Applied Soft Computing Journal 89 (2020) 106102

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Two cluster validity indices for the LAMDA clustering method

Javier Fernando Botía Valderrama [a,b,*], Diego José Luis Botía Valderrama [b,c]

[a] Grupo de Investigación en Telecomunicaciones Aplicadas (GITA), Faculty of Engineering, Department of Electronic Engineering, Universidad de Antioquia (UdeA), Calle 50 No. 73–21, 050034, Medellín, Colombia
[b] Faculty of Engineering, Department of System Engineering, Universidad de Antioquia (UdeA), Calle 50 No. 73–21, 050034, Medellín, Colombia
[c] Intelligent Information Systems Lab (IN2LAB), Faculty of Engineering, Department of System Engineering, Universidad de Antioquia (UdeA), Calle 50 No. 73–21, 050034, Medellín, Colombia

ARTICLE INFO

Article history:
Received 8 January 2019
Received in revised form 26 November 2019
Accepted 14 January 2020
Available online 23 January 2020

Keywords:
Cluster validity index
Fuzzy clustering
LAMDA
Fuzzy statistics
Data analysis

ABSTRACT

The learning algorithm and multivariable data analysis (LAMDA) is an algorithm to group quantitative and qualitative data, applying self-learning and/or directed learning. Usually, LAMDA automatically generates classes by assigning the best data partition to a class. To evaluate the data partitions generated by LAMDA, the internal evaluation is used to find the optimal number of clusters. For the LAMDA algorithm, the cluster validity (CV) is the most popular index which is based on inter-class contrast (ICC). However, other indices have not been defined for LAMDA and a comparative analysis is required to evaluate its performance. In this paper, two metrics called cluster validity index based on granulation error and the ratio of the distance (CVGED) and cluster validity index based on the ratios of covariance and distance (CVCOD) are proposed. Such indices are compared with the CV and ICC indices for two experiments: using a databases repository and selected open data and experimental laboratory data. According to the main results, CVGED and CVCOD have a better performance in compactness, separation, and coefficient of variation than ICC and CV for most of the selected repository databases but the accuracy is limited for the four indices. Nevertheless, CVCOD improves the quality of data partition when the open data and experimental laboratory data are used.

© 2020 Elsevier B.V. All rights reserved.

Mathematical expressions and symbols

See Table 1.

1. Introduction

Machine learning (ML) is defined as the capacity of a computer for obtaining knowledge from data and thus it can learn to solve complex problems [1]. Usually, ML works in three areas: supervised, unsupervised, and reinforcement learning. Taking into account the supervised and unsupervised learning, the supervised learning carries out classification tasks and the unsupervised learning makes clustering tasks. Classification is used if the assignation of a category to each sample and the exact number of groups or classes are known. By contrast, clustering is applied if the optimal number of clusters and the characteristics of the clusters are not known for a set of samples or objects. Focusing the paper on clustering, it relates a set of samples to one cluster where each one is separable and compact to another cluster. The above makes possible to differentiate one cluster from another cluster. Clustering is able to find the relation of samples and centers or prototypes through distance measures (Euclidean, Mahalanobis, among others). For definition, a prototype is an element of a set of samples that represents a simplified description of the main characteristics of data (a prototype is associated to a category) [2]. To obtain the best prototypes, a clustering algorithm randomly updates all prototypes for each iteration until the minimum error is reached. When the termination condition is fulfilled, the algorithm can generate two types of data partitions: hard or fuzzy partitions. The former shows that a sample belongs to a cluster but the same sample cannot belong to another cluster and the latter indicates that a sample belongs to all clusters according to a membership degree [3]. Considering the fuzzy partition, some fuzzy clustering algorithms that generate this kind of task are: fuzzy c-means (FCM) [4], Gustafson and Kessel means (GK-means) [5], multiple kernel fuzzy clustering (MKFC) [6], and picture fuzzy clustering [7].

Inside the fuzzy clustering, the research is focused on the learning algorithm for multivariable data analysis (LAMDA). For definition, LAMDA is a learning method to group quantitative and qualitative historical data [8]. Usually, the algorithm works as direct learning or self-learning [9]. The direct learning assigns an element of the data with a pre-defined class and the class parameters are updated according to the information of the specific

* Correspondence to: Calle 50 No. 73–21, Building 18, Office 310, 050034, Medellín, Colombia.
E-mail addresses: javier.botia@udea.edu.co (J.F.B. Valderrama), diego.botia@udea.edu.co (D.J.L.B. Valderrama).

https://doi.org/10.1016/j.asoc.2020.106102
1568-4946/© 2020 Elsevier B.V. All rights reserved.

Table 1
Mathematical symbols used for paper.

Variable	Meaning	Variable	Meaning
X	Database or historical data	$m_{c,d}$	Mean between a c- class and a d-feature
m	Center of the whole data	$\delta_{c,d}$	Standard deviation between a c-class and a d-feature
N	Number of samples	ns	Number of data previously classified in c-class
n	Sample	S_{Be}	A measure that estimates the number of elements or samples for a cluster through the sum of all element's fuzzy memberships on the cluster.
D	Number of features	$\mu_c(x_n)$	Membership degree of a x_n for a c-class
d	Feature	m_c	Mean or center of a c-class
K	Number of Classes	D_{min}	Minimum Euclidean distance among centers
c	Class	c^*	Optimal number of cluster
ϵ	Exigency degree	ICC	Inter-class contrast
α	Parameter for Yager implication	D_{is}	Dispersion among classes
Ω	Parameter for Sugeno–Hamasher implication	D^*_{min}	Minimum ultra-metric distance among fuzzy sets
γ	Parameter for Dombi implication	$I_D(c)$	Information index for a c-class
$x_{n,d}$	One datum of n- sample to d- feature	$\upsilon_c(x_n)$	Adequacy degree between the maximum membership degree for a c-class and $\mu_c(x_n)$
$\widehat{x}_{n,d}$	Normalized datum of n- sample to d- feature	μm_c	Maximum membership degree for a c-class
$xmin_d$	Minimum value of data for a d-feature	$S(i,j)$	Similarity measure between two clusters, i, j
$xmax_d$	Maximum value of data for a d-feature	$d^*(i,j)$	Ultrametric distance between two clusters, i, j
$M_{n,d,c}$	MAD function of a n-sample with respect to a d-descriptor and c-class	CM	Center matrix
$G_{n,c}$	GAD function of a n-sample with respect to a c-class	m_i, m_j	Centers for a i-cluster and j-cluster
$GTD_{n,c}$	Global Typicality Degree of a n-sample with respect to a c-class	DV_{min}	Minimum distances among clusters
$IG_{n,c}$	Intuitionistic Global Membership Degree of a n-sample with respect to a c-class	DV_{max}	Maximum distances among clusters
$G\pi_{n,c}$	Hesitancy degree for the GAD function of a n-sample with respect to a c-class	DV^*_{min}	Minimum distances among clusters based on median
$TIGAD_{n,c}$	Typicality and Intuitionistic Global Adequacy Degree of n- sample to c- class	DV^*_{max}	Maximum distances among clusters based on median
a, b	Adjustment parameters	DR	Relation between DV^*_{max} and DV^*_{min}
i, j	Two clusters, $i \neq j$	$gran(error)$	Granulation criterion
c_{max}	Maximum number of clusters	$\overline{\overline{x_n}}$	Reconstructed sample
σ, ϑ	Weighting parameters	$\Gamma\|C_i\|$	Cardinal of a class
$r_{n,d,c}$	Resemblance measure of a n-sample with respect to a d-descriptor and a c-class	$CVGED$	Cluster validity index based on granulation error and the ratio of the distance
$ds_{n,d,c}$	Dissimilarity measure of a n-sample with respect to a d-descriptor and a c-class	Σ_c	Fuzzy covariance matrix for a $c-$ class
x_dist	Complement of the distance between a normalized datum, $\widehat{x_{n,d}}$, and the adequacy degree, $\rho_{c,d}$	q	Fuzzifier parameter.
ma	Maximum adequacy degree	$GComp(c)$	Global compactness of the fuzzy partition for a c-class
$\chi[\cdot]$	Aggregation	Σ_d	Covariance matrix for data, considering d-feature
$T[\cdot]$	T-norm	GD	Global compactness of data
$S[\cdot]$	S-norm	$RComp$	Ratio of compactness
$\rho_{c,d}$	Adequacy degree between a c- class and a and d-feature	$CVCOD$	Cluster validity index based on the ratios of covariance and distance
Ac	Clustering accuracy,	x_c	Number of attributes in each object
Rc	Clustering recall	N_c	Number of samples for each cluster
w_c	Number of samples occurring in corresponding correct class	s_c	Mean of the cluster
b_c	Total number of classified samples	CoV	Coefficient of variation
B_c	Total number of samples	$\overline{h}$	Average cluster size
$F-measure$	Metric of the quality of clustering	DCV	Difference of CV values before and after clustering
E	Cluster variation	$CoV_{groundtruth}$	Coefficient of variation of the original classes vector
I	Average within cluster variation	$n_{d,c}$	Number of data of the descriptor d belonging to class c
E/I	Ratio between the cluster variation and the average within cluster variation		

sample. By contrast, the self-learning relates a sample to a specific class and LAMDA operates in unsupervised learning mode. This mode allows calculating the global membership degree of a sample with respect to a class through all contributions of its features. Such contribution is called marginal adequacy degree (MAD) and it is defined by probability functions. By unifying all MADs, the global adequacy degree (GAD) is calculated for a sample where represents a fuzzy mixed connectivity of a sample to a class. Considering the above, LAMDA generates automatically one or more class(es) through an automatic threshold called non-informative class (NIC) which compares the GAD value for NIC and the maximum GAD value for a sample. When the value of the GAD is greater than the value of the GAD-NIC, the algorithm creates a new class and otherwise, the algorithm updates the values of the GAD for the current classes. This procedure avoids the assignation of a classes number as an initial parameter.

Usually, the LAMDA algorithm has been applied in fault detection [10–12], image processing [13,14], correction of optical frequency comb spectrum [15,16], and medicine [17]. Nevertheless, several improvements have been made to the LAMDA algorithm. In this case, the GAD function was improved through geometrical interpolation [18] and the triple π operator [18–20], as well as an unification among triple π operator, the intuitionistic fuzzy sets (IFSs), and the typicality degree [21]. On the other hand, the LAMDA algorithm was used to increase the quality of the feature selection for symbolic interval data, mixed-type data and data with noise [22–24]. Although algorithm improvements have focused on the GAD function or feature selection tasks, the cluster validity index have been a certain impact to analyze

the fuzzy partition of data generated by the LAMDA algorithm. Generally speaking, the cluster validity index allows finding the best partition when clusters are distant from one another and the samples for each cluster is near to center or prototype [25]. Based on this concept, an index called cluster validity (CV) was proposed by [26]. This index calculates dispersion and the minimum ultrametric distance among clusters, using inter-class contrast (ICC) proposed by [27]. The CV index is considered as the first cluster validity index for the LAMDA algorithm and a mathematical model based on gamma functions was analyzed for the CV index [28]. This index was applied to validate transitions among functional states (current behavior of a system) for drinking water plant monitoring [29], a propylene glycol production process [30], an intensification reactor monitoring [31], an agro-system [32], and recently, in the analysis of a new improvement of GAD function [21]. However, there is not another cluster validity index for the LAMDA algorithm which new metrics have not been explored to analyze the quality of the data partition generated by self-learning in the algorithm. Therefore, it is important to define other metrics to find the most optimal exigency value where the data partition is compacted, the separation between clusters is large, and the number of clusters is optimal.

In the present paper, we proposed two cluster validity indices for the LAMDA algorithm, called cluster validity index based on granulation error and the ratio of the distance (CVGED) and cluster validity index based on the ratios of covariance and distance (CVCOD). The CVGED index considers median among clusters as well as the maximum and minimum distance among them. This condition is connected with the granulation error between the original database and the data structure obtained from the clustering results. On the other hand, the CVCOD index is defined by a ratio between the fuzzy covariance measure and the data covariance which allows measuring the compactness. The separation measure among clusters applied to the CVGED index is the same for the CVCOD index. Both indices are applied to standard databases from KEEL and UCI repositories (experiment 1) and real databases from open data sources and laboratories (experiment 2). To analyze the experimental results, several clustering evaluation metrics and other indices are taken into account in the performance of the CVGED and CVCOD indices. Such indices are compared with the ICC and CV indices in order to observe the advantages and disadvantages among them. In addition, the experiment 2 is also analyzed by human-experts or data information.

The paper has been organized as follows: Section 2 explains the LAMDA algorithm and the cluster validity indices for LAMDA. Section 3 describes the new proposal for calculating the validity index for the LAMDA algorithm. Section 4 describes the computational complexity of the LAMDA algorithm and metrics. Section 5 explains the experimental settings where the databases description and methodology are added. Section 6 shows results and discussion, comparing the two new cluster validity indices with respect to other cluster validity indices proposed in the LAMDA algorithm. Finally, conclusions, further works, and acknowledgment are mentioned.

2. Backgrounds

2.1. LAMDA

LAMDA is a fuzzy method for clustering and classification tasks. Considering the former, LAMDA calculates the adequacy between samples and a class, using historical data. To find the adequacy, the algorithm relates the contribution of features or attributes of a sample with respect to a class. The above allows establishing the global adequacy between a sample and a class [33]. To understand its operation, four steps are explained as shown below:

2.1.1. Data normalization

The first step of the LAMDA algorithm is to normalize the historical data, $X = \{x_{1,1}, \ldots, x_{n,d}, \ldots, x_{N,D}\}$, given by:

$$\widehat{x}_{n,d} = \frac{x_{n,d} - xmin_d}{xmax_d - xmin_d} \tag{1}$$

where:

- N is the number of samples or elements.
- D is the number of features.
- $x_{n,d}$ is one datum of a sample n to a feature d.
- $xmin_d$ is the minimum value of data with respect to a feature d.
- $xmax_d$ is the maximum value of data with respect to a feature d.
- $\widehat{x}_{n,d}$ is a normalized datum, being $\widehat{x}_{n,d} \rightarrow [0, 1]$.

2.1.2. Marginal adequacy degree

The second step is to calculate the marginal adequacy degree (MAD). The MAD function allows finding how similar a feature d is with respect to the same feature in a given class c [34,35]. The MAD function is defined by probabilistic functions such as binomial, binomial-distance, or Gaussian function [16,36], as shown below:

$$M^{(1)}_{n,d,c} = \rho_{c,d}^{\widehat{x}_{n,d}}(1 - \rho_{c,d})^{1-\widehat{x}_{n,d}} \tag{2}$$

$$M^{(2)}_{n,d,c} = ma^{x_{dist}} \cdot (1 - ma)^{1-x_{dist}} \tag{3}$$

$$M^{(3)}_{n,d,c} = \exp\left(-\frac{(\widehat{x}_{n,d} - m_{c,d})^2}{2\delta^2_{c,d}}\right) \tag{4}$$

where:

- $M^{(1)}_{n,d,c}$ is a binomial MAD function.
- $M^{(2)}_{n,d,c}$ is a binomial-distance MAD function.
- $M^{(3)}_{n,d,c}$ is a Gaussian MAD function.
- $\rho_{c,d}$ is the adequacy degree between a class c and a feature d, being $\rho_{c,d} \rightarrow [0, 1]$.
- $m_{c,d}$ is the mean of a class c to a feature d, being $m_{c,d} \rightarrow [0, 1]$.
- $\delta_{c,d}$ is the standard deviation of a class c to a feature d, being $\delta_{c,d} \rightarrow [0, 1]$.
- $x_{dist} = 1 - |\widehat{x}_n - \rho_{c,d}|$, is complement of the distance between a sample $\widehat{x}_n = \{\widehat{x}_{n,1}, \ldots, \widehat{x}_{n,D}\}$, and the adequacy degree, $\rho_{c,d}$.
- $ma = \max\{\rho_{c,d}^{\widehat{x}_{n,d}}(1-\rho_{c,d})\}$, is the maximum adequacy degree.

Since the LAMDA algorithm carries out an automatic generation of classes, each MAD value is compared with a threshold called non-informative class (NIC) [37]. The main task of NIC is to define an automatic threshold for generating the most relevant classes according to the MAD function:

- $\rho_{c=NIC,d} = 0.5$ then $M^{(1)}_{c=NIC,d,n} = 0.5$. In this case, the adequacy for all features is the same.
- $\rho_{c=NIC,d} = 0.5$ then $M^{(2)}_{c=NIC,d,n} = 0.3536$. In this case, the adequacy for all features is the same.
- $m_{c=NIC,d} = 0.5$ and $\delta_{c=NIC,d} = 0.25$ then $M^{(3)}_{c=NIC,d,n} = 0.5$. In this case, the adequacy for all features is the same.

Due to the operating of NIC, $\rho_{c,d}$ and $\delta_{c,d}$ are updated when a new class is generated. The updating are given by:

$$\rho_{c,d} = \rho_{c-1,d} + \left[\frac{\widehat{x}_{n,d}(c) - \rho_{c-1,d}}{ns_{c-1} + 1}\right] \tag{5}$$

$$\delta_{c,d} = \delta_{c-1,d} + \left[\frac{(\widehat{x}_{n,d}(c) - m_{c-1,d})^2}{ns_{c-1} - 1}\right] \tag{6}$$

where:

- $\rho_{c-1,d}$ is the average value for $c-1$. Eq. (5) is used to update $m_{c,d}$ from Eq. (4).
- $\widehat{x}_{n,d}(c)$ is a new datum to update $\rho_{c,d}$.
- ns_{c-1} is the number of samples classified in the class c.
- $\delta_{c-1,d}$ is the standard deviation for $c-1$.

In case that a new class is not generated, $\rho_{c,d}$ and $\delta_{c,d}$ are updated according to the following equations:

$$\rho_{c,d} = \frac{1}{n_{c,d}} \sum_{d=1}^{d=n_{c,d}} \widehat{x}_{n,d} \tag{7}$$

$$\delta_{c,d} = \frac{1}{n_{c,d}} \sum_{d=1}^{d=n_{c,d}} (\widehat{x}_{n,d} - m_{c,d})^2 \tag{8}$$

where $n_{c,d}$ is the number of data of the feature d belonging to class c.

2.1.3. Global adequacy degree

The third step is to calculate the global adequacy degree (GAD) by unifying all MAD values of a sample n with respect to a given class c, expressed by:

$$G_{n,c} = \psi\{M_{n,1,c}, \ldots, M_{n,d,c}, \ldots, M_{n,D,c}\} \tag{9}$$

where $\psi\{\cdot\}$ is an aggregation function to unify all MAD values. In lecture, several GAD functions have proposed:

1. *GAD function based on a linear interpolation* [38]: This function defines an adequacy of a sample n for a given class c, considering a fuzzy mixed connectivity based on T-norm (triangular norm) and S-norm (or T-conorm):

$$G_{n,c}^{(1)} = \epsilon \cdot T\{M_{n,1,c}, \ldots, M_{n,D,c}\} + (1-\epsilon) \cdot S\{M_{n,1,c}, \ldots, M_{n,D,c}\} \tag{10}$$

   where ϵ is an exigency value, $\epsilon \rightarrow \{0, 1\}$, $T\{\cdot\}$ is the T–norm function, and $S\{\cdot\}$ is the S–norm function. Usually, $T\{\cdot\}$ and $S\{\cdot\}$ are expressed as max–min operators, Hamasher operator, probabilistic operator, among others. However, the paper applies only max–min operators.
2. *GAD function based on a geometrical interpolation* [18,39]: This function is an algebraic expression based on a weighted mixed connectivity of T-norm and S-norm, given by:

$$G_{n,c}^{(2)} = T\{M_{n,1,c}, \ldots, M_{n,D,c}\}^{(1-\epsilon)} \cdot S\{M_{n,1,c}, \ldots, M_{n,D,c}\}^{\epsilon} \tag{11}$$

3. *GAD function based on Yager–Rybalov Triple Π Operator* [18,20,40]: This function is a full reinforcement operator called symmetrical sum, where ϵ is ignored. This GAD function is expressed as:

$$G_{n,c}^{(3)} = \frac{\prod_{d=1}^{D} M_{n,d,c}}{\prod_{d=1}^{D} M_{n,d,c} + \prod_{d=1}^{D}[1 - M_{n,d,c}]} \tag{12}$$

   To understand Eq. (12), MADs will be important for all features in a class c. If Eq. (12) allows defining an aggregation operator for all MADs then it is higher than each MAD separately calculated and the highest marginal degrees are reinforced [19]. The above is called positive reinforcement. On the other hand, if any MAD has low marginal degrees then the aggregation will be weaker than the weakest marginal degrees. The above is called negative reinforcement. By unifying both reinforcements, the total reinforcement is calculated. Due to the absence of the ϵ- parameter, the algorithm will generate a single set of classes but it does not create new classes. This condition avoids the spurious classes generation.
4. *GAD function based on the mean operator triple Π* [18]: This function is a modification of Eq. (12) but is not a complete reinforcement due to the mean is calculated between the maximum and minimum values. Nevertheless, $G_{n,c}^{(4)}$ fulfills the properties of a total reinforcement:

$$G_{n,c}^{(4)} = \frac{1}{1 + \prod_{d=1}^{D}\left[\frac{1-M_{n,d,c}}{M_{n,d,c}}\right]^{\frac{1}{D}}} \tag{13}$$

   Eq. (13) is based on the arithmetic mean, $\frac{1}{D}$, which is expressed by the number of features.
5. *Typicality and intuitionistic global adequacy degree (TIGAD)* [21]: This function is a recent modification where the typicality and intuitionistic fuzzy sets (IFSs) theory are used to create a new GAD function. The function is expressed as:

$$TIGAD_{n,c} = \sigma \cdot (IG_{n,c})^a + \vartheta \cdot (GTD_{n,c})^b \tag{14}$$

   where:

   - $IG_{n,c}$ is the intuitionistic global adequacy degree (IGAD), for a sample n associated with a class c.
   - $GTD_{n,c}$ is the global typicality degree (GTD), for a sample n associated with a class c.
   - σ and ϑ are weighting parameters.
   - a and b are adjustment parameters.

   By default, $\sigma = \vartheta = \frac{1}{2}$ and $a = b = 2$. The term $IG_{n,c}$ refers to intuitionistic global membership degree and is expressed as the sum between the GAD value and the hesitancy degree of the GAD function, $IG_{n,c} = G_{n,c} + G\pi_{n,c}$, which $G\pi_{n,c} = 1 - G_{n,c} - NG_{n,c}$, being $NG_{n,c}$ a non-membership degree function. Usually, $NG_{n,c}$ can be defined by Yager's, Sugeno–Hamasher's or Dombi's intuitionistic fuzzy complements (for more information, see [21,41]). On the other hand, the term $GTD_{n,c}$ is based on Eqs. (12) and (13). $GTD_{n,c}$ is calculated by the resemblance measure, $r_{n,d,c}$, and the dissimilarity measure, $ds_{n,d,c}$, where:

   - $r_{n,d,c}$ is associated with the MAD function, $r_{n,d,c} = M_{n,d,c}$ (see Eqs. (2), (3), (4)).
   - $ds_{n,d,c}$ is the Cauchy function with non-linear dependence between resemblance and dissimilarity, expressed as:

$$ds_{n,d,c} = \frac{1}{1 + r_{n,d,c}} \tag{15}$$

   Considering $r_{n,d,c}$ and $ds_{n,d,c}$, the GTD function can be expressed by means of Eqs. (12) and (13), as shown below [21]:

$$GTD_{n,c}^{(1)} = \frac{\prod_{d=1}^{D} r_{n,d,c} \cdot ds_{n,d,c}}{\prod_{d=1}^{D} r_{n,d,c} \cdot ds_{n,d,c} + \prod_{d=1}^{D}[(1 - r_{n,d,c}) \cdot (1 - ds_{n,d,c})]} \tag{16}$$

$$GTD_{n,c}^{(2)} = \frac{1}{1 + \prod_{d=1}^{D}\left[\frac{(1-r_{n,d,c})\cdot(1-ds_{n,d,c})}{r_{n,d,c}\cdot ds_{n,d,c}}\right]^{\frac{1}{D}}} \tag{17}$$

Eqs. (16) and (17) will be considered in this paper to calculate the TIGAD value.

2.1.4. Comparison between GAD value and GAD(NIC)

The four step is to compare the GAD value and the GAD value from NIC or *GAD*(*NIC*) which one class is created or not. Initially, the LAMDA algorithm calculates the maximum GAD value, expressed as:

$$\max GAD = \max\{G_{1,1}, \ldots, G_{n,c}, \ldots, G_{N,K}\} \tag{18}$$

where K is the maximum number of classes. Considering Eq. (18), LAMDA carries out the following comparison:

- if $\max GAD > G(NIC)_{n,c=NIC}$, then a new class is generated due to the threshold was exceeded. In this case, Eqs. (5) and (6) are applied.
- if $\max GAD \leq G(NIC)_{n,c=NIC}$, then the class is not generated and therefore, Eqs. (7) and (8) are applied.

This procedure is applied for each sample n. Taking into account NIC, the TIGAD function calculates the NIC value according to the typical parameters of Yager's, Sugeno–Hamasher's or Dombi's intuitionistic fuzzy complements [21]. Therefore, the NIC values are shown below:

- $TIGAD_{n,c=0} \rightarrow [0.3428, 0.7178]$ for Yager and Dombi's intuitionistic fuzzy complement.
- $TIGAD_{n,c=0} \rightarrow [0.3428, 0.44]$ for Sugeno–Hamasher's intuitionistic fuzzy complement.

In Figs. 1 and 2, a general scheme of LAMDA and a scheme based on TIGAD are illustrated. In addition, the appendix section shows a numerical example of the LAMDA algorithm.

2.2. ICC index

The inter-class contrast (ICC) index is a fuzzy cluster validity index, expressed as [27]:

$$ICC = \frac{S_{Be}}{N} \cdot D_{min} \cdot \sqrt{c} \tag{19}$$

where:

- S_{Be} estimates the number of elements or samples for a cluster through the sum of all element's fuzzy memberships on the cluster. The above is called between-class scatter matrix and is given by (K is the number of classes and $tr(\cdot)$ is the trace of the matrix):

$$S_{Be} = tr(S_{Be}) = \sum_{c=1}^{K}\sum_{n=1}^{N} \mu_c(x_n) \cdot \|m_c - m\|^2 \tag{20}$$

- $\mu_c(x_n)$ is the membership degree of a sample $x_n = \{x_{n,1}, \ldots, x_{n,d}, \ldots, x_{n,D}\}$, with respect to a class c, being $\mu_c(x_n) \rightarrow [0, 1]$.
- m is the center of the whole data, given by:

$$m = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{21}$$

- m_c is the mean or center of a class c, where $m_c = \{m_{c,1}, \ldots, m_{c,d}, \ldots, m_{c,D}\}$. The above is obtained from a fuzzy clustering algorithm.
- D_{min} is the minimum Euclidean distance among centers, expressed as:

$$D_{min} = \min_{1 \leq i \leq K}\left\{\min_{c+1 \leq j \leq K-1}\left\{\|m_{i,d} - m_{j,d}\|\right\}\right\} \tag{22}$$

Eq. (19) has five characteristics, as described below:

- If the centers are located at the limits of their clusters then S_{Be} has a small value. Otherwise, if the centers are located inside their clusters then S_{Be} has a large value and the quality of fuzzy partition is high.
- $\sqrt{c}$ avoids the inter-class effect from reaching the maximum ICC value for a class smaller than the optimum class.
- $\sqrt{c}$ constrains the increase of ICC value with respect to c.
- D_{min} restrains the maximum value for a class larger than the c-optimal.
- The factor $\frac{1}{N}$ compensates the relation between N and S_{Be}.

When ICC is the maximum value for $c = \{2, \ldots, \sqrt{c_{max}}\}$, then the best fuzzy partition and the optimal c are reached. The optimal fuzzy c-partition value is obtained as follows:

$$c^* = \arg \max_{2 \leq c \leq \sqrt{c_{max}}} ICC(c) \tag{23}$$

Analyzing Eq. (19), ICC estimates the quality of the placements of the centers inside clusters, considering Eq. (20) and the index calculates the distance between their centers if it is smaller than the distance among centers spanning different classes [27]. The above is considered an extension of Fisher's linear discriminate (FDL) or extended Fisher linear discriminant (EFDL) which the machine time is low and it is useful for online tasks. Due to the dependence of number of classes, ICC can generate an increase of c when ICC is high and the overlapping among clusters becomes high [28]. This aspect represents a disadvantage of the ICC index.

2.3. CV index

The cluster validity (CV) index is a fuzzy cluster validity index, expressed as [26]:

$$CV = \frac{D_{is}}{N} \cdot D^*_{min} \cdot \sqrt{c} \tag{24}$$

where:

- D_{is} is the dispersion among classes, given by:

$$D_{is} = \sum_{c=1}^{K} 1 - I_D(c) \tag{25}$$

- $I_D(c)$ is the information index for a class c which calculates the amount of information for each c. Usually, if $I_D(c)$ tends to 1, then the membership degrees are similar to historical data. Otherwise, if $I_D(c)$ tends to 0, then the membership degrees are different to historical data and the quality of fuzzy partition is low. A mathematical expression of $I_D(c)$ is defined by an entropy measure [42]:

$$I_D(c) = \frac{\sum_{n=1}^{N} \upsilon_c(x_n) \exp\left(\upsilon_c(x_n)\right)}{N \cdot \mu m_c \cdot \exp(\mu m_c)} \tag{26}$$

Being μm_c the maximum membership degree for a class c, $\mu m_c = \max_{1 \leq n \leq N}\{\mu_c(x_n)\}$; the expression $\upsilon_c(x_n)$ is the adequacy degree or the contrast index between the maximum membership degree for a class c and the membership degree of a sample n for a class c, $\upsilon_c(x_n) = \mu m_c - \mu_c(x_n)$.

- D^*_{min} is the minimum ultrametric distance among fuzzy sets without including the data value or centers, expressed as:

$$D^*_{min} = \min_{i \neq j}\{d^*(i, j)\} \tag{27}$$

Being $d^*(i, j)$ the ultrametric distance measure for two different clusters, i, j, given by:

$$d^*(i, j) = 1 - H(i, j) = 1 - \frac{\sum_{n=1}^{N} \min\{\mu_i(x_n), \mu_j(x_n)\}}{\sum_{n=1}^{N} \max\{\mu_i(x_n), \mu_j(x_n)\}} \tag{28}$$

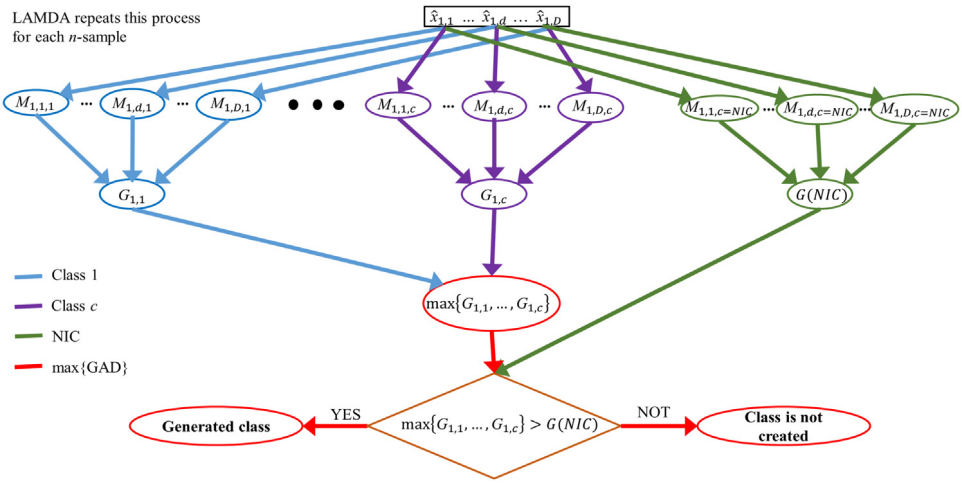

Fig. 1. General scheme of LAMDA, considering Eqs. (6) and (10)–(12).

The expression $H(i,j)$ is a fuzzy entropy measure based on [43] where the fuzzy set geometry and distances among them are taken into account. This distance allows estimating if two classes are similar or different due to the similarity index is defined as $S(i,j) = 1 - d^*(i,j)$

As in the previous index, CV is the maximum value when the optimal c is obtained. Thus, the optimal fuzzy c-partition value is given by:

$$c^* = \arg \max_{2 \le c \le \sqrt{c_{max}}} CV(c) \tag{29}$$

where c_{max} is the maximum number of clusters. Eq. (24) is a modification of Eq. (19) where several improvements are added. This index considers D_{is} and D^*_{min} through the membership degrees for each class and data values are not used. Analyzing Eq. (26), the expression $\upsilon_c(x_n) = \mu m_c - \mu_c(x_n)$ relates the membership degree for a class c with respect to the maximum membership degree for the same class which classes would be better and more similar to a Singleton function as long as D_{is} is low. Another characteristic of Eq. (24) is the ordered search of a global maximum, avoiding a local maximum when the optimal classes number is calculated. Therefore, the index evaluates the initial non-optimal fuzzy partition and the data structure is ignored [29].

3. New cluster validity indices for the LAMDA algorithm

Firstly, the definition of two indices are presented, and afterwards the importance of their expressions and interactions between them are explained below.

3.1. Proposal 1: CVGED index

Let a centers matrix, $CM = [m_{1,1}, \ldots, m_{c,d}, \ldots, m_{K,D}]$, two vectors are selected, $m_i = \{m_{i,1}, \ldots, m_{i,d}, \ldots, m_{i,D}\}$ and $m_j = \{m_{j,1}, \ldots, m_{j,d}, \ldots, m_{j,D}\}$, being i and j two classes such that $i \neq j$. Defining DV_{max} and DV_{min} as the maximum and minimum distances among clusters (using a Euclidean metric), both expression are represented as:

$$DV_{max} = \max_{i \neq j;\ i,j=[1,K]} \{\|m_i - m_j\|^2\} \tag{30}$$

$$DV_{min} = \min_{i \neq j;\ i,j=[1,K]} \{\|m_i - m_j\|^2\} \tag{31}$$

Eq. (30) represents the maximum separability likely between two clusters. If the distances among clusters are large, clusters are well-separable and the data distribution for a cluster is not overlapped with another. Meanwhile, Eq. (31) indicates the minimum separability among clusters where centers are close between them. The above allows observing those clusters whose overlapping can become critic. Although Eqs. (30) and (31) consider the separability condition for a cluster validity index, it is important to analyze the median effect in the data distribution for each cluster. According to [44], the median shows the above-average behavior of a data distribution which the separability analysis is efficient. Adding the median term in Eqs. (30) and (31), based on [44]:

$$DV^*_{max} = \frac{1}{2}[DV_{max} + med_{i \neq j;\ i,j=[1,K]}\{\|m_i - m_j\|^2\}] \tag{32}$$

$$DV^*_{min} = \frac{1}{2}[DV_{min} + med_{i \neq j;\ i,j=[1,K]}\{\|m_i - m_j\|^2\}] \tag{33}$$

The term $\frac{1}{2}$ determines an average between DV_{max} or DV_{min} and the median among clusters. By analyzing Eqs. (32) and (33), two characteristics for both equations are mentioned below:

- If $DV_{max} \rightarrow med_{i \neq j;\ i,j=[1,K]}\{\|m_i - m_j\|^2\}$ then almost the half of centers are far away among them and the number of clusters is low. In that case, the separability is maximum. Otherwise, $DV_{max} \rightarrow 0$ then Eq. (32) will be defined by median and the distance among centers calculated by an average behavior of data distribution.
- If $DV_{min} \rightarrow med_{i \neq j;\ i,j=[1,K]}\{\|m_i - m_j\|^2\}$ then almost the half of centers are close among them and the number of clusters can become high. The above indicates a high presence of spurious clusters. Otherwise, if $DV_{min} \rightarrow 0$ then Eq. (33) will be defined by median and the distance among centers by means of an average behavior of data distribution.

According to [45], the ratio of the distance among centers is expressed as the relation between the maximum and the minimum distances among centers. For hence, DR is the ratio between D^*_{max} and D^*_{min}:

$$DR = \frac{DV^*_{max}}{DV^*_{min}} = \frac{DV_{max} + med_{i \neq j;\ i,j=[1,K]}\{\|m_i - m_j\|^2\}}{DV_{min} + med_{i \neq j;\ i,j=[1,K]}\{\|m_i - m_j\|^2\}} \tag{34}$$

where DR is large if $DV^*_{max} > DV^*_{min}$. The above condition allows finding the likely maximum separation among centers.

To evaluate the compactness of clusters, the granulation error is considered in order to observe the impact of the error generated by a clustering algorithm. According to [46], the granulation

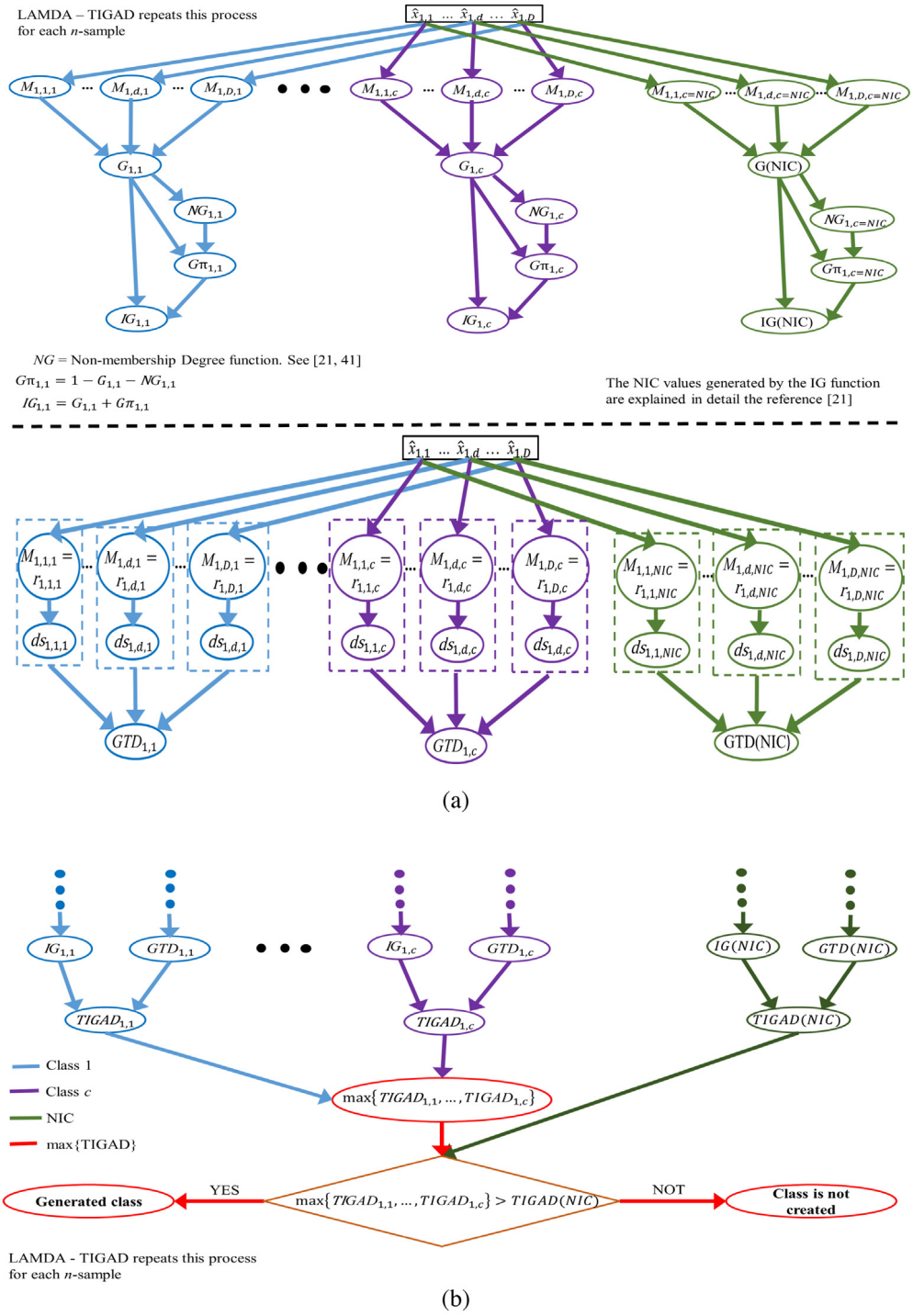

Fig. 2. General scheme of LAMDA-TIGAD based on [21]. (a) LAMDA scheme to obtain IG and GTD, (b) LAMDA scheme to obtain TIGAD.

criterion evaluates the quality of a fuzzy partition of the data and how well the centers are approached to the whole data set. For definition, $gran(error)$ is given by:

$$gran(error) = \sum_{n=1}^{N} \|x_n - \overline{\overline{x_n}}\|^2 \tag{35}$$

where $\overline{\overline{x_n}} = \{\overline{\overline{x_{n,1}}}, \ldots, \overline{\overline{x_{n,d}}}, \ldots, \overline{\overline{x_{n,D}}}\}$ is a sample reconstructed by a weighting between the corresponding cluster centers and the membership degrees. $\overline{\overline{x_n}}$ is expressed as:

$$\overline{\overline{x_n}} = \frac{1}{\Gamma|C_i|} \sum_{n=1}^{N} \mu_c(x_n) \cdot x_n \tag{36}$$

being $\Gamma|C_i| = \sum_{n=1}^{N} \mu_c(x_n)$, which represents the cardinal of a class based on Σ- count.

Eq. (35) establishes an error through the original data and the reconstructed data where the latter will depend on the number of centers and the fuzzy partition generated by the LAMDA algorithm. Obviously, a low value of $gran(error)$ is the best suitable for the original data.

By unifying Eqs. (34) and (35) and using the mathematical expression of CV index (Eq. (24)), the cluster validity index based on granulation error and the ratio of the distance (CVGED), is proposed as follows:

$$CVGED = \left(\frac{\sqrt{gran(error)}}{DR}\right) \cdot \left(\frac{\sqrt{c}}{N}\right) \tag{37}$$

CVGED is the minimum value when the optimal c is reached, defined by:

$$c^* = \arg \min_{2 \le c \le \sqrt{c_{max}}} CVGED(c) \tag{38}$$

Eq. (38) has three main characteristics:

- The factor $\frac{\sqrt{gran(error)}}{DR}$ establishes a relation between compactness and separability of clusters, where $\sqrt{gran(error)} < DR$ allows finding the best fuzzy partition due to the best compactness for each cluster and the possible maximum separation among clusters are obtained.
- The factor $\frac{\sqrt{c}}{N}$ compensates the relation between N and $\sqrt{gran(error)}$ and $\sqrt{c}$ avoids an increase of *CVGED* with respect to the number of classes.
- The term $\sqrt{gran(error)}$ represents root-mean-square error (RMSE) of the granulation error. The above improves the calculation of compactness for all clusters.

3.2. Proposal 2: CVCOD index

Let a fuzzy covariance matrix, Σ_c, expressed as [47]:

$$\Sigma_c = \frac{\sum_{n=1}^{N} \mu_c(x_n)^q [(x_n - m_c) \cdot (x_n - m_c)^{\mathrm{T}}]}{\sum_{n=1^{\mathrm{N}}} \mu_c(x_n)^q} \tag{39}$$

where q is a fuzzifier parameter. For the LAMDA algorithm, $q = 1$ due to that LAMDA does not consider this kind of parameter. To analyze the global compactness of the fuzzy partition, $GComp(c)$ is the trace of $\sum_c$ defined by [48]:

$$Gcomp(c) = \sum_{c=1}^{K} tr(\Sigma_c) \tag{40}$$

When $Gcomp(c)$ is a small value, the fuzzy partition has the best compactness. We consider the covariance matrix for data, Σ_d, for finding a relation with respect to $Gcomp(c)$:

$$\Sigma_d = \frac{\sum_{n=1}^{\mathrm{N}} (x_n - m) \cdot (x_n - m)^{\mathrm{T}}}{N - 1} \tag{41}$$

Just as Eq. (40), the global compactness of data, *GD*, is calculated as the trace of Σ_d, expressed as:

$$GD = \sum_{d=1}^{D} tr(\Sigma_d) \tag{42}$$

In order to relate Eqs. (40) and (42), we propose a ratio of compactness, *RComp*, given by:

$$RComp = \frac{GD}{Gcomp(c)} \tag{43}$$

Analyzing Eq. (43), if $GD = Gcomp(c)$ then $Rcomp = 1$ which the compactness behavior between data and the fuzzy partition are equal. The above indicates that the fuzzy sets is close to the data structure but $Gcomp(c)$ needs to be small to obtain the best fuzzy partition. Therefore, a big value of *RComp* allows finding the best compactness of fuzzy partition. By using Eqs. (34) and (43), the cluster validity index based on the ratios of covariance and distance (CVCOD) is proposed as follows:

$$CVCOD = DR \cdot Rcomp \tag{44}$$

CVCOD is the maximum value where the optimal c is reached when:

$$c^* = \arg \max_{2 \le c \le c_{max}} CVCOD(c) \tag{45}$$

Eq. (45) allows finding the best fuzzy partition when *DR* represents the likely maximum separation among clusters and *Rcomp* generates the largest value when $Gcomp(c)$ is much smaller than *GD*.

4. Computational complexity of the LAMDA algorithm and metrics

In order to analyze the computational complexity of the LAMDA algorithm and the cluster validity indices, two steps are defined: (1) By applying the LAMDA algorithm and (2) By applying the cluster validity indices.

1. *First step:* The LAMDA algorithm calculates MAD and GAD functions according to the historical data size. The MAD function generates a computational complexity $O(KND)$ but it can increase to $O(K^2ND)$ if N or D is large value. The above is related with the automatic generation of classes. On the other hand, the GAD function requires $O(KN)$ but it can increase to $O(K^2N)$ if K is a big value. By means of the computational complexity of MAD and GAD functions, the LAMDA algorithm generates $O(KND + KN) \rightarrow O(KND)$ but the automatic generation of classes can increase the complexity when $O(K^2ND + K^2N) \rightarrow O(K^2ND)$.
   When the TIGAD function is considered, the computational complexity is analyzed through *GTD* and *IG* functions. For the GTD function, the resemblance and dissimilarity measures have $O(KND)$ and therefore, the GTD function generates a computational complexity $O(KND + KN) \rightarrow O(KND)$. For The *IG* function, $G_{n,c} + G\pi_{n,c}$ is $O(KN + KN) \rightarrow O(KN)$ and therefore, *IG* has a computational complexity $O(KN)$. By considering the sum between *GTD* and *IG* functions, the TIGAD function have a computational complexity $O(KND + KN) \rightarrow O(KND)$. It is important to clarify that the TIGAD function cannot have a complexity $O(K^2ND)$ due to that the algorithm contains an automatic generation control as mentioned [21].
2. *Second step:* This step is focused on the four validity cluster indices explained in Sections 2.2, 2.3, 3.1, and 3.2. To obtain the best approach, the computational complexity is analyzed through strategy given by [49]:
   - *Computational complexity for the ICC index:* The calculation of S_{Be} generates $O(KDN + KD + ND) \rightarrow O(KDN)$ and D_{min} is a calculation between centers which the complexity is $O(K^2D)$. Therefore, the computational complexity for *ICC* is $O(KDN + K^2D) \rightarrow O(KDN)$. However, if K is large then $O(K^2DN)$.
   - *Computational complexity for the CV index:* The calculation of D^*_{min} generates $O(KN)$ due to that the ultrametric distance is a relation between two membership degrees, where the U matrix has a size of NxK. The calculation of D_{is} generates a complexity $O(KN)$ for the operation $\upsilon_c(x_n) = \mu m_c - \mu_c(x_n)$ and the operation $\exp(\upsilon_c(x_n))$ can generates a complexity $O(KNlog(KN))$. Thus, D_{is} can generate a computational complexity $O(KN + NKlog(KN))$. Keeping in mind the above, the *CV* index has a computational complexity $O(KN + KN + NKlog(KN)) \rightarrow O(KN + KNlog(KN))$ and it can increase until $O(K^2N + NK^2log(K^2N))$ if K is large.

Table 2
Summary of the computational complexity of the LAMDA algorithm and ICC, CV, CVGED and CVCOD metrics. N is the number of samples, D is the number of features, and K is the number of clusters.

Algorithm and/or metric	Computational complexity
Standard LAMDA	$O(K^2ND)$
LAMDA with TIGAD function	$O(KND)$
ICC	$O(K^2DN)$
CV	$O(K^2N + NK^2log(K^2N))$
CVGED	$O(K^2D + N)$
CVCOD	$O(KND^2 + K^2D)$

- *Computational complexity for the CVGED index:* The calculation of $gran(error)$ depends on $\Gamma|C_i|$ and $\overline{\overline{x_n}}$. The cardinal $\Gamma|C_i|$ is approached to $O(N)$ due to that the cardinal creates a vector of $1xK$ clusters and the sum is calculated around N samples. The samples reconstructed, $\overline{\overline{x_n}}$, requires $O(KND)$. Therefore, $gran(error)$ requires approximately $O(ND + KND + N) \rightarrow O(KND)$. On the other hand, the calculation of DR generates a computational complexity of $O(K^2D)$ due to that it is a relation between centers or prototypes. However, the DR calculates a median between a centers relation, which the median algorithm requires $O(N)$ and therefore, DR requires $O(N + K^2D)$. Considering the analysis of $gran(error)$ and DR, the $CVGED$ index generates $O(KND + N + K^2D) \rightarrow O(K^2D + N)$.
- *Computational complexity for the CVCOD index:* The calculation of G_{comp} requires $O(KND^2 + KD)$ due to that $O(KND^2)$ is the calculation of fuzzy covariance matrix, Σ_c, and $O(KD)$ is the computational complexity of $tr(\Sigma_c)$. On the other hand, GD requires $O(KND^2 + ND)$, where center of the whole data is included. Taking into account the computational complexity of DR, the $CVCOD$ index requires $O(KND^2 + KD + KND^2 + ND + N + K^2D) \rightarrow O(KND^2 + KD + ND + K^2D) \rightarrow O(KND^2 + K^2D)$.

In Table 2, a summary of the computational complexity is shown:

5. Experimental settings

In this section, we explain two kind of experiments to apply the ICC, CV, $CVGED$ and $CVCOD$ indices. It is important to clarify that all experiments use the LAMDA algorithm with different GAD functions explained in Section 2.1.

5.1. Experiment 1: Selected databases from UCI and KEEL

For this experiment, 16 real data sets from UCI machine learning repository (UCI, https://archive.ics.uci.edu/ml/index.php) and knowledge extraction based on evolutionary learning (KEEL, http://sci2s.ugr.es/keel/datasets.php) are used to compare the performance of the $CVGED$ and $CVCOD$ indices with respect to the ICC and CV indices. In Table 3, a description of selected databases is shown where name of the data, number of samples, number of attributes, number of classes, and data source are indicated. We selected Eqs. (6) and (10) due to that the ϵ- parameter allows generating different clusters quantities when $\epsilon \rightarrow [0, 1]$. The goal is to find a fuzzy partition where the number of clusters is equal to number of classes defined by the original databases. For this reason, if Eqs. (11) and (12) are applied then two fuzzy partitions are obtained but the number of clusters could not be equal to the number of classes of the databases. The same restriction is considered by using Eq. (13) due to that is an improvement of Eqs. (11) and (12).

Table 3
Description of real databases from UCI and KEEL repositories.

Name	Number of samples	Number of attributes	Number of classes	Source
Iris	150	4	3	UCI
Htru2	17 898	9	2	UCI
Twonorm	7400	20	2	KEEL
Ring	7400	20	2	KEEL
Vowel	990	13	11	KEEL
Wine quality white	4898	11	7	KEEL
Wine quality red	1599	11	6	KEEL
Satimage	6435	36	7	UCI
Shuttle	58 000	9	7	KEEL
Penbased	10 992	16	10	KEEL
Wisconsin	569	9	2	UCI
Yeast	1484	8	10	KEEL
Letter	20 000	16	26	KEEL
Abalone	4174	8	28	KEEL
Texture	5500	40	11	KEEL
Optdigits	5620	64	10	KEEL

To validate the performance of clustering, five evaluation metrics are applied for the best fuzzy partitions obtained by cluster validity indices. Such metrics are explained below:

- Clustering accuracy, Ac, is a metric expressed by [50]:

$$Ac = \sum_{c=1}^{K} \frac{w_c}{b_c} \tag{46}$$

where w_c is the number of samples or objects occurring in corresponding correct class and b_c is the total number of classified samples (truth ground). If $Ac = 1$, the quality of clustering is totally validated.
- Clustering recall, Rc is a metric defined as [50]:

$$Rc = \sum_{c=1}^{K} \frac{w_c}{B_c} \tag{47}$$

where B_c represents the total number of samples or objects.
- F - measure is a comprehensive metric of the quality of clustering based on Ac and Rc and can evaluate the information in clustering. F - measure is given by [51]:

$$F - measure = \frac{2 \cdot Ac \cdot Rc}{Ac + Rc} \tag{48}$$

If F-measure is high, the quality of clustering is good.
- The metric E/I is a ratio between the cluster variation, E, and the average within cluster variation, I. The ratio is given by [52]:

$$E/I = \frac{\sum_{i=1}\sum_{j=1}(m_i - m_j)^2/K^2}{\sum_{c=1}^{K} I_c/K}; \quad i \neq j \tag{49}$$

where:

$$I_c = \frac{\sum_{c=1}^{K}(x_c - s_c)^2}{N_c} \tag{50}$$

being x_c is the number of attributes in each object, N_c is the number of samples in the cluster, and $s_c = \sum x_c/N_c$, is the mean of the cluster. If E is large than I, the ratio is largest which clusters are good separated. Otherwise, if $E < I$, the quality of clustering is bad.
- The coefficient of variation, CoV, analyzes the dispersion of a data distribution, considering cluster size. This metric is defined as a ratio of the standard deviation to the mean of cluster sizes which can evaluate different cluster size. CoV

is expressed as [53]:

$$CoV = \frac{\sqrt{\sum_{c=1}^{K} \frac{N_c - \overline{h}}{K-1}}}{\sum_{c=1}^{K} N_c/K} \quad (51)$$

where $\overline{h}$ is the average cluster size. When CoV is high then the variation of data distribution will be high. Otherwise, if CoV is small then cluster sizes have a minor change, which the clustering is stable. Considering $CoV_{groundtruth}$ as the coefficient of variation of the original classes vector (from database), DCV is the difference of CV values before and after clustering, expressed by $DCV = (CoV_{groundtruth} - CoV)x100\%$ [53]. We will use DCV for the analysis of the coefficient of variation for each database and a low DCV allows finding the best data partition.

It is important to mention that each test (i.e., using one GAD and one MAD) generates 1000 iterations where ϵ-parameter is varied between 0 to 1 (intervals of 0.001, equivalent to one iteration). Therefore, 16 tests are applied for each database, considering the four cluster validity indices mentioned before.

5.2. Experiment 2: Selected open data and experimental laboratory data

In order to analyze the performance of the *CVGED* and *CVCOD* indices for other scenarios, we use the open data from Colombian government (https://datos.gov.co/ and http://medata.gov.co/) and experimental laboratory data. In Table 4, a summary of open data and experimental laboratory data are shown where name of the data, number of samples, number of attributes, and source are mentioned. It is important to clarify that the number of classes is not known and the goal is to find the optimal number of classes. Obviously, all GAD functions will be applied in the experiment and Eqs. (2) and (4) will be taken into account. Eq. (3) will not consider due to that the TIGAD function is adapted for Binomial and Gaussian MAD function [21].

5.3. Methodology for the experiments

In Figs. 3 and 4, a general scheme of methodology is illustrated. Initially, a data matrix is normalized by using Eq. (1), obtaining a normalized data matrix. This matrix is used to apply the LAMDA algorithm, generating a GAD matrix and a cluster matrix. Afterwards, one index (ICC, CV, CVGED, and CVCOD) is used to calculate the quality of clustering and its value is stored. This procedure is replied from $\epsilon = 0$ to $\epsilon = 1$, increasing the ϵ value each 0.01. The result is a matrix where each row contains the iteration value, the used ϵ value, and the calculated index value. Based on results matrix, the maximum ICC, CV, or CVCOD value or the minimum CVGED value is calculated. This calculation allows finding the best ϵ value of the LAMDA algorithm inside of results matrix. The best ϵ value is used to apply for second time the LAMDA algorithm and the main result is the best GAD matrix. Considering the best GAD matrix, the classes vector is obtained when the maximum GAD value is calculated for each row in the matrix.

In Figs. 5 and 6, the application of the methodology for the case of the TIGAD function is illustrated. The methodology is similar to Figs. 3 and 4 but the TIGAD function uses Yager's, Sugeno–Hamasher's or Dombi's intuitionistic fuzzy complements which each one considers different parameters:

- α is the parameter for Yager's intuitionistic fuzzy complements, where $\alpha \rightarrow [0, 1]$.
- Ω is the parameter for Sugeno–Hamasher's intuitionistic fuzzy complements, where $\Omega \rightarrow [0, 1]$.
- γ is the parameter for Dombi's intuitionistic fuzzy complements, where $\gamma \rightarrow [0, 0.5]$.

Due to the above, three matrices are generated for each iteration. For each matrix, the best parameter value is found according to the best value measure. In this way, the LAMDA algorithm with TIGAD function is applied again for each parameter and the final result is three TIGAD matrices and their respective classes vector. Of course this procedure requires a longer machine time but the aim is to find the most optimal values of the three mentioned parameters that allow finding the best clustering of the data.

All algorithms and cluster validity indices were tested by two laptops with the following characteristics:

- Intel Core i7-6700HQ CPU @ 2.60 GHz, 32 GB RAM, GPU: Nvidia GEFORCE GTX 980M, and Windows 10 Home Single (64 bits).
- Intel Core i7-7500U CPU @ 2.70 GHz, 6 GB RAM, GPU: Nvidia GEFORCE GTX 940MX ,Windows 10 Home Single (64 bits).

6. Results and discussion

For this section, the main results and a general discussion are shown. In order to clarify the main results, several points are mentioned below:

- The symbols ($\uparrow$) and ($\downarrow$) mean max and min value, respectively.
- These symbols are used for others metrics.
- MAD and GAD functions are represented by $M^{(\cdot)}$ and $G^{(\cdot)}$, where $(\cdot)$ indicates the kind of function.
- The TIGAD function, the GTD function, and the intuitionistic fuzzy complement function are represented as $TIGAD - GTD^{(\cdot)} - Name$, where $(\cdot)$ indicates the kind of function and *Name* is the kind of intuitionistic fuzzy complement function: Yager, Sugeno–Hamasher or Dombi. For more information about the intuitionistic fuzzy complement function, see [21,41].
- For the experiment 2, each database is entitled as *Case #*:
  - Case 1: Medellin's Population.
  - Case 2: Multidimensional index quality of life survey - Medellin.
  - Case 3: Educational Establishments of the Antioquia State.
  - Case 4: Propagation of optical comb lines spectra — ultrashort pulsed laser.
  - Case 5: Prediction about propagation of optical comb lines spectra (generated by two Mach–Zehnder Modulators).
- In order to understand the results, $M^{(1)}$ represents Eq. (2), $M^{(2)}$ represents Eq. (3), and $M^{(3)}$ represents Eq. (4). On the other hand, $G^{(1)}$ represents Eq. (10), $G^{(2)}$ represents Eq. (11), $G^{(3)}$ represents Eq. (12), and $G^{(4)}$ represents Eq. (13).
- Similar to the above, $GTD^{(1)}$ represents Eq. (16) and $GTD^{(2)}$ represents Eq. (17).
- For experiment 1, several tests were carried out by using $G^{(1)}$ and $G^{(2)}$, applying for each one $M^{(1)}$ and $M^{(2)}$.

6.1. Results and analysis for experiment 1

Initially, the best index values is calculated for each database applying Eqs. (23), (29), (38), and (45) where the optimal ϵ values are found to generate the best clustering data (these results are added in supplementary material as Table A.1). By using the optimal ϵ values, the best GAD matrices are generated and Eq. (18)

Table 4
Description of open data and experimental laboratory data.

Name	Number of samples	Number of attributes	Source
Medellin's population	357	34	Open Data (Colombia Government) [21,54,55]
Multidimensional index quality of life survey - Medellin	147	16	Open Data (Colombia Government) [56]
Educational Establishments of the Antioquia state	5050	10	Open Data (Colombia Government) [57]
Propagation of optical comb lines spectra — Ultrashort pulsed laser	2501	6	[15,21]
Predictions about propagation of optical comb lines spectra (generated by two Mach–Zehnder Modulators)	1001	19	[15]

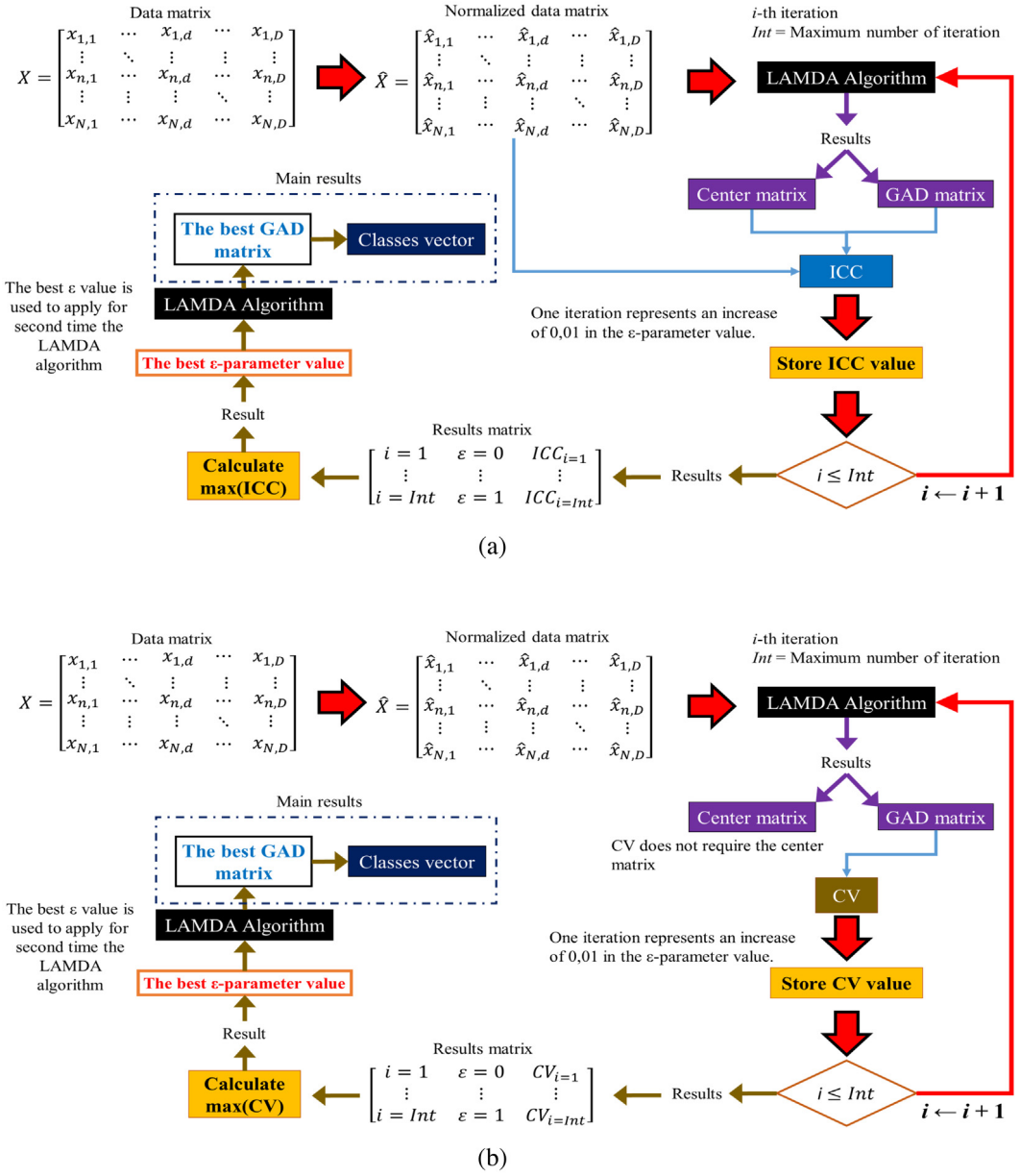

Fig. 3. (a) General methodology for the experiments using the ICC index and (b) General methodology for the experiments using the CV index.

is applied to obtain the classes vector. Considering, the best GAD and center matrices as well as the classes vector, Eqs. (46), (47), (48), (49), and (51) (considering *DCV* as the main result for the coefficient of variation) are applied to analyze the performance of

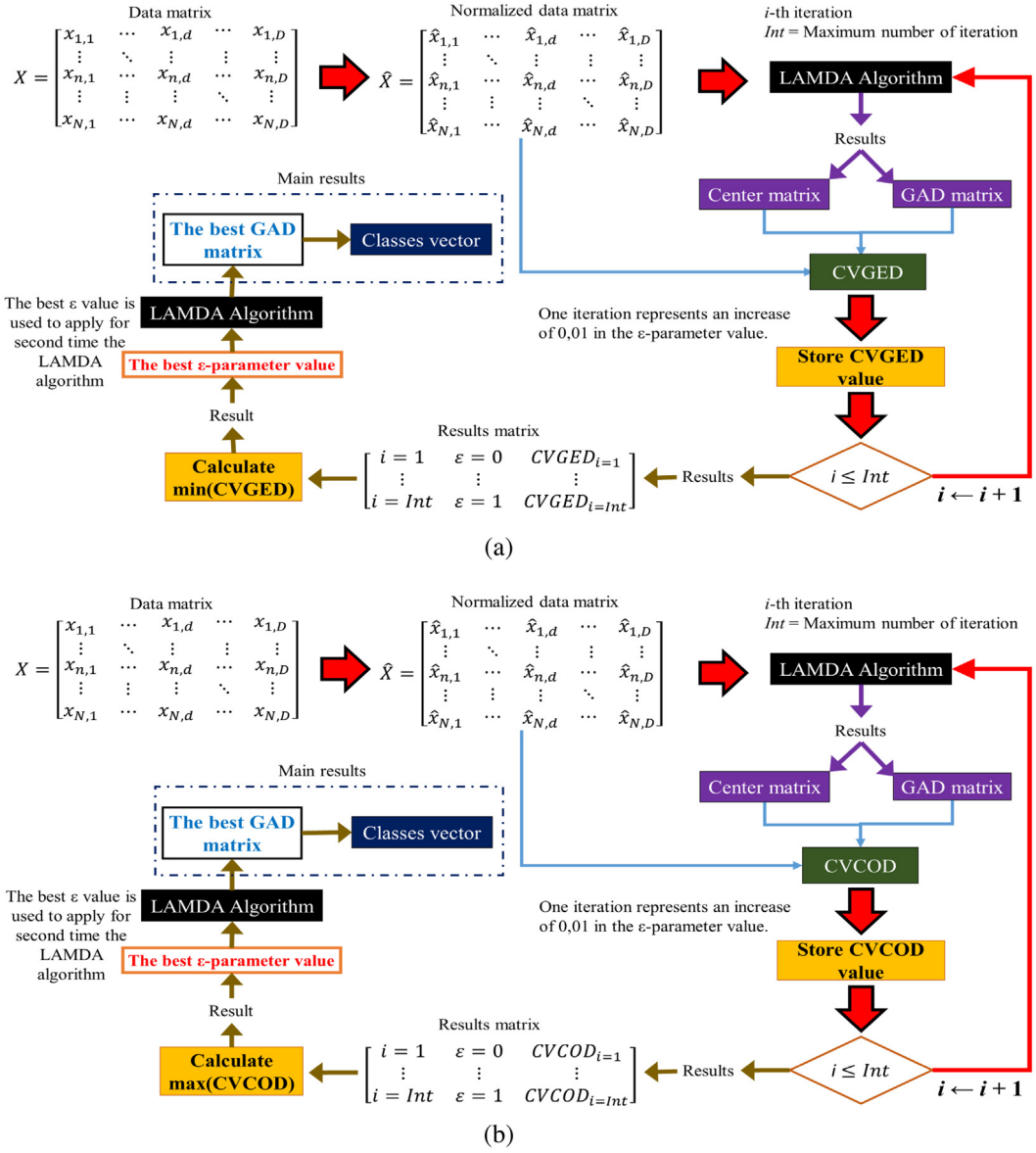

Fig. 4. (a) General methodology for the experiments using the CVGED index and (b) General methodology for the experiments using the CVCOD index.

four validity cluster indices. To use Eqs. (46) and (47), the original classes vector (ground truth) is compared with the classes vector generated by LAMDA.

Results are shown in Table 5. By analyzing results, six observations are mentioned below:

- The database "Vowell" shows that the four indices established the same set of parameters, GAD, and MAD functions. Therefore, it is the unique case with this kind of result.
- The CVGED index allowed increasing the quality of clustering for the databases "Iris", "Twonorm", "Shuttle" and "Wisconsin", considering all evaluation metrics. A similar case were identified for the CVCOD index, using the database "Iris", "Wisconsin" and "Yeast", the ICC index with the database "Iris" and "Letter", and the CV index with the database "Letter".
- By comparing the performance of E/I for all databases, the CVGED index obtained around 62.5% of the best results for E/I and the ICC and CVCOD obtained 43.75% of the best results for the same metric. In contrast, the CV index only generated 18.75% of the best results for E/I, which its performance is poor for the separability among clusters.
- Analyzing the performance of *DCV* for all databases, the CVCOD, CVGED and CV indices obtained 43.75% of the best *DCV* results but the ICC index reduced its performance with 25%.
- Analyzing the performance of F-measure for all databases, the CVCOD index obtained the highest F-measure values for 62.5% of databases and the CVGED index generated 56.25% for the same metric. In contrast, the ICC and CV indices obtained 37.5% and 25%, respectively.
- One limitation of the four indices is the accuracy due to that the databases "Vowell", "Wine Quality White", "Wine Quality Red", "Yeast", "Letter", and "Optdigits" show a *Ac* value below 0.5 for the best results of this metric. The above indicates that the performance of four indices will depend on the quality of fuzzy partition generated by the LAMDA algorithm. In addition, the type of GAD function ($G^{(1)}$ and $G^{(2)}$) and MAD function ($M^{(1)}$ and $M^{(2)}$) influence in the clustering accuracy according to data distribution.

By means of these observations, the CVGED index allows finding the best separation among clusters according to E/I which

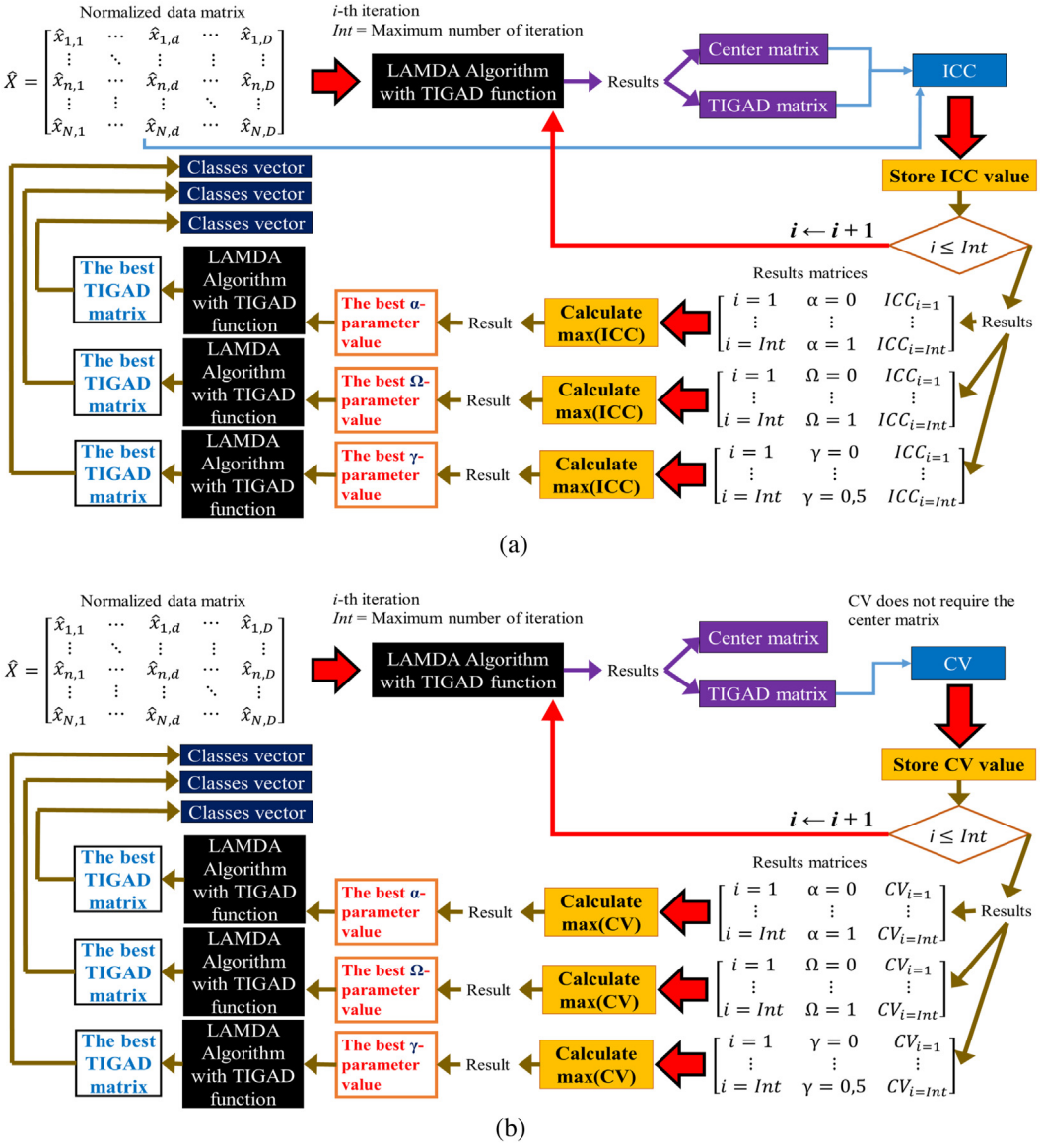

Fig. 5. (a) Methodology using TIGAD function and the ICC index and (b) Methodology using TIGAD function and the CV index.

Eq. (34) improves the quality of clustering. However, the CVGED index generated the same performance of *DCV* with respect to the CVCOD and CV indices. Despite this draw between the three indices, the CVCOD and CVGED improved the performance of $F - measure$ with respect to the ICC and CV indices but the best performance is 62.5%, indicating the indices depend on the selection of the MAD and GAD functions and the ϵ value. The above is also observed in the accuracy results where four indices generated an accuracy value below 0.5 for 6 of 16 databases. However, the rest of databases show an accuracy value greater than 0.5 (except to the value of 0.4904, using the CV index for the database "Texture"). Calculating the overall average in the accuracy values for each index (including the standard deviation), the ICC index has an average of 0.55651 ± 0.23439, the CV index has an average of 0.54291 ± 0.21987, the CVGED index has an average of 0.55452 ± 0.24221, and the CVCOD index has an average of 0.55115 ± 0.22887. *These average show that the ICC, CVGED and CVCOD indices have a similar performance but the CV indices decrease the accuracy value.*

In order to broaden the discussion of accuracy rate, we calculated the stability criterion proposed by [58]. The results of stability are illustrated in Fig. 7 where each bar represents a stability value calculated for an index. The four indices reached the best clustering stability for databases "Iris", "Htru2", and "Wisconsin" but the worst stability is observed for "Vowel", "Yeast", "Letter", and "Abalone". The above confirms that the clustering stability depends on the kind of GAD and MAD functions as well as the ϵ parameter value. One of the possible reasons of these results is the number of clusters due to that databases "Letter" and "Abalone" contain the highest cluster numbers and therefore, the LAMDA algorithm cannot reach the best data partition. When the number of clusters is below 11 clusters, the LAMDA algorithm creates better data partitions and the clustering stability increases. Another aspect is the analysis of the overall clustering stability for the four indices. In this case, the mean and standard deviation are calculated for each index, as shown below:

- ICC index, 0.5056 ± 0.27657
- CV index, 0.4880 ± 0.26467
- CVGED index, 0.5060 ± 0.28074
- CVCOD index, 0.5190 ± 0.27487

Results indicate that *the CVCOD index increases the stability with respect to the other indices in* 3.1% *with respect to the CV index,*

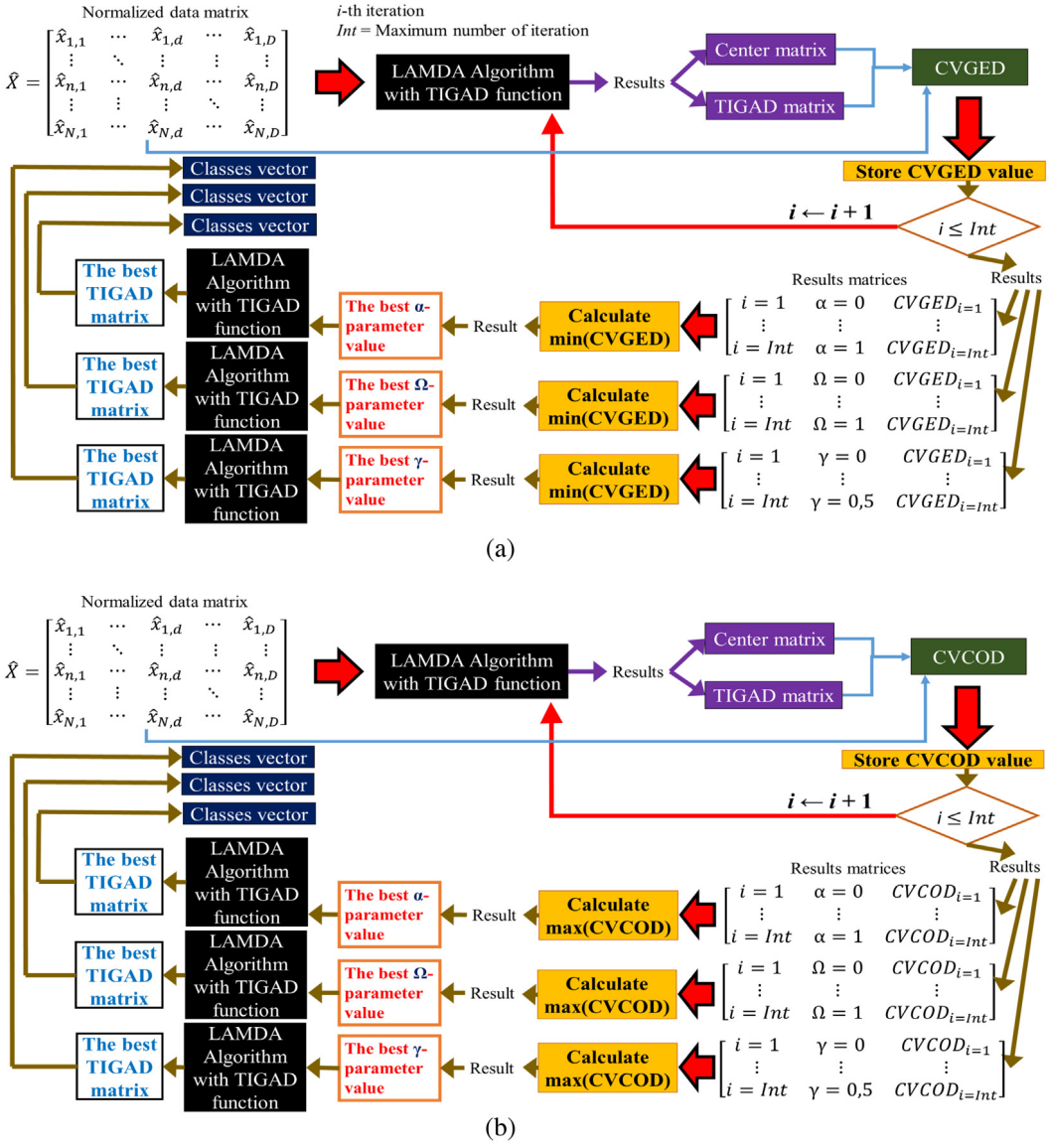

Fig. 6. (a) Methodology using TIGAD function and the CVGED index and (b) Methodology using TIGAD function and the CVCOD index.

1.34% *with respect to the ICC index, and* 1.3% *with respect to the CVGED index.*

6.2. Results and analysis for experiment 2

For the experiment 2, the best parameter values and type of MAD, GAD and GTD are found out for each database explained in Section 5.2. Afterwards, the highest value for ICC, CV, and CVCOD as well as the lowest value for CVGED are calculated through GAD and TIGAD values (see Section 2.1.4). The main goal is to observe the data partition and its distribution according to the number of clusters, comparing GAD and TIGAD functions. Results are shown in supplementary material (Tables A.2–A.6).

Selecting the best parameters and type of MAD, GAD, and GTD, a set of clustering graphics are generated through the classes vector obtained by the same strategy mentioned in Section 5.3. In Fig. 8, the best data partition for Medellín's population is illustrated. Analyzing such result, the ICC index found 10 clusters as the optimal number of clusters for both scenarios but the CV index determined 38 and 44 clusters in the same case. Nevertheless, the CVGED and CVCOD indices calculated a number of clusters in a similar range than the CV index. Considering [16], they affirmed that 10 clusters can show a data partition in terms of age but a large number of clusters is required. Therefore, the CV, CVGED and CVCOD indices fulfill the above condition due to the number of clusters is appropriate. Another observation is the CV, CVGED and CVCOD indices have the same GTD and MAD function when the TIGAD function is applied in database. The above indicates that 41 or 44 clusters is an optimal number of clusters for this kind of database.

In Fig. 9, the best data partition for quality of life survey in Medellín is illustrated. Observing such results, five cases generated the same number of clusters (3 clusters) and four of them have a similar distribution. However, the number of samples for the class 3 is very low which it could be a spurious class. Nevertheless, the CVCOD index found a better sample distribution for the class 3 if the type of MAD is Gaussian ($M^{(1)}$), the type of complement is Yager, and $GTD^{(1)}$. For the rest of the results, the number of clusters increased but 10 or 12 clusters allow analyzing other aspects of this kind of database. The above could be a quality of life scale between 1 and 10 or 1 and 12, where 1 is the worst quality and 10 or 12 is the best quality. Such scales can show more information about quality of life survey in Medellín.

Table 5
Results of experiment 1.

Name of data	Index	ϵ-parameter	GAD	MAD	E/I(↑)	DCV (%)(↓)	Ac(↑)	Rc(↑)	F-measure(↑)
Iris	ICC(↑)	0.9580	$G^{(2)}$	$M^{(1)}$	**2.4593**	**13.1148**	**0.7**	**0.7692**	**0.7330**
	CV(↑)	0.9190	$G^{(1)}$	$M^{(2)}$	0.6533	13.2665	0.6067	0.7177	0.6575
	CVGED(↓)	0.9580	$G^{(2)}$	$M^{(1)}$	**2.4593**	**13.1148**	**0.7**	**0.7692**	**0.7330**
	CVCOD(↑)	0.9580	$G^{(2)}$	$M^{(1)}$	**2.4593**	**13.1148**	**0.7**	**0.7692**	**0.7330**
Htru2	ICC(↑)	0.4450	$G^{(1)}$	$M^{(2)}$	2.5771	**0**	**1**	**1**	**1**
	CV(↑)	0.4680	$G^{(1)}$	$M^{(2)}$	2.5680	**0**	0.9499	0.9514	0.9506
	CVGED(↓)	0.6950	$G^{(2)}$	$M^{(1)}$	**2.5788**	0.12106	**1**	**1**	**1**
	CVCOD(↑)	0.6950	$G^{(2)}$	$M^{(1)}$	**2.5788**	0.12106	**1**	**1**	**1**
Twonorm	ICC(↑)	0.8080	$G^{(2)}$	$M^{(1)}$	0.1434	0.68377	0.8705	0.8854	0.8779
	CV(↑)	0.8600	$G^{(2)}$	$M^{(1)}$	0.1369	0.71431	0.8586	0.8762	0.8673
	CVGED(↓)	0.8030	$G^{(2)}$	$M^{(1)}$	**0.1442**	**0.67888**	**0.8709**	**0.8857**	**0.8783**
	CVCOD(↑)	0.0990	$G^{(1)}$	$M^{(1)}$	0.0439	0.72084	0.7047	0.7720	0.7369
Ring	ICC(↑)	0.9290	$G^{(2)}$	$M^{(1)}$	**0.0265**	1.46671	0.5361	0.6831	0.6007
	CV(↑)	0.9330	$G^{(2)}$	$M^{(1)}$	0.0203	1.31665	0.5453	0.6874	0.6081
	CVGED(↓)	0.0720	$G^{(1)}$	$M^{(1)}$	0.0195	1.26622	**0.5497**	**0.6895**	**0.6117**
	CVCOD(↑)	0.0730	$G^{(1)}$	$M^{(1)}$	0.0195	**1.26573**	0.5451	0.6873	0.6080
Vowell	ICC(↑)	0.8320	$G^{(1)}$	$M^{(1)}$	7.5620	5.44518	0.1636	0.5446	0.2517
	CV(↑)	0.8320	$G^{(1)}$	$M^{(1)}$	7.5620	5.44518	0.1636	0.5446	0.2517
	CVGED(↓)	0.8320	$G^{(1)}$	$M^{(1)}$	7.5620	5.44518	0.1636	0.5446	0.2517
	CVCOD(↑)	0.8320	$G^{(1)}$	$M^{(1)}$	7.5620	5.44518	0.1636	0.5446	0.2517
WineQuality White	ICC(↑)	0.5060	$G^{(2)}$	$M^{(2)}$	1.5559	0.57540	**0.3575**	**0.6088**	**0.4505**
	CV(↑)	0.5110	$G^{(2)}$	$M^{(2)}$	1.6155	0.55889	0.3526	0.6070	0.4461
	CVGED(↓)	0.5910	$G^{(1)}$	$M^{(1)}$	**3.7232**	**0.15100**	0.3342	0.6003	0.4294
	CVCOD(↑)	0.4380	$G^{(2)}$	$M^{(2)}$	1.7679	0.67843	0.3467	0.6048	0.4407
WineQuality Red	ICC(↑)	0.6480	$G^{(2)}$	$M^{(1)}$	**1.6717**	**0.90117**	0.3340	0.6002	0.4291
	CV(↑)	0.3710	$G^{(2)}$	$M^{(2)}$	1.3574	1.66124	**0.3971**	**0.6239**	**0.4853**
	CVGED(↓)	0.6480	$G^{(2)}$	$M^{(1)}$	**1.6717**	**0.90117**	0.3340	0.6002	0.4291
	CVCOD(↑)	0.5770	$G^{(2)}$	$M^{(1)}$	0.8310	1.70942	0.3290	0.5984	0.4245
Satimage	ICC(↑)	0.7240	$G^{(2)}$	$M^{(1)}$	**3.3558**	0.11801	0.6019	0.7152	0.6537
	CV(↑)	0.7250	$G^{(2)}$	$M^{(1)}$	3.2257	**0.05846**	0.5789	0.7037	0.6352
	CVGED(↓)	0.3080	$G^{(1)}$	$M^{(1)}$	2.2739	0.32703	**0.6159**	**0.7225**	**0.6649**
	CVCOD(↑)	0.3080	$G^{(1)}$	$M^{(1)}$	2.2739	0.32703	**0.6159**	**0.7225**	**0.6649**
Shuttle	ICC(↑)	0.4010	$G^{(2)}$	$M^{(1)}$	6.3950	0.00520	0.6918	0.7644	0.7263
	CV(↑)	0.4750	$G^{(2)}$	$M^{(1)}$	6.2012	**0.00511**	0.6919	0.7645	0.7264
	CVGED(↓)	0.4100	$G^{(2)}$	$M^{(1)}$	**6.4413**	**0.00511**	**0.7061**	**0.7728**	**0.7379**
	CVCOD(↑)	0.4100	$G^{(2)}$	$M^{(1)}$	**6.4413**	0.00520	**0.7061**	**0.7728**	**0.7379**
Penbased	ICC(↑)	0.6530	$G^{(2)}$	$M^{(1)}$	2.6143	1.50133	0.5311	0.6808	0.5967
	CV(↑)	0.6610	$G^{(2)}$	$M^{(1)}$	2.7049	**1.27510**	**0.5751**	**0.7018**	**0.6322**
	CVGED(↓)	0.5430	$G^{(1)}$	$M^{(1)}$	**2.8248**	1.35531	0.5665	0.6976	0.6253
	CVCOD(↑)	0.5430	$G^{(1)}$	$M^{(1)}$	**2.8248**	1.35531	0.5665	0.6976	0.6253
Wisconsin	ICC(↑)	0.8730	$G^{(2)}$	$M^{(1)}$	0.7485	2.21618	0.7936	0.8289	0.8108
	CV(↑)	0.9180	$G^{(2)}$	$M^{(1)}$	0.6890	2.33548	0.7804	0.8199	0.7997
	CVGED(↓)	0.8680	$G^{(2)}$	$M^{(1)}$	**0.8020**	**2.05356**	**0.8111**	**0.8411**	**0.8259**
	CVCOD(↑)	0.8680	$G^{(2)}$	$M^{(1)}$	**0.8020**	**2.05356**	**0.8111**	**0.8411**	**0.8259**
Yeast	ICC(↑)	0.6170	$G^{(2)}$	$M^{(2)}$	1.4783	2.13787	0.2884	0.5843	0.3862
	CV(↑)	0.6170	$G^{(2)}$	$M^{(2)}$	1.4783	2.13787	0.2884	0.5843	0.3862
	CVGED(↓)	0.6450	$G^{(2)}$	$M^{(2)}$	1.4843	3.06096	0.2837	0.5826	0.3816
	CVCOD(↑)	0.5900	$G^{(2)}$	$M^{(2)}$	**1.6774**	**1.92055**	**0.3019**	**0.5889**	**0.3992**
Letter	ICC(↑)	0.1960	$G^{(2)}$	$M^{(2)}$	**2.5213**	**2.24397**	**0.3106**	**0.5919**	**0.4074**
	CV(↑)	0.1960	$G^{(2)}$	$M^{(2)}$	**2.5213**	**2.24397**	**0.3106**	**0.5919**	**0.4074**
	CVGED(↓)	0.5270	$G^{(1)}$	$M^{(1)}$	1.7968	2.55391	0.2418	0.5688	0.3393
	CVCOD(↑)	0.5270	$G^{(1)}$	$M^{(1)}$	1.7968	2.55391	0.2418	0.5688	0.3393
Abalone	ICC(↑)	0.9460	$G^{(1)}$	$M^{(1)}$	22.6441	0.87312	**0.7360**	**0.7911**	**0.7626**
	CV(↑)	0.0630	$G^{(2)}$	$M^{(1)}$	**23.7290**	**0.79547**	0.7096	0.7750	0.7409
	CVGED(↓)	0.9460	$G^{(1)}$	$M^{(1)}$	22.6441	0.87312	**0.7360**	**0.7911**	**0.7626**
	CVCOD(↑)	0.9460	$G^{(1)}$	$M^{(1)}$	22.6441	0.87312	**0.7360**	**0.7911**	**0.7626**
Texture	ICC(↑)	0.5000	$G^{(2)}$	$M^{(1)}$	6.8522	3.50771	0.5807	0.7046	0.6367
	CV(↑)	0.5380	$G^{(1)}$	$M^{(1)}$	6.1691	3.63978	0.4904	0.6624	0.5635
	CVGED(↓)	0.4960	$G^{(2)}$	$M^{(1)}$	**8.0838**	3.71376	0.5127	0.6724	0.5818
	CVCOD(↑)	0.4340	$G^{(2)}$	$M^{(1)}$	7.6106	**3.09968**	**0.5875**	**0.7079**	**0.6421**
Optdigits	ICC(↑)	0.7550	$G^{(2)}$	$M^{(1)}$	**1.2405**	3.39672	0.4084	0.6283	0.4950
	CV(↑)	0.7690	$G^{(2)}$	$M^{(1)}$	1.0435	3.55350	0.3874	0.6201	0.4769
	CVGED(↓)	0.7600	$G^{(2)}$	$M^{(1)}$	0.8824	2.88139	0.4461	0.6435	0.5269
	CVCOD(↑)	0.8900	$G^{(2)}$	$M^{(2)}$	0.1473	**2.50677**	**0.4626**	**0.6505**	**0.5407**

In Fig. 10, the best data partition for educational establishments in Antioquia (Colombia) is illustrated. Initially, the number of clusters is high (51, 45, and 31 clusters) but the number of clusters decreases for the rest of data partitions. Therefore, the

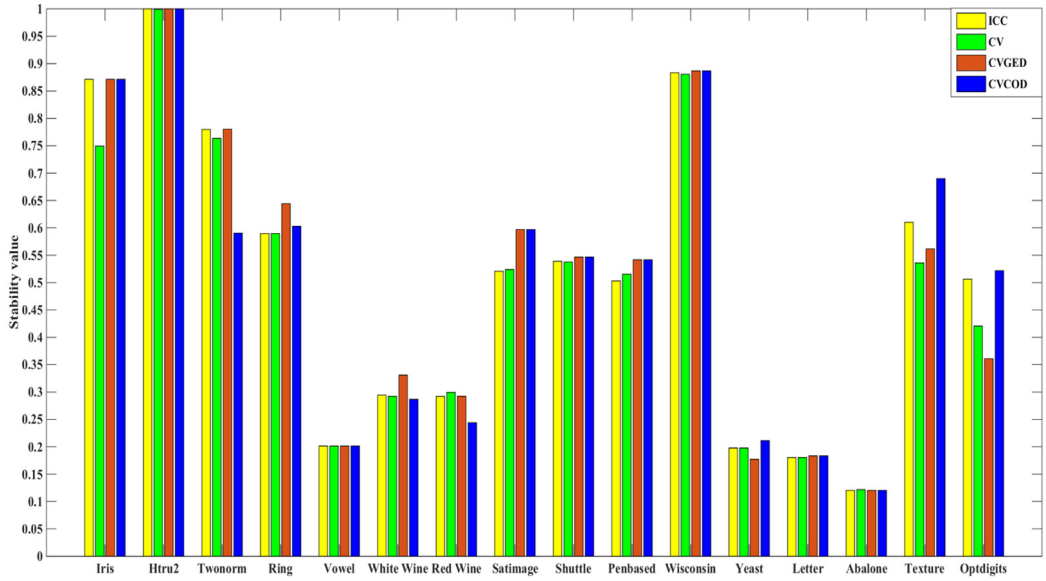

Fig. 7. Stability value for each database, comparing the ICC, CV, CVGED, and CVCOD indices — experiment 1.

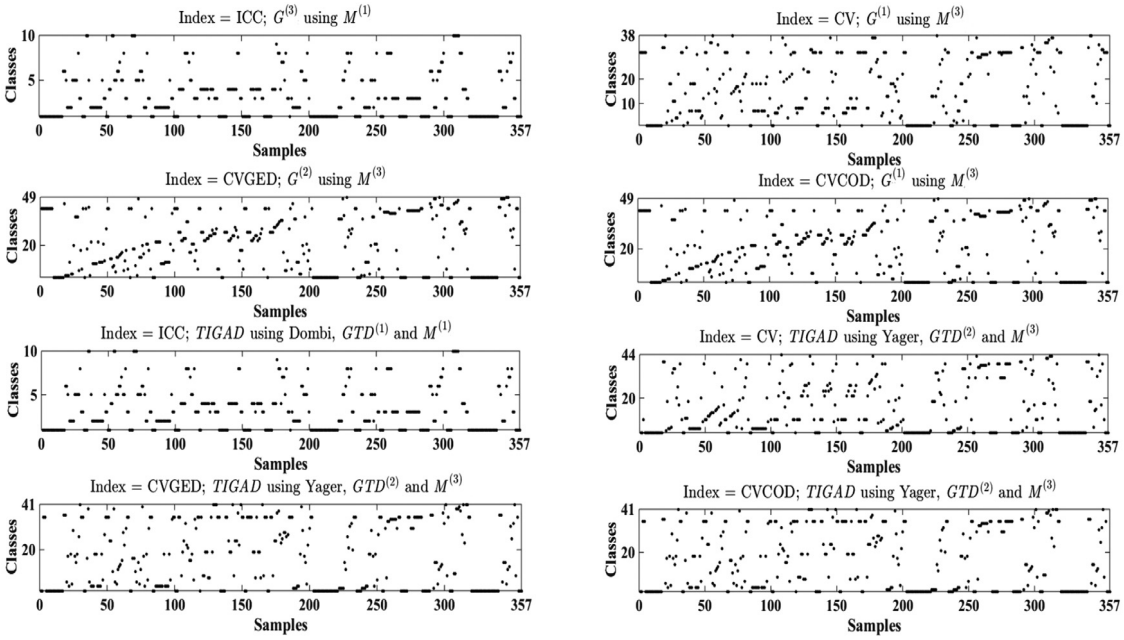

Fig. 8. The best results of the clustering — Medellín's Population.

number of clusters is reduced until 2 or 3 clusters as long as two possible scenarios are considered, as mentioned below:

- If the number of clusters is 2, then classes can represent a set of students who passed and another set of students who reproved.
- If the number of clusters is 3, then classes can show a set of students who passed, a set of students who reproved, and a set of students who deserted.

Analyzing the data information, the best partition is generated with 3 clusters due to that the original database contains information about the passed, reproved and deserted students. For the same experiment, 11 and 12 clusters were generated when the type of complement is Yager ($GTD^{(2)}$ and Gaussian function or $M^{(3)}$), and the type of complement is Dombi ($GTD^{(1)}$ and Gaussian function or $M^{(3)}$). Taking into account the above, the classes 4–11 or 4–12 can indicate others relevant data about approved, reproved, and deserted students. We suppose that this kind of information could help to a human expert in education for better decision-making. Nevertheless, we only expected 3 clusters due to that the database shows approval rate, disapproval rate, dropout rate, and number of enrollment for different levels.

In Fig. 11, the best data partition for propagation of optical comb lines spectra using an ultrashort pulsed laser is illustrated. Initially, the database is grouped in 38, 41, and 43 clusters where most of the classes are concentrated in the wavelength range whose optical frequency combs are generated. However, the optimal number of clusters is 34, according to [16]. Therefore, the CV index with Yager's intuitionistic fuzzy complement, $GTD^{(2)}$ and Gaussian MAD function $M^{(3)}$, and the CVGED and CVCOD indices with Yager's intuitionistic fuzzy complement, $GTD^{(1)}$ and Gaussian MAD function $M^{(3)}$ generate 34 clusters. Comparing such results,

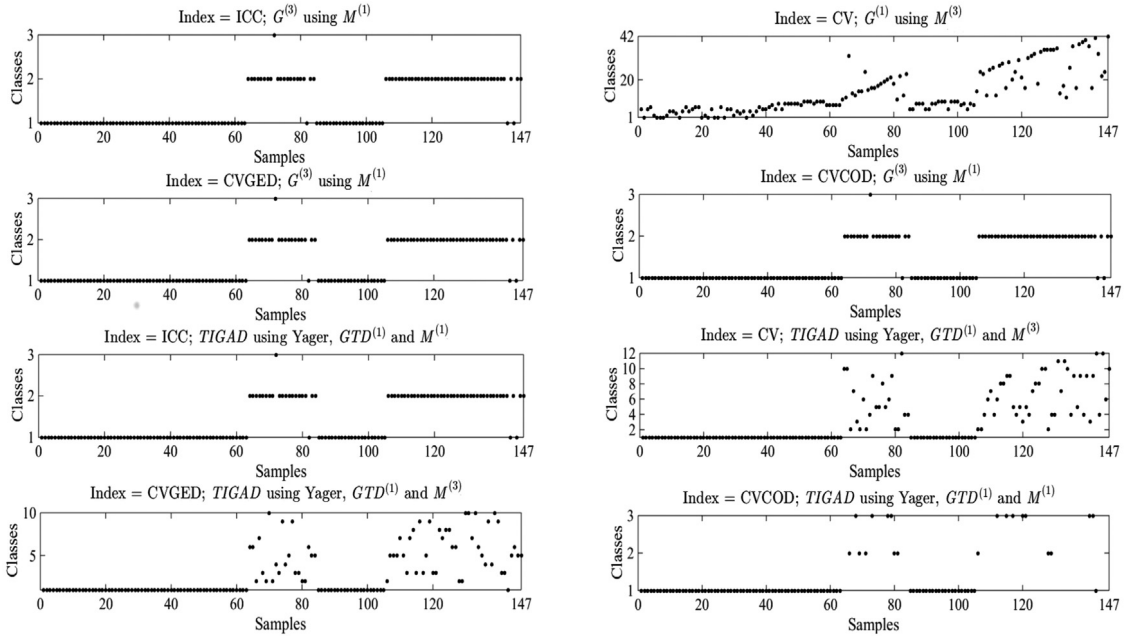

Fig. 9. The best results of the clustering – Multidimensional index: quality of life survey – Medellín.

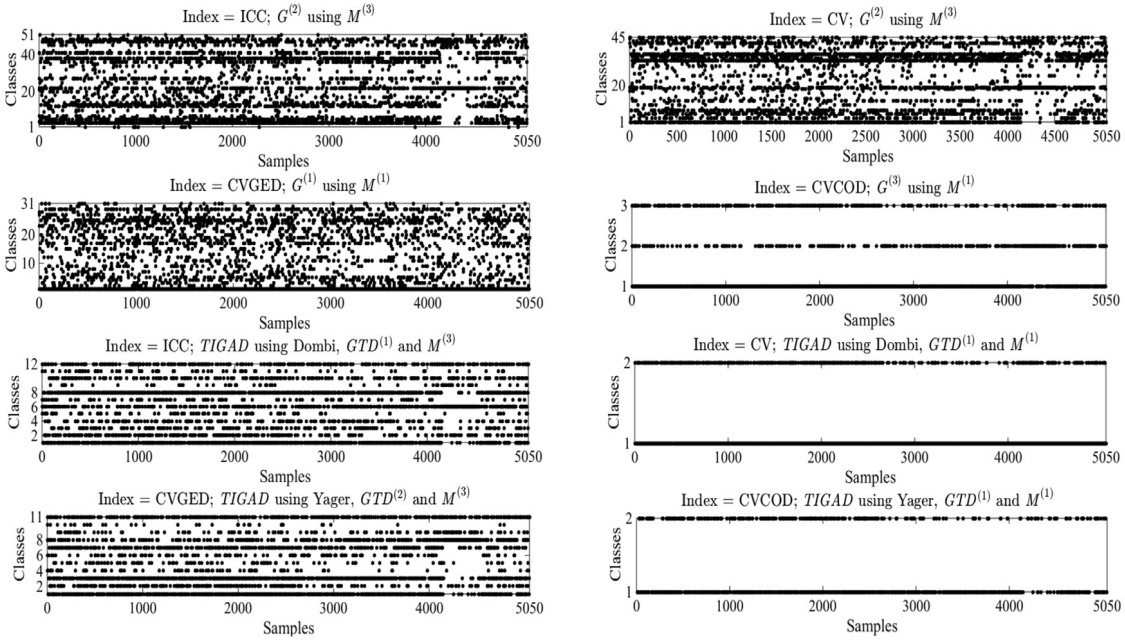

Fig. 10. The best results of the clustering — Educational Establishments of the Antioquia State.

the data partition are similar but the CVGED and CVCOD indices found the same clustering. In this manner, both indices allows selecting the most recommendable partition when the number of clusters is equal for different cluster validity indices.

In Fig. 12, the best data partition for prediction of optical comb lines spectra generated by Mach–Zehnder modulators is illustrated. As in the previous database, most of classes are created in the wavelength range where optical frequency combs are generated. However, the number of clusters is lower than the previous database due to that the spectral width is small. According to [15], predictions contain small power and phase variations obtained by optical spectra measured in real time. Considering the above, the number of clusters must be low and therefore, all data partitions generated by the TIGAD function fulfill such condition. Observing these partitions, the ICC, CVGED, and CVCOD indices identified between 7 and 8 clusters as the most optimal and 4 clusters for the CV index. However, it is necessary to select the most optimal number of clusters for the above observation. Considering expert's opinion, an idea is to compare the clustering results with respect to the normalized data and to evaluate the number of changes in phase and/or power. In that case, the best partition is generated by 7 clusters due to that the clustering results show the main changes of power and phase in the optical spectra (using the clustering results of the ICC and CVCOD indices). Otherwise, if the number of clusters is 8 then the class 3 does not provide enough information about the behavior of optical spectra.

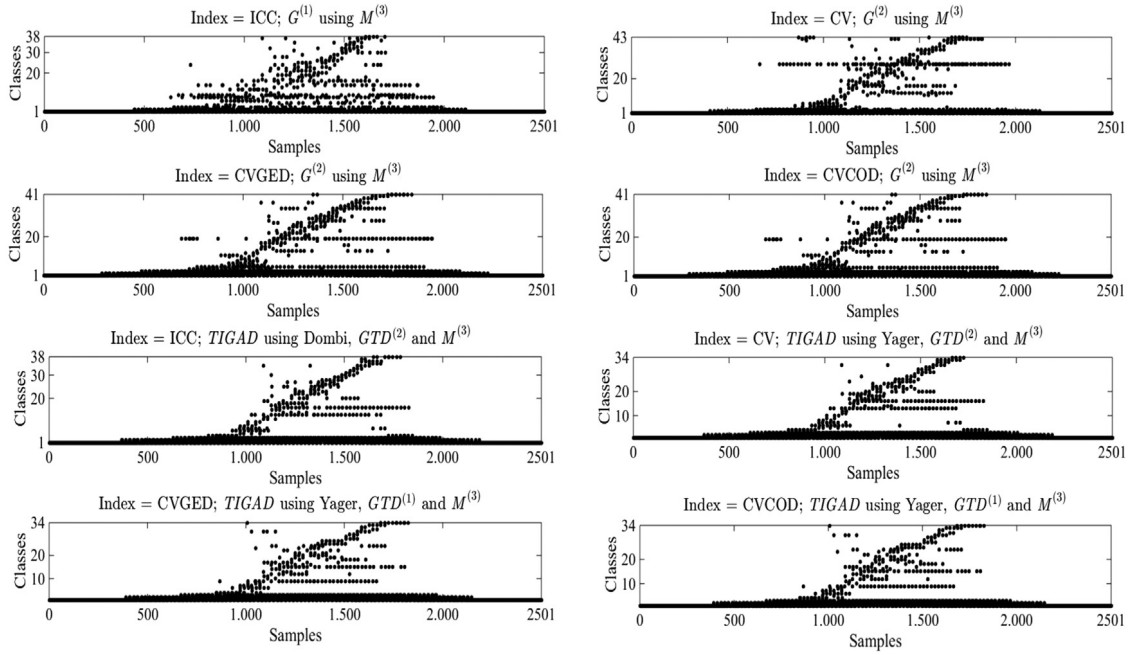

Fig. 11. The best results of the clustering – Propagation of optical comb lines spectra – ultrashort pulsed laser.

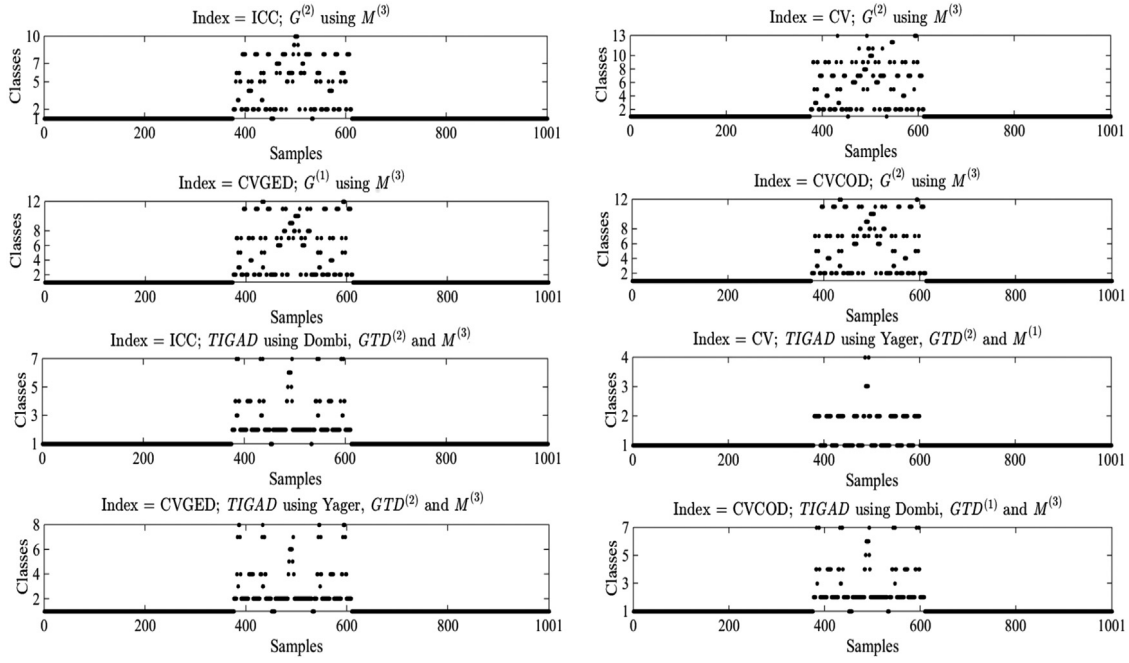

Fig. 12. The best results of the clustering — Prediction about propagation of optical comb lines spectra (generated by two Mach–Zehnder Modulators).

Taking into account the analysis of the best results for the experiment 2, several indices and metrics are used to observe other aspects of the obtained results. In Table 6, the best results of the clustering are analyzed through other indices and metrics, where each database is called with the label "case #". Metrics and indices selected are mentioned as follows (↓ is min and ↑ is max):

- *CoV* (↓) (see Eq. (51)).
- Silhouette index (↑) [59].
- Wu-and-Li index (WLI) (↓) [60].
- Bayesian information criterion (BIC) (↓) [61].
- MPE - DMPF index (↓) [62].
- Pakhira–Bandyopadhyay and Maulik index (PBM) — fuzzy granulation–degranulation (FVG) (↑) [46].
- Point symmetry index (PS) — FVG (↓) [46].
- Xie – Beni index (XB) — FVG (↓) [46].
- Fukuyama and Sugeno index (FS) — FVG (↓) [46].
- Kwon index (K) — FVG (↓) [46].
- Kim – Park and Park index or SV – FVG (↓) [46].
- Efficiency or *Eff* [63]. This metric allows finding the best fuzzy partition that reflects the hidden structure of database. Usually, $Eff \approx D$ is an optimal fuzzy partition due to the fuzzy partition is close to number of features.

According to Tables 6, 7, and 8, the CVCOD index generated the best score for 5 cases or databases and the CVGED index is ranked in the second position with respect to the number of successes. The ICC and CV indices ranked third and four positions, respectively. Analyzing each case, 5 observations are mentioned below:

- Case 1 generates a draw between the ICC and CVCOD indices but the number of clusters between them are different (10 and 49 clusters, respectively). As mentioned previously, 41 or 44 clusters is an optimal number of clusters for Medellín's population and therefore, the ICC index did not match the optimal number of clusters. Thus, the CVCOD index is close to the optimal number of clusters but we consider as an approximation.
- Case 2 shows that 3 clusters as the optimal number of clusters for multidimensional index quality of life survey — Medellín, using the CVCOD index. In the general analysis of clustering, we mentioned that 10 or 12 clusters as the optimal through database information. Comparing the two facts, all used metrics and indices are far removed from human perception in the analysis of the data. Therefore, the applied metrics and indices are focused on a small number of clusters, ignoring the needs of analysis required by a set of human experts.
- Case 3 indicates that the CVCOD index contains the best data partition with 2 clusters for educational establishments for the Antioquia state (Colombia). In the general analysis, 3 clusters is the most appropriated for grouping the number of students according to approval rate, disapproval rate and dropout rate. In this way, the CVCOD index is close to the number of clusters suggested by the general analysis.
- Case 4 shows that the best data partition is obtained by means of 34 clusters, using the CVCOD index. The same number of clusters was mentioned in the general analysis and therefore, all metrics and indices guessed with the best data partition. In fact, this result is close with respect to the human expert in the propagation of optical comb lines spectra in single-mode fiber.
- Case 5 indicates that the CVGED index generates the most optimal number of clusters (12 clusters). However, the general analysis determined 7 clusters as the most optimal number of clusters due to that classes 2–7 contain the main phase and power variations. Obviously, both results show different number of clusters but the human expert associates classes 2–12 as very small changes of power or phase in the optical spectra. In this way, 12 clusters could contain more information about the behavior of optical spectra.

For the experiment 2, the CVCOD index obtained the best performance for the 5 cases but several interpretations can be mentioned in the main analysis. Initially, the CVCOD index generated the best results by using CoV, $Silhoutte$, WLI, $PS-FVG$, $XB-FVG$, $FS-FVG$, and $K-FVG$. However, the CVCOD index tied with the CVGED index when BIC is calculated. Likewise, the CVCOD index tied with the ICC and CV indices when $MPE-DMPF$ is applied and the same index tied with the ICC index when $PBM-FVG$ and $SV-FVG$ are used. On the other hand, the CVGED index obtained the best score for Eff which this index is near of D or number of features. However, the approximation is not enough due to that the closest $Eff \approx D$ is observed in cases 4 and 5 but the rest of cases are far away. The above indicates that the CVGED index considers the data structure if the changes of data are small, as mentioned in the general analysis for case 5 and [21] (taking into account case 4).

Another factor that influenced the results is the kind of GAD, GTD and MAD function used in databases. The best results for cases 2, 3, and 4 were reached by using the TIGAD function, $GTD^{(1)}$, Yager's intuitionistic fuzzy complement, and Binomial and Gaussian MAD function. The above shows that the kind of fuzzy complement and GTD function can influence the quality of clustering. For case 1, the ICC index obtained the best result with the TIGAD function, $GTD^{(1)}$, Dombi's intuitionistic fuzzy complement, and Binomial MAD function but the CVCOD index generated a similar performance with 49 clusters. Comparing both results, the ICC index finds a minor number of clusters than CVCOD index but the general analysis mentions that between 41 and 44 clusters is the most optimal to obtain more information about database. Therefore, the CVCOD index is closer to expert's opinion than the ICC index. For case 5, the CVGED index shows a better performance than the rest of indices but the number of clusters is high (12 clusters). Although the human expert affirmed that 7 clusters is the most optimal, the CVGED index could offer more information about database. Information such as the presence of polarization mode dispersion (PMD) or self-phase modulation (SPM) could be relevant in the optical spectra data, according to [15].

6.3. Future perspectives of the LAMDA algorithm and its cluster validity measures

Based on the experiments 1 and 2, two commentaries are mentioned about the validity indices and the LAMDA algorithm:

- An alternative to improve the performance of cluster validity indices for the LAMDA algorithm is the definition of an optimization problem in order to maximize or to minimize the global solution. The above could integrate a solution based on linear programming (LP), genetic algorithms (GA) or particle-swarm optimization (PSO). To find a starting point, the use of data envelopment analysis (DEA) could be an alternative to join several cluster validity indices for LAMDA in an optimization model [64].
- Since the partitions of big data is a current scenario in clustering research, the use of Maxmin sampling algorithms and the estimation of boundary points associated with vector machines could be an alternative to apply the four indices [65]. However, a new proposal of the LAMDA algorithm for Big Data is required to apply cluster validity indices for Big Data. An alternative to begin the construction of a new LAMDA algorithm is the collaborative clustering [66] and/or deep clustering [67].

Conclusion

In this paper, two cluster validity indices, CVGED and CVCOD, are proposed for the LAMDA clustering algorithm. The CVCOD index shows the best performance for the experiments 1 and 2, where the most optimal number of clusters and quality of clustering were obtained. One advantage of the CVCOD index is to improve the stability clustering (experiment 1) and to generate the best data partition analyzed by other indices and metrics (experiment 2). Therefore, the CVCOD index can find the most optimal clustering for different kind of quantitative database, using the LAMDA algorithm. On the other hand, the CVGED index can find a data partition close to the data structure (original database), as mentioned in experiment 2 but the changes of data values must be small. In that case, a novel improvement of the CVGED index is to define the outliers ratio for each cluster, finding the best data partition with low outliers. The above may be a further work to increase the capacity for finding a data partition close to the data structure and to improve the accuracy rate when the ground truth is available in the database.

Table 6
Analysis of Experiment 2, using other indices and measures — Part 1. Case 1: Medellin's Population; Case 2: Multidimensional index quality of life survey — Medellin; and Case 3: Educational Establishments of the Antioquia State.

Case 1								
Main Index	ICC	ICC	CV	CV	CVGED	CVGED	CVCOD	CVCOD
GAD or TIGAD	$G^{(3)}$	TIGAD - $GTD^{(1)}$ - Dombi	$G^{(1)}$	TIGAD - $GTD^{(2)}$ - Yager	$G^{(2)}$	TIGAD - $GTD^{(2)}$ - Yager	$G^{(1)}$	TIGAD - $GTD^{(2)}$ - Yager
MAD	$M^{(1)}$	$M^{(1)}$	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$
Number of clusters	10	10	38	44	49	41	49	41
CoV(↓)	**0,1436**	**0,1436**	0,2993	0,3313	0,3184	0,3242	0,3184	0,3242
Silhouette(↑)	**0,3658**	**0,3658**	0,1372	0,2876	0,2701	0,2517	0,2701	0,2517
WLI(↓)	22,5646	9,7443	64,0806	14,899	5,6302	13,4219	**5,6246**	13,4219
BIC(↓)	−5018,5135	−5018,5135	−4438,3018	−4167,4637	**−3951,9527**	−4304,2514	**−3951,9527**	−4304,2514
MPE-DMPF(↓)	16,718	**16,2725**	50,0457	52,5415	76,4249	47,9023	76,4231	47,9023
PBM-FVG(↑)	2,71E−04	**6,4143E−04**	2,67E−05	8,15E−05	1,63E−04	1,07E−04	1,63E−04	1,07E−04
PS-FVG(↓)	0,0207	0,0134	0,0098	0,007	0,0047	0,0068	**0,0046**	0,0068
XB-FVG(↓)	15,2408	**6,4487**	49,6488	15,5823	42,0872	27,7248	42,0641	27,7248
FS-FVG(↓)	415,4596	175,6426	378,6744	94,7042	40,2282	80,3333	**40,2054**	80,3333
K-FVG(↓)	5456,4905	15492,9821	17781,1421	5624,3514	2317,7259	10028,053	**2317,7144**	10028,053
SV-FVG(↓)	36,1833	36,1814	259,5296	335,4749	932,0456	451,3208	932,0456	451,3208
Eff($\approx D$)	**15,7861**	10,0668	5241,5396	2874,7744	886,6066	1973,8503	885,3464	1973,8503
Number of features (D)	34	34	34	34	34	34	34	34
Number of successes	3	**5**	0	0	1	0	**5**	0
Case 2								
Main Index	ICC	ICC	CV	CV	CVGED	CVGED	CVCOD	CVCOD
GAD or TIGAD	$G^{(3)}$	TIGAD - $GTD^{(1)}$ - Yager	$G^{(1)}$	TIGAD - $GTD^{(1)}$ - Yager	$G^{(3)}$	TIGAD - $GTD^{(1)}$ - Yager	$G^{(3)}$	TIGAD - $GTD^{(1)}$ - Yager
MAD	$M^{(1)}$	$M^{(1)}$	$M^{(3)}$	$M^{(3)}$	$M^{(1)}$	$M^{(3)}$	$M^{(1)}$	$M^{(1)}$
Number of clusters	3	3	42	12	3	10	3	3
CoV(↓)	0,1414	0,1414	0,4806	0,2948	0,1414	0,2689	0,1414	**0,1258**
Silhouette(↑)	0,533	0,533	−0,0563	0,2028	0,533	0,0398	0,533	**0,6722**
WLI(↓)	0,5423	0,3405	10,4698	11,9699	0,5423	5,1788	0,5423	**0,3118**
BIC(↓)	−1000,4642	−1000,4642	**−214,9118**	−965,8197	−1000,4642	−982,7053	−1000,4642	−1003,1322
MPE-DMPF(↓)	6,7972	**6,7608**	84,0226	21,0115	6,7972	17,2334	6,7972	8,9246
PBM-FVG(↑)	8,64E−03	**9,6510E−03**	5,79E−05	4,13E−04	8,64E−03	1,93E−03	8,64E−03	1,23E−02
PS-FVG(↓)	0,0223	0,0211	**0,0081**	0,0188	0,0223	0,015	0,0223	0,0203
XB-FVG(↓)	0,6609	0,5918	16,925	7,4431	0,6609	3,3084	0,6609	**0,5435**
FS-FVG(↓)	69,5885	63,9573	69,8181	139,5073	69,5885	46,0894	69,5885	**62,2748**
K-FVG(↓)	99,0118	88,8539	2533,9562	1107,9378	99,0118	506,2265	99,0118	**82,3202**
SV-FVG(↓)	3,5545	3,5535	235,2749	33,5358	3,5545	32,3814	3,5545	3,369
Eff($\approx D$)	**2,9266**	0,9063	28,0937	33,9072	**2,9266**	53,8906	**2,9266**	1,398
Number of features (D)	16	16	16	16	16	16	16	16
Number of successes	1	2	2	0	1	0	1	**6**
Case 3								
Main Index	ICC	ICC	CV	CV	CVGED	CVGED	CVCOD	CVCOD
GAD or TIGAD	$G^{(2)}$	TIGAD - $GTD^{(1)}$ - Dombi	$G^{(2)}$	TIGAD - $GTD^{(1)}$ - Dombi	$G^{(1)}$	TIGAD - $GTD^{(2)}$ - Yager	$G^{(1)}$	TIGAD - $GTD^{(1)}$ - Yager
MAD	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$	$M^{(1)}$	$M^{(3)}$	$M^{(3)}$	$M^{(1)}$	$M^{(1)}$
Number of clusters	51	12	45	2	31	11	3	2
CoV(↓)	0,11	0,045	0,1014	**0,0241**	0,0805	0,0423	0,0272	**0,0241**
Silhouette(↑)	−0,1562	0,0282	−0,1292	**0,4481**	−0,6199	0,0027	0,0903	**0,4481**
WLI(↓)	2,0205	7,9591	7,7986	**0,8604**	139,6688	4,2154	0,9618	**0,8604**
BIC(↓)	**-17617,9546**	−20927,8231	−19037,8503	−21043,1671	−22537,2685	−20502,0332	−20462,977	−21043,1671
MPE-DMPF(↓)	38,5404	5,2611	29,6898	**0,4112**	5,6189	7,0431	1,3324	**0,4112**
PBM-FVG(↑)	1,01E−05	1,86E−05	3,10E−06	6,82E−05	8,65E−07	3,17E−05	**6,9188E−05**	6,82E−05
PS-FVG(↓)	**0,0007**	0,0036	0,0017	0,006	0,0068	0,0075	0,0044	0,006
XB-FVG(↓)	7,0542	9,3417	30,8922	0,7255	226,2233	34,5214	0,8631	**0,7254**
FS-FVG(↓)	**86,8583**	510,162	328,325	1090,3205	1094,2368	495,7207	894,9828	1090,2567
K-FVG(↓)	35824,0903	47193,635	156188,7383	3664,0761	1142547,467	174429,5038	4359,3204	**3663,8616**
SV-FVG(↓)	1012,8178	115,3614	978,9867	3,6694	1001,2917	206,1802	6,6216	3,6692
Eff($\approx D$)	36,0208	71,2656	268,8662	0,2453	**3,6064**	25,6992	0,563	0,2454
Number of features (D)	10	10	10	10	10	10	10	10
Number of successes	3	0	0	3	1	0	1	**6**

Table 7
Analysis of Experiment 2, using other indices and measures — Part 2. Case 4: Propagation of optical comb lines spectra — ultrashort pulsed laser; and Case 5: Prediction about propagation of optical comb lines spectra (generated by two Mach–Zehnder Modulators).

Case 4								
Main Index	ICC	ICC	CV	CV	CVGED	CVGED	CVCOD	CVCOD
GAD or TIGAD	$G^{(1)}$	TIGAD - $GTD^{(2)}$ - Dombi	$G^{(2)}$	TIGAD - $GTD^{(2)}$ - Yager	$G^{(2)}$	TIGAD - $GTD^{(1)}$ - Yager	$G^{(2)}$	TIGAD - $GTD^{(1)}$ - Yager
MAD	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$
Number of clusters	38	38	43	34	41	34	41	34
CoV(↓)	0,1543	0,1597	0,1683	0,1507	0,1605	**0,1503**	0,1605	**0,1503**
Silhouette(↑)	−0,4019	0,3187	−0,3034	0,2785	−0,3009	**0,3368**	−0,3009	**0,3368**
WLI(↓)	29,9305	10,0025	24,1199	8,5452	**1,5182**	4,5554	**1,5182**	4,5497
BIC(↓)	−3390,082	−2914,6266	−2896,832	−3037,9042	**−2734,7531**	−3041,8631	**−2734,7531**	−3041,5402
MPE-DMPF(↓)	26,1671	22,0137	27,1587	20,8343	25,8669	19,4634	25,8669	**19,4475**
PBM-FVG(↑)	3,97E−06	2,59E−05	4,92E−06	2,30E−05	1,33E−04	1,46E−04	1,33E−04	**1,4613E−04**
PS-FVG(↓)	0,003	0,0016	0,002	0,0022	0,0011	0,0009	0,0011	**0,0008**
XB-FVG(↓)	31,9527	9,3165	19,0225	13,389	5,1497	2,1072	5,1497	**2,0553**
FS-FVG(↓)	547,4955	93,83	363,327	129,2504	**14,7408**	20,1475	**14,7408**	20,1421
K-FVG(↓)	80074,6965	23599,1013	47723,7677	33792,3144	13815,3559	5577,3678	13815,3559	**5439,9725**
SV-FVG(↓)	458,8646	597,5842	491,7076	546,8471	1196,5211	546,847	1196,5211	540,1358
Eff($\approx D$)	542,8727	138,3692	556,7589	114,648	**9,3082**	29,4934	**9,3082**	29,4562
Number of features (D)	6	6	6	6	6	6	6	6
Number of successes	1	0	0	0	4	2	4	**7**
Case 5								
Main Index	ICC	ICC	CV	CV	CVGED	CVGED	CVCOD	CVCOD
GAD or TIGAD	$G^{(2)}$	TIGAD - $GTD^{(2)}$ - DombI	$G^{(2)}$	TIGAD - $GTD^{(2)}$ - Yager	$G^{(1)}$	TIGAD - $GTD^{(2)}$ - Yager	$G^{(2)}$	TIGAD - $GTD^{(1)}$ - Dombi
MAD	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$	$M^{(1)}$	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$	$M^{(3)}$
Number of clusters	10	7	13	4	12	8	12	7
CoV(↓)	0,1223	0,1028	0,1397	**0,0828**	0,1333	0,1115	0,1333	0,1029
Silhouette(↑)	0,2708	0,0331	0,1456	−0,0571	**0,3012**	−0,0948	**0,3012**	0,1706
WLI(↓)	3,0754	0,5599	2,1298	0,6039	**0,2395**	0,2596	0,2957	0,3909
BIC(↓)	−6429,6087	−6430,4895	−6429,6087	−7457,478	**−6283,2594**	−6529,1019	**−6283,2594**	−6436,6898
MPE-DMPF(↓)	15,2492	17,073	28,8541	**12,0846**	27,8291	17,7169	27,8264	16,6047
PBM-FVG(↑)	2,35E−04	3,99E−03	7,13E−04	2,22E−03	**5,4475E−03**	4,17E−03	5,41E−03	3,59E−03
PS-FVG(↓)	0,0054	2,90E−03	3,50E−03	4,00E−03	**0,0016**	2,70E−03	**0,0016**	3,80E−03
XB-FVG(↓)	2,9459	0,4161	2,056	0,2573	**0,2483**	0,4582	0,2505	0,702
FS-FVG(↓)	215,9366	55,6788	91,1251	278,6417	**14,0034**	40,7433	14,091	61,9548
K-FVG(↓)	2964,9666	438,1101	2104,8274	**260,7256**	397,1947	492,1107	399,3571	736,2404
SV-FVG(↓)	36,9332	19,1327	61,5059	3,848	59,6275	26,7837	59,6275	23,5485
Eff($\approx D$)	7,0812	5,7607	**13,1353**	0,1777	3,3125	4,3139	3,3391	6,2448
Number of features (D)	19	19	19	19	19	19	19	19
Number of successes	0	0	2	4	**7**	0	3	0

Table 8
Number of successes for CV, ICC, CVGED, and CVCOD indices — Experiment 2.

Main Index	CoV(↓)	Silhouette(↑)	WLI(↓)	BIC(↓)	MPE-DMPF(↓)	PBM-FVG(↑)	PS-FVG(↓)	XB-FVG(↓)	FS-FVG(↓)	K-FVG(↓)	SV-FVG(↓)	Eff($\approx D$)	
	Successes	Successes	Successes	Successes	Successes	Successes	Successes	Successes	Successes	Successes	Successes	Successes	Total of successes
ICC	1	2	0	1	2	2	1	1	1	0	2	2	15
CV	2	1	1	1	2	0	1	0	0	1	1	1	11
CVGED	1	2	2	3	0	1	1	1	2	0	0	3	16
CVCOD	3	4	4	3	2	2	3	3	3	4	2	2	**35**

In the experiment 1, the CVCOD index slightly increases the clustering stability with respect to the rest of indices. This advantage is presented by the ratio of compactness between the data covariance matrix for data and the fuzzy covariance matrix, which allows a better calculating of global compactness. However, it is important to consider the data structure and the granulation error of the CVGED index in order to increase even further the clustering stability through the CVCOD index.

When the human expert and/or the data information are available, the clustering results may close to the human perception. The above is observed by comparing the general analysis for experiment 2 and the analysis with other indices and metrics. Cases 1, 3, 4 and 5 generated the closest results with respect to human's expert which represents 80% of databases for experiment 2. However, the case 2 found a number of clusters far away of the human perception due to that the best data partition is reached for 3 clusters and the human expert determined 10 or 12 clusters. Therefore, the performance of the CVGED and CVCOD indices depend on the capacity of the LAMDA clustering algorithm to generate a data partition where the compactness and separation among clusters are optimal. One likely solution is to add a knowledge-based guidance in the LAMDA clustering

algorithm in order to include the needs of the human expert, taking advantage of the capacity of the TIGAD function to improve the quality of the diffuse data partition.

As a further work, the definition of an optimization model based on the ICC, CV, CVGED, and CVCOD indices could be increase the capacity of the LAMDA algorithm to generate the best data partition for different database. It is important to direct this idea for Big Data problems where we consider the importance of collaborative clustering and deep clustering as a key method.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Javier Fernando Botía Valderrama: Conceptualization, Methodology, Formal analysis, Writing - original draft, Project administration, Validation, Supervision. **Diego José Luis Botía Valderrama:** Software, Investigation, Writing - review & editing, Validation, Visualization, Resources.

Acknowledgments

We acknowledge to "Alcaldía Municipal de Medellín - Información y Evaluación Estratégica" and open data supplied by "Ministerio de las TICs", "Secretaría de Educación de Antioquia", "Departamento Administrativo de Planeación - Subdirección de Información y Evaluación Estratégica " and "MEData", for allowing the access and the use of the historical data about the population projections 1995–2005–2015 and 2016–2020 of Medellín city, multidimensional index quality of life survey of Medellín city, and Educational Establishments of the Antioquia state (Colombia). We thanks to the Ultrafast Optics and Optical Fiber Communications Laboratory at the Purdue University, West Lafayette, Indiana, USA, for hosting one of the authors and allowing them access to experimental equipment. Besides, we thank the University of Antioquia for its help in the use of computer equipment for the project.

Appendix

A.1. Numerical example of the LAMDA algorithm

Consider the following historical data matrix:

$$X = \begin{bmatrix} 4 & 5 & 6 & 11 \\ 1 & 7 & 2 & 20 \\ 9 & 8 & 3 & 15 \\ 9 & 8 & 2 & 15 \\ 6 & 5 & 1 & 17 \\ 8 & 19 & 22 & 5 \\ 7 & 20 & 2 & 8 \end{bmatrix}$$

This matrix contains 7 samples and 4 features. By applying the first step, the normalized data matrix is shown below:

$$\widehat{X} = \begin{bmatrix} 0.3750 & 0 & 0.2381 & 0.4 \\ 0 & 0.1333 & 0.0476 & 1 \\ 1 & 0.2 & 0.0952 & 0.6667 \\ 1 & 0.2 & 0.0476 & 0.6667 \\ 0.625 & 0 & 0 & 0.8 \\ 0.875 & 0.9333 & 1 & 0 \\ 0.75 & 1 & 0.0476 & 0.2 \end{bmatrix}$$

Considering the second and third steps, Eqs. (2) and (12) are selected to calculate the MAD and GAD values. Initially, the LAMDA algorithm considers a adequacy vector for NIC, $\rho(NIC) = \{0.5, 0.5, 0.5, 0.5\}$ (size 1×4 features), and therefore, $MAD(NIC) = [0.5, 0.5, 0.5, 0.5]$. If the user selects $\epsilon = 0.7$, then:

$$GAD(NIC) = 0.7 \cdot \min\{0.5, 0.5, 0.5, 0.5\} + (1 - 0.7) \cdot \max\{0.5, 0.5, 0.5, 0.5\} = 0.5$$

Selecting the first sample of $\widehat{X}$, i.e. $\{0.3750, 0, 0.2381, 0.4\}$, Eq. (7) is applied to update the adequacy values. Then, the adequacy values are organized as the following matrix:

$$\rho = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.4375 & 0.25 & 0.369 & 0.45 \end{bmatrix}$$

The first row of ρ is the NIC class and the second row of ρ is the ρ values of class $c = 1$. Now, the LAMDA algorithm selects the second sample of $\widehat{X}$, i.e. $\{0, 0.1333, 0.0476, 1\}$ and Eq. (2) is applied again. The results of MAD for the second sample are: $\{0.5625, 0.6478, 0.615, 0.45\}$ and therefore:

$$GAD_1 = 0.7 \cdot \min\{0.5625, 0.6478, 0.615, 0.45\} + (1 - 0.7) \cdot \max\{0.5625, 0.6478, 0.615, 0.45\} = 0.5093$$

From Eq. (18), $\max GAD = max\{0.5, 0.5093\} = 0.5093$, then $GAD_1 > GAD(NIC)$ and a new class is created, $c = 2$, and therefore, Eq. (5) is applied, generated a new updating of ρ:

$$\rho = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.4375 & 0.25 & 0.369 & 0.45 \\ 0.2917 & 0.2111 & 0.2619 & 0.6333 \end{bmatrix}$$

This procedure is carried out with the rest of samples of $\widehat{X}$. Results are shown below:

$$MAD = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.3641 & 0.211 & 0.7025 & 0.409 \\ 0.5683 & 0.225 & 0.775 & 0.389 \\ 0.5645 & 0.7166 & 0.2634 & 0.6020 \end{bmatrix}$$

$$GAD = \begin{bmatrix} 0.5 \\ 0.3585 \\ 0.3902 \\ 0.3994 \end{bmatrix}$$

$$\rho = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.2916 & 0.2111 & 0.2619 & 0.6333 \\ 0.7812 & 0.225 & 0.1607 & 0.6583 \\ 0.6875 & 0.7166 & 0.75 & 0.25 \\ 0.625 & 0.75 & 0.2738 & 0.35 \end{bmatrix}$$

The MAD matrix has a size of 4 classes per 4 features and the GAD vector has a size of 1×4 classes. To obtain the GAD matrix, the first row of the ρ matrix is ignored due to the GAD vector contains 4 classes and Eq. (2) is applied again through the ρ matrix. In this case, the mathematical operations for $n = 1$ and $n = 2$ are shown below:

for $n = 1$ from matrix $\widehat{X}$, i.e $\{0.3750, 0, 0.2381, 0.4\}$*:*

$$M^{(1)}_{n=1,d=1,c=1} = 0.2916^{0.3750}(1-0.2916)^{1-0.3750} = 0.578$$
$$M^{(1)}_{n=1,d=2,c=1} = 0.2111^{0}(1-0.2111)^{1-0} = 0.7889$$
$$M^{(1)}_{n=1,d=3,c=1} = 0.2619^{0.2381}(1-0.2619)^{1-0.2381} = 0.5767$$
$$M^{(1)}_{n=1,d=4,c=1} = 0.6333^{0.4}(1-0.6333)^{1-0.4} = 0.4563$$

$$M^{(1)}_{n=1,d=1,c=2} = 0.7812^{0.3750}(1-0.7812)^{1-0.3750} = 0.3526$$
$$M^{(1)}_{n=1,d=2,c=2} = 0.225^{0}(1-0.225)^{1-0} = 0.7750$$
$$M^{(1)}_{n=1,d=3,c=2} = 0.1607^{0.2381}(1-0.1607)^{1-0.2381} = 0.5662$$
$$M^{(1)}_{n=1,d=4,c=2} = 0.6333^{0.4}(1-0.6333)^{1-0.4} = 0.4442$$

$$M^{(1)}_{n=1,d=1,c=3} = 0.6875^{0.3750}(1-0.6875)^{1-0.3750} = 0.42$$
$$M^{(1)}_{n=1,d=2,c=3} = 0.7166^{0}(1-0.7166)^{1-0} = 0.2833$$
$$M^{(1)}_{n=1,d=3,c=3} = 0.75^{0.2381}(1-0.75)^{1-0.2381} = 0.3247$$
$$M^{(1)}_{n=1,d=4,c=3} = 0.25^{0.4}(1-0.25)^{1-0.4} = 0.4833$$

$$M^{(1)}_{n=1,d=1,c=4} = 0.625^{0.3750}(1-0.625)^{1-0.3750} = 0.4542$$
$$M^{(1)}_{n=1,d=2,c=4} = 0.75^{0}(1-0.75)^{1-0} = 0.25$$
$$M^{(1)}_{n=1,d=3,c=4} = 0.2738^{0.2381}(1-0.2738)^{1-0.2381} = 0.5757$$
$$M^{(1)}_{n=1,d=4,c=4} = 0.35^{0.4}(1-0.35)^{1-0.4} = 0.5074$$

for $n = 2$ from matrix $\widehat{X}$, i.e $\{0, 0.1333, 0.0476, 1\}$*:*

$$M^{(1)}_{n=2,d=1,c=1} = 0.2916^{0}(1-0.2916)^{1-0} = 0.7083$$
$$M^{(1)}_{n=2,d=2,c=1} = 0.2111^{0.1333}(1-0.2111)^{1-0.1333} = 0.6617$$
$$M^{(1)}_{n=2,d=3,c=1} = 0.2619^{0.0476}(1-0.2619)^{1-0.0476} = 0.7026$$
$$M^{(1)}_{n=2,d=4,c=1} = 0.6333^{1}(1-0.6333)^{1-1} = 0.6333$$

$$M^{(1)}_{n=2,d=1,c=2} = 0.7812^{0}(1-0.7812)^{1-0} = 0.2188$$
$$M^{(1)}_{n=2,d=2,c=2} = 0.225^{0.1333}(1-0.225)^{1-0.1333} = 0.6572$$
$$M^{(1)}_{n=2,d=3,c=2} = 0.1607^{0.0476}(1-0.1607)^{1-0.0476} = 0.7758$$
$$M^{(1)}_{n=2,d=4,c=2} = 0.6333^{1}(1-0.6333)^{1-1} = 0.6333$$

$$M^{(1)}_{n=2,d=1,c=3} = 0.6875^{0}(1-0.6875)^{1-0} = 0.3125$$
$$M^{(1)}_{n=2,d=2,c=3} = 0.7166^{0.1333}(1-0.7166)^{1-0.1333} = 0.3207$$
$$M^{(1)}_{n=2,d=3,c=3} = 0.75^{0.0476}(1-0.75)^{1-0.0476} = 0.2634$$
$$M^{(1)}_{n=2,d=4,c=3} = 0.25^{1}(1-0.25)^{1-1} = 0.25$$

$$M^{(1)}_{n=2,d=1,c=4} = 0.625^{0}(1-0.625)^{1-0} = 0.375$$
$$M^{(1)}_{n=2,d=2,c=4} = 0.75^{0.1333}(1-0.75)^{1-0.1333} = 0.2894$$
$$M^{(1)}_{n=2,d=3,c=4} = 0.2738^{0.0476}(1-0.2738)^{1-0.0476} = 0.6932$$
$$M^{(1)}_{n=2,d=4,c=4} = 0.35^{1}(1-0.35)^{1-1} = 0.35$$

The above procedure is applied for the rest of samples n. The MAD values can be organized in several matrices:

$$MAD(n=1) = \begin{bmatrix} 0.5078 & 0.7889 & 0.5767 & 0.4563 \\ 0.3526 & 0.7750 & 0.5662 & 0.4442 \\ 0.42 & 0.2833 & 0.3247 & 0.4833 \\ 0.4542 & 0.25 & 0.5757 & 0.5074 \end{bmatrix}$$

$$MAD(n=2) = \begin{bmatrix} 0.7083 & 0.6617 & 0.7026 & 0.6333 \\ 0.2188 & 0.6572 & 0.7758 & 0.6333 \\ 0.3125 & 0.3207 & 0.2634 & 0.25 \\ 0.375 & 0.2894 & 0.6932 & 0.35 \end{bmatrix}$$

$$MAD(n=3) = \begin{bmatrix} 0.2917 & 0.6061 & 0.6687 & 0.5279 \\ 0.7813 & 0.6052 & 0.7170 & 0.5291 \\ 0.6875 & 0.3411 & 0.2776 & 0.3606 \\ 0.6250 & 0.3114 & 0.6618 & 0.4302 \end{bmatrix}$$

$$MAD(n=4) = \begin{bmatrix} 0.2917 & 0.6061 & 0.7026 & 0.5279 \\ 0.7813 & 0.6052 & 0.7758 & 0.5291 \\ 0.6875 & 0.3411 & 0.2634 & 0.3606 \\ 0.6250 & 0.3114 & 0.6932 & 0.4302 \end{bmatrix}$$

$$MAD(n=5) = \begin{bmatrix} 0.4068 & 0.7889 & 0.7381 & 0.5678 \\ 0.4847 & 0.7750 & 0.8393 & 0.5774 \\ 0.5115 & 0.2833 & 0.25 & 0.3114 \\ 0.5160 & 0.25 & 0.7262 & 0.3961 \end{bmatrix}$$

$$MAD(n=6) = \begin{bmatrix} 0.3259 & 0.2305 & 0.2619 & 0.3667 \\ 0.6663 & 0.2443 & 0.1607 & 0.3417 \\ 0.6230 & 0.6737 & 0.75 & 0.75 \\ 0.5863 & 0.6970 & 0.2738 & 0.65 \end{bmatrix}$$

$$MAD(n=7) = \begin{bmatrix} 0.3641 & 0.2111 & 0.7026 & 0.4090 \\ 0.5683 & 0.2250 & 0.7758 & 0.3896 \\ 0.5645 & 0.7167 & 0.2634 & 0.6021 \\ 0.5501 & 0.75 & 0.6932 & 0.5743 \end{bmatrix}$$

To calculate the GAD matrix, Eq. (12) is applied for each MAD matrix with $\epsilon = 0.7$:

$MAD(n = 1)$:

$$GAD_{n=1,c=1} = 0.7 \cdot \min\{0.5078, 0.7889, 0.5767, 0.4563\}$$
$$+(1-0.7) \cdot \max\{0.5078, 0.7889, 0.5767, 0.4563\} = 0.5561$$
$$GAD_{n=1,c=2} = 0.7 \cdot \min\{0.3526, 0.7750, 0.5662, 0.4442\}$$
$$+(1-0.7) \cdot \max\{0.3526, 0.7750, 0.5662, 0.4442\} = 0.4793$$
$$GAD_{n=1,c=3} = 0.7 \cdot \min\{0.42, 0.2833, 0.3247, 0.4833\}$$
$$+(1-0.7) \cdot \max\{0.42, 0.2833, 0.3247, 0.4833\} = 0.3433$$
$$GAD_{n=1,c=4} = 0.7 \cdot \min\{0.4542, 0.25, 0.5757, 0.5074\}$$
$$+(1-0.7) \cdot \max\{0.4542, 0.25, 0.5757, 0.5074\} = 0.3477$$

$MAD(n = 2)$:

$$GAD_{n=2,c=1} = 0.7 \cdot \min\{0.7083, 0.6617, 0.7026, 0.6333\}$$
$$+(1-0.7) \cdot \max\{0.7083, 0.6617, 0.7026, 0.6333\} = 0.6558$$
$$GAD_{n=2,c=2} = 0.7 \cdot \min\{0.2188, 0.6572, 0.7758, 0.6333\}$$
$$+(1-0.7) \cdot \max\{0.2188, 0.6572, 0.7758, 0.6333\} = 0.3859$$
$$GAD_{n=2,c=3} = 0.7 \cdot \min\{0.3125, 0.3207, 0.2634, 0.25\}$$
$$+(1-0.7) \cdot \max\{0.3125, 0.3207, 0.2634, 0.25\} = 0.2712$$
$$GAD_{n=2,c=4} = 0.7 \cdot \min\{0.375, 0.2894, 0.6932, 0.35\}$$
$$+(1-0.7) \cdot \max\{0.375, 0.2894, 0.6932, 0.35\} = 0.4106$$

$MAD(n = 3)$:

$$GAD_{n=3,c=1} = 0.7 \cdot \min\{0.2917, 0.6061, 0.6687, 0.5279\}$$
$$+(1-0.7) \cdot \max\{0.2917, 0.6061, 0.6687, 0.5279\} = 0.4048$$
$$GAD_{n=3,c=2} = 0.7 \cdot \min\{0.7813, 0.6052, 0.7170, 0.5291\}$$
$$+(1-0.7) \cdot \max\{0.7813, 0.6052, 0.7170, 0.5291\} = 0.6047$$
$$GAD_{n=3,c=3} = 0.7 \cdot \min\{0.6875, 0.3411, 0.2776, 0.3606\}$$
$$+(1-0.7) \cdot \max\{0.6875, 0.3411, 0.2776, 0.3606\} = 0.4$$
$$GAD_{n=3,c=4} = 0.7 \cdot \min\{0.6250, 0.3114, 0.6618, 0.4302\}$$
$$+(1-0.7) \cdot \max\{0.6250, 0.3114, 0.6618, 0.4302\} = 0.4165$$

$MAD(n=4)$:

$$GAD_{n=4,c=1} = 0.7 \cdot \min\{0.2917, 0.6061, 0.7026, 0.5279\} + (1-0.7) \cdot \max\{0.2917, 0.6061, 0.7026, 0.5279\} = 0.4149$$
$$GAD_{n=4,c=2} = 0.7 \cdot \min\{0.7813, 0.6052, 0.7758, 0.5291\} + (1-0.7) \cdot \max\{0.7813, 0.6052, 0.7758, 0.5291\} = 0.6047$$
$$GAD_{n=4,c=3} = 0.7 \cdot \min\{0.6875, 0.3411, 0.2634, 0.3606\} + (1-0.7) \cdot \max\{0.6875, 0.3411, 0.2634, 0.3606\} = 0.3906$$
$$GAD_{n=4,c=4} = 0.7 \cdot \min\{0.625, 0.3114, 0.6932, 0.4302\} + (1-0.7) \cdot \max\{0.625, 0.3114, 0.6932, 0.4302\} = 0.4260$$

$MAD(n=5)$:

$$GAD_{n=5,c=1} = 0.7 \cdot \min\{0.4068, 0.7889, 0.7381, 0.5678\} + (1-0.7) \cdot \max\{0.4068, 0.7889, 0.7381, 0.5678\} = 0.5214$$
$$GAD_{n=5,c=2} = 0.7 \cdot \min\{0.4847, 0.7750, 0.8393, 0.5774\} + (1-0.7) \cdot \max\{0.4847, 0.7750, 0.8393, 0.5774\} = 0.5911$$
$$GAD_{n=5,c=3} = 0.7 \cdot \min\{0.5115, 0.2833, 0.25, 0.3114\} + (1-0.7) \cdot \max\{0.5115, 0.2833, 0.25, 0.3114\} = 0.3285$$
$$GAD_{n=5,c=4} = 0.7 \cdot \min\{0.5160, 0.25, 0.7262, 0.3961\} + (1-0.7) \cdot \max\{0.5160, 0.25, 0.7262, 0.3961\} = 0.3929$$

$MAD(n=6)$:

$$GAD_{n=6,c=1} = 0.7 \cdot \min\{0.3259, 0.2305, 0.2619, 0.3667\} + (1-0.7) \cdot \max\{0.3259, 0.2305, 0.2619, 0.3667\} = 0.2714$$
$$GAD_{n=6,c=2} = 0.7 \cdot \min\{0.6663, 0.2443, 0.1607, 0.3417\} + (1-0.7) \cdot \max\{0.6663, 0.2443, 0.1607, 0.3417\} = 0.3124$$
$$GAD_{n=6,c=3} = 0.7 \cdot \min\{0.6230, 0.6737, 0.75, 0.75\} + (1-0.7) \cdot \max\{0.6230, 0.6737, 0.75, 0.75\} = 0.6611$$
$$GAD_{n=6,c=4} = 0.7 \cdot \min\{0.5863, 0.6970, 0.2738, 0.65\} + (1-0.7) \cdot \max\{0.5863, 0.6970, 0.2738, 0.65\} = 0.4$$

$MAD(n=7)$:

$$GAD_{n=7,c=1} = 0.7 \cdot \min\{0.3641, 0.2111, 0.7026, 0.4090\} + (1-0.7) \cdot \max\{0.3641, 0.2111, 0.7026, 0.4090\} = 0.3585$$
$$GAD_{n=7,c=2} = 0.7 \cdot \min\{0.5683, 0.2250, 0.7758, 0.3896\} + (1-0.7) \cdot \max\{0.5683, 0.2250, 0.7758, 0.3896\} = 0.3902$$
$$GAD_{n=7,c=3} = 0.7 \cdot \min\{0.5645, 0.7167, 0.2634, 0.6021\} + (1-0.7) \cdot \max\{0.5645, 0.7167, 0.2634, 0.6021\} = 0.3994$$
$$GAD_{n=7,c=4} = 0.7 \cdot \min\{0.5501, 0.75, 0.6932, 0.5743\} + (1-0.7) \cdot \max\{0.5501, 0.75, 0.6932, 0.5743\} = 0.61$$

Finally, all GAD values are organized in the following matrix:

$$GAD = \begin{bmatrix} 0.5561 & 0.4793 & 0.3433 & 0.3477 \\ 0.6558 & 0.3859 & 0.2712 & 0.4106 \\ 0.4048 & 0.6047 & 0.4 & 0.4165 \\ 0.4149 & 0.6047 & 0.3906 & 0.4260 \\ 0.5214 & 0.5911 & 0.3285 & 0.3929 \\ 0.2714 & 0.3124 & 0.6611 & 0.4 \\ 0.3585 & 0.3902 & 0.3994 & 0.61 \end{bmatrix}$$

If the maximum GAD is calculated for each sample n, the classes vector is obtained:

- $maxGAD(n=1) = \max\{0.5561, 0.4793, 0.3433, 0.3477\} = 0.5561$ which it is located in $c=1$.
- $maxGAD(n=2) = \max\{0.6558, 0.3859, 0.2712, 0.4106\} = 0.6558$ which it is located in $c=1$.
- $maxGAD(n=3) = \max\{0.4048, 0.6047, 0.4, 0.4165\} = 0.6047$ which it is located in $c=2$.
- $maxGAD(n=4) = \max\{0.4149, 0.6047, 0.3906, 0.4260\} = 0.6047$ which it is located in $c=2$.
- $maxGAD(n=5) = \max\{0.5214, 0.5911, 0.3285, 0.3929\} = 0.5911$ which it is located in $c=2$.
- $maxGAD(n=6) = \max\{0.2714, 0.3124, 0.6611, 0.4\} = 0.6611$ which it is located in $c=3$.
- $maxGAD(n=7) = \max\{0.3585, 0.3902, 0.3994, 0.61\} = 0.61$ which it is located in $c=4$.

Therefore, the classes vector is {1, 1, 2, 2, 2, 3, 4}.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2020.106102.

References

[1] B. Ratner, Statistical and Machine-Learning Data Mining: Techniques for Better Predictive Modeling and Analysis of Big Data, second ed., CRC Press, Inc., Boca Raton, FL, USA, 2011.

[2] M.-J. Lesot, M. Rifqi, B. Bouchon-Meunier, Fuzzy prototypes: From a cognitive view to a machine learning principle, in: H. Bustince, F. Herrera, J. Montero (Eds.), Fuzzy Sets and their Extensions: Representation, Aggregation and Models, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 431–452.

[3] A.K. Jain, Data clustering: 50 years beyond k-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.

[4] J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy C-means clustering algorithm, Comput. Geosci. 10 (1984) 191–203.

[5] D.E. Gustafson, W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in: 1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes, 1978, pp. 761–766.

[6] H. Huang, Y. Chuang, C. Chen, Multiple kernel fuzzy clustering, IEEE Trans. Fuzzy Syst. 20 (1) (2012) 120–134.

[7] P.H. Thong, L.H. Son, Picture fuzzy clustering: a new computational intelligence method, Soft Comput. 20 (9) (2016) 3549–3562.

[8] J.A. Martín, R.L. de Mantaras, Controlling selectivity in nonstandard pattern recognition algorithms, the process of classification and learning the meaning of linguistic descriptors of concepts, 1982, pp. 165–175.

[9] T. Kempowsky, A. Subias, J.A. Martín, Process situation assessment: From a fuzzy partition to a finite state machine, Eng. Appl. Artif. Intell. 19 (5) (2006) 461–477.

[10] B. Lamrini, M.-V.L. Lann, A. Benhammou, E.K. Lakhal, Detection of functional states by the 'LAMDA' classification technique: application to a coagulation process in drinking water treatment, C. R. Phys. 6 (10) (2005) 1161–1168.

[11] C.V. Isaza, H.O. Sarmiento, T. Kempowsky-Hamon, M.-V. LeLann, Situation prediction based on fuzzy clustering for industrial complex processes, Inform. Sci. 279 (2014) 785–804.

[12] F.A. Ruiz, C.V. Isaza, A.F. Agudelo, J.R. Agudelo, A new criterion to validate and improve the classification process of LAMDA algorithm applied to diesel engines, Eng. Appl. Artif. Intell. 60 (2017) 117–127.

[13] J.G. Zambrano, E. Guzmán-Ramírez, O. Pogrebnyak, Search algorithm for image recognition based on learning algorithm for multivariate data analysis, in: T. Abrão (Ed.), Search Algorithms for Engineering Optimization, 2013, pp. 1–22.

[14] A. Doncescu, J. Aguilar-Martin, J.-C. Atine, Image color segmentation using the fuzzy tree algorithm T-LAMDA, Fuzzy Sets and Systems 158 (3) (2007) 230–238.

[15] J.F. Botía, Methodology for Predicting the Behavior of Optical Frequency Comb, LAP Lambert Academic Publishing, Düsseldorf, Germany, 2017.

[16] J.F. Botía, A.M. Cárdenas, C.M. Sierra, Fuzzy cellular automata and intuitionistic fuzzy sets applied to an optical frequency comb spectral shape, Eng. Appl. Artif. Intell. 62 (2017) 181–194.

[17] T. Kempowsky-Hamon, C. Valle, M. Lacroix-Triki, L. Hedjazi, L. Trouilh, S. Lamarre, D. Labourdette, L. Roger, L. Mhamdi, F. Dalenc, T. Filleron, G. Favre, J.-M.F. cois, M.-V. Le-Lann, V. Anton-Leberre, Fuzzy logic selection as a new reliable tool to identify molecular grade signatures in breast cancer – the INNODIAG study, BMC Med. Genom. 8 (3) (2015) 1–17.

[18] A. Doncescu, S. Regis, N. Kabbaj, Reinforced operators in fuzzy clustering systems, in: F. Xhafa, L. Barolli, P.J. Papajorgji (Eds.), Complex Intelligent Systems and their Applications, Springer New York, New York, NY, 2010, pp. 247–266.

[19] R. Emilion, S. Regis, A. Doncescu, A general version of the triple Π operator, Int. J. Intell. Syst. 28 (10) (2013) 957–973.

[20] C. Bedoya, J. Waissman Villanova, C.V. Isaza Narvaez, Yager–rybalov triple $I\mathcal{T}$ operator as a means of reducing the number of generated clusters in unsupervised anuran vocalization recognition, in: A. Gelbukh, F.C. Espinoza, S.N. Galicia-Haro (Eds.), Nature-Inspired Computation and Machine Learning: 13th Mexican Int. Conf. on Artificial Intelligence, MICAI 2014, Tuxtla GutiÉRrez, Mexico, November 16-22, 2014. Proceedings, Part II, Springer International Publishing, 2014, pp. 382–391.

[21] J.F.B. Valderrama, D.J.L.B. Valderrama, On LAMDA clustering method based on typicality degree and intuitionistic fuzzy sets, Expert Syst. Appl. 107 (2018) 196–221.

[22] L. Hedjazi, J. Aguilar-Martín, M.-V. Le-Lann, Similarity-margin based feature selection for symbolic interval data, Pattern Recognit. Lett. 32 (4) (2011) 578–585.

[23] L. Hedjazi, J. Aguilar-Martín, M.-V. Le-Lann, T. Kempowsky-Hamon, Membership-margin based feature selection for mixed type and high-dimensional data: Theory and applications, Inform. Sci. 322 (2015) 174–196.

[24] T. Monrousseau, L. Travé-Massuyés, M.-V.L. Lann, Processing measure uncertainty into fuzzy classifier, in: 26th International Workshop on Principles of Diagnosis, 2015, pp. 33–38.

[25] W. Wang, Y. Zhang, On fuzzy cluster validity indices, Fuzzy Sets and Systems 158 (19) (2007) 2095–2117.

[26] C. Isaza, Diagnostic Par Techniques D'Apprentissage Floues: Conception D'Méthode de Validation et D'Optimisation dans Partitions (Ph.D. thesis), LAAS/CNRS, Toulouse, France, 2007.

[27] C. Rita de Franco, L. Silva Vidal, A.J. de Oliveira Cruz, A validity measure for hard and fuzzy clustering derived from Fisher's linear discriminant, in: Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on, Vol. 2, 2002, pp. 1493–1498.

[28] J. Aguilar-Martin, Classification validity index, in: L. Magdalena, J.L. Verdegay, F. Esteva (Eds.), Enric Trillas: A Passion for Fuzzy Sets: A Collection of Recent Works on Fuzzy Logic, Springer International Publishing, Cham, 2015, pp. 261–267.

[29] C. Isaza, H. Diez-Lledo, H. Hernández de León, J. Aguilar-Martín, M.-V. Le Lann, Decision method for functional states validation in a drinking water plant, in: 10th International Symposium on Computer Applications in Biotechnology, 2007, pp. 359–364.

[30] C. Isaza, M. Lann, J. Aguilar, Diagnosis of chemical processes by fuzzy clustering methods: New optimization method of partitions, in: 18 th European Symposium on Computer Aided Process Engineering (ESCAPE 10), 2008, pp. 1–6.

[31] C. Uribe, C. Isaza, Unsupervised feature selection based on fuzzy partition optimization for industrial processes monitoring, in: 2011 IEEE Int. Conf. on Computational Intelligence for Measurement Systems and Applications (CIMSA) Proc., 2011, pp. 1–5.

[32] E. Roux, L. Travé-Massuyès, M.V. Le Lann, Applied multi-layer clustering to the diagnosis of complex agro-systems, in: DX@ Safeprocess, 2015, pp. 19–26.

[33] J.F. Botía, C. Isaza, T. Kempowsky, M.V.L. Lann, J. Aguilar-Martín, Automaton based on fuzzy clustering methods for monitoring industrial processes, Eng. Appl. Artif. Intell. 26 (4) (2013) 1211–1220.

[34] J.F. Botía, Metodología Para Establecer Las Conexiones Automáticas Entre Estados Funcionales a Partir De Agrupamiento Difuso (Master's thesis), University of Antioquia, Medellín, Colombia, 2011.

[35] L. Morales, C.A. Ouedraogo, J. Aguilar, C. Chassot, S. Medjiah, K. Drira, Experimental comparison of the diagnostic capabilities of classification and clustering algorithms for the QoS management in an autonomic IoT platform, Serv. Orient. Comput. Appl. (2019).

[36] J.M.T. Garfias, J.L. Flores, A.O. Molina, M.C.J.L.B. Avalos, A new tool for merging the information based on clustering methods, in: 2011 IEEE Electronics, Robotics and Automotive Mechanics Conf., 2011, pp. 155–160.

[37] T. Kempowsky, Surveillance de Procèdes à Based e Méthodes de Classification: Conception d'un Outil d'Aide Pour la Détection et le Diagnostic des Défaillances (Ph.D. thesis), LAAS/CNRS, Toulouse, France, 2004.

[38] N. Piera, J. Aguilar-Martín, Controlling selectivity in nonstandard pattern recognition algorithms, IEEE Trans. Syst. Man Cybern. 21 (1) (1991) 71–82.

[39] H.-J. Zimmermann, P. Zysno, Latent connectives in human decision making, Fuzzy Sets and Systems 4 (1) (1980) 37–51.

[40] R.R. Yager, A. Rybalov, Full reinforcement operators in aggregation techniques, IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 28 (6) (1998) 757–769.

[41] T. Chaira, A novel intuitionistic fuzzy c means clustering algorithm and its application to medical images, Appl. Soft Comput. 11 (2) (2011) 1711–1717, The Impact of Soft Computing for the Progress of Artificial Intelligence.

[42] E. Diez-Lledo, J. Aguilar-Martín, Proposition of NON-probabilistic entropy as reliability index for decision making, in: Artificial Intelligence Research and Development, Proc. of the 9th Int. Conf. of the ACIA, CCIA 2006, 2006, pp. 137–144.

[43] B. Kosko, Fuzzy entropy and conditioning, Inform. Sci. 40 (2) (1986) 165–174.

[44] C. Wu, C. Ouyang, L. Chen, L. Lu, A new fuzzy clustering validity index with a median factor for centroid-based clustering, IEEE Trans. Fuzzy Syst. 23 (3) (2015) 701–718.

[45] A. Starczewski, A new validity index for crisp clusters, Pattern Anal. Appl. 20 (3) (2017) 687–700.

[46] S. Bandyopadhyay, S. Saha, W. Pedrycz, Use of a fuzzy granulation – degranulation criterion for assessing cluster validity, Fuzzy Sets and Systems 170 (1) (2011) 22–42, Theme: Information processing.

[47] R. Babuka, P.J. van der Veen, U. Kaymak, Improved covariance estimation for Gustafson-Kessel clustering, in: 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE Int. Conf. on Fuzzy Systems. FUZZ-IEEE'02. Proc. (Cat. No.02CH37291), volume 2, 2002, pp. 1081–1085, vol.2.

[48] M. Bouguessa, S.-R. Wang, A new efficient validity index for fuzzy clustering, in: Proc. of 2004 Int. Conf. on Machine Learning and Cybernetics (IEEE Cat. No.04EX826), Vol. 3, 2004, pp. 1914–1919.

[49] L. Vendramin, Estudo e Desenvolvimiento de Algoritmos Para Agrupamiento Fuzzy de Dados em Cénarios Centralizados e Distribuídos (Master's thesis), Universidad de São Paulo (USP), São Carlos, Brasil, 2012.

[50] X. Liu, Q. Yang, L. He, A novel DBSCAN with entropy and probability for mixed data, Cluster Comput. 20 (2) (2017) 1313–1323.

[51] Z. Xu, X. Luo, L. Mei, C. Hu, Measuring the semantic discrimination capability of association relations, Concurr. Comput.: Pract. Exper. 26 (2) (2014) 380–395.

[52] K.C. Gull, A.B. Angadi, A methodical study on behavior of different seeds using an iterative technique with evaluation of cluster validity, in: A.K. Saini, R.K. Vyas (Eds.), ICT Based Innovations, Springer Singapore, 2018, pp. 63–74.

[53] K. Zhou, S. Yang, Exploring the uniform effect of FCM clustering: A data distribution perspective, Knowl.-Based Syst. 96 (2016) 76–83.

[54] A. Medellín, Proyecciones de Población del Municipio de Medellín-1993-2005-2015, 2013, https://www.datos.gov.co/Inclusi-n-Social-y-Reconciliaci-n/Proyecciones-de-Poblaci-n-del-Municipio-de-Medell-/7nin-7s9a.

[55] A. Medellín, Proyecciones de Población Medellín 2016–2020, 2016, https://www.datos.gov.co/Estad-sticas-frdNacionales/Proyecciones-De-Poblaci-n-Medell-n-2016-2020/imj6-7tfq.

[56] D.A. de Planeación Subdirección de Información y Evaluación Estratégica, Índice Multidimensional Encuesta Calidad de Vida 2011-2017, 2017, http://medata.gov.co/dataset/indice-multidimensional-encuesta-calidad-de-vida-2011-2017.

[57] S. de Educación de Antioquia, Sedes de los establecimientos educativos del departamento de antioquia, 2018, https://www.datos.gov.co/Educaci-n/Sedes-de-los-Establecimientos-Educativos-del-Depar/tgsp-kujm.

[58] R. Tibshirani, G. Walther, D. Botstein, P. Brown, Cluster validation by prediction strength, Technical Report, Department of Statistics, Stanford University, Stanford, USA, 2001.

[59] P.J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.

[60] C. Wu, C. Ouyang, L. Chen, L. Lu, A new fuzzy clustering validity index with a median factor for centroid-based clustering, IEEE Trans. Fuzzy Syst. 23 (3) (2015) 701–718.

[61] Q. Zhao, Cluster Validity in Clustering Methods (Ph.D. thesis), University of Eastern Finland, Joensuu, Finland, 2012.

[62] E. Rubio, O. Castillo, P. Melin, A new validation index for fuzzy clustering and its comparisons with other methods, in: 2011 IEEE Int. Conf. on Systems, Man, and Cybernetics, 2011, pp. 301–306.

[63] A. Suleman, Measuring the congruence of fuzzy partitions in fuzzy c-means clustering, Appl. Soft Comput. 52 (2017) 1285–1295.

[64] B. Kim, H. Lee, P. Kang, Integrating cluster validity indices based on data envelopment analysis, Appl. Soft Comput. 64 (2018) 94–108.

[65] P. Rathore, Z. Ghafoori, J.C. Bezdek, M. Palaniswami, C. Leckie, Approximating dunn's cluster validity indices for partitions of big data, IEEE Trans. Cybern. (2018) 1–13.

[66] A. Cornuéjols, C. Wemmert, P.G. carski, Y. Bennani, Collaborative clustering: Why, when, what and how, Inf. Fusion 39 (2018) 81–95.

[67] G.C. Nutakki, B. Abdollahi, W. Sun, O. Nasraoui, An introduction to deep clustering, in: O. Nasraoui, C.-E. Ben N'Cir (Eds.), Clustering Methods for Big Data Analytics: Techniques, Toolboxes and Applications, Springer International Publishing, 2019, pp. 73–89.