Pattern Recognition 143 (2023) 109749

Contents lists available at ScienceDirect

Pattern Recognition

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

Gaussian kernel fuzzy c-means with width parameter computation and regularization

Eduardo C. Simões, Francisco de A. T. de Carvalho*

Centro de Informática, Universidade Federal de Pernambuco, Av. Jornalista Anibal Fernandes s/n - Cidade Universitaria, Recife-PE, CEP 50740-560, Brazil

ARTICLE INFO

Article history:
Received 8 March 2022
Revised 29 November 2022
Accepted 5 June 2023
Available online 8 June 2023

Keywords:
Gaussian kernel fuzzy clustering
Kernelization of the metric
Width parameter
Entropy regularization

ABSTRACT

The conventional Gaussian kernel fuzzy c-means clustering algorithms require selecting the width hyper-parameter, which is data-dependent and fixed for the entire execution. Not only that, but these parameters are the same for every dataset variable. Therefore, the variables have the same importance in the clustering task, including irrelevant variables. This paper proposes a Gaussian kernel fuzzy c-means with kernelization of the metric and automated computation of width parameters. These width parameters change at each iteration of the algorithm and vary from each variable and from each cluster. Thus, this algorithm can re-scale the variables differently, thus highlighting those that are relevant to the clustering task. Fuzzy clustering algorithms with regularization have become popular due to their high performance in large-scale data clustering, robustness for initialization, and low computational complexity. Because the width parameters of the variables can also be controlled by entropy, this paper also proposes Gaussian kernel fuzzy c-means algorithms with kernelization of the metric and automated computation of width parameters through entropy regularization. To demonstrate their usefulness, the proposed algorithms are compared with the conventional KFCM-K algorithm and previous algorithms that automatically compute the width parameter of the Gaussian kernel.

1. Introduction

Clustering is a branch of unsupervised learning methods that separates elements from a dataset in clusters based on the dissimilarity between the elements or between the elements and the representative, either selected or calculated, of each group. It is expected that similar elements are kept in the same cluster while dissimilar elements are kept in different clusters. Clustering has been extensively applied in various areas such as image segmentation, pattern recognition, medicine, engineering, signal processing, business, geology, and taxonomy [1].

Clustering techniques can be roughly divided into two categories: hierarchical and partitioning methods. Hierarchical methods provide an output represented by a nested sequence of partitions of the data; their output is a hierarchical structure of groups known as a dendrogram. Partitioning methods aim to obtain a single partition of the data in a fixed number of clusters, typically based on the optimization of a suitable objective function. Partitioning methods were mainly developed in two different ways: hard and fuzzy. In hard clustering, the clusters are non-overlapping: Any data point belongs to one and only one cluster. The most popular partitioning hard clustering method is the k-means algorithm (KM). In fuzzy clustering, a data point belongs to all clusters with a certain fuzzy membership degree. The fuzzy memberships allow the algorithm to deal with overlapping clusters and can distinguish between points at the core and at the boundary of the cluster. The fuzzy c-means (FCM) is the most popular fuzzy clustering algorithm. Ref. [2] provides a recent survey of the various clustering algorithms.

Both KM and FCM use the Euclidean distance to compute the dissimilarity between the data points and the cluster representatives. However, they do not perform well with datasets in which the clusters are non-hyper-spherical and/or linearly non-separable. To overcome such a limitation, several approaches that can handle complex data have been proposed, including kernel-based clustering methods.

The kernel fuzzy c-means algorithm (KFCM) [3] uses the so-called “kernel trick” that allows computing Euclidean distances in the feature space through kernels in the original space [4]. Since the first developments of the KFCM algorithm, several kernel-based fuzzy clustering algorithms have been proposed. Ref. [5] presented a Gaussian kernel-based fuzzy c-means algorithm with a spatial bias correction that can automatically learn the parameters by a prototype-driven learning scheme. A fuzzy c-means algorithm with

* Corresponding author.
E-mail addresses: ecs4@cin.ufpe.br (E. C. Simões), fatc@cin.ufpe.br (F.d.A. T. de Carvalho).

https://doi.org/10.1016/j.patcog.2023.109749
0031-3203/

divergence kernel was proposed by [6] that was successfully applied to model the mean and covariance information of audio signals. A kernel generalized fuzzy c-means algorithm with spatial constraints presented by Zhao et al. [7] was successfully applied in the segmentation of gray images corrupted by noise. Ref. [8] presented an improved fuzzy c-means algorithm based on a trade-off weighted fuzzy factor and a kernel metric that was successfully applied to image segmentation. Paper [9] successfully integrated the improved possibilistic c-Means with multiple kernel learning settings under the supervision of side information.

More recently, an adaptive kernel-based fuzzy c-means clustering using spatial neighborhood constraints has been proposed to improve the efficiency of image segmentation [10]. Paper [11] introduced an entropy-like divergence kernel into fuzzy c-means algorithm with weighted local information constraints and an adaptive entropy-like noise distance for image segmentation. A partially supervised kernel-induced rough fuzzy clustering was proposed by Talukdara and Halder [12] for brain tissue segmentation in which the rough and fuzzy set handles the overlapping, vagueness, and indiscernibility of distinguishing different tissue regions. A nonlocal information self-integration optimization algorithm based on kernel-based fuzzy local information clustering algorithm, suitable for image segmentation, was proposed by Song et al. [13]. Finally, Ref. [14] proposed a kernelized total Bregman divergence-driven possibilistic fuzzy clustering algorithm with local neighborhood information aiming at the segmentation of the gray-scale image.

Kernel-based clustering algorithms are classified into two categories, depending on the position of the prototypes [4]. One is the kernelization of the metric, where the prototypes are placed in the original feature space, and the other is the clustering in feature space, in which the cluster representatives are located in the kernel space, not in the original space, and can only be obtained indirectly. This paper addresses kernel-based clustering with kernelization of the metric.

The Gaussian kernel is one of the most popular kernel functions used in kernel-based clustering algorithms. This kernel function requires tuning a single hyper-parameter, that is, the width hyper-parameter. This hyper-parameter is tuned once and for all, and it is the same for all variables. Thus, implicitly the conventional Gaussian kernel c-means assumes that the variables are equally re-scaled; therefore, they have the same importance to the clustering task.

The performance of the Gaussian kernel-based clustering algorithm depends on the selection of the width hyper-parameter, which needs to be optimized. Traditionally, empirical and cross-validation approaches have been used for that optimization [15]. Moreover, few approaches have been proposed to automatically learn the width hyper-parameter. Ref. [16] proposed Stretched KFCM, a kernel-based fuzzy clustering algorithm with an optimized width parameter that is updated according to the gradient method during each iteration process. Ref. [17] proposed FLeCK, a fuzzy clustering algorithm for dissimilarity data. FLeCK learns a suitable dissimilarity measure that is based on a Gaussian kernel function with cluster-dependent width parameters. These width parameters are learned by optimizing both the intra- and inter-cluster distances. Later, Ref. [18] provided a kernel fuzzy c-means clustering algorithm with kernelization of the metric and automated computation of width parameters using an adaptive Gaussian kernel. In this kernel-based clustering algorithm, the width parameters become variables of the suitable objective functions, change at each algorithm iteration, and differ from variable to variable but are the same for all the clusters. Thus, this algorithm is able to re-scale the variables differently, thus highlighting those that are the relevant for the clustering task.

The first contribution of this paper is to provide a new variant of the Gaussian kernel fuzzy c-means clustering algorithm of the Ref. [18], in which the width parameters become variables of a suitable objective function, changing at each algorithm iteration, differing from variable to variable and from cluster to cluster.

More recently, much attention has been directed to maximum entropy fuzzy clustering algorithms due to their low sensitivity to initialization and high clustering performance on huge datasets compared with fuzzy c-means algorithms [19]. In this regard, the second contribution of this paper is to provide Gaussian kernel fuzzy c-means clustering algorithms with kernelization of the metric and automated computation of the width hyper-parameters through entropy regularization. This paper provides the following main contributions:

- We provide a new variant of the Gaussian kernel fuzzy c-means clustering algorithm of the Ref. [18]. In this variant, the width parameters change at each algorithm iteration and differ from variable to variable, and from cluster to cluster.
- The new variant is based on a Gaussian kernel function with a local width parameters vector for each cluster, where each variable in each cluster has its own width parameter. In addition, the width parameters are computed by considering the product constraint, i.e., by considering that their product is equal to one.
- Finally, new objective functions are introduced and an algebraic solution to compute the elements that minimize the objective functions is offered.
- As to the second main contribution of the paper, a Gaussian kernel fuzzy c-means clustering algorithm with kernelization of the metric and width parameter regularization is proposed. The width parameters change at each iteration and may vary from one cluster to another.
- Our proposals add new positive entropy terms that allow an automatic adjustment of the width parameters during the optimization process to the objective function. We consider two variants of the proposed algorithm that are given by two Gaussian kernel functions, one with a global width parameters vector where each variable has its own width parameter and the other with a local width parameters vector where each variable in each cluster has its own width parameter. In addition, according to the entropy regularization terms, the width parameters are computed by considering the sum constraint, i.e., by considering that their sum is equal to one.
- Finally, new objective functions are introduced and an algebraic solution to compute the elements that minimize the objective functions is offered.

This paper is organized as follows. Section 2 reviews some basic concepts about kernel functions and describes the conventional kernel fuzzy c-Means clustering algorithm with kernelization of the metric closely related to this work. Section 3 presents the proposed Gaussian kernel fuzzy c-means with kernelization of the metric, automatic width parameters computation, and regularization. Section 4 compares the proposed algorithm with the conventional KFCM-K algorithm and with previous algorithms that automatically compute the width parameter of the Gaussian kernel [16–18]. Finally, Section 5 brings our final remarks.

2. Conventional kernel fuzzy c-means clustering algorithms

This section briefly passes on the basic concepts for kernel functions and the traditional Gaussian kernel fuzzy c-means clustering algorithm with kernelization of the metric.

Let $E = \{e_1, \ldots, e_n\}$ be a set of n objects described by p real-valued variables. Let $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a non-empty set where the object $e_k\,(1 \leq k \leq n)$ is represented by a vector $\mathbf{x}_k = (x_{k1}, \ldots, x_{kp}) \in \mathrm{R}^p$. A function $\mathcal{K} : \mathcal{D} \times \mathcal{D} \to \mathrm{R}$ is called a positive definite Kernel (or Mercer kernel) if, and only if, $\mathcal{K}$ is symmetric

(i.e., $\mathcal{K}(\mathbf{x}_k, \mathbf{x}_l) = \mathcal{K}(\mathbf{x}_l, \mathbf{x}_k)$) and if the following inequality holds [4]:

$$\sum_{l=1}^{n}\sum_{k=1}^{n} c_l c_k \mathcal{K}(\mathbf{x}_l, \mathbf{x}_k) \geq 0, \forall n \geq 2 \quad (1)$$

where $c_l, c_k \in \mathbb{R}, l, k = 1, \ldots, n$.

Let $\Phi : \mathcal{D} \rightarrow \mathcal{F}$ be a nonlinear mapping from the input space $\mathcal{D}$ to a high dimensional feature space $\mathcal{F}$. By applying the mapping Φ, the inner product $\mathbf{x}_l^T \mathbf{x}_k$ in the input space is mapped to $\Phi(\mathbf{x}_l)^T \Phi(\mathbf{x}_k)$ in the feature space. The basic notion in the kernel approaches is that the non-linear mapping Φ does not need to be explicitly specified since each Mercer kernel can be expressed as $\mathcal{K}(\mathbf{x}_l, \mathbf{x}_k) = \Phi(\mathbf{x}_l)^T \Phi(\mathbf{x}_k)$ [4].

One of the most important aspects of applications is that the so-called distance kernel trick can be used to compute Euclidean distances in $\mathcal{F}$ without explicitly knowing Φ [4,20]:

$$\begin{aligned} ||\Phi(\mathbf{x}_l) - \Phi(\mathbf{x}_k)||^2 &= (\Phi(\mathbf{x}_l) - \Phi(\mathbf{x}_k))^T (\Phi(\mathbf{x}_l) - \Phi(\mathbf{x}_k)) \\ &= \mathcal{K}(\mathbf{x}_l, \mathbf{x}_l) - 2\mathcal{K}(\mathbf{x}_l, \mathbf{x}_k) + \mathcal{K}(\mathbf{x}_k, \mathbf{x}_k) \end{aligned} \quad (2)$$

There are two major variants of the clustering strategies of kernel fuzzy c-means. One is based on the kernelization of the metric [3,4], where the clustering algorithms search for representatives of the clusters in the original space, and where the distances between objects and representatives in the feature space are obtained by means of kernels. The other, referred to as clustering in feature space [4,21], computes the squared Euclidean distances between the objects and the cluster representatives in the high dimensional kernel space $\mathcal{F}$ without explicitly computing the clusters representatives by means of the distance kernel trick. This paper addresses the kernelization of the metric variant.

2.1. Gaussian kernel fuzzy c-means with kernelization of the metric

From an initial solution, the kernel fuzzy c-means with kernelization of the metric (named KFCM-K) iteratively provides a fuzzy partition of E into c fuzzy clusters and c representatives (called prototypes) of the fuzzy clusters in two steps, by minimizing an appropriate objective function, referred to as J_{KFCM-K}, which gives the total heterogeneity of the fuzzy partition calculated as the sum of the heterogeneity in each fuzzy cluster, given by:

$$J_{KFCM-K} = \sum_{i=1}^{c}\sum_{k=1}^{n} (u_{ki})^m ||\Phi(\mathbf{x}_k) - \Phi(\mathbf{g}_i)||^2 \quad (3)$$

$$\text{s.t.} u_{ki} \geq 0, \sum_{i=1}^{c} u_{ki} = 1; \quad (4)$$

Where $\mathbf{g}_i = (g_{i1}, \ldots, g_{ip}) \in \mathbb{R}^p$ is the prototype of the ith fuzzy cluster $(1 \leq i \leq c)$ computed in the original space, $u_{ki}\,(1 \leq k \leq n)$ is the membership degree of the object k into fuzzy cluster i, and $m \in (1, \infty)$ is a parameter that controls the fuzziness of membership for each object k.

The Gaussian kernel, the most widely used in literature, is hereinafter considered:

$$\mathcal{K}(\mathbf{x}_l, \mathbf{x}_k) = \exp\left\{-\frac{||\mathbf{x}_l - \mathbf{x}_k||^2}{2\sigma^2}\right\} = \exp\left\{-\frac{1}{2}\sum_{j=1}^{p}\frac{1}{\sigma^2}(x_{lj} - x_{kj})^2\right\}, \quad (5)$$

Where σ^2 is the width hyper-parameter of the Gaussian kernel. Note that σ^2 is the same for all the variables and it is estimated once and for all, for example, such that $2\sigma^2$ is the average of the 0.1 and 0.9 quantiles of $||\mathbf{x}_i - \mathbf{x}_k||^2$, $i, k = 1, \ldots, p$, $i \neq k$ [22]. In this case, $\mathcal{K}(\mathbf{x}_k, \mathbf{x}_k) = 1, \forall k$, $\mathcal{K}(\mathbf{g}_i, \mathbf{g}_i) = 1, \forall i$, and $\|\Phi(\mathbf{x}_k) - \Phi(\mathbf{g}_i)\|^2 = 2 - 2\mathcal{K}(\mathbf{x}_k, \mathbf{g}_i)$; therefore, the objective function J_{KFMC-K} becomes:

$$J_{KFCM-K} = \sum_{i=1}^{c}\sum_{k=1}^{n} (u_{ki})^m (2 - 2\,\mathcal{K}(\mathbf{x}_k, \mathbf{g}_i)) \quad (6)$$

From an initial solution, the KFCM-K alternates two steps until the convergence. In the first step, the fuzzy partition is kept fixed. The objective function J_{KCM-K} is optimized regarding the prototypes. Thus, after setting the partial derivatives of J_{KFCM-K} w.r.t. $\mathbf{g}_i$ and after applying some algebra, the cluster prototypes are obtained as follows:

$$\mathbf{g}_i = \frac{\sum_{k=1}^{n} (u_{ki})^m \mathcal{K}(\mathbf{x}_k, \mathbf{g}_i)\mathbf{x}_k}{\sum_{k=1}^{n} (u_{ki})^m \mathcal{K}(\mathbf{x}_k, \mathbf{g}_i)}. \quad (7)$$

In the second step, the fuzzy cluster prototypes are kept fixed. Using the method of Lagrange multipliers with the restrictions that $\sum_{i=1}^{c} u_{ki} = 1$ and $u_{ki} \geq 0$, we obtain $\mathcal{L}_{KFCM-K} = J_{KFCM-K} - \sum_{k=1}^{n} \gamma_k \left(\sum_{i=1}^{c} u_{ki} - 1\right)$. After setting the partial derivatives of $\mathcal{L}_{KFCM-K}$ w.r.t. u_{ki} and γ_k to zero and applying some algebra, the membership degree u_{ki} is computed according to

$$u_{ki} = \left[\sum_{h=1}^{c}\left(\frac{1 - \mathcal{K}(\mathbf{x}_k, \mathbf{g}_i)}{1 - \mathcal{K}(\mathbf{x}_k, \mathbf{g}_h)}\right)^{\frac{1}{m-1}}\right]^{-1} \quad (8)$$

These two steps are repeated until the convergence of the KFCM-K algorithm.

3. Gaussian kernel fuzzy c-means with kernelization of the metric, automatic width parameters computation, and regularization

This section addresses the Gaussian kernel fuzzy c-means algorithms with kernelization of the metric, automatic width parameters computation, and regularization. At each iteration of the algorithms, these width parameters shift from one variable to another and may vary from one cluster to another. First, we present the previous variant of Ref. [18] in which the width parameters differ from one variable to another but are the same for all the clusters. Then, we propose a new variant in which the width parameters vary from one variable to another and from one cluster to another. In addition, these algorithms can highlight the relevant variables for the purpose of clustering.

Finally, this section provides Gaussian kernel fuzzy c-means algorithms with kernelization of the metric, and automatic width parameters computation through regularization.

3.1. Gaussian kernel fuzzy c-means with width parameters computation.

From an initial solution, in three steps, the Gaussian kernel fuzzy c-means clustering algorithm with kernelization of the metric and automatic width parameters computation (named KFCM-K-W) iteratively provides a matrix $\mathbf{U} = (u_{ki})_{\substack{1 \leq k \leq n \\ 1 \leq i \leq c}}$ of membership degrees, width parameters for the variables and a matrix of prototypes $\mathbf{G} = (\mathbf{g}_1, \ldots, \mathbf{g}_c) = (g_{ij})_{\substack{1 \leq i \leq c \\ 1 \leq j \leq p}}$ of the fuzzy clusters by the minimization of a suitable objective function, denoted as $J_{KFCM-K-W}$, which provides the total heterogeneity of the fuzzy partition computed as the sum of the heterogeneity in each fuzzy cluster, given by

$$J_{KFCM-K-W} = \sum_{i=1}^{c}\sum_{k=1}^{n} (u_{ki})^m \|\Phi(\mathbf{x}_k) - \Phi(\mathbf{g}_i)\|^2 \quad \text{s.t.} \quad u_{ki} \geq 0 \text{ and } \sum_{i=1}^{c} u_{ki} = 1 \quad (9)$$

Where $m > 1$ is the fuzzifier hyper-parameter that determines the level of cluster fuzziness.

Table 1
Computation of the width parameters: Lagrangian functions of the variants of the KFCM-K-W algorithm.

Variants of KFCM-K-W	Lagrangian functions
KFCM-K-W.1	$L_{KFCM-K-W.1} = 2\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\left(1-\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)\right) - \omega\left(\prod_{j=1}^{p}\frac{1}{s_j^2}-1\right)$
KFCM-K-W.2	$L_{KFCM-K-W.2} = 2\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\left(1-\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)\right) - \sum_{i=1}^{c}\omega_i\left(\prod_{j=1}^{p}\frac{1}{s_{ij}^2}-1\right)$

The computation of $\|\Phi(\mathbf{x}_k)-\Phi(\mathbf{g}_i)\|^2$ involves using the distance kernel trick since the non-linear mapping Φ is not explicitly known. Two variants of the KFCM-K-W algorithm given by two Gaussian kernel functions [23] are considered here:

(a) the previous KFCM-K-W.1 variant introduced in Ref. [18] is based on a Gaussian kernel function with a global width parameters vector $\mathbf{s}=(s_1^2,\ldots,s_p^2)$:

$$\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_l,\mathbf{x}_k)=\exp\left\{-\frac{1}{2}\sum_{j=1}^{p}\frac{1}{s_j^2}(x_{lj}-x_{kj})^2\right\} \tag{10}$$

Note that each variable has its own width parameter s_j^2 $(1\le j\le p)$. In this case: $||\Phi(\mathbf{x}_k)-\Phi(\mathbf{g}_i)||^2 = \mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{x}_k) - 2\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)+\mathcal{K}^{(\mathbf{s})}(\mathbf{g}_i,\mathbf{g}_i)$. Since $\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{x}_k)=1,\ \forall k$, $\mathcal{K}^{(\mathbf{s})}(\mathbf{g}_i,\mathbf{g}_i)=1,\forall i$, and $||\Phi(\mathbf{x}_k)-\Phi(\mathbf{g}_i)||^2 = 2-2\,\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)$, the objective function becomes:

$$J_{KFCM-K-W.1}=\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m(2-2\,\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)) \quad \text{s.t.} \quad u_{ki}\ge 0 \text{ and } \sum_{i=1}^{c}u_{ki}=1 \tag{11}$$

(b) The new KFCM-K-W.2 variant is based on a Gaussian kernel function with, for each cluster, a local width parameters vector $\mathbf{s}_i=(s_{i1}^2,\ldots,s_{ip}^2)$ $(1\le i\le c)$:

$$\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_l,\mathbf{x}_k)=\exp\left\{-\frac{1}{2}\sum_{j=1}^{p}\frac{1}{s_{ij}^2}(x_{lj}-x_{kj})^2\right\} \tag{12}$$

Note that each variable in each cluster has its own width parameter s_{ij}^2 $(1\le i\le c;\ 1\le j\le p)$. Also, in this case, $||\Phi(\mathbf{x}_k)-\Phi(\mathbf{g}_i)||^2 = \mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{x}_k) - 2\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)+\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{g}_i,\mathbf{g}_i)$. Since $\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{x}_k)=1,\ \forall k$, $\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{g}_i,\mathbf{g}_i)=1,\forall i$, and $||\Phi(\mathbf{x}_k)-\Phi(\mathbf{g}_i)||^2 = 2-2\,\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)$, the objective function becomes:

$$J_{KFCM-K-W.2}=\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m(2-2\,\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)) \quad \text{s.t.} \quad u_{ki}\ge 0 \text{ and } \sum_{i=1}^{c}u_{ki}=1 \tag{13}$$

3.1.1. The optimization steps of the two variants of the KFCM-K-W algorithm

In the variants of the KFCM-K-W algorithm, the global vector of width parameters $\mathbf{s}$ or the local vectors of width parameters $\mathbf{s}_i$ $(1\le i\le c)$, the matrix of prototypes $\mathbf{G}$ and the matrix $\mathbf{U}$ of membership degrees are obtained interactively from an initial solution in three steps by minimizing their respective objective functions.

Step 1: Computation of the width parameters This step provides the optimal solution to the computation of the width parameters.

In step 1 of the variants of the KFCM-K-W algorithm, the matrix of prototypes $\mathbf{G}$ and the matrix $\mathbf{U}$ of membership degrees are kept fixed. We use the method of Lagrange multipliers either with the restriction that $\prod_{j=1}^{p}\frac{1}{s_j^2}=1$ (global width parameters) or with the restriction that $\prod_{j=1}^{p}\frac{1}{s_{ij}^2}=1$ (local width parameters). Note that another useful restriction [24] is to impose $\sum_{j=1}^{p}\left(\frac{1}{s_j^2}\right)=1$. However, this latter method requires introducing and tuning an additional hyper-parameter in this optimization step, and, for sake of brevity, it is not considered here. Table 1 provides the Lagrangian functions of each variant of the KFCM-K-W algorithm.

Then, we compute the partial derivatives of the Lagrangian functions of Table 1 w.r.t ω, $\frac{1}{s_j^2}$, ω_i and $\frac{1}{s_{ij}^2}$. From setting these partial derivatives to zero and applying some algebra, Eqs. (14a) and (14b) are, respectively, the optimal solution to $\frac{1}{s_j^2}$ and $\frac{1}{s_{ij}^2}$ for the variants KFCM-K-W.1 and KFCM-K-W.2 of the KFCM-K-W algorithm.

$$\frac{1}{s_j^2}=\frac{\left\{\prod_{h=1}^{p}\left[\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)(x_{kh}-g_{ih})^2\right]\right\}^{\frac{1}{p}}}{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)(x_{kj}-g_{ij})^2}. \tag{14a}$$

$$\frac{1}{s_{ij}^2}=\frac{\left\{\prod_{h=1}^{p}\left[\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)(x_{kh}-g_{ih})^2\right]\right\}^{\frac{1}{p}}}{\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)(x_{kj}-g_{ij})^2}. \tag{14b}$$

Note that according to Eq. (14a), the closer the objects are to the set of cluster representatives for a variable j, the higher is $\frac{1}{s_j^2}$ (and therefore, the lower the width hyper-parameter s_j^2). Moreover, according to Eq. (14b), the closer the objects are to the representative of a cluster i for a variable j, the higher is $\frac{1}{s_{ij}^2}$ (and therefore, the lower the width hyper-parameter s_{ij}^2).

Step 2: Computation of the fuzzy cluster prototypes This step provides the optimal solution to the computation of the fuzzy cluster representatives.

In step 2 of the variants of the KFCM-K-W algorithm, the global vector of width parameters $\mathbf{s}$ or the local vectors of width parameters $\mathbf{s}_i$ $(1\le i\le c)$, and the matrix $\mathbf{U}$ of membership degrees are kept fixed. The objective functions $J_{KFMC-K-W.1}$ and $J_{KFMC-K-W.2}$ (see, respectively, Eqs. (11) and (13)) are optimized regarding the prototypes. From $\frac{\partial J_{KFMC-K-W.1}}{\partial \mathbf{g}_i}=0$ and $\frac{\partial J_{KFMC-K-W.2}}{\partial \mathbf{g}_i}=0$ and after applying some algebra, Eqs. (15a) and (15b) are, respectively, the optimal solution to the cluster prototypes $\mathbf{g}_i$ for the variants KFCM-K-W.1 and KFCM-K-W.2 of the KFCM-K-W algorithm.

$$\mathbf{g}_i=\frac{\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)\mathbf{x}_k}{\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)},\ (1\le i\le c) \tag{15a}$$

$$\mathbf{g}_i=\frac{\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)\mathbf{x}_k}{\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)},\ (1\le i\le c) \tag{15b}$$

Step 3: Computation of the membership degrees This step provides the optimal solution for the computation of the matrix $\mathbf{U}$ of membership degrees.

Table 2
Computation of the membership degrees: Lagrangian functions of the variants of the KFCM-K-W algorithm.

Variants of KFCM-K-W	Lagrangian functions
KFCM-K-W.1	$L_{KFCM-K-GW} = 2\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\left(1-\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)\right) - \sum_{k=1}^{n}\omega_k\left(\sum_{i=1}^{c}u_{ki}-1\right)$
KFCM-K-W.2	$L_{KFCM-K-GPW} = 2\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\left(1-\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)\right) - \sum_{k=1}^{n}\omega_k\left(\sum_{i=1}^{c}u_{ki}-1\right)$

In step 3 of the variants of the KFCM-K-W algorithm, the matrix of prototypes **G** and either the global parameters width vector **s** or the local vectors of width parameters $\mathbf{s}_1,\ldots,\mathbf{s}_c$, are kept fixed. We use the method of Lagrange multipliers with the restriction that $\sum_{i=1}^{c}u_{ki}=1, u_{ki}\geq 0$ to compute the optimal membership degrees. Table 2 provides the Lagrangian functions of each variant of the KFCM-K-W algorithm.

Then, we compute the partial derivatives of the Lagrangian functions of Table 2 w.r.t ω_k and u_{ki}. From setting these partial derivatives to zero and applying some algebra, Eqs. (16a) and (16b) are, respectively, the optimal solution to the membership degree u_{ki} for the variants KFCM-K-W.1 and KFCM-K-W.2 of the KFCM-K-W algorithm.

$$u_{ki} = \left[\sum_{h=1}^{c}\left(\frac{2-2\,\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)}{2-2\,\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_h)}\right)^{\frac{1}{m-1}}\right]^{-1} \tag{16a}$$

$$u_{ki} = \left[\sum_{h=1}^{c}\left(\frac{2-2\,\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)}{2-2\,\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_h)}\right)^{\frac{1}{m-1}}\right]^{-1} \tag{16b}$$

Note that, according to Eqs. (16a), (16b), the membership degrees of the objects to the clusters are based on the computation of the dissimilarity between the objects and cluster representatives. A variable that has a small (big) width hyper-parameter strongly (weekly) contributes to the computation of the dissimilarity between objects and cluster representatives. That is why a variable that is re-scaled by a small-width hyper-parameter is more relevant to the clustering task than a variable that is re-scaled by a big-width hyper-parameter. Therefore, KFCM-K-W.1 highlights the important variables for the whole partition whereas KFCM-K-W.2 highlights the important variables for a specific cluster.

KFCM-K-W.1 and KFCM-K-W.2 algorithms These three steps are repeated until the convergence of the KFCM-K-W.1 and KFCM-K-W.2 variants of the KFCM-K-W clustering algorithm. Algorithm 1 summarizes these steps.

Algorithm 1 KFCM-K-W.1 and KFCM-K-W.2 algorithms.

Require:
$\mathcal{D}=\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ (dataset); c (number of clusters); T (number of iterations); ϵ (threshold).
Ensure:
KCM-K-W.1: The global vector of width parameters **s**.
KCM-K-W.2: The local vectors of width parameters $\mathbf{s}_1,\ldots,\mathbf{s}_c$.
KCM-K-W.1 and KCM-K-W.2: The cluster representatives $\mathbf{g}_1,\ldots,\mathbf{g}_c$.
KCM-K-W.1 and KCM-K-W.2: The matrix **U** of membership degrees.
Initialization
KFCM-K-W.1: Set $\frac{1}{s_j^2}\leftarrow 1,\ (1\leq j\leq p)$.KFCM-K-W.2: Set $\frac{1}{s_{ij}^2}\leftarrow 1,\ (1\leq i\leq c; 1\leq j\leq p)$.
KCM-K-W.1 and KCM-K-W.2: randomly select c distinct prototypes $g_i\in\mathcal{D}\ (1\leq i\leq c)$.
KFCM-K-W.1: Compute the membership degree u_{ki}using Equation (16a).
KFCM-K-W.2: compute the membership degree u_{ki}using Equation (16b).
KFCM-K-W.1 and KFCM-K-W.2: Compute J_{NEW} from, respectively, equations (11) and (13).
repeat
    KFCM-K-W.1 and KFCM-K-W.2: Set J_{OLD} = J_{NEW};
    Step 1: computation of the width parameters
    KFMC-K-W.1: Compute the global vector of width parameters **s** from equation (14a).
    KFCM-K-W.2: Compute the local vectors of width parameters $\mathbf{s}_1,\ldots,\mathbf{s}_c$ from equation (14b).
    Step 2: Computation of the fuzzy cluster prototypes.
    KFCM-K-W.1: Compute the cluster representatives $\mathbf{g}_1,\ldots,\mathbf{g}_c$ from equation (15a).
    KFCM-K-W.2: Compute the cluster representatives $\mathbf{g}_1,\ldots,\mathbf{g}_c$ from equation (15b).
    Step 3: Computation of the membership degrees
    KFCM-K-W.1: Compute the membership degree $u_{ki}$from equation (16a).
    KFCM-K-W.2: Compute the membership degree $u_{ki}$from equation (16b).
    KFCM-K-W.1 and KFCM-K-W.2: Compute J_{NEW} from, respectively, equations (11) and (13)
until $|J_{NEW}-J_{OLD}|<\epsilon$ or $t>T$

3.2. Gaussian kernel fuzzy c-means with width parameters regularization

This section proposes the Gaussian kernel fuzzy c-means algorithms with kernelization of the metric and automatic width parameters computation through regularization. We provide two variants of the proposed algorithm (hereafter named KFCM-K-E$_W$).

Here, we consider the computation of the width parameter with entropy regularization, either globally or locally, and under the sum restriction. In this case, the objective function, denoted as $J_{KFCM-K-E_W}$, is given by

$$J_{KFCM-K-E_W} = \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\|\Phi(\mathbf{x}_k)-\Phi(\mathbf{g}_i\|^2 + T_W\mathcal{K}_{E_W} \quad \text{s.t.} \quad u_{ki}\geq 0, \sum_{i=1}^{c}u_{ki}=1 \tag{17}$$

Where $m>1$ is the fuzzifier hyper-parameter and $\mathcal{K}_{E_W}$ are suitable width parameter entropy regularization terms. The width parameter entropy regularization terms are represented by $\mathcal{K}_{E_W}$, with $E_W\in\{E_{W.1}, E_{W.2}\}$. These regularization functions are defined as follows:

$$\mathcal{K}_{E_{W.1}} = \sum_{j=1}^{p}\left(1+\frac{1}{s_j^2}\right)\ln\left(1+\frac{1}{s_j^2}\right),\ \text{s.t.:}\frac{1}{s_j^2}\geq 0, \sum_{j=1}^{p}\frac{1}{s_j^2}=1 \tag{18a}$$

$$\mathcal{K}_{E_{W.2}} = \sum_{i=1}^{c}\sum_{j=1}^{p}\left(1+\frac{1}{s_{ij}^2}\right)\ln\left(1+\frac{1}{s_{ij}^2}\right),\ \text{s.t.:}\frac{1}{s_{ij}^2}\geq 0, \sum_{j=1}^{p}\frac{1}{s_{ij}^2}=1 \tag{18b}$$

Note that since $\frac{1}{s_j^2} \geq 0$ and $\frac{1}{s_{ij}^2} \geq 0$, we have also $\mathcal{K}_{E_{W.1}} \geq 0$ and $\mathcal{K}_{E_{W.2}} \geq 0$; therefore, $J_{KFCM-K-E_W} \geq 0$.

T_W is a hyper-parameter used to control the size of the width parameters (therefore, the relevance of the variables), such that $T_W > 0$. When T_W is high, the values of the width parameters tend to be almost the same. T_W needs to be selected in advance, generally using a grid of values combined with the computation of a suitable internal validity index [25–27].

Once again, the computation of $\|\Phi(\mathbf{x}_k) - \Phi(\mathbf{g}_i)\|^2$ involves using the distance kernel trick. Two variants of the KFCM-K-E$_W$ algorithm given by two Gaussian kernel functions [23] and two regularization functions are considered here:

(a) the KFCM-K-E$_{W.1}$ variant is based on the Gaussian kernel function with a global width parameters vector $\mathbf{s} = (s_1^2, \ldots, s_p^2)$ of Eq. (10) and the width parameters entropy regularization function of Eq. (18a). Therefore, the objective function becomes:

$$\begin{aligned} J_{KFCM-K-E_{W.1}} &= 2\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\left(1-\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)\right) \\ &+T_W\sum_{j=1}^{p}\left(1+\frac{1}{s_j^2}\right)\ln\left(1+\frac{1}{s_j^2}\right) \\ &\text{s.t.} u_{ki} \geq 0, \sum_{i=1}^{c} u_{ki} = 1; \frac{1}{s_j^2} \geq 0, \sum_{j=1}^{p}\frac{1}{s_j^2} = 1 \end{aligned} \quad (19)$$

(b) the KFCM-K-E$_{W.2}$ variant is based on the Gaussian kernel function with a local parameters width vector $\mathbf{s}_i = (s_{i1}^2, \ldots, s_{ip}^2)$, $1 \leq i \leq c$, of Eq. (12) and the width parameters entropy regularization function of Eq. (18b). Therefore, the objective function becomes:

$$\begin{aligned} J_{KFCM-K-E_{W.2}} &= 2\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\left(1-\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)\right) \\ &+T_W\sum_{i=1}^{c}\sum_{j=1}^{p}\left(1+\frac{1}{s_{ij}^2}\right)\ln\left(1+\frac{1}{s_{ij}^2}\right) \\ &\text{s.t.} u_{ki} \geq 0, \sum_{i=1}^{c} u_{ki} = 1; \frac{1}{s_{ij}^2} \geq 0, \sum_{j=1}^{p}\frac{1}{s_{ij}^2} = 1 \end{aligned} \quad (20)$$

3.2.1. The optimization steps of the two variants of the KFCM-K-E$_W$ algorithm

In the variants of the KFCM-K-E$_W$ algorithm, either the global vector of width parameters $\mathbf{s}$ or the local vectors of width parameters $\mathbf{s}_i$ ($1 \leq i \leq c$), the matrix of prototypes $\mathbf{G}$ and the matrix $\mathbf{U}$ of membership degrees are interactively obtained from an initial solution in three steps by minimizing their respective objective functions.

Step 1: Computation of the width parameters This step provides the optimal solution to the computation of the width parameters.

In step 1 of the variants of the KFCM-K-E$_W$ algorithm, the matrix of prototypes $\mathbf{G}$ and the matrix $\mathbf{U}$ of membership degrees are kept fixed. We use the method of Lagrange multipliers either with the restriction that $\sum_{j=1}^{p}\frac{1}{s_j^2} = 1, \frac{1}{s_j^2} \geq 0$ (global width parameters) or that $\sum_{j=1}^{p}\frac{1}{s_{ij}^2} = 1, \frac{1}{s_{ij}^2} \geq 0$ (local width parameters), to compute the width parameters. Table 3 provides the Lagrangian functions of each variant of the KFCM-K-E$_W$ algorithm.

Taking the partial derivatives of the Lagrangian functions of Table 3 w.r.t $\frac{1}{s_j^2}$ or w.r.t $\frac{1}{s_{ij}^2}$ and setting this partial derivatives to zero, we have

$$\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)(x_{kj}-g_{ij})^2 + T_W\ln\left(1+\frac{1}{s_j^2}\right) + T_W - \omega = 0 \quad (21a)$$

$$\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)(x_{kj}-g_{ij})^2 + T_W\ln\left(1+\frac{1}{s_{ij}^2}\right) + T_W - \omega_k = 0 \quad (21b)$$

From Eqs. (21a) and (21b) it is obtained Eqs. (22a) and (22b):

$$\frac{1}{s_j^2} = \exp\left(\frac{\omega - T_W}{T_W}\right)\exp\left\{-\left(\frac{1}{T_W}\right)\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)(x_{ij}-g_{kj})^2\right\} - 1 \quad (22a)$$

$$\frac{1}{s_{ij}^2} = \exp\left(\frac{\omega_k - T_W}{T_W}\right)\exp\left\{-\left(\frac{1}{T_W}\right)\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)(x_{ij}-g_{kj})^2\right\} - 1 \quad (22b)$$

Since $\frac{1}{s_j^2}$ must be great than zero, let J be a set of indices such that $\frac{1}{s_j^2} > 0$, i.e., $J = \{j : \frac{1}{s_j^2} > 0\}$. Likewise, since $\frac{1}{s_{ij}^2}$ must be great than zero, let J be a set of indices such that $\frac{1}{s_{ij}^2} > 0$, i.e., $J = \{j : \frac{1}{s_{ij}^2} > 0\}$. In addition, let $|J|$ be the cardinal of J. Then, from Eqs. (22a) and (22b), $\sum_{h\in J}\frac{1}{s_h^2} = 1$ and $\sum_{h\in J}\frac{1}{s_{ih}^2} = 1$, we have:

$$\exp\left(\frac{\omega - T_W}{T_W}\right)\sum_{h\in J}\left[\exp\left\{-\left(\frac{1}{T_W}\right)\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)(x_{ih}-g_{kh})^2\right\}\right] - |J| = 1 \quad (23a)$$

$$\exp\left(\frac{\omega - T_W}{T_W}\right)\sum_{h\in J}\left[\exp\left\{-\left(\frac{1}{T_W}\right)\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)(x_{ih}-g_{kh})^2\right\}\right] - |J| = 1 \quad (23b)$$

Finally, this implies that for $j \in J$,

$$\frac{1}{s_j^2} = \frac{(1+|J|)\exp\left\{-\left(\frac{1}{T_W}\right)\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)(x_{ij}-g_{kj})^2\right\}}{\sum_{h\in J}\left[\exp\left\{-\left(\frac{1}{T_W}\right)\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)(x_{ih}-g_{kh})^2\right\}\right]} - 1 \quad (24a)$$

$$\frac{1}{s_{ij}^2} = \frac{(1+|J|)\exp\left\{-\left(\frac{1}{T_W}\right)\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)(x_{ij}-g_{kj})^2\right\}}{\sum_{h\in J}\left[\exp\left\{-\left(\frac{1}{T_W}\right)\sum_{k=1}^{n}(u_{ki})^m\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)(x_{ih}-g_{kh})^2\right\}\right]} - 1 \quad (24b)$$

while $\frac{1}{s_j^2} = 0$ and $\frac{1}{s_{ij}^2} = 0$ for $j \notin J$.

Following Ref. [28], the Algorithm 2 can be used to compute the optimal $\frac{1}{s_j^2}$ values.

Likewise, the Algorithm 3 can be used to compute the optimal $\frac{1}{s_{ij}^2}$ values.

Note that according to Eq. (24a), the closer the objects are to the set of cluster representatives for a variable j, the higher is $\frac{1}{s_j^2}$ (and therefore, the lower the width hyper-parameter s_j^2). Moreover,

Table 3
Computation of the width parameters: Lagrangian functions of the variants of the KFCM-K-E$_W$ algorithm.

Variants of KFCM-K-E$_W$	Lagrangian functions
KFCM-K-E$_{W.1}$	$L_{KFCM-K-E_{W.1}} = 2\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\left(1-\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)\right)+T_W\sum_{j=1}^{p}\left(1+\frac{1}{s_j^2}\right)\ln\left(1+\frac{1}{s_j^2}\right)-\omega\left(\sum_{j=1}^{p}\frac{1}{s_j^2}-1\right)$
KFCM-K-E$_{W.2}$	$L_{KFCM-K-E_{W.2}} = 2\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\left(1-\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)\right)+T_W\sum_{i=1}^{c}\sum_{j=1}^{p}\left(1+\frac{1}{s_{ij}^2}\right)\ln\left(1+\frac{1}{s_{ij}^2}\right)-\sum_{i=1}^{c}\omega_i\left(\sum_{j=1}^{p}\frac{1}{s_{ij}^2}-1\right)$

Algorithm 2 $\frac{1}{s_j^2}$ computation.

$J \leftarrow \{1,\ldots,p\}$; $Update \leftarrow 1$;
while $Update == 1$ **do**
  $Update \leftarrow 0$;
  for $j \leftarrow 1$ to p **do**
    if $j \in J$ **then**
      Compute $\frac{1}{s_j^2}^{(t)}$ as in equation (24a);
      if $\frac{1}{s_j^2}^{(t)} \leq 0$ **then**
        $\frac{1}{s_j^2}^{(t)} \leftarrow \frac{1}{s_j^2}^{(t-1)}$; $J \leftarrow J \setminus \{j\}$; $Update \leftarrow 1$;
      end if
    end if
  end for
end while

Algorithm 3 $\frac{1}{s_{ij}^2}$ computation.

for $i \leftarrow 1$ to c **do**
  $J \leftarrow \{1,\ldots,p\}$; $Update \leftarrow 1$;
  while $Update == 1$ **do**
    $Update \leftarrow 0$;
    for $j \leftarrow 1$ to p **do**
      if $j \in J$ **then**
        Compute $\frac{1}{s_{ij}^2}^{(t)}$ as in Equation (24b);
        if $\frac{1}{s_{ij}^2}^{(t)} \leq 0$ **then**
          $\frac{1}{s_{ij}^2}^{(t)} \leftarrow \frac{1}{s_{ij}^2}^{(t-1)}$; $J \leftarrow J \setminus \{j\}$; $Update \leftarrow 1$;
        end if
      end if
    end for
  end while
end for

according to Eq. (24b), the closer the objects are to the representative of a cluster i for a variable j, the higher is $\frac{1}{s_{ij}^2}$ (and therefore, the lower the width hyper-parameter s_{ij}^2).

Step 2: Computation of the fuzzy cluster prototypes This step provides the optimal solution to the computation of the fuzzy cluster representatives.

In step 2 of the variants of the KFCM-K-E$_W$ algorithm, either the global vector of width parameters $\mathbf{s}$ or the local vectors of width parameters $\mathbf{s}_i$ ($1 \leq i \leq c$), and the matrix $\mathbf{U}$ of membership degrees are kept fixed. The objective function $J_{KFMC-K-E_W}$ is optimized regarding the prototypes.

From $\frac{\partial J_{KFMC-K-E_W}}{\partial \mathbf{g}_i} = 0$ and after applying some algebra, we show in Table 4 the optimal solution to the cluster prototypes $\mathbf{g}_i$ for each variant of the KFCM-K-E$_W$ algorithm.

Step 3: Computation of the membership degrees This step provides the optimal solution for the computation of the matrix $\mathbf{U}$ of membership degrees.

Table 4
Optimal cluster prototypes for the variants of the KFCM-K-E$_W$ algorithm.

Variants of KFCM-K-E$_W$	Cluster prototypes
KFCM-K-E$_{W.1}$	Eq. (15a)
KFCM-K-E$_{W.2}$	Eq. (15b)

In step 3 of the variants of the KFCM-K-E$_W$ algorithm, either the global parameters width vector $\mathbf{s}$ or the local vectors of width parameters $\mathbf{s}_1,\ldots,\mathbf{s}_c$, and the matrix of prototypes $\mathbf{G}$ are kept fixed. We use the method of Lagrange multipliers with the restriction that $\sum_{i=1}^{c} u_{ki} = 1, u_{ki} \geq 0$ to compute the optimal membership degrees. Table 5 provides the Lagrangian functions of each variant of the KFCM-K-E$_W$ algorithm.

Then, we compute the partial derivatives of the Lagrangian functions of Table 5 w.r.t ω_k and u_{ki}. From setting these partial derivatives to zero and applying some algebra,Table 6 shows the optimal solution to u_{ki} for each variant of the KFCM-K-E$_W$ algorithm.

KFCM-K-E$_{W.1}$ and KFCM-K-E$_{W.2}$ algorithms These three steps are repeated until the convergence of the KFCM-K-E$_{W.1}$ and KFCM-K-E$_{W.2}$ variants of the KFCM-K-E$_W$ clustering algorithm. Algorithm 4 summarizes these steps.

3.3. Convergence and time complexity

The k-means algorithm can be regarded as an Expectation-Maximization (EM) algorithm that is convergent because each EM algorithm is convergent [29]. Therefore, as the KFCM-K-W and KFCM-K-E$_W$ clustering algorithms are modified versions of the classical k-means algorithm, their convergence can be proved. Section 1 of the Supplementary Material provides proof of the convergence of the proposed algorithms.

Table 7 provides the time complexity of these clustering algorithms at each iteration, where c is the number of clusters, n is the number of objects and p is the number of variables. For optimization, the kernel function can be calculated before being used with a complexity of $O(ncp)$.

4. Empirical results

This section evaluates the performance of the proposed methods compared with the conventional KFCM-K algorithm [3,4] and other works [16–18] that compute the width parameter of the Gaussian kernel automatically. Since the program code of the algorithms of Refs. [16,17] were not available, first, we compare the proposed methods with KFCM-K and previous work of Ref. [18] through benchmark datasets selected from the UCI Machine Learning Repository and according to suitable indices that computes the quality of fuzzy and hard partitions. Then, we compare the proposed methods with other works of Refs. [16,17] using indices that compute the quality of fuzzy and hard partitions, datasets, and results reported by these papers.

Table 5
Computation of the membership degrees: Lagrangian functions of the variants of the KFCM-K-E$_W$ algorithm.

Variants of KFCM-K-E$_W$	Lagrangian functions
KFCM-K-E$_{W.1}$	$L_{KFCM-K-E_{W.1}} = 2\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\left(1-\mathcal{K}^{(\mathbf{s})}(\mathbf{x}_k,\mathbf{g}_i)\right)+T_W\sum_{j=1}^{p}\left(1+\frac{1}{s_j^2}\right)\ln\left(1+\frac{1}{s_j^2}\right)-\sum_{k=1}^{n}\omega_k\left(\sum_{i=1}^{c}u_{ki}-1\right)$
KFCM-K-E$_{W.2}$	$L_{KFCM-K-E_{W.2}} = 2\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ki})^m\left(1-\mathcal{K}^{(\mathbf{s}_i)}(\mathbf{x}_k,\mathbf{g}_i)\right)+T_W\sum_{i=1}^{c}\sum_{j=1}^{p}\left(1+\frac{1}{s_{ij}^2}\right)\ln\left(1+\frac{1}{s_{ij}^2}\right)-\sum_{k=1}^{n}\omega_k\left(\sum_{i=1}^{c}u_{ki}-1\right)$

Table 6
Optimal membership degrees for the variants of the KFCM-K-E$_W$ algorithm.

Variants of KFCM-K-E$_W$	Width parameters
KFCM-K-E$_{W.1}$	Eq. (16a)
KFCM-K-E$_{W.2}$	Eq. (16b)

Algorithm 4 KFCM-K-E$_{W.1}$ and KFCM-K-E$_{W.2}$ algorithms.

Require:
$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ (dataset); c (number of clusters); T (number of iterations); ϵ (threshold);
Ensure:
KFCM-K-E$_{W.1}$: The global hyper-parameters width vector $\mathbf{s}$.
KFCM-K-E$_{W.2}$: the local vectors of width hyper-parameters $\mathbf{s}_1, \ldots, \mathbf{s}_c$.
KFCM-K-E$_{W.1}$ and KFCM-K-E$_{W.2}$: The cluster representatives $\mathbf{g}_1, \ldots, \mathbf{g}_c$.
KFCM-K-E$_{W.1}$ and KFCM-K-E$_{W.2}$: The matrix $\mathbf{U}$ of membership degrees.
Initialization
KFCM-K-E$_{W.1}$: Set $\frac{1}{s_j^2} \leftarrow \frac{1}{p}$, $(1 \le j \le p)$; KFCM-K-E$_{W.2}$: Set $\frac{1}{s_{ij}^2} \leftarrow \frac{1}{p}$, $(1 \le i \le c; 1 \le j \le p)$.
KFCM-K-E$_{W.1}$ and KFCM-K-E$_{W.2}$: Randomly select c distinct prototypes $g_i \in \mathcal{D}$ $(1 \le i \le c)$.
KFCM-K-E$_{W.1}$: Compute the membership degree u_{ki}using Equation (16a).
KFCM-K-E$_{W.2}$: compute the membership degree u_{ki}using Equation (16b).
KFCM-K-E$_{W.1}$ and KFCM-K-E$_{W.2}$: Compute J_{NEW} using, respectively, Eqs. (19) and (20).
repeat
    KFCM-K-E$_{W.1}$, and KFCM-K-E$_{W.2}$: set $J_{OLD} = J_{NEW}$;
    Step 1: Computation of the width parameters
    KFCM-K-E$_{W.1}$: Compute the global vector of width hyper-parameters $\mathbf{s}$ using Algorithm 2.
    KFCM-K-E$_{W.2}$: Compute the local vectors of width hyper-parameters $\mathbf{s}_1, \ldots, \mathbf{s}_c$ using Algorithm 3.
    Step 2: Computation of the fuzzy cluster prototypes.
    KFCM-K-E$_{W.1}$: Compute the cluster representatives $\mathbf{g}_1, \ldots, \mathbf{g}_c$ using Equation (15a).
    KFCM-K-E$_{W.2}$: Compute the cluster representatives $\mathbf{g}_1, \ldots, \mathbf{g}_c$ using Equation (15b).
    Step 3: Computation of the membership degrees
    KFCM-K-E$_{W.1}$: Compute the membership degree u_{ki}using Equation (16a).
    KFCM-K-E$_{W.2}$: Compute the membership degree u_{ki}using Equation (16b).
    KFCM-K-E$_{W.1}$ and KFCM-K-E$_{W.2}$: Compute J_{NEW} using, respectively, Eqs. (19) and (20).
until $|J_{NEW} - J_{OLD}| < \epsilon$ or $t > T$

Table 7
Time complexity of the clustering algorithms.

Algorithms			
KFCM-K-W.1 of Ref. [18]	KFCM-K-W.2	KFCMK-E$_{W.1}$	KFCM-K-E$_{W.2}$
$\mathcal{O}(ncp)$	$\mathcal{O}(ncp)$	$\mathcal{O}(ncp)$	$\mathcal{O}(ncp)$

Table 8
Datasets types.

Type	datasets
Image	Banknote Authentication, Breast Cancer (wdbc and wpbc), Image Segmentation, Iris, Landsat, Leaf, Page Blocks, Seeds, Two Circles, Vehicle Silhouettes, and Wilt
Text	German Credit, Spambase, Urban, Congressional Voting Records, Waveform, and Zoo
Medical	Brest Tissue, Heart Disease, Liver Disorders, Pima Indians Diabetes, Thyroid Disease, Vertebral Column (2 classes and 3 classes)
Chemical	Ecoli, Glass, QSAR Biodegradation, Wine Quality (red and white) and Wine
Sensor	Abalone, Connectionist Bench: Sonar, Ionosphere, Letters, Musk (V1 and V2), Pendigits and Wall-Following Robot (readings 2 and readings 4)

Metrics for hard partitions

To compute the metrics for hard partitions, first, we obtain the hard partitions by selecting the cluster with the highest membership degree for each element. The metrics for hard partitions computed are the accuracy, the F-measure [30], the adjusted Rand index (ARI) [31], the mutual normalized information (MNI) [30], and the Entropy[32]. These metrics are external indices [1] that compare the hard partition provided by the clustering algorithms with an a priori partition provided by specific expert knowledge. Further details on these external indices are available in Section 2.2 of the Supplementary Material.

Metrics for fuzzy partitions

The metrics for fuzzy partitions computed are the modified partition coefficient [33], the fuzzy variations of the rand index, proposed by Frigui [34] and Hullermeier [35], the Jaccard index [17], and the Folkes-Mallows index [36]. Details on these metrics are available in Section 2.3 of the Supplementary Material.

4.1. Comparison with KFCM-K and KFCM-K-W.1 algorithms

This section evaluates the performance of the proposed algorithms compared with the previous KFCM-K [3] and KFCM-K-W.1 [18] algorithms.

We considered 40 datasets from the UCI Machine learning Repository [37] with different numbers of objects, variables, and a priori classes. We organized them into five categories according to how the data was obtained (image, text, medical, chemical, and sensor), as shown in Table 8.

Table 9 (in which n is the number of objects, p is the number of real-valued variables and K is the number of a priori classes) summarizes these datasets. All the datasets were previously standardized according to their average and standard deviation.

We set the number of clusters to be equal to the number of a priori classes, and the parameter m as 1.1. The parameter T_W was fixed through a grid search by selecting the value that maximizes

Table 9
Summary of the datasets.

datasets	n	p	K	datasets	n	p	K	datasets	n	p	K	datasets	n	p	K
Zoo	101	16	7	Musk V1	476	166	2	Page Blocks	5473	10	5	Vehicle Silhouettes	846	18	4
Iris	150	4	3	Musk V2	6598	166	2	Heart Disease	303	13	2	Image Segmentation	2100	19	7
Leaf	340	14	30	Landsat	2000	36	6	German Credit	1000	24	2	Vertebral Column-2C	310	6	2
Wilt	4839	5	2	Letters	1972	16	26	Voting Records	435	16	2	Vertebral Column-3C	310	6	3
Wine	178	13	3	Spambase	4601	57	2	Liver Disorders	345	6	2	QSAR Biodegradation	1055	41	2
Ecoli	336	7	8	Waveform	5000	21	3	Thyroid Disease	215	5	3	Pima Indians Diabetes	768	8	2
Glass	214	9	6	Pendigits	1166	16	10	Wine Quality: red	1599	11	6	Banknote Authentication	1372	4	2
Seeds	210	7	3	Ionosphere	351	32	2	Wine Quality: white	4898	11	7	Connectionist Bench: Sonar	208	60	2
Urban	675	148	9	Two Circles	500	2	2	Breast Cancer: wdbc	569	30	2	Wall-Following: readings 2	5456	2	4
Abalone	4177	8	3	Brest Tissue	106	9	6	Breast Cancer: wpbc	198	33	2	Wall-Following: readings 4	5546	4	4

Table 10
Average performance ranking.

Metrics	KFCM-K	KFCM-K-W.1	KFCM-K-W.2	KFCM-K-E$_{W.1}$	KFCM-K-E$_{W.2}$
Execution Time	2.600	3.388	3.725	**2.075**	3.212
Accuracy	3.500	3.425	3.000	**2.525**	2.550
F Measure	3.625	3.000	2.925	2.775	**2.675**
Adjusted Rand	3.575	3.300	3.250	2.475	**2.400**
NMI	3.725	3.425	3.250	**2.225**	2.375
Entropy	3.737	3.513	3.300	**2.125**	2.325
Rand Frigui	4.450	3.513	2.962	**2.038**	**2.038**
Rand Hullermeier	4.250	3.375	2.850	**2.237**	2.288
Modified Partition Coefficient	4.900	3.925	2.850	**1.600**	1.725
Jaccard Index	4.575	3.600	2.975	1.950	**1.900**
Folkes-Mallows	4.550	3.650	2.975	1.925	**1.900**

the average of the minimum distance between the element's distinct hard clusters of 30 executions, the idea is to select the parameters that better separate the clusters. For further details, see Section 2.1 of Supplementary Material.

The term $2\sigma^2$ in the Gaussian conventional KFCM clustering method was estimated as the average of the 0.1 and 0.9 quantiles of $||\mathbf{x}_l - \mathbf{x}_k||^2$, $l \neq k$ [22].

All algorithms were implemented in C++ and run on an Intel(R) Core(TM) i7-4790 CPU @3.60GHz with 12GB of RAM, running Windows 10 Education Edition.

KFCM-K, KFCM-K-W.1, KFCM-K-W.2, KFCM-K-E$_{W.1}$, and KFCM-K-E$_{W.2}$ algorithms were run 100 times on the datasets of Table 9 until the convergence. The average and standard deviation of the execution time (Table 2 of section 2.4 of the Supplementary Material), Accuracy (Table 3 of section 2.4 of the Supplementary Material), F-measure (Table 4 of section 2.4 of the Supplementary Material), Adjusted Rand (Table 5 of section 2.4 of the Supplementary Material), NMI (Table 6 of section 2.4 of the Supplementary Material), Entropy (Table 7 of section 2.4 of the Supplementary Material), Rand Frigui (Table 8 of section 2.4 of the Supplementary Material), Rand Hullermeier (Table 9 of section 2.4 of the Supplementary Material), Modified Partition Coefficient (Table 10 of section 2.4 of the Supplementary Material), Jaccard (Table 11 of section 2.4 of the Supplementary Material), and Folkes-Mallows (Table 12 of section 2.4 of the Supplementary Material) indexes were computed for the KFCM-K, KFCM-K-W.1, KFCM-K-W.2, KFCM-K-E$_{W.1}$, and KFCM-K-E$_{W.2}$ algorithms on the datasets of Table 9.

A detailed discussion of the results according to each index is available in Section 2.4 of the Supplementary Material. In synthesis, overall, the proposed algorithms obtained the best result for most of the datasets considering every metric. The execution time obtained a more even distribution between the proposed and reference. With an emphasis on the KFCM-K-E$_W$ variants, which obtained most of the best results. In particular, the algorithms KFCM-K-E$_{W.1}$ and KFCM-K-E$_{W.2}$ had the best peformance in the datasets with many clusters (Leaf, Letters, Pendigits, and Urban) according to the average of all considered indices. The proposed algorithms are shown to be stable, with average deviations below 9.6% for every metric, especially for the KFCM-K-E$_{W.1}$, which obtained average deviations below 4.1% for every metric, demonstrating their robustness.

Table 11
Friedman test for the average executions.

Metrics	F_F	Rejects H_0
Execution Time	8.197	Yes
Accuracy	3.664	Yes
F-Measure	2.281	No
Adjusted Rand	4.913	Yes
NMI	8.330	Yes
Entropy	10.477	Yes
Rand Frigui	28.521	Yes
Rand Hullermeier	15.278	Yes
Modified Partition Coefficient	163.466	Yes
Jaccard Index	41.475	Yes
Folkes-Mallows	42.102	Yes

To identify whether the differences in the performance of the algorithms were statistically significant, we tested the null hypothesis that all models perform the same and the differences are merely random in terms of the selected metrics. For that purpose, we order the algorithms for each dataset and metric by assigning them ranks representing that order. If two or more results are the same, then their ranks are the average of the ranks they would have if they were slightly different (e.g. if the 2nd and 3rd positions were the same, then their rank would be 2.5).

After calculating the average rank of the algorithms for each metric, we apply the Friedman test [38] to compare the average ranks of algorithms under the null hypothesis, which states that all the algorithms are equivalent and so their ranks should be equal. If the null hypothesis is rejected, we proceed with the Nemenyi post-hoc test [38] to compare the distance between the average ranks to a critical distance (CD).

Table 10 summarizes the performance of these algorithms by averaging their ranking position for all datasets in Table 9.

Based on these ranks, we apply the Friedman Test (see Table 11). These results allow us to state that the only metric without a statistically significant difference was the F-measure.

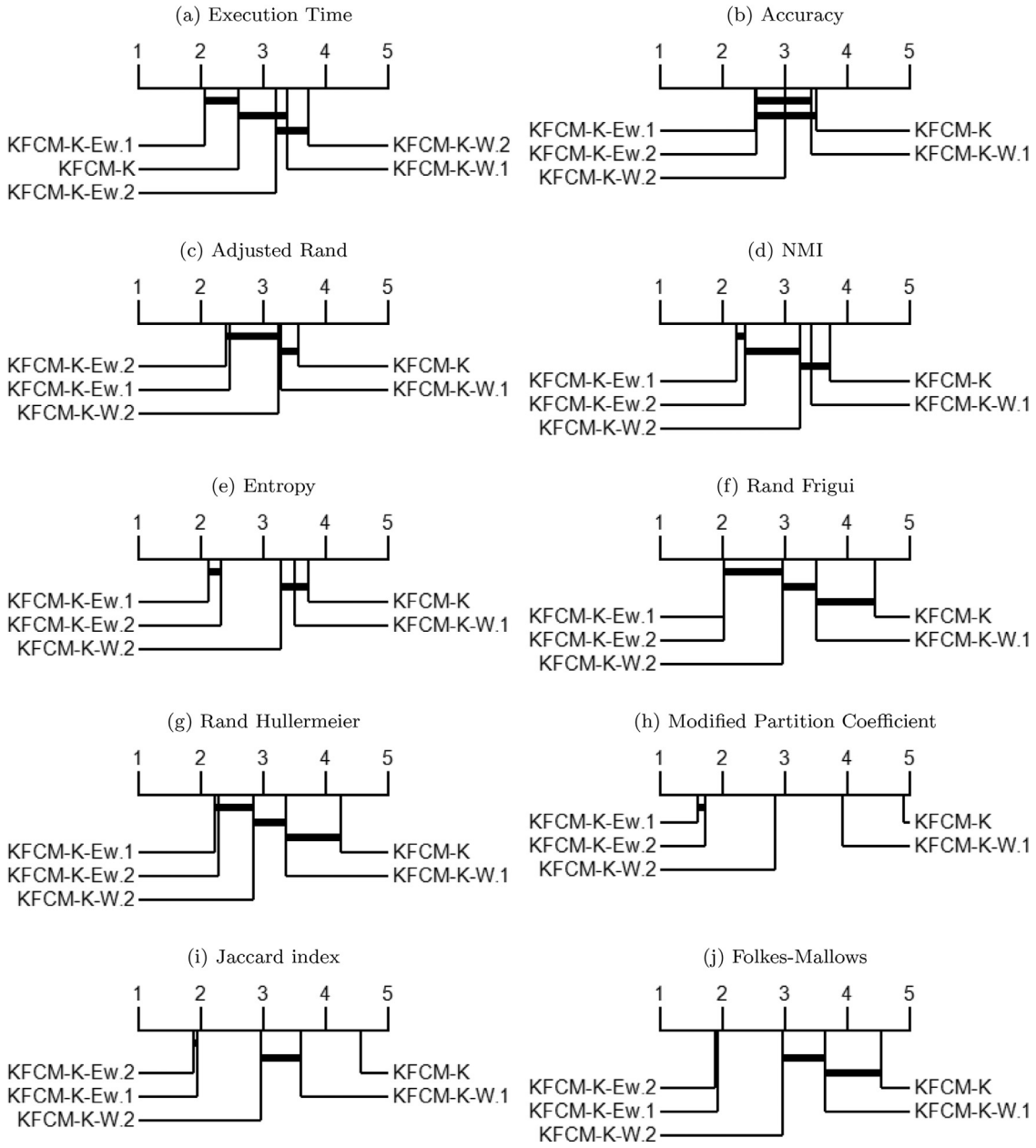

Fig. 1. Nemenyi post-hoc results for the average executions.

Considering five clustering algorithms, 40 datasets, and $\alpha = 0.05$, we have $CD = 0.964$. Thus, the Nemenyi post-hoc test provides us with the results presented in the Fig. 1.

These tables and images allow us to share the following remarks:

- Each of the proposed algorithms obtained better average performance rankings on the indices than the reference method KFCM-K, except for the execution time (KFCM-K-$E_{W.1}$ was the best and KFCM-K was the second best). The proposed algorithms were expected to be slightly slower since the reference is an algorithm with fewer steps.
- The Friedman test points out that the observed differences in the average ranking of the metrics obtained by the methods were not statistically significant for the F-measure, in which all algorithms obtained statistically similar results.
- The proposed KFCM-K-$E_{W.1}$ and KFCM-K-$E_{W.2}$ tied with the best Rand Frigui average result.
- The proposed KFCM-K-$E_{W.1}$ obtained, on average, the best execution time, accuracy, NMI, Entropy, Rand Hullermeier, and Modified Partition Coefficient of the results.
- The proposed KFCM-K-$E_{W.2}$ obtained, on average, the best F-measure, adjusted rand, Jaccard index and Folkes-Mallows index of the results.
- According to the Nemenyi post-hoc test (see Fig. 1), the proposed algorithms were statistically superior to the KFCM-K algorithm for the Rand Frigui, Rand Hullermeier, Modified Partition Coefficient, Jaccard index, and Folkes-Mallows index.
- According to the Nemenyi pos-hoc test, the KFCM-K-E_W variants were also statistically superior to the KFCM-K algorithm for the Adjusted Rand, NMI, and Entropy.
- The proposed KFCM-K-$E_{W.1}$ were also statistically superior to the KFCM-K according to the Accuracy.
- The observed differences in the average ranking of the metrics between the variants of the KFCM-K-E_W were not sta-

Table 12
Number of elements from each class.

Category	0	1	2	3	4	5	6	7	8	9
Samples	128	131	107	122	108	119	114	117	109	111

Table 13
Accuracy: Average and standard deviation (in parenthesis).

Metric	FLeCK	KFCM-K-W.1	KFCM-K-W.2	KFCM-K-E$_{W.1}$	KFCM-K-E$_{W.2}$
Pendigits	0.76 (0.052)	0.657479 (0.080384)	0.603782 (0.040164)	**0.807333 (0.016102)**	0.806587 (0.027234)
English Letters	**0.44 (0.016)**	0.287160 (0.032202)	0.189037 (0.014249)	0.310614 (0.007463)	0.312890 (0.008976)
Image Segmentation	0.59 (0.039)	0.521400 (0.042767)	0.527943 (0.038350)	**0.658995 (0.019865)**	0.625319 (0.039836)

Table 14
Jaccard index: Average and standard deviation (in parenthesis).

Metric	FLeCK	KFCM-K-W.1	KFCM-K-W.2	KFCM-K-E$_{W.1}$	KFCM-K-E$_{W.2}$
Pendigits	**0.42 (0.006)**	0.135529 (0.014194)	0.222016 (0.030753)	0.416465 (0.024044)	**0.422169 (0.029185)**
English Letters	**0.34 (0.019)**	0.028425 (0.006416)	0.039679 (0.003069)	0.099451 (0.003057)	0.097488 (0.003750)
Image Segmentation	0.35 (0.008)	0.256916 (0.044910)	0.285053 (0.030892)	**0.405741 (0.011900)**	0.375724 (0.030765)

Table 15
Folkes-Mallows index: Average and standard deviation (in parenthesis).

Metric	FLeCK	KFCM-K-W.1	KFCM-K-W.2	KFCM-K-E$_{W.1}$	KFCM-K-E$_{W.2}$
Pendigits	0.58 (0.024)	0.243144 (0.022374)	0.365398 (0.041496)	0.597760 (0.023610)	**0.599954 (0.029998)**
English Letters	**0.47 (0.029)**	0.055226 (0.012040)	0.076801 (0.005857)	0.181079 (0.005060)	0.177957 (0.006201)
Image Segmentation	0.42 (0.024)	0.409352 (0.058411)	0.446235 (0.036620)	**0.577506 (0.011942)**	0.549535 (0.032082)

tistically significant for every metric with exception of the Execution Time, were the KFCM-K-E$_{W}$.1 was superior.
- The observed differences in the average ranking of the metrics between the variants of the KFCM-K-W were not statistically significant for every metric, except for the Modified Partition Coefficient, where the proposed KFCM-K-W.2 was superior.

4.2. Comparison with FLeCK and SKFCM-opt σ algorithms

This section evaluates the performance of the proposed methods compared with other works [16,17] that compute the width parameter of the Gaussian kernel automatically.

Previous algorithm KFCM-K-W.1 [18] and the proposed algorithms KFCM-K-W.2, KFCM-K-E$_{W.1}$, and KFCM-K-E$_{W.2}$ were run on the datasets until the convergence 100 times. For each algorithm was selected the best (the most homogeneous) solution, i.e., the solution (among 100) corresponding to the minimum value of the respective objective function. Parameter m was set to 1.1.

Fuzzy Clustering With Learnable Cluster-Dependent Kernels

KFCM-K-W.1, KFCM-K-W.2, KFCM-K-E$_{W.1}$, and KFCM-K-E$_{W.2}$ were run on the datasets of Ref. [17]: Pendigits, English Letters and Image Segmentation. The metrics used to compare these algorithms were those of Ref. [17]: accuracy, Jaccard index, and Folkes-Mallows.

According to Ref. [17], we randomly selected elements from each class of the Pendigits dataset, followinbg the distribution defined in Table 12.

Tables 13–15 show the average and standard deviation of the accuracy, Jaccard and Folkes-Mallows indices, respectively. The values for the FLeCK algorithm were reported from Ref. [17].

FLeck was the best in all metrics for the English Letter dataset, but KFCM-K-E$_{W.1}$ was the best in all metrics for the segmentation dataset. KFCM-K-E$_{W.1}$ was the best in accuracy and Folkes-Mallows for the Pendigits dataset. FLeCK and KFCM-K-E$_{W.2}$ were the best in the Jaccard index for the Pendigits dataset. Finally, KFCM-K-E$_{W.1}$ algorithm was superior to FLeCK in two out of three datasets.

Kernel Parameter Optimization in Stretched Kernel-Based Fuzzy Clustering

KFCM-K-W.1, KFCM-K-W.2, KFCM-K-E$_{W.1}$, and KFCM-K-E$_{W.2}$ were run on the datasets of Ref[16].: Pendigits and iris datasets. The index used to compare these algorithms was those of Ref. [16]: accuracy.

Following Ref. [16], we randomly select 100 elements for each of the classes 1,3,5,7 of the Pendigits dataset, we will call that variation Pendigits400;

Table 16 shows the accuracy for the best execution. The values for the SKFCM-opt σ algorithm were reported from Lu et al. [16].

The SKFCM-opt σ presented the best and KFCM-K-E$_{W.2}$ the second best accuracy for both iris and Pendigits 400 datasets.

4.3. Detailed results for the iris dataset

This section considers detailed results provided by the KFCM-K-W.1 [18], KFCM-K-W.2, KFCM-K-E$_{W.1}$ and KFCM-K-E$_{W.2}$ algorithms when applied to the Iris dataset to show their usefulness. The Iris dataset describes 150 objects defined by the following four real-valued variables: sepal length, sepal width, petal length, and petal width. The objects are partitioned according to an a priori classification into three classes: Iris Setosa, Iris Versicolour, and Iris Virginica. Each a priori class has 50 objects.

These algorithms were run on the iris dataset until convergence was achieved 100 times. Here, we consider the best solution (among 100), according to the minimum value of the corresponding objective functions. Figures 1(a), 2(a), 3(a), and 4(a) (see Section 2.5 of the Supplementary Material) show the evolution of the objective functions versus iterations for the iris dataset. Algorithms KFCM-K-W.2, KFCM-K-E$_{W.1}$, and KFCM-K-E$_{W.2}$ converged faster than the algorithm KFCM-K-W.1 [18].

Table 17 shows the confusion matrix between the a priori partition of this dataset and the hard cluster partitions obtained from the fuzzy cluster partitions provided by these algorithms. One can observe the good performance of KFCM-K-W.1, KFCM-K-W.2, and KFCM-K-E$_{W.2}$ algorithms on this dataset is highlighted.

Table 16
Accuracy for the best execution.

dataset	SKFCM-opt σ	KFCM-K-W.1	KFCM-K-W.2	KFCM-K-E$_{W.1}$	KFCM-K-E$_{W.2}$
Iris	**0.9756**	0.9600	0.9533	0.8667	0.9600
Pendigits400	**0.6723**	0.5875	0.5625	0.6525	0.6550

Table 17
Iris dataset: Confusion Matrix.

		KFCM-K-W.1			KFCM-K-W.2			KFCM-K-E$_{W.1}$			KFCM-K-E$_{W.2}$		
		Cluster			Cluster			Cluster			Cluster		
		1	2	3	1	2	3	1	2	3	1	2	3
Class	Setosa	0	50	0	50	0	0	50	0	0	0	50	0
	Versicolor	48	0	2	0	3	47	0	8	42	2	0	48
	Virginica	4	0	46	0	46	4	0	38	12	46	0	4

Table 18
Iris dataset: Prototypes and the four-components vector of width parameters.

			Prototypes				Variable Weights ((s_j^2) *or* (s_{ij}^2))			
Algorithm	Cluster	Main Class	sepal length	sepal width	petal length	petal width	sepal length	sepal width	petal length	petal width
KFCM-K-W.1	1	Versicolor	5.95447	2.77495	4.33240	1.33935	2.10632	3.67971	0,31853	0.40504
	2	Setosa	4.99738	3.40633	1.46776	0.24037				
	3	Virginica	6.54649	2.99553	5.46922	2.04761				
KFCM-K-W.2	1	Setosa	5.00234	3.40962	1.46443	0.23908	2.56029	10.63942	0.13408	0.27378
	2	Virginica	6.54856	2.99709	5.45544	2.03560	2.14638	1.80043	0.39823	0.64980
	3	Versicolor	5.93439	2.76394	4.31469	1.33117	2.23798	3.04701	0.43066	0.34050
KFCM-K-E$_{W.1}$	1	Setosa	5.00092	3.41380	1.46718	0.24300	20.65688	4.00000	2.11362	4.37694
	2	Virginica	6.70553	3.09458	5.51139	2.01246				
	3	Versicolor	5.86455	2.68278	4.36586	1.38486				
KFCM-K-E$_{W.2}$	1	Virginica	6.62954	3.01383	5.55624	2.05594	17.49169	33.41129	2.15540	2.22741
	2	Setosa	5.00538	3.41677	1.46462	0.24355	7.04523	719.42446	2.28242	2.38925
	3	Versicolor	5.92058	2.74427	4.30273	1.32792	15.88814	28.71912	2.31943	2.12269

Table 18 contains the cluster prototypes and the four-components vector of width parameters computed either globally, for the whole fuzzy partition as in KFCM-K-W.1 and KFCM-K-E$_{W.1}$ algorithms, or locally, for each cluster as in KFCM-K-W.2 and KFCM-K-E$_{W.2}$.

As it has been pointed out, the closer the objects are to the representative of a given cluster concerning a given real-valued variable, the lower the width parameter of this variable is on this cluster. Let the clusters provided by the KFCM-K-E$_{W.2}$ algorithm. According to Table 17, all the objects of cluster 2 belong to the a priori class 1 Iris Setosa. The four-component vector of width parameters provided by the KFCM-K-E$_{W.2}$ algorithm to the cluster 2 is $\mathbf{s}_2 = (7.04523, 719.42446, 2.28242, 2.38925)$ (see Table 18). In this cluster, the variable petal length has the lowest width hyper-parameter, whereas the variable sepal width has the highest width hyper-parameter. The variable petal length is more relevant for cluster 2 than the variable sepal width since a small difference between an object and the representative of cluster 2 is strongly amplified on the former variable, whereas an important difference between an object and the representative of cluster 2 weakly amplified on the latter variable.

5. Final remarks and conclusions

The conventional Gaussian kernel-based clustering algorithm is very dependent on the estimation of the width hyper-parameter of the Gaussian kernel function, which is estimated once and for all and it is the same for all variables. Therefore, the variables have the same importance in the clustering task, including irrelevant variables.

The first contribution of this paper was to propose a Gaussian kernel fuzzy c-Means with kernelization of the metric and automated computation of the width parameters using adaptive Gaussian kernels. In this method, the width parameters become variables of the suitable objective function, changing at each algorithm iteration and differing from variable to variable and from cluster to cluster. Thus, this algorithm can re-scale the variables differently, thus highlighting those that are relevant to the clustering task. Moreover, much attention has been directed to maximum entropy fuzzy clustering algorithms. In this regard, the second contribution of this paper was to provide Gaussian kernel fuzzy c-means algorithms with kernelization of the metric and automated computation of the width hyper-parameters through entropy regularization. Our proposals add new positive entropy terms that allow an automatic adjustment of the width parameters during the optimization process to the objective function.

Our paper provides an evaluation of the performance of the proposed algorithm, according to suitable indices, compared with the conventional KFCM-K algorithm and other works [16–18] that compute the width parameter of the Gaussian kernel automatically. We first compared the proposed methods with KFCM-K and KFCM-K-W.1 [18] on 40 datasets from the UCI machine learning repository, with a different number of objects, variables, and a priori classes. Then, we compared the proposed methods with other works of Refs. [16,17] using indices and datasets selected by these papers.

Except for the execution time, each of the proposed Gaussian kernels c-means clustering algorithms obtained better average performance rankings compared to the KFCM-K and KFCM-K-W.1 algorithms, especially the KFCM-K-E$_W$ variants. In particular, the al-

gorithms KFCM-K-$E_{W.1}$ and KFCM-K-$E_{W.2}$ had the best peformance in the datasets with many clusters (Leaf, Letters, Pendigits, and Urban) according to the average of all considered indices.

All the proposed algorithms were statistically superior to the KFCM-K algorithm for the Rand Frigui, Rand Hullermeier, Modified Partition Coefficient, Jaccard index, and Folkes-Mallows index. The variants of the KFCM-K-E_W were also statistically superior to the KFCM-K algorithm for the Adjusted Rand, NMI, and Entropy. In addition, they were statistically superior to the KFCM-K-W.1 for the NMI, Entropy, Rand Frigui, Rand Hullermeier, Modified Partition Coefficient, Jaccard index, and Folkes-Mallows index.

The proposed algorithms obtained stable results with an average deviation lower or close to the one obtained by the reference KFCM-K-W.1, for every metric, except for the execution time of the KFCM-K-W.2, which was slightly greater than that obtained by the KFCM-K-W.1.

The observed differences in the average ranking of the metrics between the variants of the KFCM-K-W algorithm were not statistically significant for every index, except for the Modified Partition Coefficient, in which the proposed KFCM-K-W.2 is statistically superior. Finally, the observed differences in the average ranking of the metrics between the variants of the KFCM-K-E_W algorithm were not statistically significant for every index, except the execution time, in which the KFCM-K-E_W.1 is statistically superior.

As to the comparison with FLeCK [17] and SKFCM-opt σ [16] algorithms, KFCM-K-$E_{W.1}$ algorithm was superior to FLeCK in two out of three datasets and SKFCM-opt σ achieved the best accuracy in the two available datasets.

Regarding the variants of the KFCM-K-E_W algorithm, the choice of the hyper-parameter T_W is difficult and time-consuming. For further studies, we plan to integrate T_W into the objective function of the variants of the KFCM-K-E_W algorithm for it to be optimized in a new step of the algorithm. We also plan to consider other types of regularization in the framework of the proposed methods and extend the proposed approaches to Gaussian kernel-based fuzzy clustering algorithms where the cluster representatives are located in the kernel space.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome. We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us. We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property. We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept email from dppb@cin.ufpe.br

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank the anonymous referees for their careful revision, and the Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (311164/2020-0) for their partial financial support of this study.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2023.109749.

References

[1] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Upper Saddle River, NJ, 1988.
[2] A.E. Ezugwu, A.M. Ikotun, O.O. Oyelade, L. Abualigah, J.O. Agushaka, C.I. Eke, A.A. Akinyelu, A comprehensive survey of clustering algorithms: state-of-the-art machine learning applications, taxonomy, challenges, and future research prospects, Eng. Appl. Artif. Intell. 110 (2022) 104743.
[3] D. Zhang, S. Chen, A novel kernelized fuzzy c-means algorithm with application in medical image segmentation, Artif. Intell. Med. 32 (1) (2004) 37–50.
[4] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recognit. 41 (2008) 176–190.
[5] M.-S. Yang, H.-S. Tsai, A gaussian kernel-based fuzzy c-means algorithm with a spatial bias correction, Pattern Recognit. Lett. 29 (2008) 1713–1725.
[6] D.-C. Park, Classification of audio signals using fuzzy c-means with divergence-based kernel, Pattern Recognit. Lett. 30 (2009) 794–798.
[7] F. Zhao, L. Jiao, H. Liu, Kernel generalized fuzzy c-means clustering with spatial information for image segmentation, Digit. Signal Process. 23 (2013) 184–199.
[8] M. Gong, Y. Liang, J. Shi, W. Ma, J. Ma, Fuzzy c-means clustering with local information and kernel metric for image segmentation, IEEE Trans. Image Process. 22 (2013) 573–584.
[9] A.A. Abin, H. Beigy, Active constrained fuzzy clustering: a multiple kernels learning approach, Pattern Recognit. 48 (2015) 53–967.
[10] G. Hu, Z. Du, Adaptive kernel-based fuzzy c-means clustering with spatial constraints for image segmentatio, Int. J. Pattern Recognit Artif Intell. 33 (2019) 1954003.
[11] C. Wu, Z. Cao, Noise distance driven fuzzy clustering based on adaptive weighted local information and entropy-like divergence kernel for robust image segmentation, Digit. Signal Process. 111 (2021) 102963.
[12] N.A. Talukdara, A. Halder, Partially supervised kernel induced rough fuzzy clustering for brain tissue segmentation, Pattern Recognit Image Anal. 31 (2021) 91–102.
[13] Q. Song, C. Wu, X. Tian, Y. Song, X. Guo, Kernel-based fuzzy local information clustering algorithm self-integrating non-local information, Digit. Signal Process. 122 (2022) 103351.
[14] X.Z. C. Wu, Total Bregman divergence-driven possibilistic fuzzy clustering with kernel metric and local information for grayscale image segmentation, Pattern Recognit. 128 (2022) 1086862.
[15] K.C. L. Wang, Learning kernel parameters by using class separability measure, in: NIPS'02 Workshop on Kernel Machines, 2002, pp. 1–8.
[16] C. Lu, Z. Zhu, X. Gu, Parameter optimization in stretched kernel-based fuzzy clustering, in: Z. Zhou, F. Schwenker (Eds.), Partially Supervised Learning - PSL 2013, Lecture Notes in Computer Science, Vol. 8183, Springer, 2013, pp. 49–57.
[17] O. Bchir, H. Frigui, M.M.B. Ismail, Fuzzy clustering with learnable cluster-dependent kernels, Pattern Anal. Appl. 19 (2016) 919–937.
[18] F.A.T. de Carvalho, L.V.C. Santana, M.R.P. Ferreira, Gaussian kernel-based fuzzy clustering with automatic bandwidth computation, in: V.K. et al (Ed.), Artificial Neural Networks and Machine Learning - ICANN 2018 - 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4–7, 2018, Proceedings, Part I, Lecture Notes in Computer Science, Vol. 11139, Springer, 2018, pp. 685–694.
[19] X. Tao, R. Wang, R. Chang, C. Li, Density-sensitive fuzzy kernel maximum entropy clustering algorithm, Knowl. Based Syst. 166 (2019) 42–57.
[20] D. Graves, W. Pedrycz, Kernel-based fuzzy clustering and fuzzy clustering: acomparative experimental study, Fuzzy Sets Syst. 161 (2010) 522–543.
[21] R. Zhang, A.I. Rudnicky, A large scale clustering scheme for kernel K-Means, in: Proceedings of the 16th International Conference on Pattern Recognition, Vol. 1, 2002, pp. 289–292.
[22] B. Caputo, K. Sim, F. Furesjo, A. Smola, Appearence-based object recognition using SVMs: which kernel should i use? in: Proceedings of NIPS Workshop on Statistical Methods for Computational Experiments in Visual Processing and Computer vision, 2002.
[23] F.A.T. de Carvalho, E.C. Simões, L.V.C. Santana, M.R.P. Ferreira, Gaussian kernel c-means hard clustering algorithms with automated computation of the width hyper-parameters, Pattern Recognit. 79 (2018) 370–386.
[24] J. Huang, M. Ng, H. Rong, Z. Li, Automated variable weighting in k-means type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 657–668.
[25] M. Hanmandlu, O.P. Verma, S. Susan, V.K. Madasu, Color segmentation by fuzzy co-clustering of chrominance color features, Neurocomputing 120 (2013) 235–249.

[26] L. Jing, M.K. Ng, J.Z. Huang, An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data, IEEE Trans. Knowl. Data Eng. 19 (8) (2007) 1026–1041.
[27] J. Zhou, L. Chen, C.L.P. Chen, Y. Zhang, H. Li, Fuzzy clustering with the entropy of attribute weights, Neurocomputing 198 (2016) 125–134.
[28] S. Miyamoto, K. Umayahara, Fuzzy clustering by quadratic regularization, in: 1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36228), Vol. 2, IEEE, 1998, pp. 1394–1399.
[29] F. Camastra, A. Verri, A novel kernel method for clustering, IEEE Trans. Neural Netw. 27 (2005) 801–804.
[30] C. Manning, P. Raghavan, H. Schuetze, Introduction to Information Retrieval, Cambridge University Press, Cambridge, UK, 2008.
[31] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1985) 193–218.
[32] S. Pei, L. Tong, Gaussian kernel particle swarm optimization clustering algorithm, in: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016, pp. 198–204, doi:10.1109/FSKD.2016.7603174.
[33] R.N. Dave, Validating fuzzy partitions obtained through c-shells clustering, Pattern Recognit. Lett. 17 (6) (1996) 613–623.
[34] H. Frigui, C. Hwang, F. Rhee, Clustering and aggregation of relational data with applications to image database categorization, Pattern Recognit. 40 (11) (2007) 3053–3068.
[35] E. Hüllermeier, M. Rifqi, S. Henzgen, R. Senge, Comparing fuzzy partitions: a generalization of the rand index and related measures, IEEE Trans. Fuzzy Syst. 20 (3) (2012) 546–556.
[36] E.B. Fowlkes, C.L. Mallows, A method for comparing two hierarchical clusterings, J. Am. Stat. Assoc. 78 (383) (1983) 553–569.
[37] C. Black, C. Merz, UCI repository of machine learning databases, 1998, (http://www.ics.uci.edu/mlearn/MLRepository.html).
[38] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Lear. Res. 7 (2006) 1–30 .

Eduardo C. Simões has a masters degree in Computer Science by the Universidade Federal de Pernambuco, Brazil. Currently a PhD candidate in the same field by the same institution.

Francisco de A. T. de Carvalho received the PhD degree in Computer Science in 1992 from Institut National de Recherche en Informatique et en Automatique (INRIA) and Université Paris-IX Dauphine, France. From 1992 to 1998, he was a lecturer at Statistical Department at Universidade Federal de Pernambuco, Brazil. He joined the Center of Informatics at Universidade Federal de Pernambuco in 1999, where he is currently Full Professor. He is full member of the Brazilian Academy of Sciences (ABC) and full member of of the Academy of Sciences of Pernambuco (APC). He held visiting posts in several leading universities and research centers in Europe. With main research interests in symbolic data analysis, clustering analysis and machine learning he has authored over 200 technical papers in international journals and conferences. He has served as Coordinator (2005–2009) of the postgraduate program of computer science of the CIn/UFPE. He has been involved in program committees of many Brazilian and international conferences. He has also served as review of many international journals and conferences. He was member of the council (2009–2013) of the International Association for Statistical Computing (IASC). He was member of the council (2017–2020) of the Latin American Regional Section - LARS of the IASC. In 2021 He received the Scientific Merit Award of the SBC (Brazilian Computer Society). He is within the top 2% of scientists in the world in the field of Artificial Intelligence and Image Processing throughout his career and in the years of 2019 and 2020 according to a study by Plos Biology/Elsevier (https://elsevier.digitalcommonsdata.com/datasets/btchxktzyw/3). He is among the 50 most influential authors and was ranked in 14 among the most productive authors (among 13970 authors) in automatic clustering algorithms in a period of 30 years (between 1989 e 2019) according to a study by Neural Computing and Applications / Springer (https://doi.org/10.1007/s00521-020-05395-4).