NSS
Neutrosophic Sets and Systems, Vol. 55, 2023
University of New Mexico

Neutrosophic Logic-based DIANA Clustering algorithm

Azeddine Elhassouny[1,*]
[1]Mohammed V University in Rabat; azeddine.elhassouny@ensias.um5.ac.ma
*Correspondence: elhassounyphd@gmail.com;

Abstract. On the one hand, the most extensively used Hierarchical Clustering techniques are the Hierarchical Divisive Clustering (HDC) algorithms such as DIANA. Its primary goal is to build the tree of Hierarchical Agglomerative Clustering (HAC) in reverse order. On the other hand, Neutrosophy is an extension of fuzzy logic and serves as a model of uncertainty. In addition to the truth (T) and falsity (F) elements of fuzzy logic, single-valued Neutrosophic sets (SVNs) logic estimates the proportion of indeterminacy (I) for a given proposition. In this work, we propose a Neutrosophic Logic-based DIANA Clustering algorithm. Indeterminacy is added to the DIANA hierarchical clustering algorithm using single-valued Neutrosophic sets (SVNs). The suggested algorithm is named Neutro-DIANA (Neutrosophic DIANA) and is broken down into numerous steps. The experimental findings show that the suggested technique for dealing with indeterminacy is effective.

Keywords: Hierarchical Divisive Clustering (HDC), DIANA, Neutrosophic, Indeterminacy, Neutro-DIANA.

---

1. Introduction

Clustering is a subsection of unsupervised learning, which is one of the four basic subcategories of machine learning techniques. Clustering, as a learning method, is useful in numerous domains, including market segmentation [1], customer regrouping [2], Big data analysis [3], image processing [4], and so on. Clustering algorithms are classified into four types [5]: (1) K-means and K-medoids for partitioning. (2) Density-based approaches such as DBSCAN and OPTICS. (3) Model-based approaches like SOM and EM. (4) AGNES (AGglomerative NESting) and DIANA (DIvisive ANAlysis) are hierarchical approaches.

The method of organizing data points inside clusters is known as hierarchical clustering. Hierarchical clustering may be done in two ways: agglomerative (bottom-up) and divisive (top-down). In contrast to the Agglomerative method, the Divisive method Hierarchical Clustering starts with a single cluster that contains all entities, then divides the instances into a hierarchy of smaller and smaller clusters until each cluster contains just one entity or a predefined amount of entities.

Numerous hierarchical clustering techniques that use the divisive approach include TWINSPAN (Two-Way Indicator Species Analysis), MONA (divisive hierarchical MONothetic Analysis), and DIANA (DIvisive hierarchical ANAlysis). The most well-known and effective algorithm is DIANA. It is a polythetic divisive method that works with any matrix of dissimilarities. It attempts to combine a collection of data items that are comparable to one another into a single cluster, while dissimilar data objects are connected with other clusters [6].

The DIANA method creates a hierarchy of sub-clusters starting with a single cluster containing all n items. The greatest diameter cluster is split at each stage until there is just one element in each cluster. To achieve this, the algorithm looks for the element in the chosen cluster that differs from other elements on average by the most. The algorithm then reassigns items that are more closely related to the "splinter group" than to the "old group" in succeeding phases once the "splinter group" has been chosen. Two new clusters are the outcome. The greatest distance between items in the two sub-clusters determines the distance between the clusters. The average of all $1-d(i)$, or $d(i)$, is the diameter of the final group including element i divided by the diameter of the whole dataset. This is known as the divisive coefficient (DC).

As previously stated, the DIANA algorithm was designed to cope with crisp numbers, and any data issues should be addressed during the data preparation phase. However, many real-world situations are imprecise and unclear, and their data contains impurities such as imprecision, uncertainty, and so on. As an extension of fuzzy logic, Neutrosophic [7], [8]. is proposed to cope with these information flaws. To achieve this, the Neutrosophic provides a new parameter termed indeterminacy membership (I) in addition to the two values of the fuzzy logic, degree of truth-membership (T) and falsity-membership (F).

In this study, we develop the Neutrosophic set (SVNs)-based Clustering approach to address the shortcoming of fuzzy logic (sets, IFSs, and IVIFSs)-based clustering algorithms, which are unable to capture inconsistent information that corresponds to the real-world data. In a Neutrosophic setting, each element's truth, falsity, and indeterminacy (T, F, I) values are computed to identify whether or not it belongs to any given cluster. Considered to be a neutrosophic component, e is expressed as $e(T, F, I)$. The input data is initially subjected to a neutrosophic real problem formulation. This step's output is sent into the neutrosophic-based DIANA clustering method.

As you can see from this introduction, we get right into the research issue without debating the rationale for the method's selection, the necessity of the hybridization, etc. We would like to let you know that this study is a successor to a paper we wrote on a cutting-edge topic called neutrosophic and machine learning [9]. In that paper, we documented all hybrid machine learning algorithms that used Single-valued Neutrosophic Sets (SVNs) approaches, and we subsequently created a taxonomy of Neutrosophic Machine Learning algorithms. In other words, we have a list of the algorithms that have previously been used, and more details are provided in [9]. In this paper, we will concentrate on the Neutrosophic-based hierarchical clustering approach.

The rest of this paper is structured as follows: Section II delves into the background and preliminaries. The proposed Neutro-DIANA algorithm is explained in detail in Section III, while the experiments and insightful discussion of the findings are provided in Section VI. Lastly, Section VI brings this study to a close and proposes some future research areas.

1.1. *Related works*

The image segmentation technique was enhanced by Qureshi et al. [6] utilizing K-Means Clustering with Neutrosophic Logic. The technique entails converting an image into a neutrosophic collection. The neutrosophic-based k-means approach is used to segment neutrosophic images, and SVNs are used to quantify the indeterminacy in pixels of an image. To tackle the ambiguous and inconsistent information that the fuzzy is unable to handle, Vandhana et al. [10] adopted neutrosophic fuzzy hierarchical clustering. The method is used to analyze and pinpoint regions where illnesses like dengue fever are influenced by environmental and climatic factors. As an extension of the hierarchical clustering method, Sahin [11] presented a single-valued neutrosophic hierarchical clustering technique for clustering SVNSs. The technique was further expanded to categorize interval neutrosophic data. Ye [12] presented the single-valued neutrosophic minimum spanning tree (SVNMST) clustering technique as an extension of the intuitionistic fuzzy minimum spanning tree (IFMST) clustering algorithm. The approach is based on the generalized distance measure of SVNSs. H2D-FCM is a Fuzzy-based divisive hierarchical clustering technique introduced by Bordogna and Pasi [13]. It automatically estimates the number of clusters to produce and then divides the node into sub-clusters using the probabilistic Fuzzy C Means method. In Ding's study [14], Ding et al. addressed the most important topic in Hierarchical clustering algorithms: choosing the appropriate next cluster(s) to divide or merge. They determined that the average similarity approach is the best for divisive clustering and MinMaxmis the best for agglomerative clustering. In another study, Ye [15] introduced clustering algorithms for SVNs using distance-based similarity metrics in another study (Single-Valued Neutrosophic Sets). To meet the aforementioned goals, we present a novel Neutrosophic Hierarchical Divisive Clustering algorithm (n-DIANA), based on a divisive approach.

2. Background

2.1. *Single valued neutrosophic set (SVNS)*

Smarandache's notion of the neutrosophic set [8] is challenging to transfer in a genuine application and engineering challenge. As a result, Wang et al. [16, 17] established the neutrosophic set notions of SVNS (single-valued neutrosophic set) and INS (interval neutrosophic set). To execute the necessary calculus, various mathematical operations in a neutrosophic context, such as euclidean distance, average, minimum maximum, and so on, must be defined.

Definition 2.1. Consider X to be a universe discourse and $A1$ to be a single valued neutrosophic set over X. $A1$ takes the following form:

$$A_1 = \{\langle x, \mu_{A_1}(x), \omega_{A_1}(x), \nu_{A_1}(x)\rangle : x \in X\}. \tag{1}$$

where $\mu_{A_1} : X \to [0,1]$, $\omega_{A_1} : X \to [0,1]$, and $\nu_{A_1} : X \to [0,1]$, with the constraint $0 \leq \mu_{A_1}(x) + \omega_{A_1}(x) + \nu_A(x) \leq 3$, $\forall x \in X$

The values $\mu_{A_1}(x)$, $\omega_{A_1}(x)$, and $\nu_{A_1}(x)$ represent the degree of truth-membership, indeterminacy-membership and falsity-membership of x to X respectively.

Definition 2.2. For below, consider tow SVN measurements $A1$ and $A2$, where $A_1 = \{\langle x, \mu_{A_1}(x), \omega_{A_1}(x), \nu_{A_1}(x)\rangle : x \in X\}$, $A_2 = \{\langle x, \mu_{A_2}(x), \omega_{A_2}(x), \nu_{A_2}(x)\rangle : x \in X\}$

The fundamental arithmetic operations are as follows:

$$A_1 + A_2 = \{\langle x, \mu_{A_1}(x) + \mu_{A_2}(x) - \mu_{A_1}(x)\mu_{A_2}(x), \omega_{A_1}(x)\omega_{A_2}(x), \nu_{A_1}(x)\nu_{A_2}(x)\rangle : x \in X\} \tag{2}$$

$$\lambda A_1 = \{\langle x, 1 - (1 - \mu_{A_1}(x))^{\lambda}, (\omega_{A_1}(x))^{\lambda}, (\nu_{A_1}(x))^{\lambda}\rangle : x \in X \text{ and } \lambda \geq 0\}. \tag{3}$$

2.2. *DIANA (DIvisive ANAlysis)*

DIANA [18–20] is a hierarchical clustering strategy that groups items into multiple clusters, each of which contains elements that are similar to one another. The clustering method DIANA utilized in this study may be summed up as follows:

- Step 1: At first, DIANA assumes that all n observations are contained within a single cluster.
- Step 2: Divide the Clusters again and again until each cluster has just one observation.
  - Choose the pair of clusters with the greatest dissimilarity in the current cluster, which is $\{\zeta_r\}$, and $\{\zeta_s\}$, in which $d(\{\zeta_r\}, \{\zeta_s\}) = \max\{d(\zeta_i, \zeta_j)_{0 \leq i,j \leq n}\}$.
  - The cluster is divided into $(zetas)$ and $(zetar)$ clusters to generate the following clusters.
- Step 3: If all clusters are made up of a single element, break; otherwise, continue to step 2.

In comparison to Agglomerative Hierarchical Clustering, divisions in the DIANA technique are based on average distance and cophenetic distance, which are equivalent to average linkage and full linkage, respectively. The mean distance between the cluster centroid and the other objects is computed by taking the average of the Euclidean distances between the cluster centroid and each item.

Consider $\Theta = \{A_i, i = 1 \dots n\}$ as the space of n observations, and ζ as the cluster's center; Eq. 4 gives the average distance between ζ and other objects.

$$\text{Mean}(d(\zeta, \Theta \backslash \zeta)) = \frac{1}{|\Theta \backslash \zeta|} \sum_{\forall A_i \in \Theta \backslash \zeta} d(\zeta, A_i) \tag{4}$$

3. Neutro-DIANA proposed method

Neutrosophic Clustering is based on the Single-valued Neutrosophic sets (SVNs) technique, in which data points belong to several clusters with membership degrees in the range $[0, 1]$.

Definition . The neutrosophic DIANA algorithm is a clustering algorithm that uses neutrosophic logic principles and neutrosophic sets. It uses SVNs-based operations in the calculation of its clustering algorithm.

3.1. *Neutrosophic Set Formation*

Assume a dataset comprises a collection of n SVNs denoted Θ, where $\Theta = \{A_i/1 \leq i \leq n\}$ is defined in a universe of discourse X in the SVNs environment, and each object is expressed as : Let x be a vector in an n-dimensional real space Rn (the fea- ture space) and let C=c1, c2, ..., cc, be a set of class labels. A neutrosophic classifier is mapping of the type:

$$\psi\colon \mathbb{R}^n \longrightarrow \{\mathbb{TC}(x), \mathbb{IC}(x), \mathbb{FC}(x) | x \in \mathbb{R}^n\} \tag{5}$$

$$\psi\colon \mathbb{R}^n \longrightarrow \{\mathbb{TC}(x), \mathbb{IC}(x), \mathbb{FC}(x) | x \in \mathbb{R}^n\} \tag{6}$$

Let x be a vector in the n-dimensional features space $\mathbb{R}^n$, and $C = \{c_1, c_2, \cdots, c_c\}$, be a collection of class labels. A neutrosophic classifier is a sort of mapping:

$$A_i = \{\langle x_j, \mu_{A_i}(x_j), \omega_{A_i}(x_j), \nu_{A_i}(x_j)\rangle : x_j \in X\}. \tag{7}$$

We generate the Neutrosophic Distance matrix-nD0 using SVNS similarity and/or dissimilarity measurements ((8)), as indicated in the table (1) below.

3.2. *The similarity in Neutrosophic environment*

Definition 3. Euclidean Neutrosophic distance. In the Neutrosophic environment, the mapping form of euclidian distance applied to A_1 and A_2 (two SVNSs) is as follows:

TABLE 1. Neutrosophic Distance matrix-nD0

	x_1	$\cdots$	x_n
A_1	$\langle \mu_{A_1}(x_1), \omega_{A_1}(x_1), \nu_{A_1}(x_1)\rangle$	$\cdots$	$\langle \mu_{A_1}(x_n), \omega_{A_1}(x_n), \nu_{A_1}(x_n)\rangle$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
A_m	$\langle \mu_{A_m}(x_1), \omega_{A_m}(x_1), \nu_{A_m}(x_1)\rangle$	$\cdots$	$\langle \mu_{A_m}(x_n), \omega_{A_m}(x_n), \nu_{A_m}(x_n)\rangle$

$$d_{eucl} = \sqrt{\frac{1}{3}\sum_{i=1}^{n}\sum_{f=(\mu,\omega,\nu)}(f_{A_1}(x) - f_{A_2}(x))^2} \tag{8}$$

where μ, ω and ν are Neutrosophic membership functions.

Definition 4. Similarity and/or dissimilarity of SVNSs measurements. The similarity measure S_{mes} between A_1 and A_2 based on max and min operators, as described by [21], is defined as follows :

$$S_{mes} = \frac{1}{3}\sum_{i=1}^{n}\frac{\sum_{f=(\mu,\omega,\nu)}\min(f_{A_1}(x), f_{A_2}(x))}{\sum_{f=(\mu,\omega,\nu)}\max(f_{A_1}(x), f_{A_2}(x))} \tag{9}$$

The following is the definition of the dissimilarity measure:

$$DIS_{mes} = 1 - S_{mes} \tag{10}$$

3.3. *nDIANA algorithm*

Let $\{A_i // i = 1 \ldots n\}$ be a collection of n SVNs nDIANA consists on three main steps.

The nDIANA method starts by treating all n objects as a single cluster level$L(m_c = 0) = \Theta$ object. Using the dissimilarity measures (9), elements A_i are then pairwise compared among themselves, and then separated into two sub-clusters with sub-levels $L(m_{c+1} = 0)$, and $L(m_{c+2} = 0)$, respectively, based on the clusters' furthest (with maximum mean distance) sub-clusters. The subdividing operation is repeated until all clusters have a single-single item. That is, each obtained cluster has a size of 1. In each stage, we reapply the treatment on each sub-cluster recursively, and the distance between the object and the sub-cluster is taken as the average distance between the object and all components of the sub-cluster.

Step 1 : Calculate the similarity and/or dissimilarity measurements of SVNs using equations Eq.9 and/or Eq.10, and then create the Neutrosophic Distance matrix-nD0 (Table 1).

Step 2 : Each stage of the divisive algorithm requires a decision on which cluster to split. To do this, we compute the diameter as indicated in

$$\text{diam}(Q) = max_{j\in Q,h\in Q} d(A_j, A_h) \tag{11}$$

In a loop, choose just the element A_j with the greatest mean dissimilarity to all other elements in the same cluster.

$$d(A_i, \Theta \backslash A_i) = \frac{1}{|\Theta| - 1} \sum_{j \neq i} d(A_i, A_j) \tag{12}$$

$$\Theta_{new} = \Theta_{old} \backslash A_i$$

$$\bar{\Theta}_{new} = \bar{\Theta}_{,old} \cup A_i$$

$$d(A_i, \Theta \backslash A_i) - d(A_i, \bar{\Theta}) = \frac{1}{|\Theta| - 1} \sum_{j \in \Theta, j \neq i} d(A_i, A_j) - \frac{1}{|\bar{\Theta}|} \sum_{h \in \bar{\Theta}} d(A_i, A_h) \tag{13}$$

$\bar{\Theta}$ is the complement of Θ.

Step 3 : If all clusters contain only one observation, the procedure is complete; otherwise, go to step 2 using the sub-clusters formed in the previous iteration.

4. Results and Discussion

To demonstrate the usefulness of the proposed Neutrosophic DIANA method, an experiment was conducted on both the simulated and real-world datasets. For the purpose of comparison, we use the numeric example introduced by Sahin in [11]. In this case, dataset consists on five objects A_i with $1 \leq i \leq$, universe of discourse is $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$.

TABLE 2. Neutrosophic Set Formation Example

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
A1	0.2 0.05 0.5	0.1 0.15 0.8	0.5 0.05 0.3	0.9 0.55 0.0	0.4 0.4 0.35	0.1 0.4 0.9	0.3 0.15 0.5	1.0 0.6 0.0
A2	0.5 0.6 0.4	0.6 0.3 0.15	1.0 0.6 0.0	0.15 0.05 0.65	0.0 0.25 0.8	0.7 0.65 0.15	0.5 0.5 0.5	0.65 0.05 0.2
A3	0.45 0.05 0.35	0.6 0.5 0.3	0.9 0.05 0.0	0.1 0.6 0.8	0.2 0.35 0.70	0.6 0.4 0.2	0.15 0.05 0.8	0.2 0.6 0.65
A4	1.0 0.65 0.0	1.0 0.25 0.0	0.85 0.65 0.1	0.2 0.05 0.8	0.15 0.3 0.85	0.1 0.6 0.7	0.3 0.6 0.7	0.5 0.35 0.7
A5	0.9 0.2 0.0	0.9 0.4 0.0	0.8 0.05 0.1	0.7 0.45 0.2	0.5 0.25 0.15	0.3 0.3 0.65	0.15 0.1 0.75	0.65 0.5 0.8

The nDIANA algorithm begins with all observations as a single cluster, $L(m_c = 0) = \{A_i/1 \leq i \leq 5\}$, with 5 is the number of observations.

Utilize (Eq. (9), Eq.(10) to calculate the similarity and dissimilarity measures of SVNSs, and then construct Neutrosophic Distance matrix-nD0 (table 3).

To determine the similarity and dissimilarity measurements of SVNSs, use the equations ((Eq. (9), Eq.(10)), and then create the Neutrosophic Distance matrix-nD0 (table 3).

Next, use Eq. (12), Eq.(13) to compute the average distance between each element and every other element.

TABLE 3. Distance matrix–nD0

	A_1	A_2	A_3	A_4	A_5	Mean
A_1	0.000	0.661	0.563	0.636	0.508	**0.474**
A_2	0.661	0.000	0.438	0.357	0.596	0.410
A_3	0.563	0.438	0.000	0.469	0.433	0.381
A_4	0.636	0.357	0.469	0.000	0.416	0.376
A_5	0.508	0.596	0.433	0.416	0.000	0.390

Select the element with the highest distance mean, in this case (A_1), which is 0.474. As a result, the cluster $\{A_1\}$'s maximum distance from other data points is 0.474. However, before deciding to split, we must first determine which elements are closest to each new cluster. To do this, we must compute the mean distance between each element using the formulas$L(m_c = 1) = \{A_1\}$ and $L(m_c = 2) = \{\Theta \backslash A_1\}$ as given in the table 4.

TABLE 4. Distance matrix-nD01 of each observation with each cluster

	$\{A_1\}$	$\{A_2, A_3, A_4, A_5\}$
A_1	0.000	0.592
A_2	0.661	0.348
A_3	0.563	0.335
A_4	0.636	0.310
A_5	0.508	0.361

Thus, the single cluster $L(m_c = 0) = \Theta$ is split into tow clusters $L(m_c = 1) = \{A_1\}$ and $L(m_c = 2) = \Theta \backslash A_1$. With a new sub-cluster, $L(m_c = 2)$, we carry out the identical processes once more to get a new distance matrix-nD2 (able 5).

TABLE 5. Distance matrix – nD02

	A_2	A_3	A_4	A_5	Mean
A_2	0.000	0.438	0.357	0.596	0.348
A_3	0.438	0.000	0.469	0.433	0.341
A_4	0.357	0.469	0.000	0.416	0.310
A_5	0.596	0.433	0.416	0.000	0.361

From table 5 the maximum of mean distances is between A_5 and the rest at distance 0.361. Then, $L(m_c = 2) = \{A_2, A_3, A_4, A_5\}$ is split into clusters, $L(m_c = 3) = \{A_2, A_3, A_4\}$, and $L(m_c = 4) = \{A_5\}$ at a distance 0.361.

To cross check the stability of each gotten cluster $L(m_c = 3)$ and $L(m_c = 4)$, we examine the closeness of each element to both obtained cluster (table 6).

Table 6. Distance matrix-nD02 of each observation with each cluster

	$\{A_2, A_3, A_4\}$	$\{A_5\}$
A_2	0.265	0.596
A_3	0.302	0.433
A_4	0.275	0.416
A_5	0.482	0.000

The sub-cluster $L(m_c = 3) = \{A_2, A_3, A_4\}$ obtained from previous splitting need to be treated, and its Distance matrix–nD03 (see table 7).

Table 7. Distance matrix–nD03

	A_2	A_3	A_4	Mean
A_2	0.000	0.438	0.357	0.265
A_3	0.438	0.000	0.469	**0.302**
A_4	0.357	0.469	0.000	0.275

The maximum of mean distances is between A_3 and the rest of elements at distance 0.302. Then, the $L(m_c = 3) = \{A_2, A_3, A_4\}$ is split into clusters, $L(m_c = 5) = \{A_2, A_4\}$, and $L(m_c = 6) = \{A_3\}$ at a distance 0.302.

Table 8. Distance matrix-nD03 of each observation with each clusters

	$\{A_2, A_4\}$	$\{A_3\}$
A_2	0.179	0.438
A_3	0.454	0.000
A_4	0.179	0.469

Finally, because the remain cluster $L(m_c = 5) = \{A_2, A_4\}$ contains only two elements, it is divided into $L(m_c = 7) = \{A_2\}$ and $L(m_c = 8) = \{A_4\}$ at distance 0.357 and creates the single–single object in all clusters. As a result, the Neutrosophic Divisive Analysis Clustering (nDIANA) with Neutrosophic computation is terminated.

Here, we outline the specifics of the entire splitting process as implemented by our nDIANA suggested method.

At the beginning $\Theta = \{A_i/1 \leq i \leq 5\}$, all elements are in the same cluster $\{A_1, A_2, A_3, A_4, A_5\}$.

The farthest dissimilarity measure is of A_1 (Eq.14).

$$d(A_1, \Theta \backslash A_1) = \max\{d_{1 \leq i \leq 5}(A_i, \Theta \backslash A_i)\} \tag{14}$$

which is 0.474 terminates $\{A_1, A_2, A_3, A_4, A_5\}$ is split into tow clusters : $\{A_1\}$and $\{A_2, A_3, A_4, A_5\}$.

The farthest dissimilarity measure in gotten sub-cluster is of A_5 (Eq.15).

$$d(A_5, \Theta\backslash\{A_1, A_5\}) = \max\{d_{2\leq i\leq 5}(A_i, \Theta\backslash\{A_i, A_1\})\} \tag{15}$$

which is 0.361, then $\{A_2, A_3, A_4, A_5\}$ are split into tow clusters : $\{A_2, A_3, A_4\}$ and $\{A_5\}$.

The farthest dissimilarity measure in gotten sub-cluster is of A_3 (Eq.16).

$$d(A_3, \Theta\backslash\{A_1, A_5, A_3\}) = \max\{d_{i=2,4}(A_i, \Theta\backslash\{A_i, A_1, A_5\})\} \tag{16}$$

which is 0.302, then $\{A_2, A_3, A_4\}$ are split into tow clusters : $\{A_2, A_4\}$ and $\{A_3\}$.

There are only two elements left $\{A_2, A_4\}$, the distance between them is 0.357, and in this case the subdivision is automatic to two clusters which are: $\{A_2\}$ and $\{A_4\}$.

By the end, put all the results together we get :

first $((A_1), A_2, A_3, A_4, A_5)$,

next $((A_1), (A_2, A_3, A_4, (A_5)))$,

then $((A_1), (((A_2, A_4), (A_3)), (A_5)))$,

finally $((A_1), ((((A_2), (A_4)), (A_3)), (A_5)))$.

Each and every machine learning method is designed to tackle learning issues involving crisp numbers. However, all data sources produce inaccurate, imprecise, and ambiguous data that has numerous other flaws. A broad framework is provided by the single-valued Neutrosophic set (SVNs), an extension of the fuzzy logic set, to describe and model uncertain, imperfect, and imprecise data with missing and mistakes. By using machine learning algorithms designed for precise numbers, it is possible to build tidy data that is purported to be clean but really goes through a lot of creation and destruction processes simultaneously. Hence, clustering learning in a single-valued Neutrosophic environment is another way to capture and manage data noise and take it into account as an additional source of information. To handle data noise and take it into account as an extra factor, clustering learning in a single-valued Neutrosophic environment is a different technique.

5. Conclusion

In conclusion, we obtained the same outcomes when comparing the Agglomerative Hierarchical Clustering Technique and the DIANA with Neutrosophic findings on the simulated data set. And from there, we may conclude that (1) the DIANA with Neutrosophic algorithm can aggregate SVNs on a wide scale, and (2) the uncertainty information acquired by SVNs is crucial for the accomplishment of some aggregation tasks. We have created a useful approach for grouping SVNs using divisive hierarchical clustering.

To reduce data indeterminacy, the hierarchical clustering divisive DIANA method based on Neutrosophic logic is used. The suggested method's findings show that it may be utilized to produce superior outcomes on real-world data. Based on the crisp hierarchical clustering technique, we suggested a hierarchical single-value neutrosophic algorithm for SVN clustering.

References

1. Haibo Wang, Da Huo, Jun Huang, Yaquan Xu, Lixia Yan, Wei Sun, and Xianglu Li. An approach for improving the k-means algorithm on market segmentation. In 2010 International Conference on System Science and Engineering, pages 368–372. IEEE, 2010.
2. Gianfranco Chicco, Roberto Napoli, and Federico Piglione. Comparisons among clustering techniques for electricity customer classification. IEEE Transactions on power systems, 21(2):933–940, 2006.
3. Balamurugan Balusamy, Seifedine Kadry, Amir H Gandomi, et al. Big Data: Concepts, Technology, and Architecture. John Wiley & Sons, 2021.
4. S. Saraswathi and A. Allirani. Survey on image segmentation via clustering. In 2013 International Conference on Information Communication and Embedded Systems (ICICES), pages 331–335. IEEE, 2013.
5. Leonard Kaufman and Peter J Rousseeuw. Finding groups in data: an introduction to cluster analysis, volume 344. John Wiley & Sons, 2009.
6. Mohammad Naved Qureshi and Mohd Vasim Ahamad. An improved method for image segmentation using k-means clustering with neutrosophic logic. Procedia computer science, 132:534–540, 2018.
7. N. Akhtar and A. M. Vasim. A modified fuzzy c means clustering using neutrosophic logic. In 2015 Fifth International Conference on Communication Systems and Network Technologies, pages 1124–1128. IEEE, 2015.
8. S. Florentin. Neutrosophy: neutrosophic probability, set, and logic: analytic synthesis & synthetic analysis. 1998.
9. Azeddine Elhassouny, Soufiane Idbrahim, and Florentin Smarandache. Machine learning in the neutrosophic environment: A survey. Neutrosophic Sets and Systems, 28:58–68, 2019.
10. S. Vandhana and J. Anuradha. Neutrosophic fuzzy hierarchical clustering for dengue analysis in Sri Lanka. Neutrosophic Sets and Systems, 31(1):14, 2020.
11. Rıdvan Sahin. Neutrosophic hierarchical clustering algorithms. Neutrosophic Sets and Systems, 2:18 24, 2014.
12. Jun Ye. Single-valued neutrosophic minimum spanning tree and its clustering method. Journal of Intelligent Systems, 23(3):311–324, 2014.
13. Gloria Bordogna and Gabriella Pasi. Hierarchical-hyperspherical divisive fuzzy c-means (h2d-fcm) clustering for information retrieval. In 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, volume 1, pages 614–621. IEEE, 2009.
14. Chris Ding and Xiaofeng He. Cluster merging and splitting in hierarchical clustering algorithms. In 2002 IEEE International Conference on Data Mining, 2002. Proceedings., pages 139–146. IEEE, 2002.
15. Jun Ye. Clustering methods using distance-based similarity measures of single-valued neutrosophic sets. Journal of Intelligent Systems, 23(4):379–389, 2014.
16. Wang H, Smarandache F, Zhang YQ, Sunderraman R. Single valued neutrosophic sets. Multispace and Multistructure, Kalyan Mondal, and Surapati Pramanik. volume 4, pages 410–413. 2010.
17. Haibin Wang, Florentin Smarandache, Rajshekhar Sunderraman, and Yan Qing Zhang. interval neutrosophic sets and logic: theory and applications in computing: Theory and applications in computing, volume 5. Hexis, Phoenix, AZ, 2005.

18. Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd, volume 96, pages 226–231, 1996.
19. Han Jiawei, Micheline Kamber, and Morgan Kaufmann. Data mining: Concepts and techniques, 2001. University of Simon Fraser.
20. Ashish Kumar Patnaik, Prasanta Kumar Bhuyan, and KV Krishna Rao. Divisive analysis (Diana) of hierarchical clustering and gps data for level of service criteria of urban streets. Alexandria Engineering Journal, 55(1):407–418, 2016.
21. Pinaki Majumdar and Syamal Kumar Samanta. On similarity and entropy of neutrosophic sets. Journal of Intelligent & Fuzzy Systems, 26(3):1245–1252, 2014.

Received: August 09, 2022. Accepted: January 12, 2023