a local minimum as the second derivative of Equation (34) is positive.

Now, addressing the optimization of the observations ratios per cluster ϕ_{kt}, the Lagrangian function associated to Equation (30) restricted to Equation (3) is

$$L_{m,\zeta}(M, \Phi, V) = \sum_{i=1}^{N} \sum_{k=1}^{K} \phi_{kt}^{1-m} \ \mu_{ikt}^{m} d_{ikt}^2 + \frac{\zeta}{(m-1)} \ln \left(\sum_{i=1}^{N} \sum_{k=1}^{K} \phi_{kt}^{1-m} \mu_{ikt}^{m} \right) - \lambda_t \left(\sum_{i=1}^{K} \phi_{kt} - 1 \right) \tag{39}$$

Taking the derivative of Equation (39) with respect to ϕ_{kt} and equaling to zero, results in

$$\phi_{kt} = \left[\frac{1-m}{\lambda_t} \left(\sum_{i=1}^{N} \mu_{ikt}^{m} d_{ikt}^2 + \frac{\zeta \left(\sum_{i=1}^{N} \mu_{ikt}^{m} \right)}{(m-1) \left(\sum_{i=1}^{N} \sum_{g=1}^{K} \phi_{gt}^{1-m} \mu_{igt}^{m} \right)} \right) \right]^{\frac{1}{m}} \quad \forall k = 1, \ldots, K \tag{40}$$

Now, at this point of iteration t, it is possible to consider that the membership degrees μ_{ikt}, $i = 1, \ldots, N, k = 1, \ldots, K$, are known. However, the ratios' variables ϕ_{gt}, $g = 1, \ldots, K$, are still unknown, and have thus to be approximated by the corresponding $\phi_{g(t-1)}$. Then, we get

$$\phi_{kt} = \left[\frac{1-m}{\lambda_t} \left(\sum_{i=1}^{N} \mu_{ikt}^{m} d_{ikt}^2 + \frac{\zeta \left(\sum_{i=1}^{N} \mu_{ikt}^{m} \right)}{(m-1) \left(\sum_{i=1}^{N} \sum_{k=1}^{K} \phi_{g(t-1)}^{1-m} \mu_{igt}^{m} \right)} \right) \right]^{\frac{1}{m}} \quad \forall k = 1, \ldots, K \tag{41}$$

By imposing that the ϕ_{kt}, $k = 1, \ldots, K$, as given by Equation (41), fulfill the constraint in Equation (3), it follows that

$$1 = \sum_{k=1}^{K} \left(\left[\frac{1-m}{\lambda_t} \left(\sum_{i=1}^{N} \mu_{ikt}^{m} d_{ikt}^2 + \frac{\zeta \left(\sum_{i=1}^{N} \mu_{ikt}^{m} \right)}{(m-1) \left(\sum_{i=1}^{N} \sum_{g=1}^{K} \phi_{g(t-1)}^{1-m} \mu_{igt}^{m} \right)} \right) \right]^{\frac{1}{m}} \right) \tag{42}$$

and by solving in Equation (42) for the Lagrange multiplier λ_t , it is

$$\lambda_t = (1-m)\left[\sum_{k=1}^{K}\left[\sum_{i=1}^{N}\mu_{ikt}^m d_{ikt}^2 + \frac{\zeta\left(\sum_{i=1}^{N}\mu_{ikt}^m\right)}{(m-1)\left(\sum_{i=1}^{N}\sum_{g=1}^{K}\phi_{g(t-1)}^{1-m}\mu_{igt}^m\right)}\right]^{\frac{1}{m}}\right]^m \quad (43)$$

Equation (32) is then obtained by replacing Equation (43) in Equation (41). Equation (32) is also a local minimum as it is not difficult to check that the second derivative of Equation (39) is greater than zero. □

A first remark regarding the updating formulae provided by Theorem 1 has to refer to the approximation of the membership degrees and observations ratios of a given iteration t by the corresponding degrees and ratios of the previous iteration $t-1$. As shown in the previous proof, for a given k, Equations (31) and (32) for μ_{ikt} and ϕ_{kt}, respectively, depend on μ_{ikt} and ϕ_{kt} themselves. We simply avoid this difficulty by relying on the corresponding values from the previous iteration to substitute these unknown quantities, leading to Equations (36) and (41). This is a common practice when deriving iterative formulae for fuzzy clustering analysis, although one does not typically find the same unknown quantity at both sides of an expression. However, as shown by the computational study described in the next section, in this case, this strategy seems to work well in practice.

A second remark concerns the initialization of the membership degrees and observations ratios per cluster. In a similar way to the methods reviewed in Section 2, the mentioned interdependence between these quantities, as reflected in Equations (31) and (32), makes it necessary to provide alternative expressions for computing them at the beginning of the process, i.e., for $t = 0$. To this aim, we propose Equation (44) to initialize membership degrees:

$$\mu_{ik0} = \left(\frac{1}{K-1}\right)\left(1 - d_{ik0}^2 / \sum_{g=1}^{K} d_{ig0}^2\right),\ i = 1,\ldots,N,\ k = 1,\ldots,K \quad (44)$$

as well as Equation (45) for the observations ratios:

$$\phi_{k0} = \frac{1}{N}\sum_{i=1}^{N}\left[1 + sign\left(\mu_{ik0} - \max_{g}\mu_{ig0}\right)\right],\ k = 1,\ldots,K \quad (45)$$

The motivation behind Equation (44) is that it provides a normalized membership degree, which is inversely proportional to the squared distance of observation i to the kth initial seed. The factor $1/(K–1)$ enforces the degrees of a given observation for the different clusters to sum up to 1. Then, from these initial membership degrees, Equation (45) computes the proportion of observations that would be (crisply) assigned to each cluster by following the maximum-rule. These proportions obviously add up to 1 too. Therefore, membership degrees need to be initialized prior to observations ratios, and in turn the former need initial seeds to have been previously drawn. In the computational study presented in next section, we employed Equations (44) and (45) for the initialization of all those methods considering observations ratios per cluster.

A third remark refers to the amount of information gathered by the proposed method in the updating formulae for both membership degrees and observations ratios. This point gets clearer by comparing Equations (31) and (32) with the corresponding updating expressions of the Tsallis divergence-based method [45] described above, Equations (22) and (23). Notice that, in these last formulae, the Tsallis method modifies squared distances d_{ikt}^2 by a fixed addend $\zeta/(m–1)$. In contrast, in Equations (31) and (32), this fixed addend is, respectively, multiplied by $1/\left(\sum_{i=1}^{N}\sum_{k=1}^{K}\phi_{k(t-1)}^{1-m}\mu_{ik(t-1)}^m\right)$ and $\left(\sum_{i=1}^{N}\mu_{ikt}^m\right)/\left(\sum_{i=1}^{N}\sum_{k=1}^{K}\phi_{k(t-1)}^{1-m}\mu_{ik(t-1)}^m\right)$, thus taking into account membership degrees and observations ratios in the additive modification of the distance effect. Thus, the pro-

posed method seems to somehow take advantage of a greater amount of information in the updating steps than the Tsallis divergence-based method. This interesting feature provides a further motivation for the proposed methods based on Rényi divergence.

Finally, we summarize the algorithmic steps of the proposed fuzzy clustering method with Rényi relative entropy and cluster size, see Algorithm 1 below. Initial seeds V_{k0} can be selected by means of any of the several initialization methods available (see, e.g., [25]). The stopping criteria are determined through a convergence threshold $\varepsilon \in (0,1)$ and a maximum number of iterations t_{max}. Let us recall that, besides these last and the number K of clusters to be built, the algorithm also needs the input of the fuzzifier parameter $m > 1$ and the regularization parameter $\zeta > 0$.

Algorithm 1. Fuzzy C-Means with Rényi Relative Entropy and Cluster Size

Inputs: Dataset $X = (X_i)_{i=1,\ldots,N}$, number of clusters K, stopping parameters ε and t_{max}, fuzzifier parameter m and regularization parameter ζ.
Step 1: Draw initial seeds V_{k0}, $k = 1,\ldots,K$.
Step 2: Compute distances $d^2(X_i, V_{k0})$, $i = 1,\ldots,N,\ k = 1,\ldots,K$.
Step 3: Initialize μ_{ik0} by Equation (44), $i = 1,\ldots,N,\ k = 1,\ldots,K$.
Step 4: Initialize ϕ_{k0} by Equation (45), $k = 1,\ldots,K$.
Step 5: Assign $t = t + 1$, and update centroids V_{kt} by Equation (7), $k = 1,\ldots,K$.
Step 6: Compute distances d^2_{ikt}, $i = 1,\ldots,N,\ k = 1,\ldots,K$.
Step 7: Membership degrees μ_{ikt} are updated by Equation (31), $i = 1,\ldots,N,\ k = 1,\ldots,K$.
Step 8: Observations ratios per cluster are updated by Equation (32), $k = 1,\ldots,K$.
Step 9: IF $\max_{ik}\left(\left|\mu_{ikt} - \mu_{ik(t-1)}\right|\right) < \varepsilon$ or $t + 1 > t_{max}$ then stop; ELSE return to Step 5.
Output: Final centroid matrix V_t and partition matrix M_t.

3.2. Fuzzy C-Means with Rényi Divergence, Cluster Sizes and Gaussian Kernel Metric

An extension of the previous method is attained by substituting the calculation of Euclidean distances between observations and centroids in Equation (29) with a more flexible Gaussian kernel metric, defined through the kernel function $K(X_i,\ V_k) = exp\left(-\gamma\|X_i - V_k\|^2\right)$, $\gamma > 0$. As exposed in Section 2.3, by using this Gaussian kernel function, the calculation of squared distances d^2_{ik} between transformed observations and centroids in $\mathbb{R}^{n'}$ (a higher-dimensional space than the native space $\mathbb{R}^n$ originally containing the observations, $n' > n$) can be easily carried out through the expression $d^2_{ik} = 2(1 - K(X_i,\ V_k))$, without need of explicitly applying an embedding transformation $\varphi : \mathbb{R}^n \to \mathbb{R}^{n'}$. By working in such a higher-dimensionality setting, improved separability between clusters may be achieved, potentially enabling the formation of better partitions.

Then, this Gaussian kernel-based extension of the proposed method using Rényi divergence-based regularization with cluster sizes seeks to minimize the objective function

$$\min_{M,\Phi,V} J_{m,\zeta,\gamma}(M,\Phi,V) = \min_{M,\Phi,V} \sum_{i=1}^{N}\sum_{k=1}^{K} \phi_{kt}^{1-m}\mu_{ikt}^{m} 2(1 - K(X_i,\ V_{kt})) + \frac{\zeta}{(m-1)} ln\left(\sum_{i=1}^{N}\sum_{k=1}^{K} \phi_{kt}^{1-m}\mu_{ikt}^{m}\right) \tag{46}$$

subject to Equations (1) and (3). Let us remark that, due to the introduction of the kernel function $K(X_i,\ V_k) = exp\left(-\gamma\|X_i - V_k\|^2\right)$, a third parameter $\gamma > 0$ is added in Equation (46) to the two already considered in the objective function of Equation (30), i.e., the fuzzifier parameter $m > 1$ and the regularization parameter $\zeta > 0$.

As above, the Lagrange multipliers method can be applied on the objective function, Equation (46), and the mentioned constraints in order to derive iterative formulae for the membership degrees μ_{ikt} and the observations ratios per cluster ϕ_{kt}, as well as for updating the centroids V_{kt}. The same considerations above exposed regarding the orthogonality of the constraints and the lack of restrictions on the transformed centroids still hold.

Theorem 2. *The application of the Lagrange multipliers method to the objective function Equation (46) constrained by Equations (1) and (3) provides the following solutions* $\forall t > 0$.

For updating the centroids Equation (29):

$$V_{kt} = \frac{\sum_{i=1}^{N} \mu_{ik(t-1)}^{m} K\Big(X_i,\ V_{k(t-1)}\Big)\ X_i}{\sum_{i=1}^{N} \mu_{ik(t-1)}^{m} K\Big(X_i,\ V_{k(t-1)}\Big)},\ \forall k = 1, \ldots, K$$

For membership degrees:

$$\mu_{ikt} = \frac{\left[\phi_{k(t-1)}^{1-m}\left(2(1 - K(X_i,\ V_{kt})) + \frac{\zeta\left(\sum_{i=1}^{N}\sum_{k=1}^{K}\phi_{k(t-1)}^{1-m}\mu_{ik(t-1)}^{m}\right)^{-1}}{(m-1)}\right)\right]^{\frac{-1}{m-1}}}{\sum_{g=1}^{K}\left(\phi_{g(t-1)}^{1-m}\left(2\big(1 - K\big(X_i,\ V_{gt}\big)\big) + \frac{\zeta\left(\sum_{i=1}^{N}\sum_{g=1}^{K}\phi_{g(t-1)}^{1-m}\mu_{ig(t-1)}^{m}\right)^{-1}}{(m-1)}\right)\right)^{\frac{-1}{m-1}}},\ \forall i, k \quad (47)$$

For observations ratios per cluster:

$$\phi_{kt} = \frac{\left[\sum_{i=1}^{N}\mu_{ikt}^{m}2(1 - K(X_i, V_{kt})) + \frac{\zeta(\sum_{i=1}^{N}\mu_{ikt}^{m})}{(m-1)(\sum_{i=1}^{N}\sum_{k=1}^{K}\phi_{k(t-1)}^{1-m}\mu_{ikt}^{m})}\right]^{\frac{1}{m}}}{\sum_{g=1}^{K}\left[\sum_{i=1}^{N}\mu_{igt}^{m}2(1 - K(X_i, V_{gt})) + \frac{\zeta(\sum_{i=1}^{N}\mu_{igt}^{m})}{(m-1)(\sum_{i=1}^{N}\sum_{g=1}^{K}\phi_{g(t-1)}^{1-m}\mu_{igt}^{m})}\right]^{\frac{1}{m}}}, \forall k = 1, \ldots, K \quad (48)$$

Proof. Only the derivation of the expression for centroid updating is now addressed, as the expressions for membership degrees and observations ratios per cluster are derived in an analogous way to Theorem 1, simply substituting $d_{ikt}^2 = (X_i - V_{kt})^T\ (X_i - V_{kt})$ with $d_{ikt}^2 = 2\big(1 - K\big(X_i,\ V_{gt}\big)\big)$.

Thus, taking into account that $d_{ikt}^2 = 2\big(1 - K\big(X_i,\ V_{gt}\big)\big) = 2\Big(1 - exp\Big(-\gamma\|X_i - V_k\|^2\Big)\Big)$ and the centroids are unconstrained, the derivative of Equation (46) with respect to V_{kt} is equaled to zero, leading to

$$\sum_{i=1}^{N} \phi_{kt}^{1-m}\mu_{ikt}^{m}\ [-2\gamma(X_i - V_{kt})K(X_i,\ V_{kt})] = 0,\ \forall k = 1, \ldots, K \quad (49)$$

Equation (29) easily follows by solving for V_{kt} in Equation (49) and replacing μ_{ikt} with $\mu_{ik(t-1)}$ from the previous iteration, as well as V_{kt} with $V_{k(t-1)}$ in the calculation of the kernel function. It is easy to check that Equation (29) provides a local minimum as the second derivative of Equation (46) is positive. □

Similar remarks to those exposed above for the previous method also apply for this extension. Particularly, now previous-iteration centroids intervene in centroid updating, due to V_{kt} being replaced with $V_{k(t-1)}$ for the computation of Equation (29). Since the kernel function values $K(X_i, V_{kt})$ have to be computed at each iteration $t \geq 0$ to initialize and update both membership degrees and observations ratios per cluster, centroid updating can be carried out more efficiently by keeping in memory the $K(X_i, V_{kt})$ values until centroid updating at iteration $t + 1$ is performed. Likewise, both membership degrees and observations ratios need to be initialized through alternative expressions to Equations (44) and (45). To this aim, we simply substitute $d_{ik0}^2 = (X_i - V_{k0})^T(X_i - V_{k0})$ in Equation (44) with $K(X_i, V_{k0})$ in order to initialize membership degrees, leading to Equation (50),

$$\mu_{ik0} = \left(\frac{1}{K-1}\right)\left(1 - K(X_i, V_{k0}) / \sum_{k=1}^{K} K(X_i, V_{k0})\right),\ i = 1, \ldots, N,\ k = 1, \ldots, K \quad (50)$$

Equation (45) is applied without modifications for initializing observations ratios per cluster.

The Algorithm 2 corresponding to this extension, presented below, follows basically the same lines as the non-extended method (Algorithm 1), simply substituting the computation of distances $d^2(X_i, V_{kt})$ with that of kernel values $K(X_i, V_{kt})$, which entails considering a kernel parameter $\gamma > 0$ in addition to those needed by the previous method.

Algorithm 2. Fuzzy C-Means with Rényi Divergence, Cluster Sizes and Gaussian Kernel Metric

Inputs: Dataset $X = (X_i)_{i=1,\ldots,N}$, number of clusters K, stopping parameters ε and t_{max}, fuzzifier parameter m, regularization parameter ζ, and kernel parameter γ.
Step 1: Draw initial seeds V_{k0}, $k = 1, \ldots, K$.
Step 2: Compute kernel values $K(X_i, V_{k0})$, $i = 1, \ldots, N$, $k = 1, \ldots, K$.
Step 3: Initialize μ_{ik0} by Equation (50), $i = 1, \ldots, N$, $k = 1, \ldots, K$.
Step 4: Initialize ϕ_{k0} by Equation (45), $k = 1, \ldots, K$.
Step 5: Assign $t = t + 1$, and update centroids V_{kt} by Equation (29), $k = 1, \ldots, K$.
Step 6: Compute kernel values $K(X_i, V_{kt})$, $i = 1, \ldots, N$, $k = 1, \ldots, K$.
Step 7: Membership degrees μ_{ikt} are updated by Equation (47), $i = 1, \ldots, N$, $k = 1, \ldots, K$.
Step 8: Observations ratios per cluster are updated by Equation (48), $k = 1, \ldots, K$.
Step 9: IF $\max_{ik}\left(\left|\mu_{ikt} - \mu_{ik(t-1)}\right|\right) < \varepsilon$ or $t + 1 > t_{max}$ then stop; ELSE return to Step 5.
Output: Final centroid matrix V_t and partition matrix M_t.

4. Computational Study

This section describes the setup (Section 4.1) and results (Section 4.2) of the computational study carried out to analyze the performance of the proposed fuzzy clustering methods.

4.1. Experimental Configuration

The objective of this study was to analyze the performance on real data of the proposed fuzzy clustering method and its kernel extension in comparison to that of methods with a similar approach, particularly those employing relative entropy measures different to Rényi divergence. To this aim, we conducted a computational experiment consisting of the application of 10 non-hierarchical, prototype-based clustering methods on 20 well-known supervised classification datasets. All clustering methods were inputted with the known, right number K of classes or cluster to build on each dataset. A genetic differential evolution algorithm was employed to search for the optimal parameters of each method (except the K-means, which has no parameters) on each dataset. For the best parametric configuration of each method returned from the genetic algorithm, a usual supervised accuracy metric was computed comparing the method's output with the actual class labels of each dataset. This process was replicated 10 times for each method and dataset, each time using different initial seeds drawn by the k-means++ method [30] (all methods using the same seeds). The mean accuracy of each method on each dataset was finally obtained.

To provide a variety of references, besides the methods here proposed, we included in this study the standard K-means, Fuzzy C-means (FCM), and FCM with cluster sizes (FCMA) methods, along with the methods using Kullback–Leiber and Tsallis divergences described in Section 2.2, as well as their extensions through a Gaussian kernel metric. The included methods and their nomenclature and ranges considered in the search for optimal parameters are presented in Table 1.

Table 1. The range of values of the parameters of each method.

	Method\Parameter	Fuzzier (m)	Entropy (ζ)	Gaussian Kernel (γ)
K-means	K-means			
FCM	Fuzzy C-Means	[1.075,6]		
FCMA	Fuzzy C-Means with cluster size	[1.075,6]		
kFCM	Kernel Fuzzy C-Means	[1.075,6]		[0.001,10]
kFCMA	Kernel Fuzzy C-Means with cluster size	[1.075,6]		[0.001,10]
EFCA	Kullback–Leiber relative entropy with cluster size		[0.000001,10]	
Tsallis	Tsallis relative entropy with cluster size	[1.075,6]	[0.000001,10]	
kTsallis	Kernel Tsallis relative entropy with cluster size	[1.075,6]	[0.000001,10]	[0.001,10]
Renyi	Rényi relative entropy with cluster size	[1.075,6]	[0.000001,10]	
kRenyi	Kernel Rényi relative entropy with cluster size	[1.075,6]	[0.000001,10]	[0.001,10]

The 20 datasets selected for this experiment constitute usual benchmarks for cluster analysis, and were downloaded from three online repositories:

UCI machine learning website [67]. Datasets: Brest Cancer Coimbra (BCC), Blood Transfusion Service Center (Blood), Lenses, Vertebral column (Vertebral-column-2 and Vertebral-column-3), and Wholesale customers (WCD-channel and WCD-region).

Website of Speech and Image Processing Unit, School of Computing at University of Eastern Finland [68]. Datasets: Flame, Jain, Compound, and Aggregation.

Keel website at University of Granada [69,70]. Datasets: Appendicitis, Bupa, Haberman, Hayes–Roth, Hear, Iris, Sonar, and Spectfheart.

The main characteristics of these 20 datasets are summarized in Table 2. Let us remark that all these classification datasets are of supervised nature, that is, they provide the actual class label for each pattern/instance along with the available explanatory variables or features. However, these class labels were only used in this experiment once an optimal parametric configuration was selected by the differential evolution algorithm for each method, replication, and dataset. That is, each method was then fitted to the corresponding dataset with those optimal parameters, and the cluster assignments obtained (through the maximum rule) from the output partition matrix were only then compared to the actual class labels. This allowed obtaining the proportion of correctly classified instances, or classification accuracy, for each combination of method, dataset, and replication. As usual, a maximum-matching procedure was applied at this point in order to select the matching between output cluster numbers and actual classes providing the best accuracy. Finally, the mean accuracy along the 10 replications was computed for each method and dataset.

Table 2. Datasets' characteristics.

Datasets	Datum	Features	Classes	Datasets	Datum	Features	Classes
Aggregation	788	2	7	Iris	150	4	3
Appendicitis	106	9	2	Jain	373	2	2
BCC	116	10	2	Lenses	24	4	3
Blood	748	5	2	Sonar	208	60	2
Bupa	345	6	2	Spectfheart	267	44	2
Compound	399	2	6	Vertebral-column-3	310	6	3
Flame	240	2	2	Vertebral-column-2	310	6	2
Haberman	306	3	2	WCD-channel	440	8	2
Hayes–Roth	160	4	3	WCD-region	440	8	3
Heart	270	13	2	WDBC	569	30	2

Differential Evolution Algorithm

As just mentioned, a differential evolution algorithm was applied in this experiment to search for the optimal parameters of each clustering method, for each combination of dataset and replication. Let us briefly recall that differential evolution [71] algorithms are a kind of evolutionary computation technique, quite similar to genetic algorithms but exhibiting a real-valued (instead of binary-valued) codification of the genotype or elements to be optimized, which also leads to some differences in how the mutation and crossover

operators are designed and implemented. The specific differential evolution algorithm we applied features self-adaption of the algorithm's factor and crossover parameters [72,73].

Next, we describe the main details of the application of the differential evolution (DE) algorithm in the context of the present computational study. Firstly, let D denote the number of parameters to be optimized for a given clustering method, as shown in Table 1 (for instance, it is $D = 1$ for the FCM, $D = 2$ for the proposed Rényi divergence-based method, and $D = 3$ for its kernel extension). The DE algorithm considers a set or population of NP D-dimensional vectors $p^l = \left(p_1^l, \ldots, p_D^l\right), l = 1, \ldots, NP$, each providing a feasible solution of the optimization problem associated to the parameters of the clustering method. Here, we set the population size as $NP = 15D$, providing a greater set of solution vectors as the number of parameters to be searched for increases.

Along the execution of the DE algorithm, the initial population of solution vectors evolves by certain mechanisms, trying to find better solutions that reflect on a better performance of the clustering method, without being stuck in a local minimum. Particularly, the applied DE algorithm consists of the following four steps, where the last three are repeated until a preset maximum number of iterations ($G_{max} = 2000$ in our experiment) is reached:

At the initialization step, NP D-dimensional vectors p^l are randomly generated, each coordinate p_d^l being uniformly drawn in the range designated for the dth parameter to be optimized, $d = 1, \ldots, D$ (see Table 1).

At the mutation step, we applied the so-called rand-to-rand/1 strategy, which for each $l = 1, \ldots, NP$ creates a *mutant* solution vector v^l from three randomly selected vectors of the current population, following the expression

$$v^l = p^{r1} + F_l\left(p^{r2} - p^{r3}\right) \tag{51}$$

where $r1$, $r2$, and $r3$ are randomly selected integers in the interval [1,NP] and $F_l \in (0, 1)$ is a scale parameter that controls the amount of diversity being introduced in the population of solutions through mutation (see Appendix A).

At the crossover step, for each $l = 1, \ldots, NP$, a candidate solution vector q^l is generated, in such a way that for each $d = 1, \ldots, D$ it is assigned $q_d^l = v_d^l$ with probability CR_l, and $q_d^l = p_d^l$ otherwise. The crossover parameter $CR_l \in (0, 1)$ thus determines the frequency with which current population solutions and mutations mix to create diverse candidate solutions.

Finally, at the selection step, for each $l = 1, \ldots, NP$, the parametric configurations of the considered clustering method represented by the current solution p^l and the candidate solution q^l are compared through their *fitness*, that is, their performance at providing an adequate partition on the considered dataset and replication. Such fitness was assessed by the Xie–Beni validation measure [74] (see Appendix B)

$$XB(p) = \frac{\sum_{i=1}^{N}\sum_{k=1}^{K} \mu_{ik}^m d_{ik}^2}{N\left(\min\limits_{l \neq k} \|V_l - V_k\|^2\right)} \tag{52}$$

where μ_{ik}, V_k and d_{ik}^2, respectively, denote the final membership degrees, centroids, and distances from observations to centroids returned by the considered clustering method after being run with the parametric configuration p (see Appendix C). Therefore, at this selection step, the corresponding clustering method has to be run twice for each l, once with parametric configuration p^l and once with q^l, respectively, producing fitness assessments $XB\left(p^l\right)$ and $XB\left(q^l\right)$. Then, p^l is included in the DE algorithm next iteration's population if $XB\left(p^l\right) < XB\left(q^l\right)$, otherwise q^l is used instead.

We used the self-adaption scheme proposed in [75] for the DE algorithm's scale F_l and crossover CR_l parameters. This scheme allows these parameters to vary for each

population's solution vector and along the execution of the DE algorithm, enabling a more efficient search of the solution space. Particularly, at a given iteration $G = 1, \ldots, G_{max}$, the DE parameters were obtained as follows:

Scale: Draw a value $F_{l,temp}$ from a $N(0.5, 0.25)$ distribution. Then, compute the scale parameter to be used at iteration G for the lth population member as

$$F_{l,G} = \begin{cases} 1, \ if \ F_{l,temp} > 1 \\ 0.1, \ if \ F_{l,temp} < 0.1 \\ F_{l,temp}, \ otherwise \end{cases} \quad (53)$$

Crossover: Draw two U(0,1) random numbers $rand_1$ and $rand_2$, and assign $CR_{l,temp} = 0.1 + rand_1 \cdot 0.8$. Then, compute the crossover parameter to be used at iteration G for the lth population member as

$$CR_{l,G} = \begin{cases} CR_{l,temp} \ if \ rand_2 < 0.1 \\ CR_{l,G-1} \ \ otherwise \end{cases} \quad (54)$$

4.2. Results

Table 3 presents the mean accuracy, along the 10 replications, attained by each clustering method on each dataset considered in this experiment. The best performance for each dataset is highlighted in bold. A first impression of these results is that two methods seem to stand out among the others, each achieving the best performance on four datasets: the kernel extension of the proposed Rényi divergence-based method (kRenyi) and the method based on Tsallis divergence (Tsallis).

Table 3. Results of the computational study. Table cells present the mean accuracy of the 10 experimental replications attained by each method on each dataset.

Datasets	K-Means	FCM	kFCM	FCMA	kFCMA	EFCA	Tsallis	kTsallis	Renyi	kRenyi
Flame	84.83	84.17	81.96	89.00	88.96	78.29	86.71	86.79	69.63	**89.13**
Jain	88.20	87.13	89.28	90.19	90.13	80.62	**90.48**	85.74	83.75	90.13
Compound	58.22	55.21	59.45	50.75	50.13	46.39	62.98	47.07	65.54	**66.92**
Aggregation	**77.58**	67.77	53.57	57.54	57.35	45.01	42.21	47.92	63.65	67.16
Haberman	50.65	51.96	52.29	50.72	50.88	**64.38**	51.27	57.35	62.48	50.59
Sonar	54.81	**55.34**	54.90	54.09	53.80	53.37	52.60	53.51	54.09	53.94
Hayes–Roth	41.44	43.94	43.06	42.44	42.13	42.19	44.19	42.38	40.88	**44.94**
Bupa	54.61	50.72	55.77	55.65	55.65	45.57	**57.80**	57.39	56.23	55.65
Appendicitis	81.04	74.53	**87.74**	78.21	76.51	83.49	82.64	80.94	80.00	77.74
Iris	77.47	**85.33**	63.73	85.20	82.07	61.00	80.07	71.67	75.87	84.87
Lenses	51.67	50.83	49.17	57.08	52.50	**58.75**	55.42	57.50	53.33	51.67
Heart	71.00	78.89	**81.11**	79.85	79.85	74.96	79.59	54.44	80.00	80.00
Vertebral-column-3	47.29	58.71	55.84	57.65	56.19	48.19	**59.65**	51.52	50.39	58.16
Vertebral-column-2	65.55	66.77	64.84	67.94	67.87	65.19	54.19	67.39	60.71	**68.13**
WDBC	**92.79**	**92.79**	85.89	92.32	91.76	64.90	91.81	79.47	74.71	92.14
BCC	51.64	50.86	50.26	52.67	52.67	48.62	51.38	**55.17**	52.76	52.59
WCD-channel	56.57	56.36	56.14	57.18	57.66	60.39	57.23	**62.16**	57.75	57.43
WCD-region	49.55	43.82	53.64	45.73	46.50	57.45	**65.95**	52.11	52.00	44.57
Blood	58.82	56.42	59.36	53.60	53.90	**64.09**	57.09	60.80	62.01	57.19
Spectfheart	62.73	59.18	65.54	66.67	68.88	73.78	72.73	**77.83**	66.67	70.19

In order to descriptively compare the aggregated performance of the methods in all datasets in terms of central tendency measures, Table 4 shows the mean and median accuracy attained by each method along all datasets, as well as the related standard deviation. Notice then that the kRenyi method achieves the best mean and median performance, while the proposed Renyi method ranks second in terms of median accuracy. Indeed, there seems to be a clear shift of performance in terms of median accuracy between the proposed methods (Renyi and kRenyi) and the rest. Table 4 also shows that only the proposed

method (Renyi) seems to exhibit a positive synergy with the usage of a Gaussian kernel extension, as it is the only method to improve on both its mean and median accuracy when applying such extension. Let us also note that the proposed Renyi method is the one with the lowest accuracy variability along the different datasets, as measured by the standard deviations shown in Table 4. In this sense, the combination of a relatively high median accuracy with a relatively small variability may be taken to suggest that the proposed Renyi method presents a quite robust performance.

Table 4. Mean and median accuracy attained by each method on all datasets and related standard deviations.

	K-Means	FCM	kFCM	FCMA	kFCMA	EFCA	Tsallis	kTsallis	Renyi	kRenyi
Mean	63.82	63.54	63.18	64.22	63.77	60.83	64.8	62.46	63.12	**65.66**
Median	58.52	57.56	57.75	57.36	56.77	60.69	58.72	57.45	62.24	**62.54**
Std. Desv.	15.09	15.17	14.16	15.84	15.69	12.68	15.47	13.51	**11.48**	15.39

In order to analyze possible statistically significant differences among the methods, we applied two non-parametrical multiple comparisons statistical tests, the Friedman ranks test [76,77] and the Friedman aligned ranks tests [78]. Both tests constitute non-parametric alternatives to the parametric ANOVA test for multiple comparisons. The main difference between both Friedman tests is that the former (Friedman ranks test) treats each dataset as an experimental block, thus considering that the different samples provided by the results of each method are dependent, while the latter (Friedman aligned ranks test) removes this assumption and considers all samples to be independent, allowing for a certain comparability between the performances of the different methods on different datasets (see Appendix D). That is, the former procedure tests for differences in the mean ranks controlling for differences due to datasets, while the second tests for differences in mean (aligned) ranks, without taking into account dataset differences.

Table 5 shows the mean ranks attained by each method. Notice that the kernel extension of the proposed method, kRenyi, obtains the lowest rank, consistently with the best central tendency measures already observed in Table 4. However, the related Friedman ranks test provides a p-value of 0.5089, and thus it is not significant at a significance level of $\alpha = 0.05$. Therefore, this test does not allow concluding that significant differences exist between the 10 analyzed clustering methods when controlling for datasets differences.

Table 5. Average ranks of the compared methods in the Friedman rank test.

Method	Avg. Rank
kRenyi	4.6
Tsallis	4.8
FCMA	4.85
kTsallis	5.1
Renyi	5.5
kFCM	5.7
kFCMA	5.75
FCM	6.025
K-means	6.225
EFCA	6.45

In turn, Table 6 presents the mean aligned rank computed for each method. Again, the kRenyi method obtains the lowest rank. However, now the related Friedman aligned ranks test provides a p-value of 0.0304, being therefore significant at a significance level of $\alpha = 0.05$. This leads to concluding that statistically significant differences exist among the 10 methods when performance comparability between datasets is allowed.

Table 6. Average aligned ranks of the compared methods in the Friedman aligned rank test.

Method	Avg. Aligned Rank
kRenyi	78.575
Tsallis	80.9
FCMA	95.25
kFCMA	100.625
kTsallis	103.2
Renyi	105.475
FCM	106.375
kFCM	106.45
K-means	110.3
EFCA	117.85

Due to the significance of the previous aligned ranks multiple comparisons test, and given that the proposed kRenyi method obtains the best aligned rank, we next applied a post hoc test on the results of the aligned rank test to check for statistically significant differences between the kRenyi method and the rest. That is, we take kRenyi as a control method, and study the significance of the aligned rank differences between this control and the other methods. Table 7 presents the *p*-values provided by this test for each comparison versus the kRenyi method. Only the comparison of this last with *EFCA*, the Kullback–Leiber divergence-based method, is significant at a significance level of $\alpha = 0.05$, although the *p*-value obtained for the comparison with *K*-means is also small and would be significant at a significance level of $\alpha = 0.1$. Other comparisons' *p*-values are also relatively small, pointing to a potential superiority of the kRenyi method over the FCM and its kernel extension (kFCM), as well as over the proposed, non-kernel extended Renyi method or the kernel extension of the Tsallis divergence-based method (kTsallis).

Table 7. Post hoc *p*-values for the comparison of the kRenyi method versus the other methods.

Method	*p*-Value
EFCA	0.0319
K-means	0.0830
kFCM	0.1278
FCM	0.1288
Renyi	0.1416
kTsallis	0.1785
kFCMA	0.2283
FCMA	0.3623
Tsallis	0.8989

To compare the computational efficiency of the proposed methods, Table 8 presents the mean times expended by each considered method on the benchmark datasets along the 10 experimental replications. Let us remark that the mean computational costs shown in this table were calculated by applying each method on each dataset with the corresponding parametric configuration returned by the DE algorithm at each replication. That is, these computational costs do not include execution times of the DE algorithm; instead, they represent the mean execution times of each single method on each dataset. In this sense, it is important to notice that actual execution times of a method on a given dataset may vary considerably from one replication to another depending on the interaction between the initial seeds and the parametric configuration selected for the method at each replication: in some replications, convergence of a method may occur much more quickly than in other replications. This helps explain the relatively large standard deviations presented in Table 8, as well as the fact that lower complexity methods, e.g., the K-means, do not obtain systematically lower computational costs than greater complexity methods, e.g., FCM or the proposed Renyi and kRenyi methods. This variability also makes it difficult to extract

bold conclusions from the data presented in Table 8. Although the Renyi method exhibits the largest mean cost along all datasets (64.2 ms), its kernel extension kRenyi (which obtained the best results in term of classification accuracy) almost halves its execution times (39.3 ms), even presenting the best cost performance on several datasets. In turn, this global mean execution time of the kRenyi method only doubles that of the K-means (18 ms) and represents a 50% increment with respect to that of the FCM (26.3 ms). Furthermore, the kRenyi method improves on the mean computational cost of other divergence-based methods, such as Tsallis and kTsallis. The observed differences in computational costs among all the considered methods are rather small on the benchmark datasets of this experiment, and they are rarely greater than an order of magnitude.

Table 8. Computational cost of the different methods on the benchmark datasets. Table cells present the mean time in milliseconds (ms) expended by each method on each dataset with the parametric configuration returned by the DE algorithm for the 10 replications. In brackets are the standard deviations obtained for the 10 experimental replications. The lowest mean cost for each dataset is highlighted in bold.

Datasets	K-Means	FCM	kFCM	FCMA	kFCMA	EFCA	Tsallis	kTsallis	Renyi	kRenyi
Aggregation	56 (37.9)	132 (18.1)	91.2 (15.8)	122 (34.1)	152 (59.1)	17.2 (25.8)	62 (52.9)	126 (11.4)	179.6 (92.2)	**3.2 (1.7)**
Appendicitis	14.8 (13.3)	17.6 (7.4)	**10 (3.9)**	38.8 (9.4)	42.8 (8.2)	16.8 (16.9)	55.6 (10.7)	32 (23.3)	53.2 (12.9)	46 (9.5)
BBC	12.4 (4.8)	**10.8 (3.3)**	25.2 (8.2)	13.2 (9.8)	17.6 (16.9)	18.8 (15.9)	39.2 (18.7)	18.8 (19.3)	17.2 (11.2)	12 (2.7)
Blood	18.8 (7.3)	**9.2 (3.3)**	12 (2.7)	50 (4.7)	56 (7.5)	23.6 (12.6)	57.2 (8.7)	97.6 (36.7)	71.2 (22.8)	64 (6)
Bupa	17.6 (5.7)	**15.6 (4.8)**	17.2 (8.2)	29.2 (6.3)	35.6 (8.7)	26.8 (17.4)	19.6 (22.7)	44.4 (18.5)	30.8 (7.3)	45.2 (13.2)
Compound	32 (12.2)	76 (10.2)	52.8 (12.2)	90.8 (32.1)	90 (27.6)	10.8 (16.4)	71.2 (4.9)	114 (38)	140.8 (46.5)	**2.4 (2.1)**
Flame	12.4 (3)	12.4 (1.3)	12.8 (5.9)	38 (6.6)	36 (2.7)	**10 (11)**	40.8 (7)	51.2 (6.5)	50.4 (11.3)	43.2 (1.7)
Haberman	14.4 (6.3)	9.2 (1.9)	10.4 (2.1)	35.6 (3)	40.8 (6.2)	**4.8 (3.7)**	41.2 (4.2)	50.4 (9.3)	49.6 (10.2)	45.2 (5)
Hayes–Roth	17.6 (6.9)	45.6 (11)	37.6 (13.5)	47.6 (14.2)	55.6 (22.8)	24 (18.5)	71.2 (20.4)	18 (25.7)	70 (12)	**0.4 (1.3)**
Heart	14 (6.9)	**9.2 (1.9)**	12 (3.3)	20.8 (14.3)	14.8 (3.8)	22.4 (21.4)	17.2 (6)	28.8 (28)	14.4 (4.3)	16.8 (3.2)
Iris	15.6 (7.2)	**13.6 (3.4)**	23.6 (14.7)	49.2 (12.2)	43.2 (10.6)	15.2 (20.6)	55.2 (17.3)	54.4 (13.8)	58 (16.5)	56.4 (11.7)
Jain	11.6 (3.5)	10.4 (2.8)	**8 (1.9)**	40 (6.3)	40.8 (1.7)	8.8 (10.8)	45.2 (8.7)	51.2 (10.6)	44 (8.6)	48.4 (2.3)
Lenses	8.4 (3)	31.6 (5.5)	31.2 (4.9)	28 (13.2)	26 (19.9)	14.8 (16.8)	14.8 (6.8)	8.8 (13)	18 (16.6)	**0.8 (1.7)**
Sonar	**16.4 (4.8)**	23.2 (14.9)	37.6 (11.8)	35.2 (7.7)	34 (17.7)	23.6 (14.3)	35.6 (8.1)	30.8 (25.2)	48.4 (19)	45.2 (22.2)
Spectfheart	13.2 (4.6)	14.4 (2.1)	13.6 (2.1)	28.4 (4.4)	28.8 (12.8)	38.8 (14.4)	68 (17.6)	**7.6 (12.9)**	32.8 (7.3)	48.8 (22.8)
Vertebral-column-2	12.4 (4.8)	9.6 (3.9)	9.6 (2.1)	40.8 (7)	42.4 (3.9)	**8.8 (7.5)**	62 (4.7)	44.8 (21.6)	58 (13)	49.2 (4.6)
Vertebral-column-3	20 (7.1)	34.8 (9.4)	33.6 (12.2)	46.8 (5.7)	52.4 (6.9)	**8.8 (14.2)**	60 (13.5)	70.4 (15.6)	79.6 (15.8)	58 (3.4)
WCD_channel	18.8 (26)	8 (1.9)	**5.6 (2.1)**	50.4 (11.8)	58.8 (15)	7.2 (3.2)	56.8 (12.6)	46.8 (18.1)	62.8 (15.1)	66.4 (17.9)
WCD_region	17.2 (6.3)	20.4 (4.4)	43.6 (5.5)	49.2 (13.6)	55.6 (12.1)	**2.8 (1.9)**	30.4 (28.5)	91.2 (27.2)	105.6 (10.4)	56.8 (20.7)
WDBC	**16.8 (1.7)**	22.8 (6)	23.2 (17.5)	64 (10.3)	68.8 (8.2)	60 (25.1)	70.4 (12)	73.2 (41.6)	98.8 (20.6)	76.8 (14.7)
Mean	**18**	**26.3**	**25.5**	**45.9**	**49.6**	**18.2**	**48.7**	**53**	**64.2**	**39.3**

To sum up, the results of the computational study carried out consistently point to a relatively good performance of the proposed Rényi divergence-based methods, especially in the case of the Gaussian kernel-extended kRenyi method. Although the evidence extracted from this experiment is not fully conclusive from a statistical point of view, particularly on a dataset-by-dataset basis, there seems however to be enough support to at least conclude that kRenyi performs significantly better than the Kullback–Leiber divergence-based method on an all-datasets basis. This conclusion was somehow to be expected since Rényi divergence is indeed a generalization of Kullback–Leiber divergence. Moreover, a close-to-significance superiority of kRenyi is also suggested with respect to *K*-means, FCM (with and without kernel extension), and the kernel-extended Tsallis divergence-based method. The small increment (if any) in computational cost of the proposed methods seems to compensate for the improvement in classification performance, especially in the case of the kRenyi method.

5. Conclusions

This paper delves into the usage of entropy measures as regularization functions penalizing the formation of too overlapping partitions in fuzzy clustering, building on some previous proposals applying relative entropy measures, or divergences, to that regularization aim. In this sense, this work particularly focuses on the application of Rényi divergence between fuzzy membership degrees of observations into clusters, on the one side, and observations ratios per cluster, on the other side, in the context of fuzzy clustering problems considering cluster size variables. Since Rényi divergence (as also happens with Tsallis divergence) provides a generalization of several other, more specific divergence measures, particularly of Kullback–Leiber divergence, its application in fuzzy clustering seems interesting in order to devise more general and potentially more effective

methods. This led us to the proposal of two fuzzy clustering methods exhibiting a Rényi divergence-based regularization term, the second method extending the first one through the consideration of a Gaussian kernel metric instead of the standard Euclidean distance.

An extensive computational study was also carried to illustrate the feasibility of our approach, as well as to analyze the performance of the proposed clustering methods in comparison with that of several other methods, particularly some methods also applying divergence-based regularization and Gaussian kernel metrics. The results of this study, although not fully conclusive from a statistical point of view, clearly point to a comparatively good performance of the proposed method, and particularly of its Gaussian kernel extension, which significantly improves on the performance of Kullback–Leiber divergence-based clustering methods.

Future research by the authors following this work is ongoing in the direction of studying the application on the methods here proposed of other distances or metrics different from the Euclidean and the Gaussian kernel metrics. We are particularly interested in the effect of the Mahalanobis distance on our methods to deal with non-spherical clusters and its potential synergy with Rényi-based regularization.

Author Contributions: Conceptualization, J.B. and J.M.; methodology, J.B.; software, J.B.; validation, J.M., D.V. and J.T.R.; formal analysis, J.B.; investigation, J.B.; resources, J.B.; data curation, J.B.; writing—original draft preparation, J.B. and D.V.; writing—review and editing, J.T.R., D.V. and J.B.; visualization, J.B.; supervision, J.M., J.T.R. and D.V.; project administration, J.M.; and funding acquisition, J.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Government of Spain (grant PGC2018-096509-B-100), and Complutense University of Madrid (research group 910149).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Such diversity is convenient in order to adequately explore the solution space and avoid getting stuck in a local minimum. When F_l is close to 0, the population will tend to converge too quickly to a non-optimal solution with low diversity, and, when F_l is close to 1, the population will tend to converge too slowly to a non-optimal solution with low diversity as well. Therefore, F_l should be kept away from 0 or 1.

Appendix B

Notice that the Xie–Beni validation measure can be interpreted as the intra-cluster variance divided by the minimum distance between centroids, and thus it only uses information from the clustering process, not employing knowledge of the actual class labels. The motivation behind employing this validation measure as the fitness criterion of the DE algorithm instead of the objective function of the clustering method being considered is that the former provides a fitness metric external to the clustering method, avoiding overfitting issues such as shrinking the entropy regularization parameter ζ to 0.

Appendix C

The parameter m in Equation (52) is obtained as the first coordinate of the solution vector p, except for the EFCA method (which implicitly considers $m = 1$) and the K-means (which has no parameters to optimize).

Appendix D

This is reflected in the different procedures employed by either test when computing average ranks for each method: On the one hand, the Friedman ranks test proceeds by ranking the methods' performance on each dataset, i.e., from the 1st to the 10th for each dataset, and later computes the average rank of each method. On the other hand, the Friedman aligned ranks test first calculates the mean accuracy of all methods on each

dataset, and then subtracts it from each method's accuracy. This process is repeated for all datasets, and then a single ranking of all obtained differences is computed (i.e., from the 1st to the 200th, as there are 10 methods and 20 datasets). These ranks are known as aligned ranks. Finally, the average aligned rank of each method is calculated.

References

1. Anderberg, M.R. *Cluster Analysis for Application*; Academic Press: New York, NY, USA, 1972.
2. Härdle, W.; Simar, L. *Applied Multivariate Statistical Analysis*, 2nd ed.; Springer Berlin Heildeberg: Berlin/Heildeberg, Germany, 2007.
3. Johnson, J.W.; Wichern, D.W. *Applied Multivariate Statistical Analysis*; Prentice Hall: Upper Saddle River, NJ, USA, 1998.
4. Srivastava, M.S. *Methods of Multivariate Statistics*; John Wiley & Sons, Inc.: New York, NY, USA, 2002.
5. Johnson, S.C. Hierarchical clustering schemes. *Psychometrika* **1967**, *32*, 241–254. [CrossRef]
6. Ward, J.H. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* **1963**, *58*, 236–244. [CrossRef]
7. Forgy, E. Clustering analysis of multivarate data: Efficiency vs. interpretability of classification. *Biometrics* **1965**, *21*, 768–769.
8. MacQueen, J.B. Some methods of classifications and analysis of multivariate observations. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 21 June–18 July 1965; pp. 281–297.
9. Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Clustering Analysis*; John Wiley & Sons, Inc.: New York, NY, USA, 1990.
10. Park, H.-S.; Jun, C.-H. A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* **2009**, *36*, 3336–3341. [CrossRef]
11. Chen, Y. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 790–799.
12. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: An efficient data clustering method for very large databases. *ACM Sigmod Rec.* **1996**, *25*, 103–114. [CrossRef]
13. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; pp. 226–231.
14. Ankerst, M.; Breuning, M.M.; Kriegel, H.-P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmond Rec.* **1999**, *28*, 49–60. [CrossRef]
15. Schaeffer, S.E. Graph Clustering. *Comput. Sci. Rev.* **2007**, *1*, 27–64. [CrossRef]
16. Ng, A.Y.; Jordan, M.I.; Weiss, Y. On spectral clustering: Analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* **2001**, *14*, 849–856.
17. von Luxburg, U.A. Tutorial of spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [CrossRef]
18. Liu, J.; Han, J. Spectral clustering. In *Data Clustering: Algorithms and Applications*; Aggarwal, C., Reddy, C., Eds.; CRC Press Taylor and Francis Group: London, UK, 2014; pp. 177–200.
19. Wang, W.; Yang, J.; Muntz, R. STING: A statistical information grid approach to spatial data mining. In Proceedings of the 23rd VLDB Conference, Athens, Greece, 25–29 August 1997; pp. 186–195.
20. Sheikholeslami, G.; Chatterjee, S.; Zhang, A. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In Proceedings of the 24th VLDB Conference, New York, NY, USA, 24–27 August 1998; pp. 428–439.
21. Miyamoto, S.; Ichihashi, H.; Honda, K. Algorithms for fuzzy clustering. In *Methods in C-Means Clustering with Applications*; Kacprzyk, J., Ed.; Springer Berlin Heidelberg: Berlin/Heidelberg, Germany, 2008; Volume 299.
22. Kohonen, T. *Self-Organizing Maps*, 2nd ed.; Springer: Berlin, Germany, 1997.
23. Bottou, L.; Bengio, Y. Convergence properties of the k-means algorithms. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 27–30 November 1995; pp. 585–592.
24. Cebeci, Z.; Yildiz, F. Comparison of k-means and fuzzy c-means algorithms on different cluster structures. *J. Agric. Inform.* **2015**, *6*, 13–23. [CrossRef]
25. Celebi, M.E.; Kingravi, H.A.; Vela, P.A. A Comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Syst. Appl.* **2013**, *40*, 200–210. [CrossRef]
26. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A K-means clustering algorithm. *J. R. Stat. Society. Ser. C* **1979**, *28*, 100–108. [CrossRef]
27. Jain, A.K. Data Clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [CrossRef]
28. Steinley, D. K-means clustering: A half-century synthesis. *Br. J. Math. Stat. Psychol.* **2006**, *59*, 1–34. [CrossRef] [PubMed]
29. Selim, S.Z.; Ismail, M.A. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Trans. Pattern Anal. Mach. Intell.* **1984**, *6*, 81–87. [CrossRef]
30. Arthur, D.; Vassilvitskii, S. K-means++: The advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'07), New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.
31. Zadeh, L.A. Fuzzy sets. Information and control. *J. Symb. Log.* **1965**, *8*, 338–353.
32. Dunn, J.C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separed clusters. *Cybern. Syst.* **1973**, *3*, 32–57.
33. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Plenium Press: New York, NY, USA, 1981.
34. Amo, A.; Montero, J.; Biging, G.; Cutello, V. Fuzzy classification systems. *Eur. J. Oper. Res.* **2004**, *156*, 495–507. [CrossRef]

35. Bustince, H.; Fernández, J.; Mesiar, R.; Montero, J.; Orduna, R. Overlap functions. *Nonlinear Anal. Theory Methods Appl.* **2010**, *72*, 1488–1499. [CrossRef]
36. Gómez, D.; Rodríguez, J.T.; Montero, J.; Bustince, H.; Barrenechea, E. n-Dimensional overlap functions. *Fuzzy Sets Syst.* **2016**, *287*, 57–75. [CrossRef]
37. Castiblanco, F.; Franco, C.; Rodríguez, J.T.; Montero, J. Evaluation of the quality and relevance of a fuzzy partition. *J. Intell. Fuzzy Syst.* **2020**, *39*, 4211–4226. [CrossRef]
38. Li, R.P.; Mukaidono, M. A Maximun Entropy Approach to fuzzy clustering. In Proceedings of the 4th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE/IFES 1995), Yokohama, Japan, 20–24 March 1995; pp. 2227–2232.
39. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 623–656. [CrossRef]
40. Miyamoto, S.; Kurosawa, N. Controlling cluster volume sizes in fuzzy C-means clustering. In Proceedings of the SCIS & ISIS 2004, Yokohama, Japan, 21–24 September 2004; pp. 1–4.
41. Ichihashi, H.; Honda, K.; Tani, N. Gaussian Mixture PDF Approximation and fuzzy C-means clustering with entropy regulation. In Proceedings of the Fourth Asian Fuzzy Systems Symposium, Tsukuba, Japan, 31 May–3 June 2000; pp. 217–221.
42. Kullback, S.; Leibler, R.A. On information theory. *Ann. Math. Statist.* **1951**, *22*, 79–86. [CrossRef]
43. Tsallis, C. Possible Generalization of Boltzmann-Gibbs Statistics. *J. Stat. Phys.* **1988**, *52*, 478–479. [CrossRef]
44. Kanzawa, Y. On possibilistic clustering methods based on Shannon/Tsallis-entropy for spherical data and categorical multivariate data. In *Lectures Notes in Computer Science*; Torra, V., Narakawa, Y., Eds.; Springer: New York, NY, USA, 2015; pp. 115–128.
45. Zarinbal, M.; Fazel, M.H.; Turksen, I.B. Relative entropy fuzzy C-means clustering. *Inf. Sci.* **2014**, *260*, 74–97. [CrossRef]
46. Rényi, A. On measures of Entropy and information. In Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 20 June–30 July 1960; pp. 547–561.
47. Jenssen, R.; Hild, K.E.; Erdogmus, D. Clustring using Renyi's entropy. In Proceedings of the International Joint Conference on Neural Networks, Porland, OR, USA, 26 August 2003; pp. 523–528.
48. Popescu, C.C. A Clustering model with Rényi entropy regularization. *Math. Rep.* **2009**, *11*, 59–65.
49. Ruspini, E. A new approach to clustering. *Inform. Control* **1969**, *15*, 22–32. [CrossRef]
50. Pal, N.R.; Bezdek, J.C. On cluster validity for the fuzzy C-means model. *IEEE Trans. Fuzzy Syst.* **1995**, *3*, 370–379. [CrossRef]
51. Yu, J. General C-means clustering model. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1197–1211.
52. Yu, J.; Yang, M.S. Optimality test for generalized FCM and its application to parameter selection. *IEEE Trans. Fuzzy Syst.* **2005**, *13*, 164–176.
53. Jain, A.; La, M. Data clustering: A user's dilemma. *Lect. Notes Comput. Sci.* **2005**, *3776*, 1–10.
54. Huang, D.; Wang, C.D.; Lai, J.H. Locally weighted ensemble clustering. *IEEE Trans. Cybern.* **2018**, *48*, 1460–1473. [CrossRef]
55. Huang, D.; Wang, C.D.; Lai, J.H.; Kwoh, C.K. Toward multidiversified ensemble clustering of high-dimensional data: From subspaces to metrics and beyond. *IEEE Trans. Cybern.* **2021**. [CrossRef] [PubMed]
56. Hartle, R.V. Transmission of information. *Bell Syst. Tech. J.* **1928**, *7*, 535–563. [CrossRef]
57. Bennett, C.H.; Bessette, F.; Brassard, G.; Salvail, L.; Smolin, J. Experimental quantum cryptography. *J. Crytol.* **1992**, *5*, 3–28. [CrossRef]
58. Cover, T.M.; Thomas, J. *Elements of Information*; Jonh Wiley & Sons: New Jersey, NJ, USA, 2006.
59. Gray, R.M. *Entropy and Information Theory*; Springer: New York, NY, USA, 2010.
60. Van Erven, T.; Harremoës, P. Rényi divergence and Kullback-Leibler divergence. *IEEE Trans. Inf. Theory* **2014**, *7*, 3797–3820. [CrossRef]
61. Bhattacharyya, A. On a measure of divergence between two multinomial populations. *Sankhya Indian J. Stat.* **1946**, *7*, 401–406.
62. Ménard, M.; Courboulay, V.; Dardignac, P.A. Possibilistic and probabilistic fuzzy clustering: Unification within the framework of the non-extensive thermostatistics. *Pattern Recognit.* **2003**, *36*, 1325–1342. [CrossRef]
63. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory—COLT '92, Pittsburgh, PA, USA, 27–29 July 1992; pp. 144–152.
64. Graves, D.; Pedrycz, W. Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study. *Fuzzy Sets Syst.* **2010**, *4*, 522–543. [CrossRef]
65. Vert, J.P. *Kernel Methods in Computational Biology*; The MIT Press: Cambridge, MA, USA, 2004.
66. Wu, K.L.; Yang, M.S. Alternative C-means clustering algorithms. *Pattern Recognit.* **2002**, *35*, 2267–2278. [CrossRef]
67. UCI Machine Learning Repository, University of California. Available online: https://archive.ics.uci.edu/ml/index.php (accessed on 26 January 2021).
68. School of Computing University of Eastern Finland. Available online: http://cs.joensuu.fi/sipu/datasets (accessed on 26 January 2021).
69. Keel. Available online: www.keel.es (accessed on 26 January 2021).
70. Alcalá-Fernández, J.; Fernández, A.; Luego, J.; Derrac, J.; García, S.; Sánchez, L. KEEL data-mining software tool: Data set repository, integration of agorithms and experimental analysis framework. *J. Mult. Valued Log. Soft Comput.* **2011**, *17*, 255–287.
71. Price, K.; Storn, R. *Differential Evolution—A Simple and Efficient Adaptative Scheme for Global Optimization over Continuous Spaces*; Technical Report TR-95-012; International Computer Science Institute: Berkeley, CA, USA, 1995.
72. Brest, J.; Greiner, S.; Boskovic, B.; Mernik, M.; Zumer, V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **2006**, *10*, 646–657. [CrossRef]

73. Qinqin, F.; Xuefeng, Y. Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies. *IEEE Trans. Cybern.* **2015**, *46*, 2168–2267.
74. Xie, X.L.; Beni, G. A Validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 841–847. [CrossRef]
75. Liu, B.; Yang, H.; Lancaster, J.M. Synthesis of coupling matrix for diplexers based on a self-adaptive differential evolution algorithm. *IEEE Trans. Microw. Theory Tech.* **2018**, *66*, 813–821. [CrossRef]
76. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 674–701. [CrossRef]
77. Friedman, M. A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.* **1940**, *11*, 86–92. [CrossRef]
78. Hodges, J.L.; Lehmann, E.L. Ranks methods for combination of independent experiments in analysis of variance. *Ann. Math. Stat.* **1962**, *33*, 482–497. [CrossRef]