



Similarity measures for interval-valued fuzzy sets based on average embeddings and its application to hierarchical clustering [☆]

Noelia Rico ^{a,*}, Pedro Huidobro ^b, Agustina Bouchet ^b, Irene Díaz ^a

^a Department of Computer Science, University of Oviedo, Spain

^b Department of Statistics, University of Oviedo, Spain

ARTICLE INFO

Article history:

Received 28 February 2022

Received in revised form 30 July 2022

Accepted 3 October 2022

Available online 10 October 2022

Keywords:

Interval-valued fuzzy sets

Similarity measure

Hierarchical clustering

Interval-based data

Embedding function

Weather data mining

ABSTRACT

Clustering algorithms create groups of objects based on their similarity. As objects are usually defined by data points, this similarity is commonly measured by a distance function. When the objects are defined by variables that are intervals, it is more difficult to determine how to measure the similarity between the objects of the dataset. In this work, we propose some similarity measures between intervals based on average embedding functions. Using these, new similarity measures between interval-based objects are proposed. All the proposed similarities are based on measuring the similarity between the objects variable by variable and then averaging the obtained results to get a single value. By its definition, the objects can be considered as interval-valued fuzzy sets (IVFS), so the similarities introduced are proved to be valid similarities for IVFS. The measures proposed are used in a hierarchical clustering algorithm with the aim of grouping the objects of the dataset into different clusters based on their similarity to interval-valued data. The described process is applied to real data regarding the Spanish weather in order to cluster the provinces of Spain based on the interval temperature of each month in 2021, showing different results than the ones obtained using non-interval-valued data.

© 2022 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Clustering methods aim to create clusters of objects in a dataset by grouping the objects according to some criterion of similarity [40]. The purpose of these methods is to minimize intra-group distance and, at the same time, to maximize inter-group distance [34]. The agglomerative hierarchical clustering algorithm starts assuming that each object belongs to an individual cluster. Then, it performs pairwise comparisons between all the clusters of the dataset in order to measure their similarity. This comparison makes possible to determine the two most similar ones, which are grouped together into a single cluster. This pairwise comparison between clusters is repeated sequentially until all the objects are merged into one single cluster [22]. The result of the hierarchical clustering algorithm is commonly represented as a tree graph called *dendrogram*.

[☆] This research has been partially supported by Spanish MINECO projects TIN2017-87600-P (Noelia Rico, Agustina Bouchet and Irene Díaz) and PGC2018-098623-B-I00 (Pedro Huidobro). Pedro Huidobro and Noelia Rico are also supported by the Severo Ochoa predoctoral grant program by the Principality of Asturias (PA-20-PF-BP19-169 and PA-20-PF-BP19-167). The code to reproduce this research is available at <https://github.com/noeliarico/embclust/>.

* Corresponding author.

E-mail addresses: noeliarico@uniovi.es (N. Rico), huidobropedro@uniovi.es (P. Huidobro), bouchetagustina@uniovi.es (A. Bouchet), sirene@uniovi.es (I. Díaz).

Usually, the number of clusters is determined once the complete dendrogram is built. When this number is set, it is possible to define the objects in each cluster by using the dendrogram, cutting the tree at the level corresponding to the desired number of clusters. This presents an advantage in relation with partitioning clustering algorithms, when the number of clusters must be set beforehand [2].

In the process of creating the groups, this algorithm is classically applied to create clusters of objects from a dataset where each object is defined by variables which value commonly take a real number. For this reason, the similarity between the objects can be measured by using point-based distances such as the Euclidean distance. Also it is common to find dataset of categorical variables, for which distances such as the Jaccard distance are common [19]. In this work, we focus on interval-valued datasets, for which their objects are defined by a set of variables (all the objects by the same variables) that take an interval value. In the literature, some clustering methods to deal with this kind of data can be found. For instance, Lingras and West [25] propose a version of the k -means algorithm to develop interval clusters of web visitors using rough set theory. In the same year, de Souza and de Carvalho [11] introduce some methods for clustering with interval data. Those methods are extensions of the standard dynamic cluster method. They present a version of the adaptive city-block distance and two algorithms that converge to a value as an effect of the fitting between the kind of representation of the clusters and the characteristics of the distance functions. Some years later, Chavent et al. [7] propose two methods based also on the dynamical clustering method. Both methods compare two vectors of intervals, the first one utilizing a distance that is based on the Hausdorff distance and the second one using a dissimilarity. In addition, methods for fuzzy clustering can be also found, for example de Carvalho [10] presents two methods, adaptive and non-adaptive fuzzy c -means for partitioning symbolic interval data. Other authors like Guh et al. [17] introduce the use of interval-valued fuzzy relations in order to apply them to hierarchical clustering. In this case, they not use interval value data, they used matrices formed by interval-valued fuzzy sets, where the intervals express the value of the similarity between the objects. Researchers as Galdino and Marciel [14] consider some arithmetic operations between intervals in order to define the Euclidean distance as an interval. On the other hand, Ramos-Guajardo [32] proposes some methods based on the Jaccard index and on statistical hypothesis tests. Furthermore, Vo-Van et al. [38] show a clustering algorithm that can automatically choose the number of clusters and designate the outlier intervals into separated clusters. Then they apply it in detecting the abnormal images, and the images contaminated with noise.

In some real situations, it could be hard to find the proper value to express some variable or it is not known precisely. Nevertheless, we can avoid this problem by considering intervals. Interval-valued data has attracted very quickly the attention of multiple researchers since they could witness their high possibility for diverse applications [24,33,18,39]. In order to compare intervals, the most used similarities are the Jaccard similarity [19] and the Dice similarity [12].

The fact that many machine learning algorithms require the definition of similarities between the objects motivates an increasing study of these metrics. It is common to find similarity measures that are fitted to a concrete problems. This can be seen in many areas like data medicine [37], pattern recognition [27], decision-making [41], approximate reasoning [31], fuzzy analysis [42], image processing [5], among others. In the literature, it is possible to find several proposals of similarity functions between objects of a dataset, with different properties and their own strengths and weaknesses. The most common approach to describe a similarity measure is considering the grade of similarity as a number between 0 and 1, where 0 means that the objects are not similar while 1 denotes the objects compared are identical. Another interesting point about similarity measures is that it is not so hard to obtain a distance function by transforming the similarity measure [31,15].

In the seventies Zadeh [43], Grattan-Guinness [16], Jahn [20], and Sambuc [35] introduced independently the concept of interval-valued fuzzy sets (IVFS). Since then, authors have used them in a diverse fields like, for example, [35] for medical diagnosis in thyroidian pathology, [1] in convexity, [4] in approximate reasoning, or [8] in logic.

The main aim of this paper is to define new similarity measures in order to apply to hierarchical clustering. In this way, we propose these new similarity measures between two objects defined by interval variables. It allows to define a similarity matrix that represents the pairwise similarity between each pair of objects of the dataset. Then, using this similarity matrix, the hierarchical clustering algorithm can be applied in order to determine the clusters that should be merged together in each level of the tree. These similarity functions are based on averaging embedding functions proposed by Bouchet et al. [3]. These embedding functions are a generalization of the notion of *subsethood* Kabir et al. [23]. Note that the objects defined by interval variables can be characterized as IVFS. For this reason, the similarities proposed are proved to be valid similarities for IVFS.

This paper is organized as follows. Section 2 gives an overview of hierarchical clustering method for creating data partitions. Section 3 presents embedding functions and defines the similarity using the average of these functions. After that, Section 4 deals with similarity measures between interval-valued fuzzy sets. To continue, the clustering algorithm with similarities based on interval-valued data is presented and illustrated with a toy example in Section 5. In Section 6, an application to the clustering algorithm using the proposed similarities to interval-valued data of the Spanish temperatures in 2021 is shown. Finally conclusions are drawn in the last section.

2. Hierarchical clustering

Considering a dataset \mathbb{X} of objects n , the purpose of any clustering algorithm is to group these objects into ℓ different clusters \mathcal{C} . *Hard clustering* algorithms determine that each object belongs uniquely to one cluster such that $\mathcal{C}_1 \cup \dots \cup \mathcal{C}_m = \mathbb{X}$.

Agglomerative hierarchical clustering algorithms are a kind of hard clustering algorithm that provides a partition for any number ℓ of clusters between 1 (i.e. one single cluster that contains the complete dataset) and n (each of the objects in \mathbb{X} is a cluster of one single object). The clustering process is done iteratively starting with each object in a single cluster. Then, these clusters are merged such that, at each step of the algorithm, the two most similar clusters are combined into one single cluster. For this reason hierarchical clustering algorithms are shown in a binary tree-based representation called *dendrogram*. This representation has all the individual clusters on the leaves and a single cluster with the whole dataset on the root of the tree. The algorithm builds this dendrogram from leaves to root. Then, the dendrogram can be used to group the dataset into any number of clusters by cutting the tree at the desired level.

How the similarity is measured in order to determine the clusters to merge relies on how the data in \mathbb{X} is defined. Usually, the datasets \mathbb{X} are a set of objects, such that each object $\vec{x}_i \in \mathbb{X}$ is a vector defined by n variables $\vec{x}_i = (x_i^1, \dots, x_i^n)$ with $x_i^k \in \mathbb{R}$ for $1 \leq k \leq n$. For this reason, the similarity is commonly measured by means of a distance function, considering that the closest that two objects are according to the distance, the most similar they are. Although the Euclidean distance is usually the one used in practice, many variations have been proposed in the literature [6,26,29] and this is still a hot topic in research due to the impact that this has on the results of the clustering algorithm.

How to measure the similarity between two objects is the first step towards using the algorithm. However, it is necessary to take into account that, after the first iteration when there is at least one cluster, and after the second (when more than one cluster may exist) it is necessary to establish how to compare the clusters.

A *linkage method* is a method to measure the similarity between two clusters. Depending on the linkage method chosen, the algorithm will result in different sets of clusters [28]. Many linkage methods have been defined in the literature. The *single* linkage method defines the similarity between the cluster as the similarity between the two most similar objects. In the same fashion *complete* linkage method uses the distance between the most different objects of the clusters. A step further, that considers more information, is the *average* linkage method that computes the average between all pairs of objects in the dataset. There are also some more elaborated methods such as the *centroid*, which calculates the centroid of each cluster (i.e. for each variable the average of all the points that belong to the cluster) and computes the distance between the two clusters using their centroids. Other methods are based on the variance such as Ward's method. In the next subsection, hierarchical clustering for interval-valued data is defined.

2.1. Hierarchical clustering applied to interval-valued data

For objects defined by variables that take real values, the measure of similarity between objects is usually done by computing a distance between the objects. However, for data where each data point is defined by intervals, it is not straightforward how to measure the similarity between two objects.

Let us denote this kind of datasets by \mathbb{Y} where each object $z_i \in \mathbb{Y}$ is defined by n variables such that $z_i = \{z_i^1, \dots, z_i^n\}$, where each variable is an interval $z_i^k = [\underline{z}_i^k, \overline{z}_i^k]$ having that $\underline{z}_i^k \leq \overline{z}_i^k$ with $1 \leq k \leq n$. Before starting, we should normalize the data set in each variable, obtaining that every interval is contained on the unit interval. We will denote as $\mathbb{Y}^{[0,1]}$ the set of all normalized intervals. Keeping this idea on mind, we can interpret each object $y_i \in \mathbb{Y}^{[0,1]}$ as an interval-valued fuzzy set.

Let X be the referential set. An IVFS on X is defined by a mapping $a : X \rightarrow L([0, 1])$ such that $a(x) = [\underline{a}(x), \overline{a}(x)]$, where $L([0, 1])$ denotes the family of closed intervals included on the unit interval. Then, it is completely characterized by two mappings, \underline{a} and \overline{a} , from X into $[0, 1]$ such that $\underline{a}(x) \leq \overline{a}(x), \forall x \in X$. We will denote the set of all interval-valued fuzzy sets as $IVFS(X)$ [36].

Thus, given a dataset $\mathbb{Y}^{[0,1]}$, each object $y_i \in \mathbb{Y}^{[0,1]}$ is defined by a set of n variables, where each variable is an interval $y_i^k = [\underline{y}_i^k, \overline{y}_i^k]$ such that $\underline{y}_i^k \leq \overline{y}_i^k$. We would like to point out that there is a natural relation between $\mathbb{Y}^{[0,1]}$ and $IVFS(X)$. If we identify the variables with the elements of X , every object of $\mathbb{Y}^{[0,1]}$ could be noticed as an interval-valued fuzzy set on X and vice versa.

If $X = \{x_1, \dots, x_n\}$, the interval-valued fuzzy set a has n possible values, $\{a(x_1), \dots, a(x_n)\}$, contained on the unit interval $[0, 1]$. From the previous set it is possible to construct $(a(x_1), \dots, a(x_n))$ such that each object y_i with n variables, where $y_i^k = [\underline{y}_i^k, \overline{y}_i^k]$ is the value of the variable k , with $1 \leq k \leq n$, so $y_i \in \mathbb{Y}^{[0,1]}$. Therefore, we identify each $a(x_k)$ as y_i^k . Let us see an example.

Example 1. Let us suppose that we have 3 variables and the following object $z_i \in \mathbb{Y}$:

	z_i^1	z_i^2	z_i^3
z_i	[2, 4]	[5, 7]	[4, 9]

First, we should normalize the dataset, so the objects are now $y_i \in \mathbb{Y}^{[0,1]}$:

	y_i^1	y_i^2	y_i^3
y_i	$[0, \frac{2}{7}]$	$[\frac{3}{7}, \frac{5}{7}]$	$[\frac{2}{7}, 1]$

Then, we can construct an IVFS a on $X = \{x_1, x_2, x_3\}$ as:

	x_1	x_2	x_3
a	$[0, \frac{2}{7}]$	$[\frac{3}{7}, \frac{5}{7}]$	$[\frac{2}{7}, 1]$

where $a(x_1) = [0, \frac{2}{7}]$ and so on.

Thus, to cluster this kind of data, the first step is to define how to measure the similarity between two objects. Therefore, it is necessary to define a similarity measure. We propose to measure the similarity between two objects using, variable by variable, similarities based on embedding functions for intervals. Then, the individual similarities obtained for each variable are averaging to get a single value. By doing this, a similarity matrix \mathbf{S} can be obtained comparing every pair of objects $y_i, y_j \in \mathbb{Y}^{[0,1]}$, and using \mathbf{S} the hierarchical algorithm can be applied with the aim of finding groups in the data. Further details about how to measure this similarity and to apply the algorithm are given in the next sections.

3. Similarity measures defined based on embedding functions

When dealing with intervals, the notion of embedding is a cornerstone. It is based on a partial order such that, for $a = [\underline{a}, \bar{a}]$ and $b = [\underline{b}, \bar{b}]$, a is contained in b if and only if $\underline{b} \leq \underline{a} \leq \bar{a} \leq \bar{b}$. Bouchet et al. [3] proposed some embedding measures for intervals based on the following definition:

Definition 1. The function $E : L([0, 1]) \times L([0, 1]) \rightarrow [0, 1]$ is an embedding on $L([0, 1])$, if for any $a, b, c \in L([0, 1])$ the following properties are hold:

- A1. $E(a, b) = 1$ if and only if $a \subseteq b$
- A2. If $a \cap b = \emptyset$, then $E(a, b) = E(b, a) = 0$
- A3. If $b \subseteq c$, then $E(a, b) \leq E(a, c)$

It should be noticed that an embedding E is not a commutative map, e.g. if $a = [0.2, 0.3]$ and $b = [0, 1]$, then $E(a, b) = 1$ but $E(b, a)$ must be different from 1 since b is not included on a . Some embedding functions satisfying Definition 1 are introduced below. The first one, which is based on the interval width, is E_w . It is defined as follows:

$$E_w(a, b) = \begin{cases} 1 & \text{if } w(a) = 0, a \subseteq b \\ 0 & \text{if } w(a) = 0, a \not\subseteq b \\ \frac{w(a \cap b)}{w(a)} & \text{if } w(a) \neq 0 \end{cases}$$

is an embedding for intervals.

Example 2. Consider the intervals $[0.2, 0.3]$ and $[0, 1]$, the embedding $E_w(a, b)$ is applied over them showing that

$$\begin{aligned} E_w(a, b) &= E_w([0.2, 0.3], [0, 1]) = \frac{0.1}{0.1} = 1, \\ E_w(b, a) &= E_w([0, 1], [0.2, 0.3]) = \frac{0.1}{1} = 0.1, \end{aligned}$$

which shows that embeddings are not commutative.

After constructing embeddings based on the interval width, in [3] it is also developed a method for obtaining interval embeddings based on implications. Here we present some of these intervals embeddings based on some well-known implications:

- This embedding applies the Lukasiewicz implication in the endpoints of the interval.

$$E_{LK}(a, b) = \begin{cases} 0 & \text{if } a \cap b = \emptyset \\ \min(1 - \underline{b} + \underline{a}, 1 - \bar{a} + \bar{b}, 1) & \text{otherwise} \end{cases}$$

- Here, the Fodor implication is used to build the following operator

$$E_{FD}(a, b) = \begin{cases} 0 & \text{if } a \cap b = \emptyset \\ 1 & \text{if } a \subseteq b \\ \max(1 - \bar{a}, \bar{b}) & \text{if } \underline{b} < \underline{a} \leq \bar{b} < \bar{a} \\ \max(1 - \underline{b}, \underline{a}) & \text{if } \underline{a} < \underline{b} \leq \bar{a} < \bar{b} \\ \min\{\max(1 - \bar{a}, \bar{b}), \max(1 - \underline{b}, \underline{a})\} & \text{if } b \subset a \end{cases}$$

- The Gödel implication help to construct this embedding

$$E_{GD}(a, b) = \begin{cases} 0 & \text{if } a \cap b = \emptyset \\ 1 & \text{if } a \subseteq b \\ \bar{b} & \text{if } \underline{b} \leq \underline{a} \leq \bar{b} < \bar{a} \\ \underline{a} & \text{otherwise} \end{cases}$$

- Using the Goguen implication, it is possible to arrive to the following map

$$E_{GG}(a, b) = \begin{cases} 0 & \text{if } a \cap b = \emptyset \\ 1 & \text{if } a \subseteq b \\ \frac{\bar{b}}{\underline{a}} & \text{if } a \cap b \neq \emptyset \wedge \underline{b} = 0 \\ \min\left(\frac{\bar{b}}{\underline{a}}, \frac{\underline{a}}{\bar{b}}\right) & \text{otherwise} \end{cases}$$

- The following embedding is based on the Rescher implication

$$E_{RS}(a, b) = \begin{cases} 1 & \text{if } a \subseteq b \\ 0 & \text{otherwise} \end{cases}$$

Bouchet et al. [3] pointed out that $E_{RS} \leq E_{GD} \leq E_{FD} \leq E_{LK}$ and $E_{RS} \leq E_{GD} \leq E_{GG} \leq E_{LK}$. However it is not possible to establish an order relation between E_{GG} and E_{FD} .

Nevertheless, it is plausible to establish a relation between intervals which are included one in each other.

Proposition 1. The embeddings E_w , E_{LK} , E_{FD} , E_{GD} , E_{GG} and E_{RS} satisfy that given $a, b, c \in L([0, 1])$ such that $a \subseteq b \subseteq c$, then $E(c, a) \leq E(b, a)$.

Proof. Let the embeddings E_w , E_{LK} , E_{FD} , E_{GD} , E_{GG} and E_{RS} :

- $E_w(b, a) = \frac{w(b \cap a)}{w(b)} \geq \frac{w(c \cap a)}{w(c)} = E_w(c, a)$
- $E_{LK}(b, a) = \min(1 - \underline{a} + \underline{b}, 1 - \bar{b} + \bar{a}, 1) \geq \min(1 - \underline{a} + \underline{c}, 1 - \bar{c} + \bar{a}, 1) = E_{LK}(c, a)$
- E_{FD}
 - Case 1 if $c \subseteq a$ ($a = c = b$), $E_{FD}(b, a) = E_{FD}(c, a)$
 - Case 2 if $a \subset c$,
 - if $b = c$, thus $E_{FD}(b, a) = E_{FD}(c, a)$
 - if $b \subset c$, then $E_{FD}(b, a) = \min\{\max(1 - \bar{b}, \bar{a}), \max(1 - \underline{a}, \underline{b})\} \geq \min\{\max(1 - \bar{c}, \bar{a}), \max(1 - \underline{a}, \underline{c})\} = E_{FD}(c, a)$ as $1 - \bar{b} \geq 1 - \bar{c}$ and $\underline{b} \geq \underline{c}$.
- E_{GD}
 - Case 1 if $b \subseteq a$ ($a = b$), $E_{GD}(b, a) = 1 \geq E_{GD}(c, a)$
 - Case 2 if $b \not\subseteq a$ and $\underline{a} = \underline{b} \leq \bar{a} < \bar{b}$, $E_{GD}(b, a) = \bar{a} \geq E_{GD}(c, a)$
 - Case 3 if $b \not\subseteq a$ and not $\underline{a} = \underline{b} \leq \bar{a} < \bar{b}$, $E_{GD}(b, a) = \underline{b} \geq E_{GD}(c, a)$

- E_{GG}

- Case 1 if $b \subseteq a$ ($a = b$), $E_{GG}(b, a) = 1 \geq E_{GG}(c, a)$
- Case 2 $\bar{b} = 0 \Rightarrow b \subseteq a \Rightarrow$ case 1
- Case 3 if $\underline{a} = 0$, $E_{GG}(b, a) = \frac{\bar{a}}{b} \geq E_{GG}(c, a)$
- Case 4 if $b \not\subseteq a$, $\bar{b} \neq 0$ and $\underline{a} \neq 0$, $E_{GG}(b, a) = \min\left(\frac{b}{\underline{a}}, \frac{\bar{a}}{b}\right) \geq E_{GG}(c, a)$

- $E_{RS}(b, a) = 0 = E_{RS}(c, a)$

In addition, it is interesting to analyze $E(c, a)$ and $E(c, b)$. At a first sight, it seems reasonable that $E(c, a)$ should be smaller or equal to $E(c, b)$ as c is closer to b than a .

Proposition 2. The embeddings $E_w, E_{LK}, E_{FD}, E_{GD}, E_{GG}$ and E_{RS} satisfy that given $a, b, c \in L([0, 1])$ such that $a \subseteq b \subseteq c$, then $E(c, a) \leq E(c, b)$.

Proof. Let $a, b, c \in L([0, 1])$ such that $a \subseteq b \subseteq c$:

- $E_w(c, b) = \frac{w(c \cap b)}{w(c)} = \frac{w(b)}{w(c)} \geq \frac{w(a)}{w(c)} = \frac{w(c \cap a)}{w(c)} = E_w(c, a)$
- $E_{LK}(c, b) = \min(1 - \underline{b} + \underline{c}, 1 - \bar{c} + \bar{b}, 1) \geq \min(1 - \underline{a} + \underline{c}, 1 - \bar{c} + \bar{a}, 1) = E_{LK}(c, a)$
- E_{FD}

- Case 1 if $c \subseteq a$ ($a = c = b$), $E_{FD}(c, b) = 1 \geq E_{FD}(c, a)$
- Case 2 if $a \subset c$,

* if $b = c$, thus $E_{FD}(c, b) = E_{FD}(c, a)$

* if $b \subset c$, then $E_{FD}(c, b) = \min\{\max(1 - \bar{c}, \bar{b}), \max(1 - \underline{b}, \underline{c})\} \geq \min\{\max(1 - \bar{c}, \bar{a}), \max(1 - \underline{a}, \underline{c})\} = E_{FD}(c, a)$ because $\bar{b} \geq \bar{a}$ and $1 - \underline{b} \geq 1 - \underline{a}$.

- E_{GD}

- Case 1 if $c \subseteq b$ ($c = b$), $E_{GD}(c, b) = 1 \geq E_{GD}(c, a)$
- Case 2 if $c \not\subseteq b$ and $\underline{b} = \underline{c} \leq \bar{b} < \bar{c}$, $E_{GD}(c, b) = \bar{a} \geq E_{GD}(c, a)$
- Case 3 if $c \not\subseteq b$ and not $\underline{b} = \underline{c} \leq \bar{b} < \bar{c}$, $E_{GD}(c, b) = \underline{b} \geq E_{GD}(c, a)$

- E_{GG}

- Case 1 if $c \subseteq b$ ($c = b$), $E_{GG}(c, b) = 1 \geq E_{GG}(c, a)$
- Case 2 $\bar{c} = 0 \Rightarrow c \subseteq b \Rightarrow$ case 1
- Case 3 if $\underline{b} = 0$, $E_{GG}(c, b) = \frac{\bar{b}}{c} \geq E_{GG}(c, a)$
- Case 4 if $c \not\subseteq b$, $\bar{c} \neq 0$ and $\underline{b} \neq 0$, $E_{GG}(c, b) = \min\left(\frac{c}{\underline{b}}, \frac{\bar{b}}{\bar{c}}\right) \geq E_{GG}(c, a)$

- E_{RS}

- Case 1 if $c \subseteq b$ ($c = b$), $E_{GG}(c, b) = 1 \geq E_{GG}(c, a)$
- Case 2 if $c \not\subseteq b$, $E_{RS}(c, b) = 0 = E_{RS}(c, a)$

Next, let us recall the definition of similarity function.

Definition 2. The function $S : L([0, 1]) \times L([0, 1]) \rightarrow [0, 1]$ is a similarity measure for intervals if the following properties hold for any $a, b, c \in L([0, 1])$ [9,21]:

- S1: $0 \leq S(a, b) \leq 1$
- S2: $S(a, b) = S(b, a)$
- S3: $S(a, b) = 1$ if and only if $a = b$
- S4: If $a \subseteq b \subseteq c$ then $S(a, c) \leq S(a, b)$ and $S(a, c) \leq S(b, c)$

We propose the use of embedding function to measure the similarity between two intervals. As it was shown before, embedding maps are not commutative, so given a and b two intervals, we will consider the average of $E(a, b)$ and $E(b, a)$ to measure the similarity between a and b .

Proposition 3. Let E be an embedding function. Let $\mathcal{S}_E : L([0, 1]) \times L([0, 1]) \rightarrow [0, 1]$ be the mean of $E(a, b)$ and $E(b, a)$, i.e., $\mathcal{S}_E(a, b) = \frac{E(a, b) + E(b, a)}{2}$. If the embedding satisfies that given $a, b, c \in L([0, 1])$ such that $a \subseteq b \subseteq c$, we have that $E(c, a) \leq E(b, a)$ and $E(c, a) \leq E(c, b)$, then the operator \mathcal{S}_E is a similarity function.

Proof. We should check if \mathcal{S}_E fulfills the axioms for being a similarity function.

- Consider $a, b \in L([0, 1])$, then $0 \leq \mathcal{S}_E(a, b) \leq 1$ as $0 \leq E(a, b) \leq 1$ and $0 \leq E(b, a) \leq 1$.
- \mathcal{S}_E is symmetric by construction.
- $\mathcal{S}_E(a, b) = \frac{E(a, b) + E(b, a)}{2} = 1 \iff E(a, b) = E(b, a) = 1 \iff a = b$
- For $a, b, c \in L([0, 1])$, if $a \subseteq b \subseteq c$, then

$$\mathcal{S}_E(a, c) = \frac{E(a, c) + E(c, a)}{2} = \frac{1 + E(c, a)}{2} \leq \frac{1 + E(b, a)}{2} = \frac{E(a, b) + E(b, a)}{2} = \mathcal{S}_E(a, b)$$

$$\mathcal{S}_E(a, c) = \frac{E(a, c) + E(c, a)}{2} = \frac{1 + E(c, a)}{2} \leq \frac{1 + E(c, b)}{2} = \frac{E(b, c) + E(c, b)}{2} = \mathcal{S}_E(b, c)$$

Therefore, considering the similarity defined using embeddings verified in Proposition 3 we can define the similarities $\mathcal{S}_w, \mathcal{S}_{LK}, \mathcal{S}_{FD}, \mathcal{S}_{GD}, \mathcal{S}_{GG}$ and \mathcal{S}_{RS} .

Corollary 1. The operator $\mathcal{S}_E(a, b) = \frac{E(a, b) + E(b, a)}{2}$ is a similarity function for the embeddings $E_w, E_{LK}, E_{FD}, E_{GD}, E_{GG}$ and E_{RS} .

Proof. It is evident using Propositions 1 and 2.

Once we have achieved these results, one could wonder what happen when we use an aggregation function instead of the arithmetic mean. Not every aggregation function generates a similarity measure. For instance, if we consider the maximum the axiom S3 is not fulfilled as it is shown in the following example.

Example 3. If we consider the interval given in Example 2, it is evident that:

$$\max(E_w([0.2, 0.3], [0, 1]), E_w([0, 1], [0.2, 0.3])) = 1$$

as $E_w([0.2, 0.3], [0, 1]) = 1$, but the intervals are clearly different.



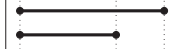
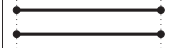
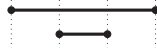
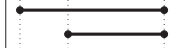
In Section 4, we will use the previous result in order to define the similarity between two IVFS. This similarity will be then used to compare objects of the dataset $\mathbb{Y}^{[0,1]}$.

3.1. An intuitive interpretation of the similarities

Considering the similarities proposed based on embedding functions, let us give an intuition of how these similarities behave depending on the input intervals in order to contribute towards the explainability of the results obtained with each function. Consider the following pairs of intervals, that are classified attending to their relation. The possible relations are graphically represented in Table 1 and are:

- Case 1: no overlapping intervals.
- Case 2: identical intervals.
- Case 3: non empty intersection.
- Case 4: different intervals but one interval within another.
- Case 5: different intervals but one interval within another, sharing the left endpoint.
- Case 6: different intervals but one interval within another, sharing the right endpoint.

Table 1
Possible relations of a pair of intervals.

Case 1	Case 3	Case 5
		
Case 2	Case 4	Case 6
		

Consider now all the embedding functions E_W , E_{LK} , E_{FD} , E_{GD} , E_{GG} and E_{RS} , and a (b) an interval representing the top (bottom) interval of each cases proposed in Table 1. On the one hand, if the intersection between them is empty, the result of any embedding function is always 0 attending to their definitions for $a \cap b = \emptyset$. On the other hand, if a and b are the same interval, $a = b$, then the result of the embedding is 1. Following this, by definition, all the similarities proposed are 1 if both intervals are the same and 0 if there is no intersection between them. Both situations correspond with Case 1 and 2 in Table 1.

As the similarities are defined based on the mean of $E(a, b)$ and $E(b, a)$, as shown in Definition 4, it can be observed that for all the measures proposed if one of the intervals is contained within the other, $a \subseteq b$ or $b \subseteq a$ with $a \neq b$, the similarity value is at least 0.5 and moreover never can be 1 (Case 4, 5 and 6).

The similarity based on the Rescher embedding is the most restrictive of all the similarities proposed, as it only can be: 0 if any of the intervals is completely contained within the other (Case 1 and 3); 0.5 if one is completely contained in the other but they are different intervals (Case 4, 5 and 6); 1 if both are in the same interval (Case 2). The remaining similarities are not so restrictive.

The width similarity focuses on maximizing the intersection of both intervals. If one interval is contained within the other (Case 4, 5, and 6), the similarity is guaranteed to be greater than 0.5. In this case, the remaining 0.5 depends on the width of the intervals and which percentage of the larger interval occupies the short one. On the other hand, if none is contained within the other but they intersect (Case 3), the value of the similarity depends on the percentage of overlapping in relation to the width of both intervals.

The similarity based on the Lukasiewicz embedding can be summarized for Case 3, 5 and 6 as $1 - \frac{|a-b|+|\bar{a}-\bar{b}|}{2}$. It should be noticed that the value of the similarity is increased when the left endpoints are closer or the right endpoints are closer. For the Case 4 we have 0.5 as one interval is inside another plus $\frac{1}{2} \min(1 - \underline{b} + \underline{a}, 1 - \bar{a} + \bar{b})$, so the similarity value is larger when the left or the right endpoints are close between themselves, respectively.

If we consider the similarity using the Fodor similarity, the Case 3 is represented by $\frac{1}{2} \max(1 - \bar{a}, \bar{b}) + \frac{1}{2} \max(1 - \underline{b}, \underline{a})$. For the cases where one interval is included in the other one, the value is at least 0.5 and the other 0.5 varies depending on $\frac{1}{2} \min\{\max(1 - \bar{a}, \bar{b}), \max(1 - \underline{b}, \underline{a})\}$. It should be remarked that $\max(1 - \bar{a}, \bar{b})$ is increased when \bar{a} is decreased or \bar{b} is increased; and $\max(1 - \underline{b}, \underline{a})$ is increased when \underline{b} is reduced or \underline{a} is grown.

Case 3 using the God el similarity is $\frac{\underline{a} + \bar{a}}{2}$, that is, the middle point of the first interval. When one interval is totally contained, in Cases 4 and 6 we have that the remaining 0.5 is obtained as $\frac{\underline{a}}{2}$, so it grows when \underline{a} increases; and in the Case 5 is obtained as $\frac{\bar{b}}{2}$ which is larger when \bar{b} is raised.

For the similarity obtained with the Goguen embedding, if we consider that all the endpoints are different from 0, the Case 3 is described with $\frac{1}{2} \min(\frac{\bar{b}}{\bar{a}}, \frac{\underline{a}}{\underline{b}}) + \frac{1}{2} \min(\frac{\underline{a}}{\underline{b}}, \frac{\bar{b}}{\bar{a}})$. The Cases 4, 5 and 6, where there is one interval completely included in the other one, are 0.5 plus $\frac{1}{2} \min(\frac{\bar{b}}{\bar{a}}, \frac{\underline{a}}{\underline{b}})$. In these four cases, the value of the similarity depends on the proportion between the endpoints.

4. Similarity between interval-valued fuzzy sets

In subSection 2.1, we considered an object as an interval-valued fuzzy set in a set X of dimension n . In this way, let us recall the following definition:

Definition 3. A real function $\mathbb{S} : IVFS(X) \times IVFS(X) \rightarrow [0, 1]$ is a similarity measure for interval-valued fuzzy sets, for short IVFS similarity measure, if the following properties hold [44]:

-  1: $\mathbb{S}(a, a^c) = 0$ if a is a crisp set
-  2: $\mathbb{S}(a, b) = \mathbb{S}(b, a)$
-  3: $\mathbb{S}(a, b) = 1$ if and only if $a = b$

§4: If $a \subseteq b \subseteq c$ then $\mathbb{S}(a, c) \leq \mathbb{S}(a, b)$ and $\mathbb{S}(a, c) \leq \mathbb{S}(b, c)$

for any $a, b, c \in IVFS(X)$.

After that definition, we can claim that the similarity measures for intervals we are considering could be also similarities for interval-valued fuzzy sets.

Proposition 4. $\mathcal{S}_w, \mathcal{S}_{LK}, \mathcal{S}_{FD}, \mathcal{S}_{GD}, \mathcal{S}_{GG}$ and \mathcal{S}_{RS} are IVFS similarity measures when X is a uni-punctual set, i.e., $X = \{x\}$.

Proof.

- §1: If $a \in IVFS(X)$ is a crisp set, it means that for all values the membership value is 0 or 1. As the intersection of $[0, 0]$ and $[1, 1]$ is \emptyset , then the embedding value is 0.
- §2: Immediate by S2.
- §3: Immediate by S3.
- §4: If $a \subseteq b \subseteq c$, we already know that $\mathcal{S}_E(a(x), c(x)) \leq \mathcal{S}_E(a(x), b(x))$. Now we will study the relation between $\mathcal{S}_E(a(x), c(x))$ and $\mathcal{S}_E(b(x), c(x))$ for a fixed $x \in X$, so for simplicity we will write $\mathcal{S}_E(a, c)$ instead of $\mathcal{S}_E(a(x), c(x))$. We have that $\mathcal{S}_E(a, c) = \frac{1+E(c,a)}{2} \leq \frac{1+E(c,b)}{2} = \mathcal{S}_E(b, c)$ as $E(c, a) \leq E(c, b)$ because by Proposition 2 we know that $E(c, a) \leq E(c, b)$.

The main drawback of the similarity measure \mathcal{S}_E is that it is only valid when X is uni-punctual.

Taking into account that objects in $\mathbb{Y}^{[0,1]}$ could be seen as an interval-valued fuzzy set, and these objects have more than one variable, it is necessary to define how to measure the similarity between two objects.

Definition 4. Consider $y_i, y_j \in \mathbb{Y}^{[0,1]}$, the object similarity function $\mathcal{S}_E : L([0, 1])^n \times L([0, 1])^n \rightarrow [0, 1]$ is

$$\mathcal{S}_E(y_i, y_j) = \frac{\sum_{k=1}^n \mathcal{S}_E(y_i^k, y_j^k)}{n}.$$

As it has been stated in Section 4, the objects of $\mathbb{Y}^{[0,1]}$ are IVFS. Therefore, it is necessary to check if our proposal is a similarity measure between any interval-valued fuzzy sets on X of dimension n .

Proposition 5. The operator \mathcal{S}_E introduced in Definition 4 is an IVFS similarity measure on a set X of dimension n .

Proof. Let us suppose that $X = \{x_1, \dots, x_n\}$.

- §1: If $a \in IVFS(X)$ is a crisp set, $\mathcal{S}_E(a, a^c) = \frac{\sum_{k=1}^n \mathcal{S}_E(a(x_k), a^c(x_k))}{n} = 0$ as $\mathcal{S}_E(a(x_k), a^c(x_k)) = 0$.
- §2: Immediate.
- §3: Immediate.
- §4: If $a \subseteq b \subseteq c$, we already know for a fixed $x_k \in X$ that $\mathcal{S}_E(a(x_k), c(x_k)) \leq \mathcal{S}_E(a(x_k), b(x_k))$ and $\mathcal{S}_E(c(x_k), a(x_k)) \leq \mathcal{S}_E(c(x_k), b(x_k))$, with $1 \leq k \leq n$. Thus,

$$\mathcal{S}_E(a, c) = \frac{\sum_{k=1}^n \mathcal{S}_E(a(x_k), c(x_k))}{n} \leq \frac{\sum_{k=1}^n \mathcal{S}_E(a(x_k), b(x_k))}{n} = \mathcal{S}_E(a, b) \text{ as } \mathcal{S}_E(a(x_k), c(x_k)) \leq \mathcal{S}_E(a(x_k), b(x_k))$$

and

$$\mathcal{S}_E(a, c) = \frac{\sum_{k=1}^n \mathcal{S}_E(a(x_k), c(x_k))}{n} \leq \frac{\sum_{k=1}^n \mathcal{S}_E(b(x_k), c(x_k))}{n} = \mathcal{S}_E(b, c) \text{ as } \mathcal{S}_E(a(x_k), c(x_k)) \leq \mathcal{S}_E(b(x_k), c(x_k)).$$

In a similar way that it happened with embeddings in Example 3, observe that not every aggregation of similarities is a similarity measure.

Example 4. Let us consider the similarity measure defined in Proposition 3 with the width function and the following intervals:

	x_1	x_2
A	[0.2,0.3]	[0.4,0.6]
B	[0,1]	[0.4,0.6]
$E_w(A(x_i), B(x_i))$	1	1
$E_w(B(x_i), A(x_i))$	0.1	1
$\mathcal{S}_{E_w}(A(x_i), B(x_i))$	0.55	1

However, if we use the maximum as the aggregation function, we obtain that:

$$\mathcal{S}_E(A, B) = \max(\mathcal{S}_{E_w}(A(x_1), B(x_1)), \mathcal{S}_{E_w}(A(x_2), B(x_2))) = 1$$

which shows that it is not a similarity measure as the axiom $\mathbb{S}3$ is not fulfilled.

In the next section, these similarity measures are applied to hierarchical clustering.

5. Interval hierarchical clustering using \mathcal{S}_E

Consider a dataset \mathbb{Y} , in order to apply a hierarchical clustering algorithm, it is necessary to obtain the similarity between every pair of objects of the dataset. In this work, similarity measures \mathcal{S}_E based on some embedding functions (E_w , E_{LK} , E_{FD} , E_{GD} , E_{GG} and E_{RS}) are considered. Recall that these embedding functions are defined for intervals between 0 and 1. Thus, in order to apply the similarity measures \mathcal{S}_E defined using these embedding functions it is necessary to normalize the dataset such that \mathbb{Y} is transformed into $\mathbb{Y}^{[0,1]}$, where each variable is an interval in range $[0, 1]$. If we consider two objects $y_i, y_j \in \mathbb{Y}^{[0,1]}$, the similarity between these objects is obtained using the similarity based on the embedding function variable by variable (see Definition 4). First, the values $\mathcal{S}_E(y_i^k, y_j^k)$ with $0 \leq k \leq n$ are obtained for each variable. Then, these values are aggregated using the mean to obtain a single value $\mathcal{S}_E(y_i, y_j)$ of the similarity between the two objects.

Using this, it is possible to compare pairwise all the objects $y_i, y_j \in \mathbb{Y}^{[0,1]}$ and to define a similarity matrix \mathbf{S}^E of dimension $m \times m$, such that each element \mathbf{S}_{ij}^E represents the similarity between y_i and y_j according to \mathcal{S}_E . It is straightforward that this matrix is symmetric.

Once the matrix \mathbf{S}^E is obtained, the clustering algorithm can be applied. At each iteration, this matrix must be considered to determine the two most similar clusters that will be merged together into one single cluster.

The algorithm obtained from the considerations explained is shown in Algorithm 1. This algorithm describes the full process and it can be divided into two steps: obtaining the similarity matrix (lines 1 to 9) and using this matrix for creating the clusters (lines 10 to 14).

Algorithm 1 Hierarchical clustering using similarity measures based on embedding functions

Input: Dataset $\mathbb{Y}^{[0,1]}$ of m objects defined by n intervals

- 1: Establish the embedding function E in which the similarity will be based
- 2: Establish the linkage method μ
 - ▷ Define the matrix \mathbf{S}^E of dimension $m \times m$ computing the similarity for each pair of objects
- 3: **for all** $y_i, y_j \in \mathbb{Y}^{[0,1]} : y_i \neq y_j$ **do** ▷For each pair of objects of the dataset
- 4: **for all** $k \in \{1, \dots, n\}$ **do** ▷For each variable of the dataset (defined by an interval)
- 5: $s_{ij}^k = \mathcal{S}_E(y_i^k, y_j^k)$ ▷Measure the similarity of the two objects in that variable
- 6: **end for**
- 7: $\mathcal{S}_E(y_i, y_j) = \frac{\sum_{k=1}^n s_{ij}^k}{n}$ ▷Similarity between two objects
- 8: $\mathbf{S}_{ij}^E = \mathcal{S}_E(y_i, y_j)$ ▷Update the matrix
- 9: **end for**
 - ▷ Use the similarity matrix \mathbf{S}^E as input for hierarchical clustering
- 10: $\ell = m$ ▷Initially each object has its own cluster, the algorithm starts with the leaves
- 11: **while** $\ell > 0$ **do** ▷Build the tree from bottom to top
- 12: Determine the two closest clusters using μ
- 13: Merge the closest clusters into one
- 14: **end while**

In Example 5, Algorithm 1 is illustrated using S_w and the single linkage method.

Table 2

Interval-valued toy dataset \mathbb{Y} representing temperatures given in Example 5 (left) and the same dataset normalized $\mathbb{Y}^{[0,1]}$ by column so the intervals of each variable are in range $[0, 1]$ (right). To obtain the normalized dataset the endpoints of the intervals are normalized using the maximum right end point and the minimum left endpoint for each variable.

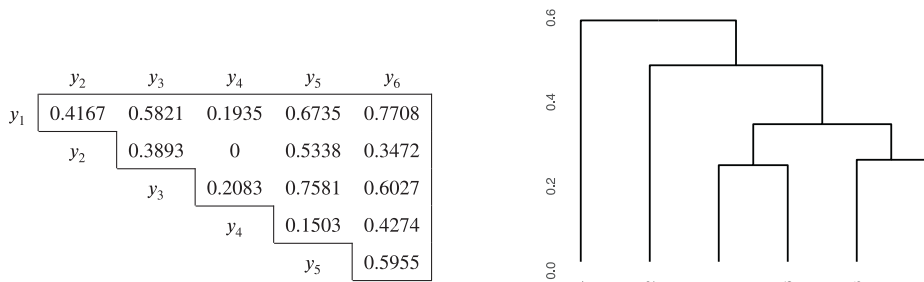
\mathbb{Y}	Temperature	Precipitation		$\mathbb{Y}^{[0,1]}$	Temperature	Precipitation
z_1	[5,11]	[16,21]	→	y_1	[0.3,0.6]	[0.32,0.42]
z_2	[6,12]	[0,8]		y_2	[0.35,0.65]	[0,0.16]
z_3	[7,16]	[7,30]		y_3	[0.4,0.85]	[0.14,0.6]
z_4	[-1,6]	[19,50]		y_4	[0,0.35]	[0.38,1]
z_5	[4,19]	[4,21]		y_5	[0.25,1]	[0.08,0.42]
z_6	[2,11]	[13,25]		y_6	[0.15,0.6]	[0.26,0.5]

Example 5. Let us consider the artificial toy dataset shown in Table 2 (left). This dataset contains 6 objects, representing 6 different cities. For each city, two different variables are given: temperature and precipitation. Each of these variables is defined by an interval showing the minimum and maximum values of the temperature and precipitation respectively for one week. The dataset (left) is normalized (right) to apply the similarities obtained.

The objects of the dataset are going to be compared using the similarity based on the width embedding function E_w . Let us denote S^w the similarity matrix with the pairwise comparison between the data using E_w . Notice that, as this matrix is symmetric, it can be summarized using only $tr(m)$, being tr the triangular number. The first step is to apply the similarity function variable by variable. Then the final results are aggregated using the average. The results obtained for the pairwise comparison of each variable are shown below.

Temperature						Precipitation					
	y_2	y_3	y_4	y_5	y_6		y_2	y_3	y_4	y_5	y_6
y_1	0.8333	0.5556	0.1548	0.7	0.8333	y_1	0	0.6087	0.2323	0.6470	0.7083
	y_2	0.6944	0	0.7	0.6944		y_2	0.0842	0	0.3676	0
		y_3	0	0.8	0.4444			y_3	0.4165	0.7161	0.7609
			y_4	0.2095	0.5079				y_4	0.0910	0.3468
				y_5	0.6222					y_5	0.5687
											y_6

The results obtained for each variable shown above must be now aggregated in order to obtain the similarity matrix. This corresponds to applying Definition 4. Below are shown the resulting matrix (left) and the results obtained from the aggregation and the hierarchical clustering resulting from applying the single linkage method (right).



6. Application to real data

In this section, the hierarchical clustering algorithm using the similarity defined between IVFS is applied to the data of the AEMET (after its Spanish name Agencia Estatal de Meteorología)¹, which is the State Meteorological Agency of the Spanish Government.

¹ <https://opendata.aemet.es/>

It is usually considered that in Spain there are, broadly speaking, five different types of climates depending on the location²:

- the climate of the Atlantic coast, which is humid and rainy, usually cold but does not have very extreme temperatures.
- the climate of the central plateau, quite arid and moderately continental, with relatively cold winters and hot summers.
- the Mediterranean climate of the southern and eastern coastal regions which is mild and sunny all the year.
- the mountainous climate of the mountains, more or less cold depending on altitude.
- the climate of the regions of Spain that are close to Africa, mild in winter and very hot in summer.

The aim of this section is to apply the hierarchical clustering algorithm to interval-valued data, check whether any of the proposed similarity measures are able to detect these different climates, and compare the results using non-interval-valued data.

6.1. Creation of the dataset

The raw data obtained from the AEMET has been transformed in order to build an interval-valued dataset. For each of the 50 provinces and the two autonomous cities in Spain (Ceuta and Melilla), data has been collected from the daily records of the AEMET, which register the maximum and minimum temperature of each day in each station. As the number of stations varies depending on the city, for this dataset we have selected for each province the data corresponding to the capital. Using this, for each month, the daily data has been aggregated in order to obtain an interval for each month and city. The resulting dataset \mathbb{V} has $m = 52$ objects, corresponding as previously said to the 50 provinces and 2 autonomous cities in Spain. Each object $z_i \in \mathbb{V}$ is defined by $n = 12$ variables, one for each month of the year. The value of $z_i^k = [\underline{z}_i^k, \overline{z}_i^k]$ where \underline{z}_i^k is the minimum temperature and \overline{z}_i^k is the maximum temperature. As explained before, for the data to work with any of the similarity measures the intervals must be in the range $[0, 1]$, therefore the transformation $\mathbb{V}^{[0,1]}$ of the dataset must be obtained. The resulting dataset is shown in Table 3. The provinces are numerated in alphabetical order and the location of these provinces in the map is shown in Fig. 2.

A graphical distribution of this data is shown in Fig. 1. Here the intervals of all the objects are shown divided by objects i.e. provinces. Therefore, each box represents a different province containing 12 lines, one representing the interval for each month. Similar boxes indicate a similar range of temperatures in the provinces represented.

6.2. Design of the experiments

The aim of this application is to compare the results obtained by the hierarchical clustering algorithm using a classic approach for objects defined by real variables in relation to the results obtained when the proposed similarity measures are used for interval-valued data where each object is an IVFS. Therefore, the hierarchical clustering algorithm is applied in three different problems:

- Create groups based on the minimum temperature of each month in each province.
- Create groups based on the maximum temperature of each month in each province.
- Create groups based on the interval of minimum and maximum temperatures each month.

For the first and second problems, the Euclidean distance using the linkage methods single, complete and average are compared. The best linkage method is then used for the interval problem. In the interval problem the clustering is done several times, each of them using a different similarity measure of the ones proposed in this work.

6.3. Validation of the clusters

Despite its popularity, validation for clustering algorithms remains a difficult problem, making the evaluation of the results obtained very complicated. This is due to the lack of knowledge about the real groups of the data. When the real grouping is unknown, internal clustering validation metrics are applied, as they only use information regarding the characteristics of the clusters obtained in order to get quantitative results that makes the comparison between different partitions possible.

The *Dunn index* [13] is an internal clustering validation measure that is based on the distance between each of the objects in the clusters in relation to the objects in other clusters. To this aim, the minimum pairwise distance of two objects in two different clusters is taken as the inter-cluster separation (*min.separation*). Then, for each cluster, the distance between each pair of objects in the same cluster is considered in order to obtain the maximal intra-cluster distance (*max.diameter*) as a measure of compactness. Using this the Dunn Index is computed as follows:

² Iberian climate atlas: https://www.aemet.es/es/conocermas/recursos_en_linea/publicaciones_y_estudios/publicaciones/detalles/Atlas-climatologico

Table 3

Interval-valued data created from the AEMET data of years 2021 with the minimum and maximum temperatures of each province selecting the closest station to the province capital.

		January	February	March	April	May	June	July	August	September	October	November	December
1	Álava	[0.21, 0.57]	[0.32, 0.60]	[0.25, 0.66]	[0.28, 0.64]	[0.33, 0.75]	[0.39, 0.81]	[0.40, 0.86]	[0.41, 0.86]	[0.42, 0.82]	[0.29, 0.68]	[0.30, 0.56]	[0.30, 0.59]
2	Albacete	[0.14, 0.62]	[0.32, 0.64]	[0.30, 0.65]	[0.33, 0.66]	[0.42, 0.78]	[0.51, 0.83]	[0.51, 0.88]	[0.52, 0.95]	[0.49, 0.80]	[0.38, 0.73]	[0.31, 0.63]	[0.29, 0.62]
3	Alicante	[0.33, 0.76]	[0.40, 0.68]	[0.40, 0.69]	[0.41, 0.75]	[0.50, 0.85]	[0.56, 0.79]	[0.60, 0.89]	[0.61, 0.90]	[0.56, 0.82]	[0.49, 0.76]	[0.41, 0.72]	[0.39, 0.68]
4	Almería	[0.30, 0.72]	[0.35, 0.69]	[0.36, 0.67]	[0.40, 0.68]	[0.42, 0.81]	[0.51, 0.83]	[0.56, 0.92]	[0.55, 0.92]	[0.51, 0.78]	[0.45, 0.74]	[0.32, 0.71]	[0.35, 0.73]
5	Asturias	[0.28, 0.59]	[0.35, 0.64]	[0.32, 0.73]	[0.33, 0.63]	[0.38, 0.68]	[0.45, 0.72]	[0.49, 0.74]	[0.49, 0.73]	[0.46, 0.79]	[0.41, 0.72]	[0.35, 0.57]	[0.32, 0.63]
6	Ávila	[0.15, 0.60]	[0.28, 0.57]	[0.25, 0.62]	[0.29, 0.60]	[0.35, 0.74]	[0.42, 0.78]	[0.44, 0.87]	[0.46, 0.89]	[0.41, 0.75]	[0.33, 0.68]	[0.28, 0.53]	[0.28, 0.61]
7	Badajoz	[0.24, 0.58]	[0.33, 0.62]	[0.30, 0.77]	[0.37, 0.72]	[0.36, 0.86]	[0.43, 0.87]	[0.49, 0.93]	[0.50, 0.96]	[0.46, 0.88]	[0.40, 0.77]	[0.31, 0.64]	[0.30, 0.61]
8	Barcelona	[0.34, 0.64]	[0.43, 0.61]	[0.40, 0.59]	[0.43, 0.62]	[0.47, 0.69]	[0.57, 0.76]	[0.58, 0.75]	[0.59, 0.79]	[0.55, 0.74]	[0.50, 0.68]	[0.39, 0.65]	[0.38, 0.63]
9	Burgos	[0.09, 0.56]	[0.28, 0.57]	[0.24, 0.68]	[0.26, 0.63]	[0.35, 0.79]	[0.39, 0.81]	[0.40, 0.86]	[0.39, 0.89]	[0.37, 0.79]	[0.28, 0.69]	[0.24, 0.53]	[0.24, 0.55]
10	Cáceres	[0.24, 0.58]	[0.35, 0.60]	[0.32, 0.69]	[0.38, 0.68]	[0.40, 0.81]	[0.48, 0.84]	[0.51, 0.90]	[0.53, 0.96]	[0.49, 0.84]	[0.42, 0.74]	[0.34, 0.61]	[0.35, 0.61]
11	Cádiz	[0.37, 0.59]	[0.47, 0.62]	[0.44, 0.66]	[0.51, 0.67]	[0.51, 0.74]	[0.57, 0.77]	[0.59, 0.79]	[0.60, 0.86]	[0.59, 0.79]	[0.53, 0.74]	[0.42, 0.66]	[0.43, 0.63]
12	Cantabria	[0.35, 0.60]	[0.41, 0.64]	[0.39, 0.74]	[0.39, 0.70]	[0.42, 0.65]	[0.48, 0.68]	[0.52, 0.66]	[0.52, 0.70]	[0.51, 0.81]	[0.46, 0.71]	[0.39, 0.59]	[0.38, 0.63]
13	Castellón	[0.32, 0.73]	[0.42, 0.68]	[0.38, 0.64]	[0.39, 0.66]	[0.49, 0.81]	[0.56, 0.82]	[0.57, 0.80]	[0.58, 0.87]	[0.55, 0.78]	[0.49, 0.77]	[0.40, 0.71]	[0.40, 0.68]
14	Ceuta	[0.39, 0.65]	[0.45, 0.62]	[0.42, 0.65]	[0.50, 0.67]	[0.51, 0.78]	[0.54, 0.79]	[0.58, 0.83]	[0.60, 0.89]	[0.58, 0.80]	[0.54, 0.74]	[0.44, 0.69]	[0.46, 0.62]
15	Ciudad Real	[0.22, 0.62]	[0.31, 0.63]	[0.31, 0.71]	[0.37, 0.67]	[0.42, 0.82]	[0.49, 0.85]	[0.54, 0.92]	[0.55, 0.95]	[0.49, 0.82]	[0.39, 0.73]	[0.31, 0.60]	[0.31, 0.56]
16	Córdoba	[0.28, 0.64]	[0.34, 0.63]	[0.31, 0.71]	[0.39, 0.69]	[0.38, 0.78]	[0.46, 0.84]	[0.50, 0.93]	[0.51, 0.98]	[0.49, 0.83]	[0.44, 0.74]	[0.29, 0.61]	[0.33, 0.65]
17	Cuenca	[0.15, 0.57]	[0.31, 0.61]	[0.29, 0.64]	[0.30, 0.65]	[0.38, 0.75]	[0.45, 0.81]	[0.47, 0.89]	[0.48, 0.93]	[0.42, 0.79]	[0.34, 0.71]	[0.29, 0.59]	[0.27, 0.64]
18	Gerona	[0.25, 0.61]	[0.32, 0.63]	[0.29, 0.64]	[0.31, 0.67]	[0.39, 0.78]	[0.50, 0.84]	[0.51, 0.90]	[0.52, 0.87]	[0.48, 0.80]	[0.40, 0.71]	[0.29, 0.64]	[0.29, 0.63]
19	Granada	[0.22, 0.65]	[0.29, 0.62]	[0.25, 0.69]	[0.33, 0.71]	[0.36, 0.80]	[0.45, 0.85]	[0.48, 0.93]	[0.51, 0.97]	[0.43, 0.86]	[0.37, 0.77]	[0.26, 0.62]	[0.27, 0.65]
20	Guadalajara	[0.15, 0.59]	[0.34, 0.62]	[0.28, 0.68]	[0.33, 0.67]	[0.39, 0.78]	[0.47, 0.82]	[0.48, 0.89]	[0.48, 0.92]	[0.45, 0.80]	[0.36, 0.71]	[0.32, 0.61]	[0.30, 0.59]
21	Guipuzcoa	[0.31, 0.57]	[0.40, 0.65]	[0.35, 0.71]	[0.36, 0.69]	[0.40, 0.76]	[0.47, 0.77]	[0.49, 0.72]	[0.50, 0.71]	[0.49, 0.80]	[0.43, 0.70]	[0.37, 0.58]	[0.36, 0.64]
22	Huelva	[0.30, 0.64]	[0.38, 0.64]	[0.38, 0.74]	[0.46, 0.71]	[0.43, 0.82]	[0.51, 0.85]	[0.53, 0.89]	[0.55, 0.93]	[0.52, 0.83]	[0.48, 0.79]	[0.36, 0.70]	[0.36, 0.66]
23	Huesca	[0.16, 0.54]	[0.28, 0.57]	[0.27, 0.64]	[0.28, 0.65]	[0.31, 0.69]	[0.39, 0.79]	[0.42, 0.85]	[0.42, 0.87]	[0.39, 0.76]	[0.31, 0.66]	[0.28, 0.57]	[0.25, 0.57]
24	Islas Baleares	[0.36, 0.67]	[0.42, 0.64]	[0.41, 0.67]	[0.42, 0.70]	[0.51, 0.76]	[0.57, 0.83]	[0.62, 0.81]	[0.63, 0.86]	[0.60, 0.81]	[0.52, 0.73]	[0.44, 0.69]	[0.41, 0.66]
25	Jaén	[0.28, 0.61]	[0.38, 0.65]	[0.34, 0.73]	[0.43, 0.70]	[0.41, 0.79]	[0.51, 0.85]	[0.57, 0.92]	[0.56, 0.97]	[0.53, 0.86]	[0.46, 0.74]	[0.32, 0.61]	[0.35, 0.63]
26	La Coruña	[0.35, 0.57]	[0.41, 0.64]	[0.40, 0.72]	[0.40, 0.66]	[0.41, 0.67]	[0.47, 0.70]	[0.52, 0.76]	[0.53, 0.73]	[0.49, 0.81]	[0.45, 0.71]	[0.43, 0.58]	[0.41, 0.63]
27	La Rioja	[0.26, 0.60]	[0.32, 0.61]	[0.30, 0.66]	[0.32, 0.68]	[0.35, 0.79]	[0.46, 0.84]	[0.47, 0.90]	[0.47, 0.92]	[0.45, 0.78]	[0.32, 0.69]	[0.28, 0.57]	[0.32, 0.58]
28	Las Palmas	[0.52, 0.70]	[0.49, 0.68]	[0.55, 0.71]	[0.57, 0.69]	[0.58, 0.68]	[0.59, 0.74]	[0.60, 0.74]	[0.61, 0.75]	[0.62, 0.72]	[0.60, 0.72]	[0.55, 0.68]	[0.54, 0.70]
29	León	[0.19, 0.56]	[0.27, 0.55]	[0.25, 0.64]	[0.28, 0.63]	[0.33, 0.75]	[0.39, 0.77]	[0.42, 0.81]	[0.41, 0.84]	[0.40, 0.77]	[0.35, 0.68]	[0.28, 0.58]	[0.28, 0.58]
30	Lleida	[0.22, 0.62]	[0.33, 0.63]	[0.30, 0.68]	[0.32, 0.70]	[0.42, 0.81]	[0.49, 0.88]	[0.51, 0.90]	[0.51, 0.92]	[0.48, 0.82]	[0.37, 0.74]	[0.29, 0.63]	[0.29, 0.61]
31	Lugo	[0.24, 0.55]	[0.29, 0.62]	[0.27, 0.73]	[0.29, 0.65]	[0.30, 0.76]	[0.35, 0.77]	[0.41, 0.79]	[0.43, 0.78]	[0.41, 0.81]	[0.35, 0.71]	[0.29, 0.59]	[0.26, 0.62]
32	Madrid	[0.20, 0.58]	[0.38, 0.58]	[0.32, 0.66]	[0.36, 0.65]	[0.41, 0.77]	[0.50, 0.81]	[0.51, 0.88]	[0.51, 0.92]	[0.47, 0.78]	[0.42, 0.69]	[0.33, 0.58]	[0.33, 0.56]
33	Málaga	[0.35, 0.72]	[0.42, 0.67]	[0.39, 0.70]	[0.48, 0.74]	[0.49, 0.86]	[0.53, 0.85]	[0.60, 0.95]	[0.60, 0.87]	[0.55, 0.86]	[0.51, 0.76]	[0.40, 0.74]	[0.41, 0.68]
34	Melilla	[0.40, 0.64]	[0.44, 0.64]	[0.43, 0.64]	[0.48, 0.69]	[0.51, 0.79]	[0.56, 0.76]	[0.61, 0.86]	[0.59, 0.90]	[0.58, 0.78]	[0.54, 0.72]	[0.45, 0.69]	[0.44, 0.64]
35	Murcia	[0.30, 0.74]	[0.40, 0.69]	[0.35, 0.70]	[0.42, 0.74]	[0.48, 0.85]	[0.54, 0.85]	[0.60, 0.97]	[0.62, 1.00]	[0.55, 0.85]	[0.49, 0.78]	[0.36, 0.72]	[0.37, 0.70]
36	Navarra	[0.24, 0.58]	[0.30, 0.62]	[0.30, 0.68]	[0.32, 0.66]	[0.37, 0.78]	[0.44, 0.81]	[0.47, 0.88]	[0.47, 0.91]	[0.46, 0.83]	[0.36, 0.70]	[0.31, 0.56]	[0.30, 0.57]
37	Orense	[0.23, 0.60]	[0.31, 0.63]	[0.29, 0.75]	[0.34, 0.72]	[0.34, 0.80]	[0.40, 0.83]	[0.46, 0.88]	[0.46, 0.87]	[0.44, 0.86]	[0.39, 0.76]	[0.32, 0.63]	[0.30, 0.63]
38	Palencia	[0.18, 0.53]	[0.29, 0.55]	[0.26, 0.66]	[0.27, 0.63]	[0.33, 0.76]	[0.39, 0.77]	[0.39, 0.83]	[0.42, 0.87]	[0.41, 0.81]	[0.30, 0.67]	[0.27, 0.51]	[0.27, 0.53]
39	Pontevedra	[0.29, 0.53]	[0.34, 0.63]	[0.35, 0.74]	[0.38, 0.68]	[0.37, 0.70]	[0.42, 0.79]	[0.47, 0.83]	[0.49, 0.82]	[0.48, 0.75]	[0.43, 0.73]	[0.34, 0.62]	[0.39, 0.63]
40	Salamanca	[0.19, 0.60]	[0.31, 0.61]	[0.29, 0.72]	[0.32, 0.65]	[0.37, 0.77]	[0.45, 0.80]	[0.47, 0.85]	[0.48, 0.90]	[0.43, 0.82]	[0.35, 0.71]	[0.30, 0.56]	[0.29, 0.62]
41	Santa Cruz de Tenerife	[0.51, 0.69]	[0.49, 0.67]	[0.52, 0.75]	[0.55, 0.71]	[0.56, 0.77]	[0.57, 0.77]	[0.60, 0.79]	[0.62, 0.85]	[0.61, 0.77]	[0.58, 0.75]	[0.56, 0.71]	[0.52, 0.71]
42	Segovia	[0.19, 0.57]	[0.32, 0.57]	[0.27, 0.63]	[0.27, 0.63]	[0.36, 0.73]	[0.42, 0.78]	[0.43, 0.87]	[0.44, 0.89]	[0.37, 0.77]	[0.36, 0.68]	[0.28, 0.52]	[0.29, 0.65]
43	Sevilla	[0.29, 0.65]	[0.36, 0.66]	[0.33, 0.76]	[0.44, 0.74]	[0.41, 0.83]	[0.50, 0.89]	[0.55, 0.96]	[0.54, 0.99]	[0.53, 0.90]	[0.47, 0.81]	[0.32, 0.67]	[0.34, 0.68]
44	Soria	[0.15, 0.55]	[0.28, 0.57]	[0.25, 0.64]	[0.28, 0.63]	[0.32, 0.71]	[0.40, 0.76]	[0.43, 0.85]	[0.41, 0.89]	[0.40, 0.76]	[0.29, 0.68]	[0.27, 0.59]	[0.26, 0.62]
45	Tarragona	[0.32, 0.73]	[0.39, 0.67]	[0.38, 0.68]	[0.39, 0.73]	[0.46, 0.81]	[0.52, 0.87]	[0.57, 0.89]	[0.56, 0.91]	[0.53, 0.81]	[0.43, 0.74]	[0.38, 0.64]	[0.35, 0.68]
46	Teruel	[0.00, 0.63]	[0.28, 0.64]	[0.26, 0.66]	[0.29, 0.66]	[0.37, 0.75]	[0.43, 0.81]	[0.44, 0.86]	[0.45, 0.93]	[0.40, 0.78]	[0.31, 0.72]	[0.26, 0.60]	[0.23, 0.63]
47	Toledo	[0.11, 0.59]	[0.33, 0.63]	[0.31, 0.69]	[0.35, 0.69]	[0.41, 0.82]	[0.47, 0.86]	[0.49, 0.94]	[0.52, 0.97]	[0.47, 0.83]	[0.39, 0.72]	[0.32, 0.61]	[0.31, 0.60]
48	Valencia	[0.32, 0.70]	[0.43, 0.67]	[0.39, 0.70]	[0.42, 0.69]	[0.51, 0.83]	[0.57, 0.79]	[0.60, 0.90]	[0.61, 0.89]	[0.55, 0.79]	[0.51, 0.77]	[0.40, 0.73]	[0.41, 0.67]

Table 3 (continued)

		January	February	March	April	May	June	July	August	September	October	November	December
49	Valladolid	[0.21, 0.56]	[0.30, 0.59]	[0.28, 0.70]	[0.30, 0.66]	[0.37, 0.79]	[0.44, 0.81]	[0.46, 0.86]	[0.47, 0.90]	[0.44, 0.82]	[0.34, 0.71]	[0.29, 0.55]	[0.29, 0.55]
50	Vizcaya	[0.29, 0.61]	[0.38, 0.64]	[0.32, 0.74]	[0.33, 0.70]	[0.38, 0.77]	[0.44, 0.82]	[0.47, 0.83]	[0.49, 0.74]	[0.48, 0.89]	[0.40, 0.75]	[0.37, 0.62]	[0.33, 0.69]
51	Zamora	[0.21, 0.59]	[0.31, 0.58]	[0.29, 0.71]	[0.30, 0.66]	[0.37, 0.80]	[0.45, 0.82]	[0.46, 0.87]	[0.47, 0.89]	[0.42, 0.81]	[0.35, 0.69]	[0.30, 0.57]	[0.27, 0.56]
52	Zaragoza	[0.08, 0.60]	[0.32, 0.66]	[0.26, 0.67]	[0.31, 0.65]	[0.40, 0.78]	[0.44, 0.85]	[0.46, 0.90]	[0.47, 0.93]	[0.41, 0.81]	[0.31, 0.73]	[0.26, 0.58]	[0.24, 0.57]

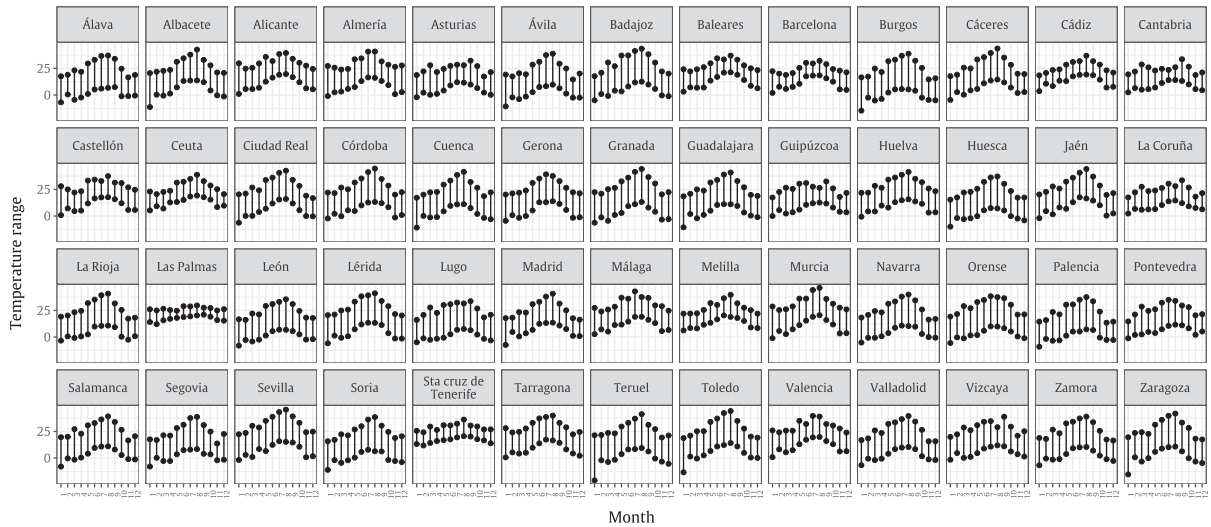


Fig. 1. Interval distribution of temperatures of the 52 provinces of Spain by month.

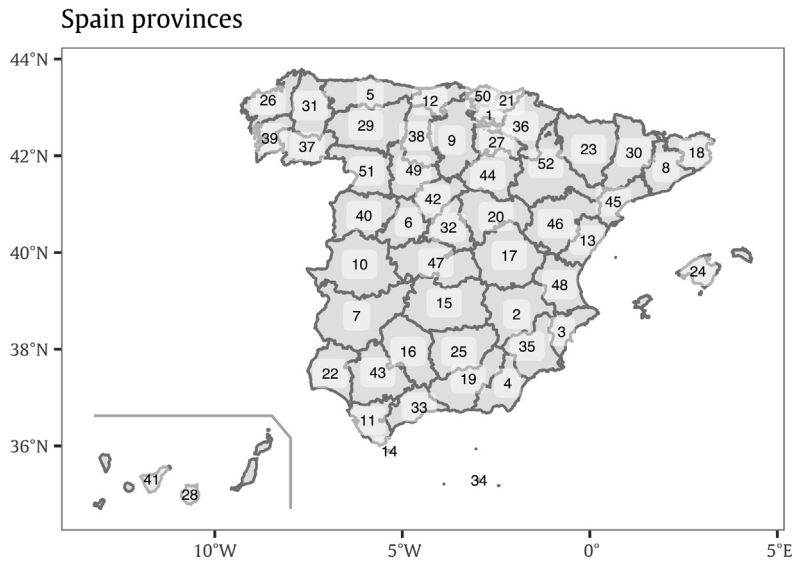


Fig. 2. Provinces of Spain, numbered in alphabetical order.

$$\text{DunnIndex} = \frac{\min.\text{separation}}{\max.\text{diameter}}.$$

If the clusters obtained are compact and a well-separated partition, then the diameter of the clusters is expected to be small and the distance between the clusters should be large. Therefore, the larger the value obtained for the Dunn Index, the better.

Note that, in order to apply this index it is necessary, in the classical sense to measure the distances. In this work we applied the index using the *dissimilarities* from the similarities that matches how the clusters were created. The similarities proposed in this work are thus transformed to dissimilarities measures [30] such that the dissimilarity between two objects is considered equal to 1 minus the similarity between the objects.

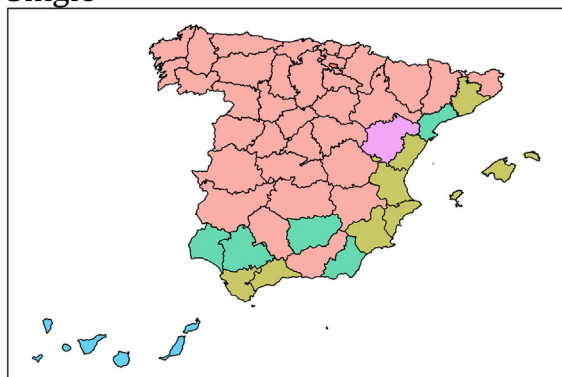
6.4. Results

The results obtained for the dataset, experiments and validation described in the previous sections are shown below. First of all, Fig. 3 presents the results obtained applying the linkage methods only to the minimum temperature (left endpoints of

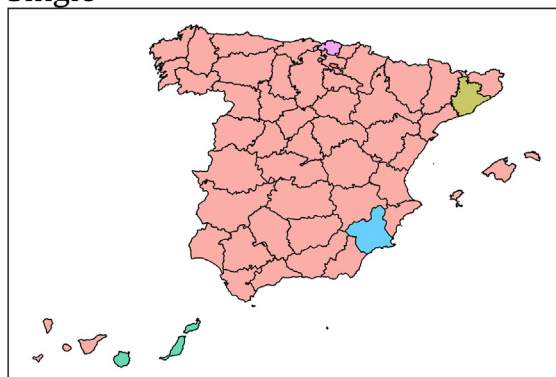
Minimum temperature

Maximum temperature

Single



Single



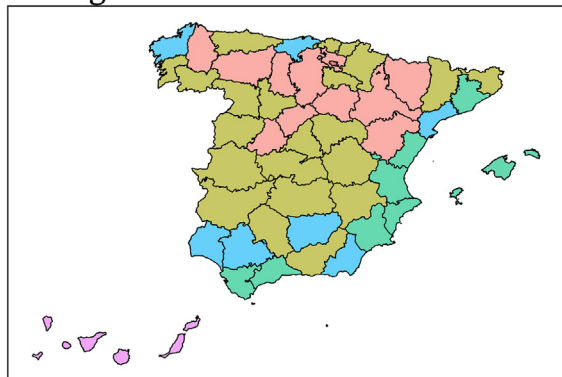
Complete



Complete



Average



Average

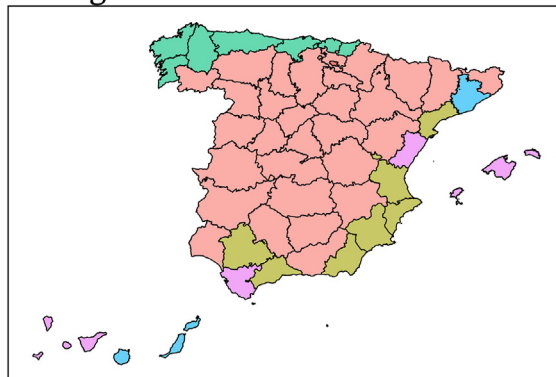


Fig. 3. Clustering results obtained with the single, complete and average methods for the minimum (left) and maximum (right) temperatures of the weather in Spain 2021 by month.

the intervals, shown in the left column) and to the maximum temperature (right endpoints of the intervals, shown in the right column). This means that the hierarchical clustering algorithm is applied to real data. In this case, we can observe that the linkage method visibly performs worse than the other ones.

The values of the Dunn Index obtained for these partitions are given in [Table 4](#). It can be observed that the partition that leads to the best value of the Dunn Index for both problems is that created with the average linkage method. For this reason, this linkage method is selected and used to explore the results obtained with the similarities based on embedding functions proposed in this work.

Table 4
Results obtained using the hierarchical clustering algorithm using non interval valued data and the Euclidean distance to measure the similarity between the clusters. Two different problems are considered: group the provinces based on its 1) minimum temperature and 2) maximum temperature.

	Minimum	Maximum
Single	0.2261	0.1009
Complete	0.1464	0.1973
Average	0.2563	0.2444

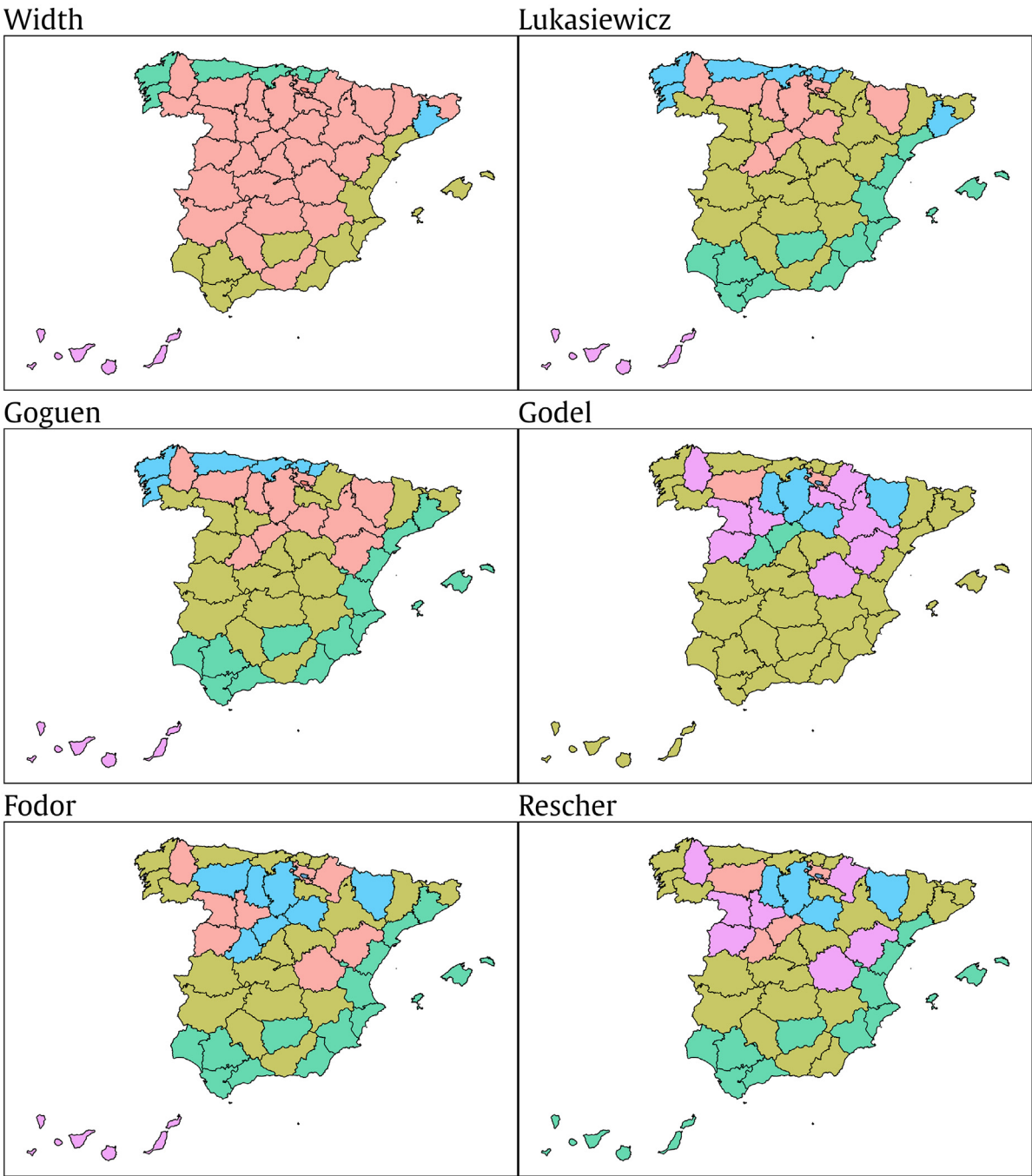


Fig. 4. Clustering results obtained with the average using the different similarity measures proposed in order to group the provinces attending to the weather considering the interval temperatures of each province in Spain 2021 by month.

Table 5

Results obtained using the hierarchical clustering algorithm using interval valued data and different similarities.

	Width	Lukasiewicz	Goguen	Godel	Fodor	Rescher
Average	0.3086	0.3855	0.4505	0.7536	0.6636	0.5652

The results obtained applying the linkage method with all the similarities proposed based on embeddings following Algorithm 1 in order to cluster the provinces using the intervals of the temperatures for each month are now discussed. The maps obtained for the partitions created using different embedding functions are shown in Fig. 4. The metric of the Dunn index for these is shown in Table 5. Notice how this metric shows that the clusters are better shaped in relation to the ones obtained considering real data.

In contrast to the results obtained using hierarchical clustering, most of these maps show how the algorithm using interval-valued data and the metrics proposed works better at identifying the Mediterranean coast. The fact that these similarity measures take into account both extremes of the interval instead of only one of the values helps to this aim. To comment a few, the width similarity function, taking into account the overlapping of two intervals, benefits the creation of clusters between those provinces that have a similar shape in Fig. 1. With this it identifies all the coasts of the peninsula and different weather in the Canary Islands, which ranges are very different. Lukasiewicz is also able to do this as it bases its similarity in the mean between the difference of the maximum temperatures and minimum temperatures. The Rescher function creates more exotic groups. Recall that this measure was more strict, benefiting those intervals that are completely embedded into another.

7. Discussion

In the last years, the use of similarity functions has been increased in order to compare intervals, being Jaccard and Dice similarities the most common measures used in the literature. However, in some circumstances these measures could be helpless [23,45]. In this way, we propose the use of averaging embeddings to define similarity measures.

In this paper, we have analyzed the behavior of the proposed similarities in SubSection 3.1. In case there is no intersection between the intervals, the value of the similarity is 0 while when one interval is completely included in the other one, the similarity value is at least 0.5. Depending on the context, we should choose the embedding used to define the similarity based on its strengths and limitations. For instance, the similarity based on the Rescher embedding has important limitations which have been studied and described in the previous section.

The functions reviewed from the theoretical point of view are then used to measure the similarities between objects in an algorithm of hierarchical clustering, which can be used on data having each variable defined by intervals. The methodology proposed in Algorithm 1 does not variate the complexity of the classic hierarchical clustering algorithm, as the only part modified regards the computing of the similarity measures between the objects of the dataset, and this also must be computed using any other measure.

The clustering algorithm proposed is applied to real data of the Spanish weather, having for each province and month the range of minimum and maximum temperatures. The results show that the use of averaging functions are useful for identifying more precise information in comparison with the traditional measures that only take one number of the interval as representative. Of course, among the similarities based on averaging embeddings, appear also different behaviors. For these reasons, in the case these would be applied to different datasets, they must be compared using validation measures in order to select the one that yields to the best partitions, as has been done in this paper using the Dunn Index.

8. Conclusion

In this work we point out that interval-valued data could be noticed as interval-valued fuzzy sets. This point of view allows the comparison of the objects of an interval-valued dataset treating them as IVFS. Keeping this idea in mind, we propose similarity measures for intervals based on average embeddings that allow us to define new similarity measures for IVFS. With them, it is possible to build similarity matrices for objects of interval-valued datasets. This similarity matrix allows the use of the hierarchical clustering algorithm for data defined using interval-valued data, where each object is defined as a set of interval-valued variables. As following the hierarchical clustering algorithm the clusters are created based on similarity measures, the measures proposed can be applied in order to obtain clusters of IVFS. This method is applied to a real problem in order to group the provinces of Spain based on their range temperature. The results show that better results are reached when considering the interval data than when considering the extremes by themselves, as more information can be used in order to create the clusters in the first case.

In the future, we would like to develop a more general approach, involving an arbitrary aggregation function, and then study the properties that aggregation functions must satisfy in order to obtain new similarity measures.

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.

References

- [1] E. Ammar, J. Metz, On fuzzy convexity and parametric fuzzy optimization, *Fuzzy Sets Syst.* 49 (2) (1992) 135–141.
- [2] M.C.N. Barioni, H. Razente, A.M.R. Marcelino, A.J.M. Traina, C. Traina Jr., Open issues for partitioning clustering methods: an overview, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4 (3) (2014) 161–177.
- [3] A. Bouchet, M. Sesma-Sara, G. Ochoa, H. Bustince, S. Montes, and I. Díaz, Measures of embedding for interval-valued fuzzy sets. In revision process.
- [4] H. Bustince, P. Burillo, Mathematical analysis of interval-valued fuzzy relations: Application to approximate reasoning, *Fuzzy Sets Syst.* 113 (2) (2000) 205–219.
- [5] H. Bustince, C. Marco-Detchart, J. Fernández, C. Wagner, J.M. Garibaldi, Z. Takáč, Similarity between interval-valued fuzzy sets taking into account the width of the intervals and admissible orders, *Fuzzy Sets Syst.* 390 (2020) 23–47.
- [6] C.C. Chang, P.W. Lu, and J.Y. Hsiao, A hybrid method for estimating the euclidean distance between two vectors. In *First International Symposium on Cyber Worlds, 2002. Proceedings*, pp. 183–190, 2002.
- [7] M. Chavent, F.A.T. de Carvalho, Y. Lechevallier, R. Verde, New clustering methods for interval data, *Computat. Stat.* 21 (2) (2006) 211–229.
- [8] C. Cornelis, G. Deschrijver, E.E. Kerre, Implication in intuitionistic fuzzy and interval-valued fuzzy set theory, *Int. J. Approx. Reason.* 35 (2004) 55–95.
- [9] I. Couso, H. Bustince, From fuzzy sets to interval-valued and atanassov intuitionistic fuzzy sets: A unified view of different axiomatic measures, *IEEE Trans. Fuzzy Syst.* 27 (2) (2019) 362–371.
- [10] F.A.T. de Carvalho, Fuzzy c-means clustering methods for symbolic interval data, *Pattern Recogn. Lett.* 28 (4) (2007) 423–437.
- [11] R.M.C.R. de Souza, F.A.T. de Carvalho, Clustering of interval data based on city-block distances, *Pattern Recogn. Lett.* 25 (3) (2004) 353–365.
- [12] L.R. Dice, Measures of the amount of ecologic association between species, *Ecology* 26 (3) (1945) 297–302.
- [13] J.C. Dunn, A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters, *J. Cybern.* 3 (3) (1973) 32–57.
- [14] S. Galdino, P. Maciel, Hierarchical cluster analysis of interval-valued data using width of range euclidean distance, in: *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2019, pp. 1–6.
- [15] A.A. Goshtasby, *Similarity and Dissimilarity Measures*, Springer, London, London, 2012, pp. 7–66.
- [16] I. Grattan-Guinness, Fuzzy membership mapped onto intervals and many-valued quantities, *Math. Logic Q.* 22 (1) (1976) 149–160.
- [17] Y.-Y. Guh, M.-S. Yang, R.-W. Po, E.S. Lee, Interval-valued fuzzy relation-based clustering with its application to performance evaluation, *Comput. Math. Appl.* 57 (5) (2009) 841–849.
- [18] D.S. Guru, B.B. Kiranagi, P. Nagabhushan, Multivalued type proximity measure and concept of mutual similarity value useful for clustering symbolic patterns, *Pattern Recogn. Lett.* 25 (10) (2004) 1203–1213.
- [19] P. Jaccard, Nouvelles recherches sur la distribution florale, *Bull. Soc. Vaud. Sci. Nat.* 44 (1908) 223–270.
- [20] K.U. Jahn, Intervall-wertige mengen, *Mathematische Nachrichten* 68 (1975) 115–132.
- [21] Q. Jiang, J. Xin, S.-J. Lee, S. Yao, A new similarity/distance measure between intuitionistic fuzzy sets based on the transformed isosceles triangles and its applications to pattern recognition, *Expert Syst. Appl.* 116 (2018) 08.
- [22] S.C. Johnson, Hierarchical clustering schemes, *Psychometrika* 32 (3) (1967) 241–254.
- [23] S. Kabir, C. Wagner, T. Craig Havens, D.T. Anderson, A similarity measure based on bidirectional subethood for intervals, *IEEE Trans. Fuzzy Syst.* 28 (11) (2020) 2890–2904.
- [24] K.K. Lai, S.Y. Wang, J.P. Xu, S.S. Zhu, Y. Fang, A class of linear interval programming problems and its application to portfolio selection, *IEEE Trans. Fuzzy Syst.* 10 (6) (2002) 698–704.
- [25] P. Lingras, C. West, Interval set clustering of web users with rough k-means, *J. Intell. Inform. Syst.* 23 (1) (2004) 5–16.
- [26] R. Lustig, Angle-average for the powers of the distance between two separated vectors, *Mol. Phys.* 65 (1) (1988) 175–179.
- [27] H.B. Mitchell, On the dengfeng–chuntian similarity measure and its application to pattern recognition, *Pattern Recogn. Lett.* 24 (16) (2003) 3101–3104.
- [28] F. Murtagh, A survey of recent advances in hierarchical clustering algorithms, *Comput. J.* 26 (4) (1983) 354–359.
- [29] I. Olkin, F. Pukelsheim, The distance between two random vectors with given dispersion matrices, *Linear Algebra Appl.* 48 (1982) 257–263.
- [30] E. Pekalska, P. Paclik, R.P.W. Duin, A generalized kernel approach to dissimilarity-based classification, *J. Mach. Learn. Res.* 2(Dec):175–211, 2001.
- [31] S. Raha, N.R. Pal, K.S. Ray, Similarity-based approximate reasoning: methodology and application, *IEEE Trans. Syst., Man, Cybern.-Part A: Syst. Humans* 32 (4) (2002) 541–547.
- [32] A.B. Ramos-Guajardo, A hierarchical clustering method for random intervals based on a similarity measure, *Comput. Stat.* (2021) 1–33.
- [33] Y. Ren, Y. Liu, J. Rong, R. Dew, Clustering interval-valued data using an overlapped interval divergence, *Proceedings of the Eighth Australasian Data Mining Conference-Volume 101* (2009) 35–42.
- [34] L.R. Rokach, O. Maimon, Clustering methods, in: *Data mining and knowledge discovery handbook*, Springer, 2005, pp. 321–352.
- [35] R. Sambuc, *Function phi-floous, application a l'aide au diagnostic en pathologie thyroïdienne*, These de Doctorat en Medicine (1975).
- [36] I.B. Turksen, Fuzzy sets and systems and its applications in production research, in: *Toward the Factory of the Future*, Springer, 1985, pp. 649–656.
- [37] S. Vilar, R. Harpaz, H.S. Chase, S. Costanzi, R. Rabadan, and C. Friedman, Facilitating adverse drug event detection in pharmacovigilance databases using molecular structure similarity: application to rhabdomyolysis, *J. Am. Med. Inform. Assoc.* 18(Supplement_1):i73–i80, 2011.
- [38] T. Vo-Van, L. Ngoc, T. Nguyen-Trang, An efficient robust automatic clustering algorithm for interval data, in: *Communications in Statistics-Simulation and Computation*, 2021, pp. 1–15.
- [39] C. Wagner, S. Miller, J.M. Garibaldi, D.T. Anderson, T.C. Havens, From interval-valued data to general type-2 fuzzy sets, *IEEE Trans. Fuzzy Syst.* 23 (2) (2014) 248–269.
- [40] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, *Ann. Data Sci.* 2 (2) (2015) 165–193.
- [41] Z. Xu, Some similarity measures of intuitionistic fuzzy sets and their applications to multiple attribute decision making, *Fuzzy Optimiz. Decision Making* 6 (2) (2007) 109–121.
- [42] L. Xuecheng, Entropy, distance measure and similarity measure of fuzzy sets and their relations, *Fuzzy Sets Syst.* 52 (3) (1992) 305–318.
- [43] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning–I, *Inform. Sci.* 8 (3) (1975) 199–249.
- [44] W. Zeng and Q. Yin, Similarity measure of interval-valued fuzzy sets and application to pattern recognition. In *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 1, pp. 535–539, IEEE, 2008.
- [45] P. Huidobro, N. Rico, A. Bouchet, S. Montes, I. Díaz, A New Similarity Measure for Real Intervals to Solve the Aliasing Problem, in: *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Springer, 2022.