International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN:2347-4890, ijournals.in/ijshre
Volume 14 Issue 10, October 2026

A Fuzzy Hierarchical Clustering Based on the Maximum Distance to the Center of Gravity

Author: Mputu Losala Lomo Denis-Robert

Affiliation: Université Pédagogique Nationale (UPN), Kinshasa, Democratic Republic of Congo

E-mail: denis-robert.mputu@upnrdc.net

https://doi.org/10.26821/ijshre.14.10.2026.141001

ABSTRACT

This paper proposes a new approach to Fuzzy Agglomerative Hierarchical Clustering based on the maximum distance to the center of gravity (CAHFD), falling within the field of automatic classification. This contribution introduces a hierarchical approach based on a membership function using the maximum distance to the center of gravity. Furthermore, the paper presents an independent innovation: a new distance expression, the Corrected Distance (D_{Co}), combining the J_M (or JM) index and the Euclidean distance. Unlike non-hierarchical partitioning methods such as fuzzy C-means, which do not produce any dendrogram, CAHFD incorporates a tree-like structure that preserves nuanced fuzziness regarding boundary objects and faithfully restores the topological progressivity of the data.

Keywords: Automatic classification, Fuzzy hierarchical clustering, CAHFD, Maximum distance to the center of gravity, Membership degree, Corrected Distance (D_{Co}), J_M index, Dendrogram, Fuzzy C-means.

1. INTRODUCTION

Classification methods consist in finding classes such that individuals within the same class are as similar as possible (intra-class homogeneity) and those in different classes are as dissimilar as possible (inter-class heterogeneity). Classification can be classical or fuzzy. It is classical when each individual belongs to a single class. Conversely, it is said to be fuzzy if each individual simultaneously belongs to all classes with given membership degrees whose sum equals 1. The works of (Chesneau, 2018; Dagnelie, 1986) provide essential elements on classification.

A distinction is made between supervised classification and unsupervised (or automatic) classification. In supervised classification, classes are defined in advance, and new individuals must be classified one by one into one of these pre-labeled classes. Meanwhile, in unsupervised or automatic classification (clustering), classes are not fixed beforehand but are determined progressively.

The typology of methods distinguishes between hierarchical classification (hierarchical approach) and non-hierarchical or partitioning classification, also known as flat classification (partitive approach). The former includes Hierarchical Agglomerative Clustering (HAC) and Hierarchical Divisive Clustering (HDC), while the latter includes, in a classical environment, the k-means method (or moving centers) and the dynamic clouds method, and in a fuzzy environment, the Fuzzy c-means method ([8], [11], [14], [21], [24], [26]).

In the following, we present our new and essential proposed approaches: the corrected distance (D_{Co}), applied notably in classical hierarchical agglomerative clustering (CAH_{DCo}), and fuzzy hierarchical agglomerative clustering based on the maximum distance to the center of gravity (CAHFD, from French, Classification Ascendante Hiérarchique Floue basée sur la Distance maximale au centre de gravité), which notably utilizes D_{Co}.

To bypass version incompatibility constraints or the unavailability of external packages on certain platforms, all calculations were implemented within the R environment. Due to the unavailability of certain standard packages, custom functions were developed, more specifically for the implementation of the CAHFD approach. In this regard, the assistance of a language model (Gemini, Google) was also utilized to optimize the syntax of certain code blocks, with the entirety of the methodological design, mathematical validations, and final implementation remaining under the sole responsibility of the author.

2. MATERIALS AND METHODS

2.1. Classical Clustering Method

Classification (clustering) is a statistical method that consists in finding classes such that individuals within the same class are as similar as possible (intra-class homogeneity) while those in different classes are as dissimilar as possible (inter-class heterogeneity). Among classification methods, there is Agglomerative Hierarchical Clustering (AHC), which we will use in the following. AHC allows grouping individuals into a certain number of classes derived from a hierarchy of partitions. Applications of the AHC method can be found notably in [16].

Carrying out an AHC involves the following steps:

2.1.1. Constitution of the data table

This involves setting up the data collected from individuals based on the selected variables. This may be preceded by transforming the initial data into standardized (or centered-standardized) data in the case of heterogeneous data, or into centered data. Centered data brings the origin of the axes to the center of gravity, while standardized data eliminates the influence of one unit of measurement over the others.

2.1.2. Calculation of pairwise distances between individuals

This step of AHC consists in calculating pairwise distances between individuals, leading to the grouping of two individuals having the smallest distance (first grouping).

At this level, it is appropriate to choose a distance function or index and use it to determine the distance between pairwise individuals defined for m variables in order to conclude whether or not they resemble each other ([1], [16], [23]).

Definition 2.1 (Distance)

Let I be a set of points. A distance on I is a mapping d: $I \times I \rightarrow R^+$ satisfying the following three properties for all i, k, l ∈ I:

· d(i, k) = 0 if i = k (Separation)

· d(i, k) = d(k, i) (Symmetry)

· d(i, l) ≤ d(i, k) + d(k, l) (Triangle inequality)

Definition 2.2 (Metric space)

A metric space is a set equipped with a distance. It is therefore a pair (I, d) where I is a set and d is a distance on I.

2.1.2.1. Distance expressions

There are various distance expressions [8]. Let w_{ij} and w_{kj} be the respective values of individuals i and k for variable X_j, we have:

1) Manhattan distance (City-block distance):

$$d(i,k) = \sum_{j=1}^{m} |w_{ij} - w_{kj}| \quad (2.1)$$

2) Euclidean distance:

$$d(i,k) = \sqrt{\sum_{j=1}^{m} (w_{ij} - w_{kj})^2} \quad (2.2)$$

3) Minkowski distance:

$$d(i,k) = \sqrt[p]{\sum_{j=1}^{m} |w_{ij} - w_{kj}|^p} \quad (2.3)$$

This is the p-distance. If p = 2, this distance is equivalent to the Euclidean distance.

4) Chebyshev distance:

$$d(i,k) = \sqrt[p]{\sum_{j=1}^{m} |w_{ij} - w_{kj}|^p} \quad (2.4)$$

5) Squared Euclidean distance:

$$d(i,k) = \sum_{j=1}^{m} (w_{ij} - w_{kj})^2 \quad (2.5)$$

6) Canberra distance [25]:

$$d(i,k) = \sum_{j=1}^{m} \frac{|w_{ij}-w_{kj}|}{|w_{ij}+w_{kj}|} \quad (2.6)$$

ou encore

$$d(i,k) = \sum_{j=1}^{m} \left|\frac{w_{ij}-w_{kj}}{w_{ij}+w_{kj}}\right| \quad (2.7)$$

2.1.2.2. Number of Distances to Calculate

Proposition 2.1 (Number of distances to calculate [16])

The number N of distances to calculate between n given individuals is:

$$N = \frac{n(n-1)}{2} \quad (2.8)$$

Proof: (Cf. [16])

2.1.3. Calculation of distances between groups of individuals ([1], [8], [25])

The different aggregation indices are:

1) Minimum distance or single linkage or nearest neighbor:

$$d(I_1, I_2) = d(i,k) \quad (2.9)$$

2) Maximum distance or complete linkage or diameter:

$$d(I_1, I_2) = d(i,k) \quad (2.10)$$

3) Average distance or average linkage:

$$d(I_1, I_2) = \sum_{i=1}^{n_1} \sum_{i=1}^{n_2} d(i,k) \quad (2.11)$$

where n_1 (respectively n_2) is the number of individuals in I_1 (respectively in I_2).

4) Median distance:

All distances between the respective individuals of two studied groups are calculated, and then their median value is determined.

5) Centroid distance:

$$d(I_1, I_2) = d(G_1, G_2) \quad (2.12)$$

where G_1 and G_2 are the respective centroids of I_1 and I_2.

6) Ward's method:

$$d(I_1, I_2) = \frac{n_1.n_2}{n_1+n_2} d^2(G_1, G_2) \quad (2.13)$$

where G_1 (respectively G_2) is the centroid of group I_1 (respectively I_2) with size n_1 (respectively n_2).

7) Distance by weighted sums [16]:

$$d(I_1, I_2) = \frac{1}{pq} \sqrt{\sum_{j=1}^{m} \left(p \sum_{i=1}^{q} y_{ij} - q \sum_{i=1}^{p} x_{ij}\right)^2} \quad (2.14)$$

where I_1 and I_2 are two groups of individuals each potentially containing a single element, p and q respectively the number of individuals in I_1 and I_2, m the number of variables, and x_{ij} and y_{ij} respectively the various coordinates of the points (individuals) of I_1 and I_2.

We proposed this expression in our previous works. It combines the Euclidean distance between two points (special case) and the centroid distance between two groups of points (general case) (Cf. [16]).

2.1.4. Graphical representation: The dendrogram

After aggregation comes the graphical representation of Agglomerative Hierarchical Clustering called a dendrogram (or classification tree or hierarchical tree). This tree is generally binary and positioned as an upside-down tree. Going from bottom to top, isolated individuals are found at the bottom constituting the terminal elements of the tree, followed by groups of individuals resulting from groupings, called nodes of the tree. Each of the nodes constitutes a partition ([3], [13], [15]).

2.1.5. Cutting the Dendrogram

For the same dendrogram, several partitions can be extracted. In this case, the best one must be sought because only a single one is needed. The problem of partition quality arises. To this end, the notion of inertia is used. A partition is best if the individuals within the same class are closest to each other (intra-class inertia is small: minimization of intra-class inertia) and those from two different classes are furthest apart (inter-class inertia is large: maximization of inter-class inertia) ([3], [13]).

To obtain an optimal partition, the dendrogram is cut at the level where the aggregation index makes a significant jump when moving from one partition to another using a horizontal line. This leads to the determination of classes forming a partition.

The number of classes depends notably on the researcher's assessment ([7], [8]).

2.1.6. Interpretation of a partition

After determining the best partition and cutting the dendrogram, its interpretation is carried out, consisting of: 1) calculating the mean, variance, and standard deviation of each variable; 2) characterizing (or describing) the classes by their individuals by determining those that are most typical (paragons and extremes) or by their variables by determining those that are most important; and finally 3) graphically representing the classes of individuals.

2.2. Fuzzy clustering method: Fuzzy C-Means

2.2.1. Presentation of the Fuzzy C-Means method

Fuzzy c-Means (FCM) is a non-hierarchical fuzzy clustering algorithm introduced by Dunn (1974) to extend the k-means method to a fuzzy environment. This was generalized by other authors, principally [4], [5], [6] and (22].

The fuzzy c-means algorithm leads to a partition where each individual belongs to each class (characterized by its center of gravity) with a membership degree lying within the interval [0, 1]. For each individual, the sum of the membership degrees to the different classes equals 1.

To determine the different fuzzy classes whose number is known in advance, Fuzzy c-means proceeds by minimizing intra-class distances and maximizing inter-class distances while

assigning each individual i a specific membership value to each class. It uses an iterative process by minimizing an objective function ([27], [28]).

2.2.2. Characteristics of the membership degree matrix

Let μ_{ik} represent the membership degree of individual i to the k^{th} class. The matrix $U_{q\times n} = (\mu_{ik})_{1\le i\le n,\ 1\le k\le q}$ is called the membership degree matrix corresponding to the fuzzy k-partitions of n individuals if and only if it satisfies the following two conditions:

```
# Fuzzy C-means function
fcm_dero <- function(X, c = 2, m = 2, max_iter = 100, tol = 1e-5) {
  X <- as.matrix(X)
  n <- nrow(X)
  p <- ncol(X)
  # Step 1: Random initialization of the membership matrix U
  set.seed(123)
  U <- matrix(runif(n * c), nrow = n, ncol = c)
  U <- sweep(U, 1, rowSums(U), "/") # Normalization so that each row sums to 1
```

- $\mu_{ik} \in [0, 1]$ and $0 < \sum_{i=1}^{n} \mu_{ik} < n, \forall k \in \{1,...,q\}, \quad \forall i \in \{1,...,n\}.$ (2.15)
- $\sum_{k=1}^{q} \mu_{ik} = 1, \forall i \in \{1,...,n\}.$ (2.16)

2.2.3. Steps of Fuzzy C-means

The main steps of the fuzzy c-means algorithm are:

1) arbitrarily setting an initial membership matrix,
2) calculating the class centroids,
3) updating the membership matrix taking into account the positions of the centroids,
4) convergence of the algorithm toward a local minimum, which determines the stopping condition of the algorithm ([10]).

2.2.4. Principle of the Fuzzy C-means algorithm

The principle is to minimize the inter-group variance ([12]), which amounts to minimizing the objective function ([27]) presented in the following theorem:

Theorem 2.1

Let the objective function be:

$$J(G,U,I) = \sum_{k=1}^{q} \sum_{i=1}^{n} \mu_{ik}^{m} d^2(x_i, G_k) \quad (2.17)$$

The solutions to the Fuzzy c-means problem consisting in minimizing the objective function J(G, U, I) are described in the following formulas:

$$G_k = \frac{\sum_{i=1}^{n} \mu_{ik}^{m} x_i}{\sum_{i=1}^{n} \mu_{ik}^{m}} \quad (2.18)$$

and

$$\mu_{ik} = \left[\left(\frac{d(x_i, G_k)}{\sum_{j=1}^{q} d(x_i, G_j)} \right)^{\frac{2}{m-1}} \right]^{-1} \quad (2.19)$$

where $G = \{G_1, ..., G_k, ..., G_q\}$ is the set of q centers of gravity of the q classes, U is the membership degree matrix, I is the set of n individuals, q ($1 \le k \le q$) is the number of classes, n ($1 \le i \le n$) is the number of individuals, $m \in]1, 2]$ is the fuzziness exponent or fuzzification parameter ([10]); the higher its value, the greater the fuzziness of the partition. The value most commonly used in algorithms is 2.

Proof ([27])

The fuzzy parameter m allows amplifying the differences between distant points and central points. If m = 1, we are in the case of ordinary classification.

2.2.5. Implementation of Fuzzy C-means in R

2.2.5.1. Fuzzy C-means function (fcm_dero)

The R function, fcm_dero, help to use Fuzzy c-means and determinate automatically different individuals classes:

```
for (iter in 1:max_iter) {
U_old <- U
# Step 2: Calculation of cluster centers (V) weighted by U^m
    Um <- U^m
    V <- t(Um) %*% X / colSums(Um)
# Step 3: Updating the membership matrix U
    dist_mat <- matrix(0, nrow = n, ncol = c)
    for (i in 1:c) {
      dist_mat[, i] <- sqrt(rowSums((X - matrix(V[i, ], nrow = n, ncol = p, byrow = TRUE))^2))
    }
    dist_mat[dist_mat == 0] <- 1e-10
    expon <- 2 / (m - 1)
    dist_inv <- 1 / dist_mat
    dist_inv_exp <- dist_inv^expon
    U <- dist_inv_exp / rowSums(dist_inv_exp)
    # Convergence test
    if (max(abs(U - U_old)) < tol) break
  }
  cluster <- max.col(U)
  names(cluster) <- rownames(X)
  # Assign names of individuals and columns to matrices for readability
  colnames(U) <- paste0("Cluster", 1:c)
  rownames(V) <- paste0("Cluster", 1:c)
  colnames(V) <- colnames(X)
  # Display results
  cat("\n--- Cluster Centers (V) ---\n")
  print(V)
  cat("\n--- Membership Matrix (normalized with names) ---\n")
  print(round(U, 4))
  cat("\n--- Majority Assignment Class ---\n")
  print(cluster)
  return(list(centers = V, membership = U, cluster = cluster))
}
```

2.2.5.2. Automatic calculation with fcm_dero function:

Simply copy and paste the code of the function above into the R console and then input the data to be classified.

```
# --- Application ---
data_eleves <- data.frame(Maths = c(6, 7, 2, 8, 2),
Francais = c(4, 8, 3, 9, 6))
rownames(data_eleves) <- c("a", "b", "d", "e", "f")
# Execution of the function
result <- fcm_dero(data_eleves, c = 2, m = 2)
```

2.2.5.3. Results and class composition:

After validating, R returns the matrix of membership degrees while specifying the membership classes of the individuals:

Individual	Cluster1	Cluster2	Assignment
a	0.3461	0.6539	2
b	0.9938	0.0062	1
d	0.0412	0.9588	2
e	0.9801	0.0199	1
f	0.0907	0.9093	2

Two classes: Cluster 1 ={b, e}; Cluster 2 ={a, d, f}.

2.3. Our perspective on clustering in relation to inequality reduction

In our humble opinion, we consider that the results of Agglomerative Hierarchical Clustering can be used at the interpretation level to calculate the proportions and shares of individuals in the distribution of a given resource based on the total values of the individuals. This therefore constitutes a new element in clustering interpretation.

Clustering will serve to highlight the closest individuals among whom inequalities will be reduced. Because, in our view, the

closest individuals—that is, those belonging to the same class—should help each other before seeking rescue from others who are more distant. In addition, individuals belonging to the same population should assist one another before another population comes to their rescue. Thus, individuals show solidarity within their respective classes and then across the entire population. This justifies the double reduction of inequalities. This concerns individuals who have not contributed to creating the resource to be shared, unlike the case where individuals have contributed to creating the resource. In the latter case, there is no question, in our view, of reducing inequalities ([16], [17], [18], [19], [20]).

3. RESULTS

We have proposed: 1) The corrected distance, which is a Euclidean distance weighted by our correction index J_M, 2) the maximum distance to the center of gravity, which is a maximization of distances between individuals belonging to a closed ball, 3) the construction of the membership function based on the maximum distance to the center of gravity and 4) the construction of the classical Hierarchical Agglomerative Clustering method using the corrected distance denoted D_{Co} and the fuzzy one based on the maximum distance to the center of gravity denoted CAHFD.

3.1. Corrected distance

3.1.1. Definition of the Corrected distance

Proposition 3.1 (Corrected distance)

Let a set of n individuals be characterized by an m-dimensional space (variables). For two distinct individuals A and B, whose coordinates are respectively A = (X_{1A}, X_{2A}, ..., X_{mA}) and B = (X_{1B}, X_{2B}, ..., X_{mB}), the corrected distance between A and B, denoted $D_{Co}(A, B)$, is defined by:

$$D_{Co}(A,B) = \sqrt{\sum_{j=1}^{m} \left[J_M(j).\left(X_{jB} - X_{jA}\right)\right]^2} \qquad (3.1)$$

Where

$$J_M(j) = \frac{\sum_{i=1}^{n} (X_{ij} - X_{1j})}{\sum_{i=1}^{n} X_{ij}} \qquad (3.2)$$

or

$$J_M(j) = 1 - \frac{X_{1j}}{\underline{X}_j} \qquad (3.3)$$

is the inequality reduction index or correction index corresponding to variable j, with X_{1j} being the minimum observed value and

$$\underline{X}_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij} \qquad (3.4)$$

the arithmetic mean of dimension j (assuming $\underline{X}_j \neq 0$ for all j).

3.1.2. Special case

In the case where the arithmetic mean of a variable X_j is zero ($\underline{X}_j = 0$), we apply a translation by a constant $\varepsilon > 0$ to all values X_{ij} for the concerned variable, such that $X'_{ij} = X_{ij} + \varepsilon$.

The correction index is adjusted and then becomes:

$$J_M(j) = 1 - \frac{X_{1j}+\varepsilon}{\underline{X}_j+\varepsilon} \qquad (3.5)$$

Since $\underline{X}_j = 0$, the formula simplifies to:

$$J_M(j) = 1 - \frac{X_{1j}+\varepsilon}{\varepsilon} \qquad (3.6)$$

3.1.3. Implementation of Corrected distance calculation

The R code implementing the calculation of the corrected distance and the J_M index that we proposed, taking into account the handling of the special case where the mean is zero via the application of translation, is presented as follows:

```
# 1. Corrected distance function
dist_corrigee <- function(x, epsilon = 1) {
n <- nrow(x)
m <- ncol(x)
# 2. Calculation of JM(j) for each column j
jm_values <- sapply(1:m, function(j) {
col_j <- x[, j]
moyenne_j <- mean(col_j)
min_j <- min(col_j)
# Case 1: mean different from 0
if (moyenne_j != 0) {
return(1 - (min_j / moyenne_j))
}
# Case 2: mean equal to 0
else {
return(1 - ((min_j + epsilon) / epsilon))
}
})
# 3. Calculation of corrected distance
print("weights JM calculated:")
print(jm_values)
d_matrix <- matrix(0, nrow = n, ncol = n)
for (i in 1:(n - 1)) {
for (k in (i + 1):n) {
d <- sqrt(sum((jm_values * (x[k, ] - x[i, ]))^2))
d_matrix[i, k] <- d
d_matrix[k, i] <- d
}
}
# Conversion to dist object
d_resultat <- as.dist(d_matrix)
# Insertion of row names (A, B, C, ...)
attr(d_resultat, "Labels") <- rownames(x)
print("Corrected distances (DCo):")
return(d_resultat)
}
```

To apply this function, copy and paste the code of the dist_corrigee function into the R console. Then, enter the data set X and finally call the dist_corrigee(X) function. *Example 2.1 (Automatic calculation of the corrected distance in R)*

Let us calculate the corrected distance (D_{Co}) between points A(2, 3) and B(4, 8).

 # 1. Preparation of data after copying and pasting the dist_corrigee function code.

X <- matrix(c(2, 3, 4, 8), nrow = 2, byrow = TRUE)

rownames(X) <- c("A", "B")

colnames(X) <- c("X1", "X2")

 # 2. Call to the dist_corrigee function

dist_corrigee(X)

The R software returns:

[1] "weights JM calculated:"

[1] 0.3333333 0.4545455

[1] "Corrected distances (DCo):"

A

B 2.368488

Therefore, the corrected distance between A and B is D_{Co}(A, B) = 2.368488. The Euclidean distance is 5.385164.

3.1.4. Fundamental property of the corrected distance

The corrected distance is distinguished from the classical Euclidean distance by an adaptive weighting driven by the correction index J_M(j) (between 0 and 1). It satisfies the following properties:

3.1.4.1. Contraction and equality condition:

The corrected distance is systematically less than or equal to the Euclidean distance. Equality is strict only when the correction index equals 1 for all dimensions presenting a deviation.

3.1.4.2. Filtering and analytical resilience:

The corrected distance minimizes the contribution of stable dimensions (where J_M is low) and only releases the full value of the deviation for critical dimensions (where J_M approaches 1). It thus avoids the oversensitivity of the Euclidean distance to minor variations.

3.1.4.3. Impact in data analysis:

The corrected distance highlights groupings invisible to the Euclidean distance, isolates distant structures (anomaly detection), and transforms a simple separation measure into a true configuration analysis.

3.2. Membership function by maximum distance

3.2.1. Maximum distance to the center of gravity

Before formalizing the maximum distance, let us recall the definition of the underlying geometric structure.

3.2.1.1. Notion of closed ball

Definition 3.1 (Closed ball)

Let I = {1, ..., i, ..., n} ⊂ $\mathbb{R}^m$ be the set of n individuals, where each individual i is represented by an m-tuple of coordinates in the space $\mathbb{R}^m$. Let G = (g_1, ..., g_j, ..., g_m) ∈ $\mathbb{R}^m$ be the center of gravity (barycenter) of the cloud of points of I. The closed ball centered at G with radius r > 0 is given by:

$$\underline{B}(G,r) = \{i \in I / d(i,G) \leq r\}, I \subset \underline{B} \qquad (3.7)$$

It is the set of individuals located at a distance of at most r from the center of gravity G.

Remark 3.1

If the closed ball is defined on the continuous space $\mathbb{R}^m$, we have:

$$\underline{B}(G,r) = \{x \in R^m / d(x,G) \leq r\} \qquad (3.8)$$

3.2.1.2. Definition and formalization of the maximum distance to the center of gravity

Definition 3.1

Let I = {1, 2, ..., n} be a set of individuals in an m-dimensional space and let G = (g_1, ..., g_j, ..., g_m) be the center of gravity (barycenter) of this cloud of points. The maximum distance to the center of gravity, denoted d_{max}, is defined by the maximization optimization of the distances separating each individual i from the center G:

$$d_{max} = max_{i \in I} d(i,G) \qquad (3.9)$$

Metric precision: The distance d(i, G) used for this maximization is not fixed. It can be: 1) The classical Euclidean distance, measuring the raw geometric deviation, or 2) the corrected distance (D_{Co}), which incorporates weighting by the correction index J_M to account for disparities and the criticality of variables.

3.2.1.3. *Illustration of the maximum distance*

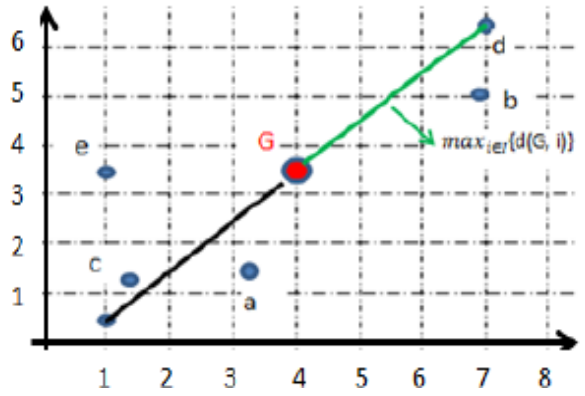


Fig 1: Illustration of the maximum distance to the center of gravity of a group

Source: Our design

3.2.1.4. Fundamental properties

1. Envelope (Minimal closed ball)

Proposition 3.1. (Minimal envelope radius)

Let I be a set of individuals included in the space $\mathbb{R}^m$ (I ⊂ $\mathbb{R}^m$) and G ∈ $\mathbb{R}^m$ the center of gravity of this point cloud. The maximum distance d_{max} represents the minimal radius of the closed ball $\underline{B}(G, d_{max})$ centered at G that contains and encompasses the entirety of the population I:

$$\forall\, i \in I,\ d(i,G) \leq d_{max} \qquad (3.10)$$

Proof. By definition, the maximum distance d_{max} is the greatest distance separating an individual i of the set I from the center of gravity G. Thus, for any individual i ∈ I, the distance d(i, G) cannot exceed this upper bound, which implies by definition that all elements of I belong to the closed ball $\underline{B}(G, d_{max})$. Moreover, d_{max} is the smallest possible radius allowing this complete inclusion, hence its minimal character. ■

2. Upper bound of the inter-individual diameter

Proposition 3.2. (Distance between two individuals and maximum distance)

Let I be a set of individuals included in the space $\mathbb{R}^m$, G the center of gravity of the point cloud of I, and d_{max} the maximum distance to the center of gravity. Consider the closed ball $\underline{B}(G, d_{max})$. Then, for any two individuals i and i' located inside this structure, the distance separating them is bounded by twice this maximum distance:

$$d(i,i') \leq 2d_{max},\ \forall\, i, i' \in \underline{B}(G, d_{max}) \qquad (3.10)$$

with

$$\underline{B}(G, d_{max}) = \{i \in I / d(i,G) \leq d_{max}\} \qquad (3.11)$$

Proof. Since $d_{max} = max_{i \in I} d(i,G)$, the diameter of the ball is equal to $2d_{max}$. According to the triangle inequality, for any individuals i, i' ∈ I, we have: d(i, i') ≤ d(i, G) + d(G, i').

Now, by definition of the closed ball, d(i, G) ≤ d_{max} and d(G, i') ≤ d_{max}. Consequently: d(i, i') ≤ $2d_{max}$.

This demonstrates that the distance between two individuals of $\underline{B}(G, d_{max})$ does not exceed $2d_{max}$. ■

Corollary 3.1.

For a closed ball of arbitrary radius r, the distance between any two individuals is bounded by twice that radius:

$$d(i,i') \leq 2r,\ \forall i, i' \in \underline{B}(G,r) \qquad (3.12)$$

2. Universal normalization (Role of universal normalizer)

Proposition 3.3. (Absolute scale factor)

Let I be a set of individuals included in the space $\mathbb{R}^m$ and d_max the maximum distance to the center of gravity. The term $2d_{max}$ serves as an absolute scale factor to bound and normalize fuzzy membership functions within the interval [0, 1].

$$0 \leq \frac{d(i,i')}{2d_{max}} \leq 1,\ \forall i, i' \in \underline{B}(G,r) \qquad (3.13)$$

Proof. Based on the property of upper-bounding the inter-individual diameter, for all individuals i, i' belonging to the closed ball $\underline{B}(G, d_{max})$, the distance separating them is bounded by $2d_{max}$ (d(i, i') ≤ $2d_{max}$). Moreover, due to the positivity of a distance, 0 ≤ d(i, i'). Dividing the entire inequality by the strictly positive term $2d_{max}$, we obtain: $0 \leq \frac{d(i,i')}{2d_{max}} \leq 1$.

This ratio thus guarantees a systematic normalization of the distance within the closed interval [0, 1], allowing it to be used directly for evaluating fuzzy membership degrees. ■

3.2.2. *Membership function by maximum distance*

3.2.2.1. *Definition and Properties*

Proposition 3.4. (Definition of the membership function by maximum distance (μ_{dm}))

Let μ_{dm} be the function associating each inter-individual distance with a degree of resemblance or membership in the interval [0, 1]:

μ_{dm}: IxI → [0, 1]

(i, i') ↦ μ_{dm}(i, i') with

$$\mu_{dm}(i,i') = \begin{cases} 1 - \frac{d(i,i')}{2d_{max}} & if\ 0 \le d(i,i') < 2d_{max} \\ 0 & if\ d(i,i') \ge 2d_{max} \end{cases} \quad (3.14)$$

Thus, the membership function can be simply expressed as:

$$\mu_{dm}(i,i') = max\left(0, 1 - \frac{d(i,i')}{2d_{max}}\right) \quad (3.15)$$

Where dmax represents the maximum distance to the center of gravity and the factor 2dmax serves as a universal normalizer, ensuring that the extreme distance between two entities never exceeds the structural upper-bound threshold.

This formulation naturally ensures that the degree of membership remains strictly bounded within the interval [0, 1] across all distance ranges.

Proof. To demonstrate that $\mu_{dm}(i,i') \in [0,1]$, we examine the two possible cases for the expression inside the maximum operator:

1) Case 1 (d(i, i') < 2d_{max}): Since d(i, i') ≥ 0 and d_{max} > 0, the ratio $\frac{d(i,i')}{2d_{max}}$ is such that:$0 \le \frac{d(i,i')}{2d_{max}} \le 1$. Multiplying by -1 and adding 1, we obtain:

In this case, the quantity is strictly positive, so the maximum operator yields:

2) Case 2 (d(i, i') ≥ 2d_{max}):

We have: $\frac{d(i,i')}{2d_{max}} \ge 1 \iff 1 - \frac{d(i,i')}{2d_{max}} \le 0$.

Since the expression is less than or equal to 0, taking the maximum with 0 gives:

$$\mu_{dm}(i,i') = max\left(0, 1 - \frac{d(i,i')}{2d_{max}}\right) = 0$$

Thus, for any pair of entities (i, i'), the membership function $\mu_{dm}(i,i')$ remains strictly bounded within the interval [0, 1]. ■

3.2.2.2. *Properties*

Proposition 3.5. (Fundamental properties of the membership function)

For all individuals i, i' ∈ K ⊆ I, with center of gravity G_K:
1) Bounds: For any individual i, the membership degree is strictly between 0 and 1:∀ i, μ_{dm}(i, K) ∈ [0, 1]

2) Normalization: The membership degree reaches its absolute maximum if and only if the individual coincides with the center of gravity:

μ_{dm}(i, K) = 1 ⟺ d(i, K) = 0 ⟺ i = G_K

3) Annihilation (Boundary): Beyond the maximum diameter of the closed structure, the membership degree vanishes completely:

If d(i, G_K) ≥ 2d_{max}, then μ_{dm}(i, K) = 0

4) Monotonicity: The function is linear and decreasing; the greater the distance from the center, the more the membership degree μ_{dm}(i, K) decreases proportionally.

3.2.2.3. *Illustration of the membership function μ_{dm}*

Let us illustrate the membership function μ_{dm} below:

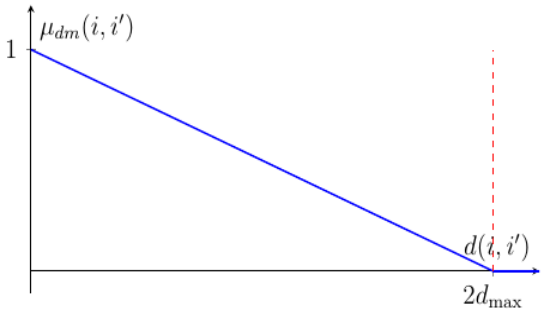


Fig 2: Illustration of the linear membership function

Source: Self-made using latex

Indeed, the abscissa axis (x) represents the distance d(i,i') (if i' = i), which varies from 0 to 2d_{max}. The ordinate axis (y) represents the membership degree μ_{dm}(i, K) = μ_{dm}(i, i'), which varies from 0 to 1. The shape of the curve: for d(i,i') = 0, we have μ_{dm}(i, i') = 1, for d(i,i') = 2d_{max}, we have μ_{dm}(i, i') = 0. Between the two, the function decreases linearly.

Thus, the graph is a decreasing straight line connecting the point (0, 1) to (2d_{max}, 0).

It is a monotonically decreasing function. Through this function, we can calculate the degree of membership of an individual relative to another individual i' knowing the distance between them and the maximum distance to the center of gravity of the point cloud.

3.3. Classical Agglomerative Hierarchical Clustering using the corrected distance (CAH_{Co})

The goal is to start from the corrected distances we proposed to build the dendrogram and cut it into the desired number of classes (for an example, see the application).

3.3.1. *Reminder of the general formula for the corrected distance*

The alert distance D_{Co}(A, B), which weights raw deviations by their relative divergence, is expressed by:

$$D_{Co}(A,B) = \sqrt{\sum_{j=1}^{m} \left[J_M(j).\left(X_{jB} - X_{jA}\right)\right]^2} \quad (3.16)$$

Where

$$J_M(j) = 1 - \frac{X_{1j}}{\underline{X_j}} \quad (3.17)$$

with X_{1j} being the minimum value observed for dimension j and $\underline{X_j}$ the arithmetic mean of dimension j.

3.3.2. *Corrected distance between two groups*

The distance between two groups is the corrected distance between their centroids, weighted by the criticality indices J_M(j):

$$D_{Co}(G_1,G_2) = \sqrt{\sum_{j=1}^{m} \left[J_M(j).\left(g_{j2} - g_{j1}\right)\right]^2} \quad (3.18)$$

This measure aggregates the gap between the "average profiles" of the two groups while taking into account the relative sensitivity of the system.

3.3.3. *Grouping conditions for individuals*

In this Agglomerative Hierarchical Classification (AHC) approach, the conditions for grouping individuals (or groups) rely on minimizing the corrected distance between individuals and/or groups (group centroids), modulated by the relative sensitivity of the system via J_M.

3.3.4. *Implementation of CAH_{Co} in R*

3.3.4.1. *Function cah_co*

```
cah_co <- function(X, epsilon = 1, methode =
"centroid") {
# 1. Conversion and dimensions
x <- as.matrix(X)
```

```
n <- nrow(x)
m <- ncol(x)
# 2. Calculation of J_M(j) for each column j
jm_values <- sapply(1:m, function(j) {
col_j <- x[, j]
moyenne_j <- mean(col_j)
min_j <- min(col_j)
if (moyenne_j != 0) {
return(1 - (min_j / moyenne_j))
} else {
return(1 - ((min_j + epsilon) / epsilon))
}
})
print("Calculated JM weights:")
print(jm_values)
# 3. Calculation of the corrected distance matrix
d_matrix <- matrix(0, nrow = n, ncol = n)
for (i in 1:(n - 1)) {
for (k in (i + 1):n) {
d <- sqrt(sum((jm_values * (x[k, ] - x[i, ])) ^ 2))
d_matrix[i, k] <- d
d_matrix[k, i] <- d
}
}
# Conversion to dist object
d_resultat <- as.dist(d_matrix)
if (!is.null(rownames(x))) {
attr(d_resultat, "Labels") <- rownames(x)
}
#  4.  Hierarchical  classification  and  dendrogram
plotting
hc <- hclust(d_resultat, method = methode)
plot(hc,
main = "Dendrogram using Corrected distance", sub =
"", xlab = "Groups", ylab = "Corrected distance")
# 5. Interactive prompt to the user for the number of
classes to cut the dendrogram
k_classes <- as.numeric(readline(prompt = "Into how
```

```
many classes do you want to cut the dendrogram? : "))
# Cutting and displaying corresponding groups
groupes <- cutree(hc, k = k_classes)
print(paste("Result  of  cutting  into",  k_classes,
"classes:"))
print(groupes)

# Visual addition of cut rectangles on the dendrogram
rect.hclust(hc, k = k_classes, border = 2: (k_classes
+ 1))
# Invisibly returns the hc object and groups
invisible(list(hclust = hc, groupes = groupes))
}
```

3.3.4.2. Automatic calculation with cah_co function

Individuals are defined in the space $\mathbb{R}^2$ by the variables y_1 and y_2: a = (6, 4); b = (7, 8); d = (2, 3); e = (8, 9); f = (2, 6). Simply copy and paste the code of the function above into the R console and then input the data to be classified.

 # --- Application ---

 # Definition of initial data (individuals and variables)
X <- matrix(c(6, 4, 7, 8, 2, 3, 8, 9, 2, 6), nrow = 5, byrow = TRUE, dimnames = list(c("a", "b", "d", "e", "f"), c("y1", "y2")))

 # Call to the cah_co function

cah_co(X)

After validation, R returns the JM values, the dendrogram, and the prompt asking for the number n at which to cut this dendrogram. Once the number n is entered, the dendrogram is displayed, cut into n classes (In our case n=2).

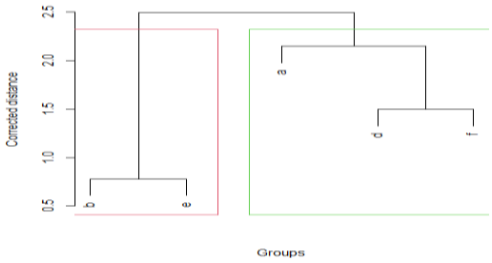


Fig. 3. Dendrogram based on corrected distance

Source: Self-made using R

[1] "Result of cutting into 2 classes:"

a	*b*	*d*	*e*	*f*
1	*2*	*1*	*2*	*1*

K_1={a, d, f}; K_2={b, e}

3.4. Fuzzy Agglomerative Hierarchical Classification based on maximum distance to the center of gravity (CAHFD)

We propose a fuzzy agglomerative hierarchical classification method named Fuzzy Agglomerative Hierarchical Classification based on the maximum distance to the center of gravity, abbreviated as CAHFD. It is therefore a fuzzy extension of classical hierarchical classification. It relies on transforming inter-individual distances into degrees of membership, according to a membership function defined from the maximum distance to the center of gravity.

This method combines the rigor of AHC, the flexibility of membership degrees from fuzzy logic, and the possibility of using a classical or corrected distance between individuals located within a closed ball.

3.4.1. Steps of the CAHFD method

3.4.1.1. Preliminary step: Variable analysis and homogeneity

-Measurement on the same scale: The variables X_j (with j = 1, …, m) must be measured on the same scale (for example, on the same grading or scoring scale).

-Homogeneity verification: If the variables are homogeneous, this avoids the preliminary transformation into standardized variables, thereby allowing the raw original values to be preserved directly for distance calculations.

3.4.1.2. Calculation of arithmetic means per dimension

For each dimension j (with j = 1, …, m), the arithmetic mean of the observations is calculated:

$$\underline{X_j} = \frac{1}{n}\sum_{i=1}^{n} X_{ij} \qquad (3.19)$$

The center of gravity G of all individuals is defined by the means of each dimension:

$$G = (\underline{X_1}, .., \underline{X_m}) \qquad (3.20)$$

3.4.1.3. Calculation of basic characteristics and weighting coefficients $J_M(j)$

-Minimum value per dimension: $X_{1j} = \min_{i \in \{1, \ldots, n\}} \{X_{ij}\}$

- Original weighting factor:

$$J_M(j) = 1 - \frac{X_{1j}}{\underline{X_j}} \qquad (3.21)$$

- Weighting used in the corrected distance (squared factor): $[J_M(j)]^2$

3.4.1.4. Calculation of inter-individual corrected distance and distance to the center of gravity

- Corrected distance between two individuals i and i':

$$D_{Co}(i, i') = \sqrt{\sum_{j=1}^{m} \left[J_M(j).\left(X_{ij} - X_{i'j}\right)\right]^2} \qquad (3.22)$$

-Corrected distance of each individual to the center of gravity G:

$$D_{Co}(i, G) = \sqrt{\sum_{j=1}^{m} \left[J_M(j).\left(\underline{X_j} - X_{ij}\right)\right]^2} \qquad (3.23)$$

-Determination of the maximum distance:

$d_{max} = \max_{i \in I} D_{Co}(i, G)$ (3.24)

-Calculation of twice the maximum distance: $2d_{max}$

3.4.1.5. *Transformation of distances into fuzzy membership degrees*

The fuzzy membership degree $\mu_{dm}(i,i')$ between two individuals i and i' is obtained by the similarity relation:

$$\mu_{dm}(i,i') = 1 - \frac{D_{Co}(i,i')}{2d_{max}} \quad (3.25)$$

3.4.1.6. *Fuzzy grouping and aggregation criterion (Complete linkage)*

The degree of adhesion between two groups K_1 and K_2 is defined by:

$$\mu_{dm}(K_1,K_2) = max\{\mu_{dm}(i,i')\}, i \in K_1, i' \in K_2 \quad (3.26)$$

3.4.1.7. *Complement of the similarity*

This measure represents the transition from similarity (or membership degree) $\mu_{dm}(K_k)$ to a dissimilarity (or complementarity scale).

$$Diss(K_k) = 1 - \mu_{dm}(K_k) \quad (3.27)$$

3.4.1.8. *Graphical representation (Fuzzy dendrogram) and cut*

We use the complements of groups degrees to construct CAHFD Fuzzy dendrogram,

For manually draw the CAHFD Fuzzy Dendrogram we proceed as follows:

1) Height values (Y-axis): different complements of similarity.

*2) Horizontal layout (X-axis):*Align the individuals from left to right in the grouping order.

3) Drawing the branches in connecting individuals according to grouping level.

4) Dendrogram cut: Draw a horizontal line to divide the tree into q classes (C_1, …, C_k, …, C_q)

3.4.1.9. *Determination of normalized membership degrees*

The normalization of this degree for each class C_k (with q classes) is given by:

$$\mu_{norm}(i,C_k) = \frac{\mu_{dm}(i,C_k)}{\sum_{r=1}^{q} \mu_{dm}(i,C_r)} \quad (3.28)$$

with

$$\sum_{k=1}^{q} \mu_{norm}(i,C_k) = 1 \quad (3.29)$$

the global normalization condition and

$$\mu_{dm}(i,C_k) = max_{i' \in C_k}\{\mu_{dm}(i,i')\} \in [0,1[\quad (3.30)$$

the raw adherence degree of an individual i to a class C_k.

3.4.2. *Implementation of CAHFD*

3.4.2.1. *R function for CAHFD*

We proposed a R function for CAHFD nommed cahfd_dendrogram and presented as follows:

```
cahfd_dendrogram <- function(donnees) {
  X <- as.matrix(donnees)
  n <- nrow(X)
  noms <- rownames(X)
  # Interactive prompt for the number of classes q (allowed up to n)
  reponse <- readline(prompt = "How many classes do you want to cut into? ")
  q <- as.integer(reponse)
 if (is.na(q) || q < 2 || q > n) {
    stop("Error: The number of classes q must be an integer between 2 and n.")
  }
# 1: Basic CAHFD calculations
  x_bar <- colMeans(X)
  min_x <- apply(X, 2, min)
  JM <- 1 - (min_x / x_bar)
  JM_sq <- JM^2
  G <- x_bar
  d_max <- max(sqrt(rowSums(sweep(X, 2, G, "-")^2 * matrix(JM_sq, nrow = n, ncol = ncol(X), byrow = TRUE))))
  two_d_max <- 2 * d_max

  D_Co_mat <- matrix(0, nrow = n, ncol = n, dimnames = list(noms, noms))
  for (i in 1:n) {
    for (j in 1:n) {
      D_Co_mat[i, j] <- sqrt(sum(JM_sq * (X[i, ] - X[j, ])^2))
    }
  }
  Mu_dm <- 1 - (D_Co_mat / two_d_max)
# 2. Initialization
  groupes <- as.list(noms)
  names(groupes) <- noms
  Sim_mat <- Mu_dm
  hauteurs_fusions <- numeric(n - 1)
  fusions_arbre <- vector("list", n - 1)

  # Preliminary simulation to retrieve the exact branch order
  Sim_mat_sim <- Mu_dm
  groupes_sim <- as.list(noms)
  names(groupes_sim) <- noms

  for (step in 1:(n - 1)) {
    k_g <- length(groupes_sim)
    diag(Sim_mat_sim) <- -Inf
    Sim_mat_sim[is.na(Sim_mat_sim)] <- -Inf
    max_val <- -Inf
    best_pair <- c(1, 2)
    for (i in 1:k_g) {
      for (j in 1:k_g) {
        if (i < j && Sim_mat_sim[i, j] >= max_val) {
          max_val <- Sim_mat_sim[i, j]
          best_pair <- c(i, j)
        }
      }
    }
    g1 <- best_pair[1]
    g2 <- best_pair[2]
    fusions_arbre[[step]] <- c(names(groupes_sim)[g1], names(groupes_sim)[g2])
    nouveau_nom <- paste0("F", step)
    nouveau_membre <- c(groupes_sim[[g1]], groupes_sim[[g2]])
    groupes_sim[[nouveau_nom]] <- nouveau_membre
    groupes_sim[c(g1, g2)] <- NULL

    k_new <- length(groupes_sim)
    noms_actuels <- names(groupes_sim)
    nouv_sim <- matrix(0, nrow = k_new, ncol = k_new)
    for (i in 1:k_new) {
      for (j in 1:k_new) {
        if (i == j) {
          nouv_sim[i, j] <- 1
        } else {
          membres_i <- groupes_sim[[i]]
          membres_j <- groupes_sim[[j]]
          nouv_sim[i, j] <- max(Mu_dm[membres_i, membres_j])
        }
      }
    }
    Sim_mat_sim <- nouv_sim
    colnames(Sim_mat_sim) <- rownames(Sim_mat_sim) <- noms_actuels
  }
# Recursive function to order leaves without crossing
  obtenir_ordre_feuilles <- function(noeud_idx) {
    if (noeud_idx < 0) return(noms[-noeud_idx])
    paire <- fusions_arbre[[noeud_idx]]
    gauche <- if (paire[1] %in% noms) paire[1] else obtenir_ordre_feuilles(as.numeric(sub("F", "", paire[1])))
    droite <- if (paire[2] %in% noms) paire[2] else obtenir_ordre_feuilles(as.numeric(sub("F", "", paire[2])))
    return(c(gauche, droite))
  }

  ordre_optimal <- unlist(obtenir_ordre_feuilles(n - 1))
  pos_x <- numeric(n)
  names(pos_x) <- ordre_optimal
  for (idx in 1:n) {
    pos_x[ordre_optimal[idx]] <- idx
  }
```

```
  h_actuelle <- rep(0, n)
  names(h_actuelle) <- noms

  # 3. Graphic configuration
  plot.new()
  plot.window(xlim = c(0.5, n + 0.5), ylim = c(0, 1))
  axis(2, las = 1)
  title(main = "CAHFD Fuzzy Hierarchical Dendrogram",
xlab = "Individuals", ylab = "Dissimilarity (1 -
mu_dm)")

  final_classes <- NULL
# Special case if q == n
  if (q == n) {
    final_classes <- as.list(noms)
    names(final_classes) <- paste0("{", noms, "}")
  }

  # 4. Real agglomeration loop and graphical plotting
  for (step in 1:(n - 1)) {
    k_g <- length(groupes)

    if (k_g == q && is.null(final_classes)) {
      final_classes <- vector("list", q)
      class_labels <- character(q)
      for (idx in 1:q) {
        final_classes[[idx]] <- groupes[[idx]]
            class_labels[idx]     <-     paste0("{",
paste(groupes[[idx]], collapse = ", "), "}")
      }
      names(final_classes) <- class_labels
    }
diag(Sim_mat) <- -Inf
    Sim_mat[is.na(Sim_mat)] <- -Inf
    max_val <- -Inf
    best_pair <- c(1, 2)
    for (i in 1:k_g) {
      for (j in 1:k_g) {
          if (i < j && Sim_mat[i, j] >= max_val) {
          max_val <- Sim_mat[i, j]
          best_pair <- c(i, j)
```

```
   # Final display of labels in strict fusion order
  axis(1, at = pos_x[noms], labels = noms, tick =
FALSE, line = -0.5)
# 5. Calculation and Display of Normalized Membership
Degrees
  class_names <- names(final_classes)
   Mu_dm_classes <- matrix(0, nrow = n, ncol = q,
dimnames = list(noms, class_names))
  for (k in 1:q) {
    k_members <- final_classes[[k]]
    for (i in 1:n) {
      indiv_name <- noms[i]
      if (indiv_name %in% k_members) {
           other_members    <-    setdiff(k_members,
indiv_name)
        if (length(other_members) > 0) {
                Mu_dm_classes[i,        k]        <-
max(Mu_dm[indiv_name, other_members])
        } else {
          Mu_dm_classes[i, k] <- 1
        }
      } else {
        Mu_dm_classes[i, k] <- max(Mu_dm[indiv_name,
k_members])
      }
    }
  }
Mu_norm       <-       sweep(Mu_dm_classes,       1,
rowSums(Mu_dm_classes), "/")

  cat(paste0(" NORMALIZED MEMBERSHIP DEGREES (q = ",
q, " classes) \n"))
print(round(Mu_norm, 4))

  results <- list(
    q = q,
 Mu_dm = Mu_dm,
     Mu_dm_classes = Mu_dm_classes,
    Mu_norm = Mu_norm,
    final_classes = final_classes
  )

  return(invisible(results))
}
       }
      }
    }
    g1 <- best_pair[1]
```

```
    g2 <- best_pair[2]

    h_fusion <- 1 - max_val
    hauteurs_fusions[step] <- h_fusion
    nom_g1 <- names(groupes)[g1]
    nom_g2 <- names(groupes)[g2]
    x1 <- pos_x[nom_g1]
    x2 <- pos_x[nom_g2]
    h1 <- h_actuelle[nom_g1]
    h2 <- h_actuelle[nom_g2]

    # Plotting proper segments
    segments(x1, h1, x1, h_fusion, col = "blue", lwd
= 2)
    segments(x2, h2, x2, h_fusion, col = "blue", lwd
= 2)
    segments(x1, h_fusion, x2, h_fusion, col = "blue",
lwd = 2)

    nouveau_nom <- paste0("F", step)
    nouveau_membre <- c(groupes[[g1]], groupes[[g2]])
    nouveau_x <- (x1 + x2) / 2
    pos_x[nouveau_nom] <- nouveau_x

    h_actuelle[nouveau_nom] <- h_fusion
    groupes[[nouveau_nom]] <- nouveau_membre

 groupes[c(g1, g2)] <- NULL

    k_new <- length(groupes)
    noms_actuels <- names(groupes)
nouv_sim <- matrix(0, nrow = k_new, ncol = k_new)
for (i in 1:k_new) {
      for (j in 1:k_new) {
        if (i == j) {
          nouv_sim[i, j] <- 1
        } else {
          membres_i <- groupes[[i]]
          membres_j <- groupes[[j]]
           nouv_sim[i,   j]   <-   max(Mu_dm[membres_i,
membres_j])
        }
      }
    }
    Sim_mat <- nouv_sim
    colnames(Sim_mat) <- rownames(Sim_mat) <-
   noms_actuels
  }
if (is.null(final_classes)) {
    stop("The specified class level q could not be
reached.")
  }
```

3.4.2.2. *Complexity of CAHFD algorithm*

The overall complexity of the CAHFD method is broken down into two main phases:

- Initial Phase: The computation of the corrected distances over the m variables results in a temporal complexity of $O(n^2xm)$ and a spatial complexity of $O(n^2)$ for matrix storage.

- Agglomeration Phase: The iterative agglomeration algorithm and fusion simulation present a complexity of $O(n^3)$ due to the successive recalculations of the similarity matrices.

In summary, the overall temporal complexity of the CAHFD method is $O(n^3 + n^2xm)$, while its spatial complexity is $O(n^2)$.

3.4.3. *Automatic resolution*

3.4.3.1. *Problem statement*

Five students a, b, d, e, f obtained grades out of 10 points respectively in Mathematics (y_1) and French (y_2): a = (6, 4); b = (7, 8); d = (2, 3); e = (8, 9) et f = (2, 6). Classify using the CAHFD method under the R software.

3.4.3.2. *Solution using R function cahfd_dendrogram*

 # Simply copy and paste the code of the function above (cahfd_dendrogram) into the R console and then input the data to be classified.

```
notes_etudiants <- data.frame(
Maths = c(6, 7, 2, 8, 2),
Francais = c(4, 8, 3, 9, 6)
)
rownames(notes_etudiants) <- c("a", "b", "d", "e", "f")
cahfd_dendrogram(notes_etudiants)
```

#After validation, R automatically returns the question: How many classes do you want to cut into?

#We write the number of de classes we need and valide, R returns results of the CAHFD method, identical to those calculated manually, including the normalized membership degrees to classes K_1 = {d, f} and K_2 = {a, b, e}, as well as the fuzzy hierarchical dendrogram.

NORMALIZED MEMBERSHIP DEGREES (q = 2 classes)

	{d, f}	*{a, b, e}*
a	*0.4624*	*0.5376*
b	*0.2807*	*0.7193*
d	*0.5878*	*0.4122*
e	*0.1676*	*0.8324*
f	*0.6043*	*0.3957*

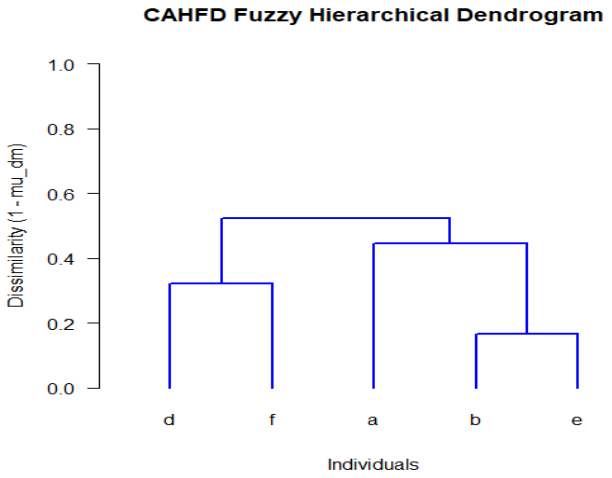


Fig 5. CAHFD Fuzzy Dendrogram using corrected distance

Source: Author using R

Remark 3.1. On hclust vs. Step-by-Step CAHFD: If we use R's built-in hclust function on the fuzzy dissimilarity matrix, it applies standard agglomerative hierarchical clustering algorithms (such as complete linkage). While this provides a clean and standard dendrogram, it relies on standard ultrametric tree-building routines. In contrast, our custom step-by-step CAHFD implementation strictly follows the exact fuzzy logic rules, maximum membership linkage, and sequential merging specific to the methodology, ensuring complete alignment with the manual theoretical framework.

3.4.4. *Manual resolution of the CAHFD method*

3.4.4.1. *Problem statement (Cf. 3.4.3.1.)*

3.4.4.2. *Preliminary step: Variable analysis and homogeneity*

The variables (y_1 for Math and y_2 for French) are measured on the same scale (grades out of 10) and are homogeneous. Therefore, transforming the raw variables into standardized variables is unnecessary.

3.4.4.3. *Calculation of arithmetic means per dimension*

For each dimension (j = 1 for Math, j = 2 for French):

$\underline{X_1} = \frac{6+7+2+8+2}{5} = \frac{25}{5} = 5,\ \underline{X_2} = 6.$

The center of gravity G is: $G = (\underline{X_1}, \underline{X_2}) = (5, 6)$.

3.4.4.4. *Calculation of base characteristics and weighting coefficients $J_M(j)$*

- Minimum values: $\min_i(X_{i1}) = 2$ and $\min_i(X_{i2}) = 3$.
- Weighting factors or correction factors:

$J_M(1) = 1 - \frac{2}{5} = 0.6, J_M(2) = 1 - \frac{3}{6} = 0.5.$

-Squares of the weighting factors:

$[J_M(1)]^2 = (0.6)^2 = 0.36, [J_M(2)]^2 = (0.5)^2 = 0.25.$

3.4.4.5. *Calculation of corrected distances between individuals and to the center of gravity*

The general formula for the corrected distance is:

$$D_{Co}(i,i') = \sqrt{0.36(X_{i1} - X_{i'1})^2 + 0.25(X_{i2} - X_{i'2})^2}$$

- Inter-individual distances:

$D_{Co}(i,i')$	a	b	d	e	f
a	0	2.088	2.451	2.773	2.600
b	2.088	0	3.905	0.781	3.162
d	2.451	3.905	0	4.686	1.500
e	2.773	0.781	4.686	0	3.900
f	2.600	3.162	1.500	3.900	0

- Corrected distances to the center of gravity G(5, 6):

$D_{Co}(G,i)$	a	b	d	e	f
G	1.166	1.562	2.343	**2.343**	1.800

- Maximum distance and its double:

$d_{max} = 2.343 \Rightarrow 2d_{max} = 4.686.$

3.4.4.6. *Transformation of distances into Fuzzy membership degrees*

$\mu_{dm}(i,i')$	a	b	d	e	f
a	1	0.554	0.477	0.408	0.445
b	0.554	1	0.167	**0.833**	0.325
d	0.477	0.167	1	0.000	0.680
e	0.408	**0.833**	0.000	1	0.168
f	0.445	0.325	0.680	0.168	1

3.4.4.7. *Fuzzy clustering and aggregation criterion (Complete linkage)*

$\mu_{dm}(i, K_k) = max_{i \in K_k}\{\mu_{dm}(i,i')\} \in [0,1[$

(1) 1st grouping: $\mu_{dm}(b,e) = 0.833$ then K_1 = {b, e}.

(2) 2d grouping: $\mu_{dm}(d,f) = 0.680$ K_2 = {d, f}.

Indeed, $\mu_{dm}(a,\{b,e\}) = \max\{\mu_{dm}(a,b), \mu_{dm}(a,e)\}$

=max {0.554, 0.408}= 0.554

$\mu_{dm}(d,\{b,e\}) = \max\{\mu_{dm}(d,b),\ \mu_{dm}(d,e)\}$

=max {0.167, 0.000}= 0.167

$\mu_{dm}(f,\{b,e\}) = \max\{\mu_{dm}(f,b), \mu_{dm}(f,e)\}$

=max { 0.325, 0.168}= 0.325

$\mu_{dm}(i,i')$	a	{b, e}	d	f
a	1	0.554	0.477	0.445
{b, e}	0.554	1	0.167	0.325
d	0.477	0.167	1	**0.680**
f	0.445	0.325	**0.680**	1

$\mu_{dm}(d,f) = 0.680$ K_2 = {d, f}.

(3) 3th grouping: $\mu_{dm}(a,\{b,e\}) = 0.554$ K_3 = {a, {b, e}}.

Indeed, $\mu_{dm}(a,\{d,f\}) = \max\{\mu_{dm}(a,d), \mu_{dm}(a,f)\}$

=max {0.477, 0.445}= 0.477

$\mu_{dm}(i,i')$	a	{b, e}	{d, f}
a	1	**0.554**	0.477
{b, e}	**0.554**	1	0.167
{d, f}	0.477	0.167	1

$\mu_{dm}(a,\{b,e\}) = 0.554$ K_3 = {a, {b, e}}.

(4) 4th grouping: $\mu_{dm}(\{d,f\},\{a,\{b,e\}\}) = 0.477$

K_4 = {{d, f}, {a, {b, e}}}.

Indeed, $\mu_{dm}(\{d,f\},\{a,\{b,e\}\})$=

max$\{\mu_{dm}(d,a), \mu_{dm}(d,\{b,e\}), \mu_{dm}(f,a), \mu_{dm}(f,\{b,e\})\}$

= max{0.477, 0.167, 0.445, 0.325}=0.477

$\mu_{dm}(i,i')$	{a, {b, e}}	{d, f}
{a, {b, e}}	1	**0.477**
{d, f}	**0.477**	1

$\mu_{dm}(\{d,f\},\{a,\{b,e\}\}) = 0.477$ K_4 = {{d, f}, {a, {b, e}}}.

3.4.4.8. *Complement of the similarity*

This measure represents the transition from similarity (or membership degree) $\mu_{dm}(K_j)$ to a dissimilarity (or complementarity scale).

$$Diss(K_j) = 1 - \mu_{dm}(K_j)$$

-Diss(K_1) = Diss({b, e}) =1 - 0.833 = 0.167
-Diss(K_2) = Diss({d, f}) =1 - 0.680 = 0.320
-Diss(K_3) = Diss({a, {b, e}}) =1 - 0.554 = 0.446
- Diss(K_4) = Diss({{d, f}, {a, {b, e}}})=1-0.477 = 0.523

3.4.4.9. *Graphical representation (Fuzzy dendrogram) and cut*

We use the complements (or dissimilarities) of groups degrees to construct CAHFD Fuzzy dendrogram.

For manually draw the CAHFD Fuzzy Dendrogram we proceed as follows:

1) Height values (Y-axis): Diss(K_1) = 0.167, Diss (K_2) = 0.320, Diss (K_3) = 0.446, Diss (K_4) = 0.523.

2) Horizontal layout (X-axis): Align the individuals from left to right in the order: d, f, b, e, a.

3) Drawing the branches: Level 0.167: connect b and e, level 0.320: connect d and f, level 0.446: connect cluster {b, e} to individual a, level 0.523: connect cluster {d, f} to cluster {a, b, e} to complete the tree.

4) Dendrogram cut: Draw a horizontal line (e.g., at 0.5) to divide the tree into two classes: {d, f} and {a, b, e}.

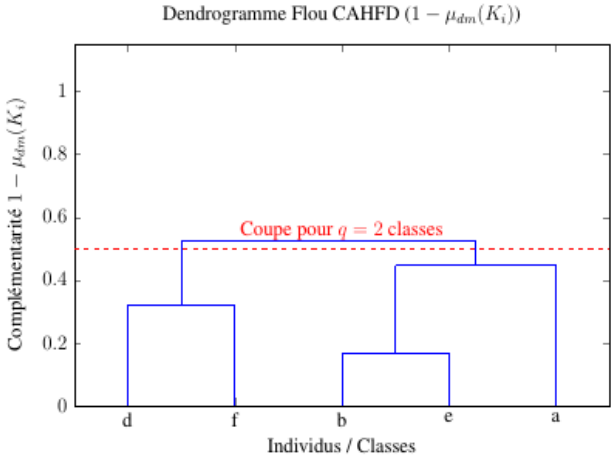


Fig. 4. CAHFD fuzzy dendrogram cut in two classes

Source: Our design using latex code

3.4.4.10. *Determination of normalized membership degrees*

The normalization of this degree for each class C_k (with q classes) is given by:

$$\mu_{norm}(i,C_k) = \frac{\mu_{dm}(i,C_k)}{\sum_{r=1}^{q} \mu_{dm}(i,C_r)}$$

with

$\sum_{k=1}^{q} \mu_{norm}(i,C_k) = 1$ the global normalization condition and

$\mu_{dm}(i,C_k) = max_{i' \in C_k}\{\mu_{dm}(i,i')\} \in [0,1[$ the raw adherence degree of an individual i to a class C_k.

Let's calculate normalized membership degrees by individual:

* For individual a:

- Raw degrees: To C_1 = {d, f}:

$\mu_{dm}(a,C_1) = max(\mu_{dm}(a,\ d),\ \mu_{dm}(a,f))$ =max(0.477, 0.445) = 0.477

To C_2 = {b, e, a}:

$\mu_{dm}(a,C_2) = max(\mu_{dm}(a,\ b),\ \mu_{dm}(a,e), \mu_{dm}(a,a))$

=max(0.554, 0.408, 1.0) = 0.554 ∈ [0, 1[

- Sum of raw degrees: 0.477 + 0.554 = 1.031

- Normalized degrees:

$$\mu_{norm}(a, C_1) = \frac{0.477}{1.031} = 0.463,$$

$$\mu_{norm}(a, C_2) = \frac{0.554}{1.031} = 0.537$$

Verification:

$$\sum_{j=1}^{2} \mu_{norm}(a, C_j) = 0.463 + 0.537 = 1.0$$

* For individual b: The normalized degrees are

$$\mu_{norm}(b, C_1) = \frac{0.325}{1.158} = 0.281,$$

$$\mu_{norm}(b, C_2) = \frac{0.833}{1.158} = 0.719$$

* For individual d: The normalized degrees are

$$\mu_{norm}(d, C_1) = \frac{0.680}{1.157} = 0.588,$$

$$\mu_{norm}(d, C_2) = \frac{0.477}{1.157} = 0.412$$

* For individual e: The normalized degrees are

$$\mu_{norm}(e, C_1) = \frac{0.168}{1.001} = 0.168,$$

* For individual f: The normalized degrees are

$$\mu_{norm}(f, C_1) = \frac{0.680}{1.125} = 0.604,$$

$$\mu_{norm}(f, C_2) = \frac{0.445}{1.125} = 0.396$$

Therefore, CAHFD normalized membership degrees are:

Individuals	C_1 ={d, f}	C_2 = {b, e, a}
a	0.463	0.537
b	0.281	0.719
d	0.588	0.412
e	0.168	0.832
f	0.604	0.396

Note: These results can be used within the framework of the PRRC/F and PRRG/F methods of our fair division approach APCR in a fuzzy environment [16], [17], [18], [19], [20].

3.4.5. *Synthetic table of Fuzzy c-means and CAHFD results*

	Fuzzy c-means		CAHFD	
Indivi-duals	{b, e}	{a, d, f}	{d, f}	{a, b, e}
a=(6, 4)	0.346	0.654	0.463	0.537
b=(7, 8)	0.994	0.006	0.281	0.719
d=(2, 3)	0.041	0.959	0.588	0.412
e=(8, 9)	0.980	0.020	0.168	0.832
f=(2, 6)	0.091	0.909	0.604	0.396
J_M	*0.916*	*0.988*	*0.601*	*0.316*

Note: In light of the J_M results above, for Fuzzy c-means, the values are very high (0.916 and 0.988). This shows that this method produces membership degree distributions that are very sharp, close to the extremes (0 or 1), which reflects more unequal results.

For CAHFD: The J_M values are significantly lower (0.601 and 0.316). This indicates that the method generates more moderate, nuanced, and balanced membership degrees, thus proving to be less unequal and more favorable to intermediate or boundary profiles.

3.4.6. *Complementary graphic: Diagram of normalized membership degrees in CAHFD*

To visually illustrate the continuity and overlap inherent in CAHFD, we can design a stacked bar chart or an individual membership profile highlighting the continuous and overlapping nature of fuzzy subsets for each individual. This is a graphical representation of the normalized membership degrees:

C_1={d, f}=(0.463, 0.281, 0.588, 0.168, 0.604).

C_2={a, b, e}=(0.537,0.719, 0.412, 0.832, 0.396)

Below is the structure of the code (in R) to plot this diagram (Stacked bar chart):

```
# Individual data
individus <- c('a', 'b', 'd', 'e', 'f')
C1 <- c(0.463, 0.281, 0.588, 0.168, 0.604)
C2 <- c(0.537,0.719, 0.412, 0.832, 0.396)

# Creating a matrice for the stacked bar chart
matrice_data <- rbind(C1, C2)
colnames(matrice_data) <- individus
# Plotting the graph
```

```
barplot(matrice_data,
        main = " Stacked bar chart of normalized membership degrees in CAHFD",
        ylab = "Normalized membership degree",
        xlab = "Individuals (i)",
        col = c("#2b5c8f", "#d95f02"),
        ylim = c(0, 1.2),
        legend.text = c("Class C1 {d, f}", "Class C2 {a, b, e}"),
        args.legend = list(x = "topright"))
```

R returns the following graph:

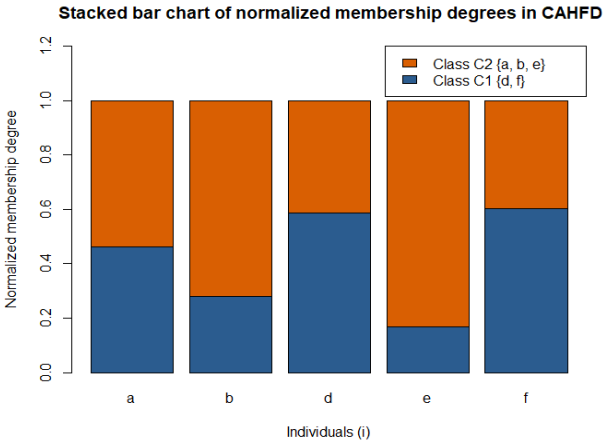


Fig. 6: CAHFD stacked bar chart

Source: Author using R software

Note: The stacked bar chart clearly highlights the fuzzy nature of the partition, showing how each individual's total membership (normalized to 1) is distributed between the two reference classes: Class C_1 ({d, f}, represented in blue) and Class C_2 (\{a, b, e\}, represented in orange/red). Individuals b and e emerge as the most characteristic members of Class C_2 ({a, b, e}), whereas d and f position themselves more clearly within Class C_1 ({d, f}). Individual *a* occupies a more balanced, intermediate position between the two clusters, illustrating the smooth transitions characteristic of fuzzy clustering methods.

4. DISCUSSION

The discussion focuses on the existing methods (Euclidean distance, Fuzzy c-means) in the literature and our innovative methods (Corrected distance, membership function via maximum distance and Fuzzy Agglomerative Hierarchical Clustering based on maximum Distance at the center of gravity (CAHFD)).

1) Euclidean distance versus corrected distance.

The comparative analysis between Euclidean distance and Corrected distance is the follow:

(1) Distance Contraction: It is clearly observed that the corrected distance (2.368) is significantly lower than the classical Euclidean distance (5.385). This perfectly illustrates fundamental property (Contraction and equality condition),

which states that the corrected distance is systematically less than or equal to the Euclidean distance thanks to weighting by the J_M index (which filters the impact of stable dimensions).

(2) Mitigation of Oversensitivity: The classical Euclidean distance sharply amplifies discrepancies (notably on the second dimension, where the difference is 5). The corrected distance weights this discrepancy using the J_M weights (which are 0.601 and 0.316 here), thereby avoiding oversensitivity to minor variations and offering a more nuanced, equitable reading of the spatial configuration.

2) Membership function via maximum distance.

Unlike traditional fuzzy clustering techniques—such as Fuzzy c-means, where membership degrees rely on exponential ratios of standard Euclidean distances governed by an arbitrary fuzzification coefficient—our proposed approach derives membership entirely from the structural topology of the data point cloud. By utilizing the maximum distance to the center of gravity (d_{max}), the membership function (μ_{dm}) establishes a rigorous, natural link between individual dispersion and cluster boundaries.

This construction ensures key analytical properties: membership reaches its absolute maximum ($\mu_{dm} = 1$) exclusively at the barycentre, decreases linearly and proportionally with distance, and undergoes complete annihilation ($\mu_{dm} = 0$) beyond the critical structural diameter ($2d_{max}$). Consequently, this methodology bypasses empirical fuzzification parameters, offering a transparent, geometrically bounded framework that faithfully captures the true ambiguity of boundary objects.

3) Fuzzy c-means method versus CAHFD method

The comparative analysis between the fuzzy c-means method and the CAHFD method is the following:

(1) Fundamental similarities

- Both approaches fall under unsupervised automatic classification.
- Both produce the same individuals in the selected cluster, thereby validating a coherent base partition into two distinct groups (C_1 and C_2).

- Both lead to a flexible modeling where the sum of an individual's membership degrees across all classes equals 1

(2) Major methodological divergences

- Structure type: The fuzzy c-means method is non-hierarchical and proceeds by iterative partitioning around mobile centers, whereas CAHFD is hierarchical (bottom-up) via successive fusions.

- Parameterization: Fuzzy c-means yields results that depend on the chosen fuzzification parameter value. In contrast, CAHFD is free from any arbitrary parameter, constituting a noticeable gain in accuracy and time.

- Visual representation: Fuzzy c-means produces no dendrogram, whereas CAHFD provides one, allowing a clear visualization of the chronology and aggregation order of subgroups.

- Behavior of membership degrees: Most cross-membership degrees are minimal with fuzzy c-means, which generates sharper distributions. CAHFD produces improved and nuanced membership degrees.

(3) Comparative analysis using the inequality index J_M

- For fuzzy C-means, the J_M values are very high (0.916 and 0.988). This shows that this method produces very sharp membership degree distributions, close to the extremes (0 or 1), reflecting more unequal results.

- For CAHFD: The J_M values are significantly lower (0.601 and 0.316). This indicates that the method generates more moderate, nuanced, and balanced membership degrees, thus proving to be less unequal and more favorable to intermediate or boundary profiles.

4) Analytical interpretation of individuals (Typicality versus ambiguity)

-Strongly anchored individuals (b, e) : They form the core of the classes with clear memberships, although moderated by the continuous nature of the fuzzy model (e.g., individual e strongly belongs to Class C2 with 0.832, and b with 0.719).

-Boundary or pivot individuals (a, d, f) : Individuals such as *a* (0.537 in C2 vs 0.463 in C_1), *d* (0.588 in C1 vs 0.412 in C2), and *f* (0.604 in C1 vs 0.396 in C2) illustrate the power of CAHFD by revealing topological hesitation and smooth transitions, whereas geometric partitioning methods harden the decision.

5. CONCLUSION

In summary, the comparison between fuzzy c-means and CAHFD highlights a fundamental choice in classification: on one hand, geometric optimization based on moving centers; on the other hand, tree-based topological exploration.

While fuzzy c-means seeks rigid compactness—at the risk of making fuzziness disappear by yielding overly extreme scores—CAHFD relies on the dendrogram and the maximum distance to the center of gravity. This method eliminates the arbitrary choice of a fuzzification parameter and provides a true membership function bounded by natural limits ($2d_{max}$). Thanks to the moderation of the J_M index and the corrected distance, CAHFD stands out as an effective tool for properly capturing the complexity and nuances of objects located at the boundaries of groups.

6. ACKNOWLEDGMENTS

We thank professor MABELA Rostin of Université de Kinshasa (UNIKIN), professor NGOIE Ruffin-Bénoît of Institut Supérieur Pédagogique de Mbanzangungu (ISP/Mbanzangungu), professor SUMATA Claude and professor INDENGE José of Université Pédagogique Nationale (UPN) for their wise observations.

7. REFERENCES

[1] Béatrice de Tilière, Analyse Statistiques Multivariée, 2009.

[2] (http://proba.jussien.fr/detiline/cours/polycop-Bio.pdf, consulted 20/02/2016).

[3] F. Benzecri, La classification ascendante hiérarchique d'après un exemple de données économiques, Les cahiers de l'analyse des données, tome 3, n°3, Dunod, 1985.

[4] J. C. Bezdek, Pattern recognition with fuzzy objective function algorithms, New York, Plenum, 1981.

[5] J. C. Bezdek, C. Coray, Gunderson R. et Watson J., Detection and characterization of cluster substructure, SIAM Journal of Applied Mathematics, vol. 40, n°2, 339-372, 1981.

[6] J. C. Bezdek et S. K. Pal, Fuzzy models for pattern recognition, IEEE press, New York, 1992.

[7] F. Chavalier et J. Le Ballac, la Classification, Université Rennes 2012-2013.

[8] C. Chesneau, Eléments de classification, université de Caen, 2018.

[9] P. Dagnelie, Théorie et méthodes statistiques : Analyse statistique à plusieurs dimensions, Les presses Agronomiques de Gembloux, 1986.

[10] J.C. Dunn , A fuzzy relative ISODATA process and its use in detecting compact well-separated clusters, Journal of Cybernetics, vol. 3, n° 3, 32-57, 1974.

[11] P. Gambette, Classification supervisée et non supervisée, cours, Master 1, université Marne-la-Vallée, 2014.

[12] M. Grabisch, Classification et reconnaissance des formes dans Logique floue, ARAGO 14, Observatoire Français des Techniques Avancées, Masson 1994.

[13] F. Husson et J. Josse, Analyse de données avec R, Complémentarité des méthodes d'Analyse Factorielle et de Classification, Agrocampus Rennes, Marseille, 2010.

[14] B. Iooss et V. Verrier, Introduction à l'analyse des correspondances et à la classification, EDF R&D, Cours, Toulouse, 2011.

[15] L. Lebart, A. Morineau, J.-P Fenelon, Traitement des données statistiques : méthodes et programmes, 2è éd, Dunod, Paris, 1982.

[16] D.-R. Mputu. Application de la Classification Ascendante Hiérarchique à la Répartition des Ressources Budgétaires dans la ville-province de Kinshasa, Dissertation DEA, UPN, Kinshasa, 2022. (https://mpra.ub.uni-muenchen.de/113774/, consulted 09/12/2022)

[17] D.-R. Mputu. The clustering method applied in a fair division process, International Journal of Scientific Research and Innovative Technology (IJSRIT), volume 10, Issue 1, pp 1-29, March, 2023.

[18] D.-R. Mputu. Fair division approach based on clustering and reduction of inequalities, iJournals: International Journal of Software and Hardware Research in Engineering (IJSHRE), Volume 12, Issue 3, pp 14-28, March, 2024. (https://ijournals.in/journal/index.php/ijshre/article/view/237, consulted 15/04/2024 at 20h00)

[19] D.-R. Mputu et R. Mabela. Fair Division: Corrected Proportional Rule, iJournals: International Journal of Software and Hardware Research in Engineering (IJSHRE), Volume 13, Issue 8, pp 1-11, August 2025. (DOI: 10.26821/IJSHRE.13.08.2025.130602).

[20] D.-R. Mputu. Application of Classical and Fuzzy Clustering to the Fair Division of Homogeneous Divisible Resources. Mathematics [math]. Université Pédagogique Nationale (UPN), Thèse de doctorat, Kinshasa, Janvier 2026. (https://hal.science/tel-05488208/)

[21] J.P. Nakache, J. Cofais, Approche pragmatique de la classification : Arbres hiérarchiques, Partitionnements, Editions Technip, Paris, 2005.

[22] N. R. Pal et J. C. Bezdek, On cluster validity for the fuzzy c-means model, IEEE transactions on fuzzy systems, vol. 3, n°3, august, 370-379, 1995.

[23] J. Pontier, A.-B. Dufour et M. Normand, Statistique et Mathématiques Appliquées, éditions ellipses, Bruxelles, 1990.

[24] G. Saporta, Analyse discriminante, classification supervisée, scoring, Editions technip, Paris, 2009.

[25] L. Sousa and J. Gama, The application of hierarchical clustering algorithms for recognition using Biometrics of the hand, IJAERS, vol-1, Issue 7, december, 2014.

[26] J.-H. Sublemontier, Classification non supervisée : de la multiplicité des données à la multiplicité des analyses, Thèse, Université d'Orléans, 2012.

[27] Tran Dinh Khang, Nguyen Duc Vuong, Manh-Kien Tran and Michael Fowler, Fuzzy C-Means Clustering algorithm with multiple fuzzification coefficients, MDPI, Algorithms, 13, 158, 2020. (www.mdpi.com/journal/algorithms) (https://www.mdpi.com/1999-4893/13/7/158/htm, consulted 22/05/2022)

[28] X. L. Xie et G. A. Beni, Validity measure for fuzzy clustering, IEEE transactions on pattern analysis and machine intelligence, vol. 13, n°8, august, 841-846 1991.