

PretopoMD: Pretopology-based Mixed Data Hierarchical Clustering

Loup-Noé Levy^{1*†}, Guillaume Guerard^{2,3*†}, Sonia Djebali² and Soufian Ben Amor³

¹Energisme, 88 Avenue du Général Leclerc, Boulogne-Billancourt, 92100, France.

²Léonard de Vinci Pôle Universitaire, Research Center, 12 Avenue Léonard de Vinci, Paris La Défense, 92916, France.

³LI-PARAD Laboratory EA 7432, Versailles University, 55 Avenue de Paris, Versailles, 78035, France.

*Corresponding author(s). E-mail(s): loup-noe.levy@energisme.com; guillaume.guerard@devinci.fr;

Contributing authors: sonia.djebali@devinci.fr; soufian.ben-amor@uvsq.fr;

†These authors contributed equally to this work.

Abstract

This article presents a novel pretopology-based algorithm designed to address the challenges of clustering mixed data without the need for dimensionality reduction. Leveraging Disjunctive Normal Form, our approach formulates customizable logical rules and adjustable hyperparameters that allow for user-defined hierarchical cluster construction and facilitate tailored solutions for heterogeneous datasets. Through hierarchical dendrogram analysis and comparative clustering metrics, our method demonstrates superior performance by accurately and interpretably delineating clusters directly from raw data, thus preserving data integrity. Empirical findings highlight the algorithm's robustness in constructing meaningful clusters and reveal its potential in overcoming issues related to clustered data explainability. The novelty of this work lies in its departure from traditional dimensionality reduction techniques and its innovative use of logical rules that enhance both cluster formation and clarity, thereby contributing a significant advancement to the discourse on clustering mixed data.

Keywords: Mixed Data, Clustering, Machine Learning, Pretopology

1 Introduction

Clustering is a fundamental task in unsupervised machine learning, aimed at grouping similar data points based on inherent similarities in large datasets. While numerous clustering methods have been developed for numerical data, challenges arise when datasets consist of a mixture of numerical, ordinal, and categorical features. Mixed data clustering is critical in various domains, such as the energy sector, biology, medicine, marketing, and economics [1–6]. In these fields, datasets often present rich and heterogeneous characteristics, requiring methods that can accommodate different data types without oversimplification.

One of the primary difficulties in clustering mixed data is the diversity in scales, distributions, and similarity measures that characterize different feature types. Traditional clustering techniques, which often rely on a single distance measure or assume data homogeneity, may fail to capture the complex interactions present in heterogeneous datasets. An alternative strategy involves preprocessing the mixed data into a purely numerical format, thereby enabling the use of advanced numerical clustering algorithms. However, this conversion process may lead to significant information loss and an oversimplified representation of the underlying data structure.

The increasing need for robust mixed data clustering methods has spurred significant research. Yet, many existing techniques depend on dimensionality reduction—which can obscure the original data complexity—or lack the flexibility and interpretability required for real-world applications. To overcome these limitations, our work presents a novel clustering algorithm based on the mathematical framework of Pretopology. This approach offers several distinctive advantages over traditional methods.

The innovations and contributions of our paper are as follows:

- Direct Handling of Mixed Data: Our algorithm is specifically designed to operate directly on mixed datasets without requiring prior dimensionality reduction. This ensures that the intrinsic complexity of the data is preserved, leading to more authentic and reliable clustering outcomes.
- Customizable Cluster Formation via Logical Rules: A key innovation is the incorporation of user-defined logical rules for cluster construction. This feature provides increased flexibility, allowing the clustering process to be tailored to the specific characteristics of the dataset under study.
- Hierarchical Flexibility Through Multiple Hyperparameters: Multiple hyperparameters govern the conditions for clustering and division. This flexibility enables the generation of customized hierarchical structures that can reveal underlying patterns at various levels of granularity.
- Explainability of the Clustering Process: The algorithm inherently produces a hierarchical dendrogram, offering transparent insights into how clusters are formed and how different hyperparameters influence the segmentation. This level of explainability is particularly valuable in applications where interpretability is essential for validation and decision-making.

By addressing the inherent challenges of clustering mixed data and explicitly incorporating innovations that include direct data handling, flexible rule-based construction, customizable hierarchical structuring, and enhanced explainability, our work contributes a robust and interpretable tool to the field of unsupervised machine learning.

The remainder of the paper is organized as follows. Section 2 provides an extensive review of the state-of-the-art in mixed data clustering methods and dimensionality reduction techniques, highlighting their respective strengths and limitations. In Section 3, we introduce our novel pretopology-based clustering algorithm, detailing both its theoretical foundation and its operational mechanics. Section 4 presents experimental results and discussions, demonstrating the effectiveness of our approach across diverse mixed datasets without the need for prior dimensionality reduction.

2 Literature review

2.1 Machine Learning Models

Mixed data clustering involves grouping datasets with a combination of numerical, categorical, and other types of features. In this section, we offer an overview of prominent open-source methods in mixed data clustering.

Partitioning Clustering

Partitioning clustering aims to divide a dataset into a predefined number of non-overlapping clusters. Two notable methods in this category are:

- **K-Prototypes** [7]: An extension of traditional K-Means, K-Prototypes handles both numerical and categorical features. It combines K-Means for numerical features with the K-Modes method [8] for categorical features, utilizing a model-based approach for clustering categorical data.
- **Convex K-Means** [9]: This algorithm refines centroids with convex hull iteratively, aiming for convergence or a predefined maximum number of iterations.

Model-Based Clustering

Model-based clustering leverages statistical models to describe data distribution within clusters, accommodating various data types. Key methods include:

- **MixtComp** [10]: A statistical approach for clustering mixed data, MixtComp combines model-based clustering and Bayesian approaches. It models mixed data as a mixture of multivariate distributions, utilizing latent variables to capture complex data structures.
- **KAMILA** (KAy-means for MIxed LArge data) [11]: Extending the standard K-means algorithm, KAMILA handles mixed data through a combination of K-means clustering and the Gaussian-multinomial mixture model. It avoids strong parametric assumptions and balances the effects of numerical and categorical data.

- **ClustMD** (Clustering of Mixed Data) [12]: A model-based clustering method specifically designed for mixed data, ClustMD employs a latent variable model fitted with the Expectation-Maximization algorithm.

Hierarchical Clustering

Hierarchical clustering is widely used for grouping similar data points based on similarity or distance measures. Notable hierarchical methods for mixed data include:

- **Phillip and Ottaway** [13]: This method proposes hierarchical clustering for mixed data based on Gower's similarity measure. It separates categorical and numerical features, computes similarities in each feature space, and combines them to create a similarity matrix, facilitating meaningful clustering of mixed data.

2.2 Deep Learning Models

Deep learning models—including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers—have demonstrated remarkable capabilities in automating the extraction of high-level features from raw data. They have achieved state-of-the-art performance in areas such as image recognition, natural language processing, and speech recognition by learning complex, hierarchical representations. These models can autonomously capture and refine intricate patterns from large-scale datasets, which not only minimizes the need for extensive manual feature engineering but also leads to considerable successes in diverse domains. Advanced techniques—such as contrastive learning, deep clustering, and methods incorporating self-labeling and pseudo-label refinement—further extend their capabilities in challenging unsupervised or semi-supervised tasks [14, 15].

Despite these strengths, several significant drawbacks have led us to decide against the development of deep learning models for the current project. A primary concern is their substantial dependency on large, well-annotated datasets. Many deep learning architectures require a critical mass of labeled data to generalize well; when available data is limited, these models become prone to overfitting. This problem is compounded in advanced methods—such as multi-view techniques and soft contrastive losses—that often demand extra iterative training stages (like pseudo-label generation and refinement) and further exacerbate data requirements.

In addition to issues with data dependency, the computational demands of deep learning models are considerable. Training these models typically requires specialized hardware (such as GPUs or TPUs) along with extended processing time and high memory consumption. This level of computational overhead makes them less feasible in environments where such resources are constrained—a concern that is well documented in recent investigations [16, 17]. Moreover, these models tend to be very sensitive to hyperparameter choices including architectural design, learning rates, and regularization strategies. This sensitivity not only renders the development and tuning process time-consuming and error-prone but also impedes the efficient deployment of these large models in production settings.

Another notable drawback is the limited interpretability of deep learning models. Their complex, multilayered structures often result in a "black box" effect,

making it difficult to understand the basis of their decision-making. This lack of transparency is particularly problematic for applications requiring clear, explainable insights. Although emerging hybrid architectures and automated hyperparameter tuning methods [18–20] show promise in mitigating some of these issues, they remain largely experimental and have yet to fully address the underlying challenges.

Given these considerations—the extensive data requirements, high computational demands, and limited interpretability—our current project, which operates under restricted data and resource constraints while demanding robust transparency, motivates the use of alternative methods. By prioritizing approaches that can deliver greater efficiency, robustness, and explainability, our strategy is more closely aligned with the project’s operational requirements and long-term goals.

2.3 Dimensionality Reduction

Dimensionality reduction techniques prove valuable in applying advanced numerical data clustering methods to datasets with a mix of numerical and categorical features. However, caution must be exercised as these techniques may lead to substantial information loss and compromise the preservation of intrinsic relationships, impacting the accuracy and interpretability of results.

In existing literature, only Factor Analysis of Mixed Data (FAMD) addresses mixed data dimensionality reduction. Nevertheless, we have adapted two numerical data dimensionality reduction techniques for mixed data:

- **Factor Analysis of Mixed Data (FAMD)** [21]: An extension of Principal Component Analysis (PCA) for continuous variables and Multiple Correspondence Analysis (MCA) for categorical variables, providing a low-dimensional representation of mixed data.
- **Uniform Manifold Approximation and Projection (UMAP)**: Originally designed for numerical data, we applied UMAP to mixed data by employing the Huang distance metric, suitable for mixed data. UMAP focuses on preserving local data structure, making it efficient for large-scale datasets.
- **Pairwise Controlled Manifold Approximation (PaCMAP)** [22]: Initially designed for numerical data, PaCMAP aims to preserve both local and global structures. We adapted PaCMAP for clustering mixed data by utilizing FAMD instead of PCA in the initialization phase.

2.4 Mixed data cluster evaluation indices

Key measures for data clustering, categorized into external and internal indices, facilitate the evaluation of clustering algorithms on mixed data. For this paper, three internal measures are employed after FAMD reduction due to the absence of a clustering evaluation method for mixed data. Since clustering is unsupervised learning, the focus is on internal indices independent of ground truths.

Calinski-Harabasz Index [23]. Also known as the Variance Ratio Criterion, this index calculates the ratio of between-cluster dispersion to within-cluster dispersion. A higher Calinski-Harabasz score suggests better-defined clusters.

Silhouette Coefficient [24]. This index evaluates the compactness and separation of the clusters. A higher Silhouette Coefficient indicates well-defined clusters. Since the Silhouette coefficient computes pairwise distances, we use it after FAMD reduction or with the Gower distance.

Davies Bouldin Index [25]. This index is the average of the maximum ratio between the distance of a point from the center of its group and the distance between two centers of groups.

It is crucial to recognize that in mixed data clustering, cluster evaluation indices have limitations, including information loss from dimensionality reduction, incompatibility with non-Euclidean spaces, and sensitivity to data characteristics like outliers and skewness. Further details on additional external and internal measures are available in the GitHub repository¹.

2.5 Summary

Find in Table 1 the pros and cons of the presented algorithms where $\#k$? refers to the hyperparameters k to indicate the number of output clusters.

3 Theory of Pretopology

In this section, we delve into the key concepts and definitions of pretopology as outlined in the article, including pretopological space and pseudo-closure. Although we won't extensively cover the origins of pretopology, it's crucial to grasp that the concept of a pretopological space arises by relaxing the assumptions of topological spaces, enabling the modeling of discrete structures [26].

3.1 Pretopological space

Definition 1. *A pseudoclosure function, denoted as $a : \wp(U) \rightarrow \wp(U)$ on a set U , is defined by:*

- $a(\emptyset) = \emptyset$
- $\forall A \mid A \subseteq U : A \subseteq a(A)$

where $\wp(U)$ is the power set of U .

This function establishes a relation between any set of elements and a larger set, forming the basis of a pretopological space, and by the way, a hierarchy. An illustrative example of a pseudoclosure function is depicted in Figure 1.

Definition 2. *A tuple $(U, a(.))$, where U is a set of elements and $a(.)$ is a pseudoclosure function on U , constitutes a pretopological space.*

The most general pretopological space is determined by the above definition. Introducing additional conditions yields more specific pretopological spaces:

Definition 3. Isotony. *If $\forall A, B \mid A \subseteq U, B \subseteq U : A \subseteq B \implies a(A) \subseteq a(B)$, then we get a pretopological space of type V .*

Definition 4. Union Preservation. *If $\forall A, B \mid A \subseteq U, B \subseteq U : a(A \cup B) = a(A) \cup a(B)$, then we get a pretopological space of type V_D .*

¹<https://github.com/ClementCornet/Benchmark-Mixed-Clustering>

Method	Type	#k?	Key Strengths	Key Limitations
K-Prototypes	Partitional	Yes	Combines Euclidean (numerical) and matching (categorical) distances in a unified framework.	Simplistic categorical distance measure; requires careful tuning of k and balancing hyperparameter γ .
Convex K-Means	Partitional	Yes	Refines centroid positions via convex hull concepts; produces more coherent clusters with improved convergence.	Additional convex-combination steps add computational overhead; still requires a predefined cluster count.
MixtComp	Model-Based	Yes	Provides a rigorous probabilistic framework; models mixed data with latent variables for rich interpretability.	Computationally intensive; potential convergence issues due to Bayesian estimation.
Kamila	Model-Based	Yes	Balances numerical and categorical data using Gaussian–multinomial models; scalable for large datasets.	Sensitive to initialization and parameter settings; requires pre-specification of the number of clusters.
ClustMD	Model-Based	Yes	Comprehensive modelling that handles heterogeneity and missing data; models statistical properties of both data types.	Higher computation time; convergence can be challenging for some datasets.
Phillip & Ottaway	Hierarchical	No	Produces an interpretable dendrogram revealing multi-level nested cluster structures without pre-defining k .	Sensitive to merging criteria; may not scale well for very large datasets.
PretopoMD	Hierarchical	No	Directly handles mixed data without dimensionality reduction; offers customizable logical rules and multiple hyperparameters; highly explainable via a dendrogram.	Computationally intensive due to iterative pseudoclosure and adjacency matrix construction; performance sensitive to hyperparameter settings.

Table 1 Summary of mixed data clustering methods.

Definition 5. Union Distribution. If $\forall A \mid A \subseteq U : a(A) = \bigcup_{x \in A} a(x)$ then we get a pretopological space of type V_S .

In pretopology, the concept of closure aligns with its definition in topology [27]:

Definition 6. A part F of U will be a closure of U if and only if $a(F) = F$

Proposition 1. Intersection of Closures. In a pretopological space of type V , the intersection of closures is a closure.

Proposition 2. Existence of Closure and Opening. In a pretopological V – type space, the closure and opening of any part of U still exists.

Proposition 3. Smallest Closure. In a pretopological space of type V , the closure of a part A of U is the smallest closure containing A . Denoted $F(A)$.

Proposition 4. Every Set Has a Closure. In a pretopological space of type V , every set has a closure.

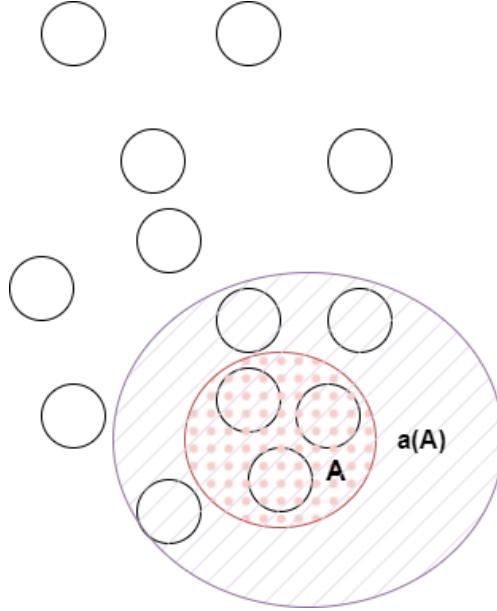


Fig. 1 Example of a pseudoclosure function.

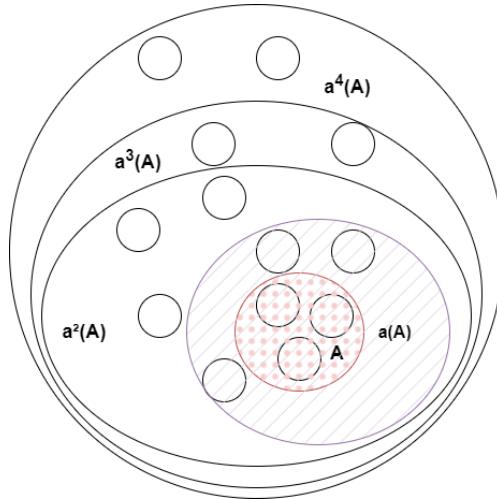


Fig. 2 Closure of set A , $a^4(A) = F(A)$.

The closure in a pretopological space of type V can be obtained by iteratively applying the pseudoclosure operator to the set and its subsequent images until expansion ceases. Figure 2 illustrates an example of this process.

If a pretopological space is of type V_D , and for every $A \subseteq U : a(A) = a(a(A))$, the pseudoclosure function is termed idempotent [28]. In such cases, a topology is obtained. Notably, in a finite space, $V_S = V_D$ [29]. Additionally, in pretopological spaces of

type V_D , the pseudoclosure of a set is entirely determined by the pseudoclosures of its singletons. When the space is finite, the pseudoclosure can be represented as a graph, establishing pretopology as a generalization of graph theory [28]. Figure 3 shows the relation between the two.

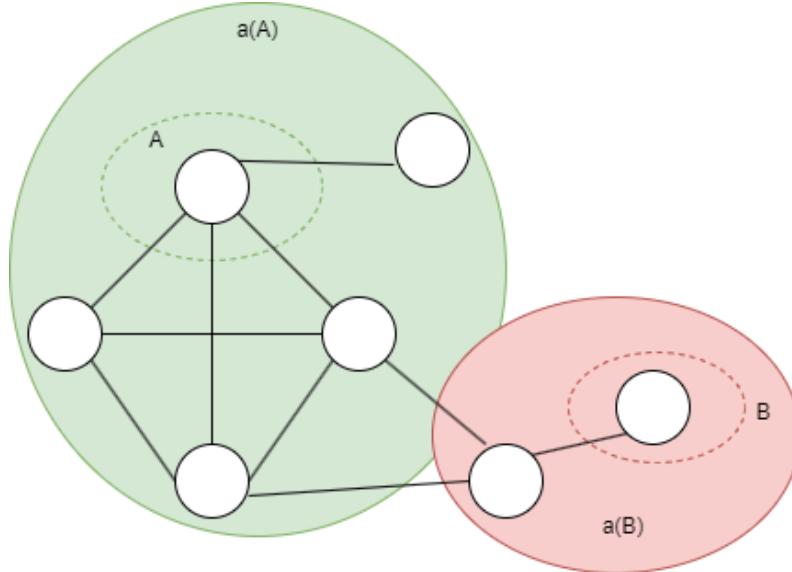


Fig. 3 Pseudoclosure function on a graph.

An alternative characterization involves the concepts of prefilter and neighborhood. To comprehend this, let's introduce the following definitions:

Definition 7. We say that a set \mathcal{F} of $\wp(\wp(U))$ is a prefilter over U , if:

$$\forall F \in \mathcal{F}, \forall H \in \wp(U), F \subset H \implies H \in \mathcal{F} \quad (1)$$

Definition 8. \mathcal{F} of $\wp(\wp(U))$ is a filter over U , if it is a prefilter stable under finite intersection, i.e.

$$\forall F \in \mathcal{F}, \forall G \in \mathcal{F}, F \cap G \in \mathcal{F} \quad (2)$$

In other words, and restricting ourselves to a finite space, a filter is the family of all supersets of a set B , while a prefilter is the family of supersets of every member B_i of a family of sets B . The family of sets B is called the basis of the prefilter. We can see in Figure 4 an example of a filter and a prefilter with basis $B = 1, 4, 2, 4$.

If we have a set U and, for every $x \in U$, there exists a prefilter $V(x)$ such that every member of $V(x)$ contains the element x , we can define a pseudoclosure function as follows:

$$\forall A \subseteq U, a(A) = \{x \in U \mid \forall V \in V(x), V \cap A \neq \emptyset\} \quad (3)$$

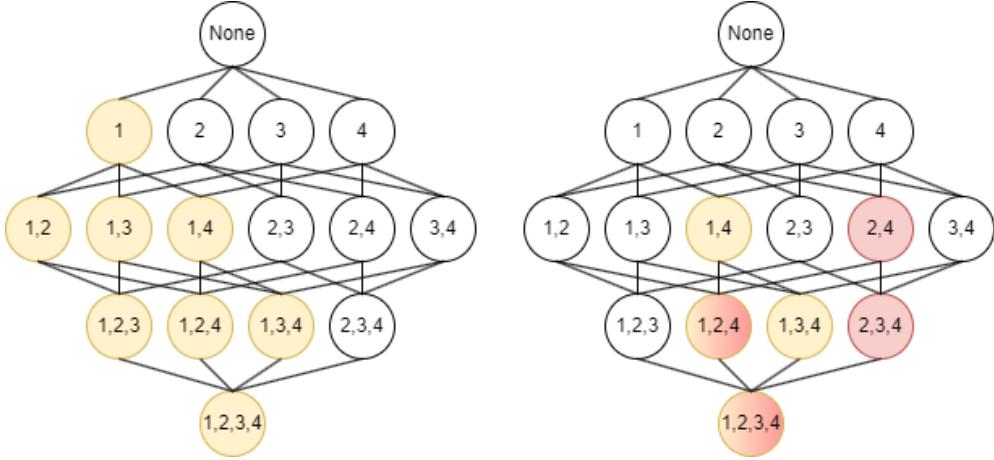


Fig. 4 Filters vs Prefilters.

Here, $V(x)$ is termed the family of neighborhoods of x , and each set in the family is referred to as a neighborhood of x . Figure 5 shows a graphical representation of this definition of the pseudoclosure.

Conversely, if we have a pseudoclosure function $a(\cdot)$ in a pretopological space of type V , the family of sets given by:

$$V(x) = \{V \subset U | x \in i(V)\} \quad (4)$$

where $i(A) = a(A^c)^c$, is a prefilter.

A proposition asserts that no two families of prefilters or pseudoclosure functions define the same pretopological space, demonstrating the interchangeability of these characterizations [29].

3.2 Framework

We introduce a framework for formalizing a pretopological space, building upon the work of Julio Laborde [28]. The pretopological space is defined by a tuple $(G, \Theta, DNF(\cdot))$, where:

- $G = G_1(V_1, E_1), G_2(V_2, E_2), \dots, G_n(V_n, E_n)$ represents a collection of n weighted directed graphs.
- $\Theta = \theta_1, \theta_2, \dots, \theta_n$ is a set of n thresholds, each associated with a specific graph.
- $DNF(\cdot) : (\wp(U), U) \rightarrow True, False$ is a boolean function defined as a positive disjunctive normal form involving the n boolean functions $V_1(A, x), \dots, V_n(A, x)$, each associated with a graph. The truth value depends on the set A and the element x .

To determine if an element $x \in U$ belongs to the pseudoclosure of a set A , the following steps are followed:

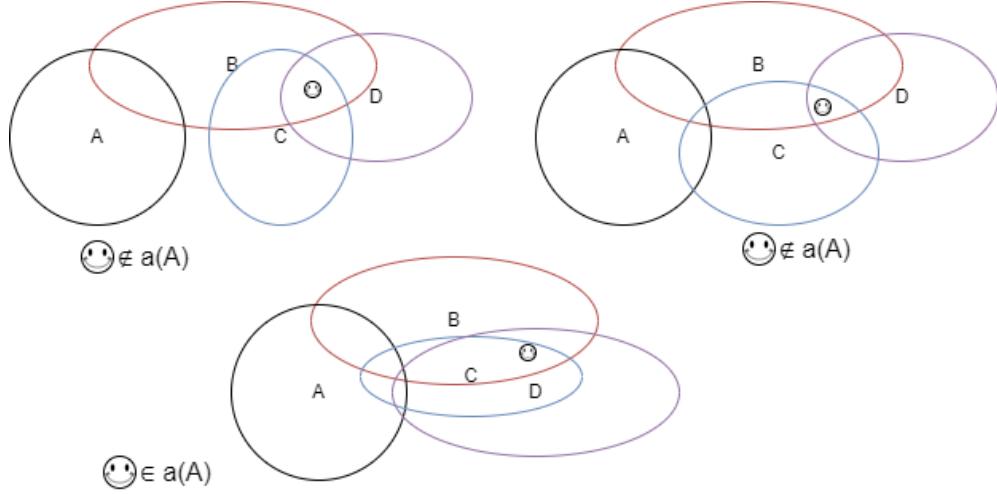


Fig. 5 Neighborhood definition of a pretopology.

- **Boolean Function Evaluation.** For each $V_i(A, x)$, $V_i(A, x) = \text{True}$ if and only if $\sum_{e_{xy} \in G_i, y \in A} w(e_{xy}) \geq \theta_i$, where e_{xy} denotes the edge from x to y , and $w(e)$ represents the weight of the edge e .
- **Pseudoclosure Membership.** The element $x \in U$ belongs to the pseudoclosure of A if and only if the $DNF(\cdot)$ evaluates to True.

In essence, this process checks whether the sum of the edge weights connecting the element x to the elements within A is greater than the threshold associated with the graph in each graph. If this condition is met, the boolean variable corresponding to that graph takes the value True; otherwise, it takes the value False. If $DNF(\cdot)$ evaluates to True given the values of the boolean functions $V_i(A, x)$, then the element belongs to the pseudoclosure.

3.3 PretopoMD Algorithm

This section outlines the Python library's algorithms designed for constructing closures and building hierarchical clustering of mixed data.

The algorithm 1, provided as pseudocode in Algorithm 1, is organized into four stages:

- **Identify Seeds.** Identify a family of elementary subsets, referred to as seeds.
- **Construct Closures.** Construct closures of seeds through iterative application of the pseudoclosure function.
- **Create Adjacency Matrix.** Create the adjacency matrix representing relationships between all recognized subsets, including intermediate ones.
- **Determine Quasi-Hierarchy.** Determine the quasi-hierarchy by applying the corresponding algorithm to the adjacency matrix.

At this point, we need to define three hyperparameters:

Algorithm 1 QuasistructuralAnalysis: Algorithm for building a quasi-hierarchy from pretopological space.

Require: $((U, a(.)), d, seed_Func(.), th_{qh})$
Ensure: $Sets_{qh}, Adj_{qh}$
 $seed_List \leftarrow Set_Seeds((U, a), d, seed_Func)$
 $Sets_{ipc} \leftarrow Iterative_Pseudoclosure((U, a), seed_List)$
 $Attr \leftarrow Attraction_Matrix(Sets_{ipc})$
 $Sets_{qh}, Adj_{qh} \leftarrow QuasiHierarchy(Sets_{ipc}, Attr, th_{qh})$

- $seed_Func(.)$: Determines a set of nearby elements for a given element, constituting a seed.
- d : Specifies the size of the seeds.
- th_{qh} : Threshold above which two sets are considered related in the hierarchy. This hyperparameter is required by the *Quasi_Hierarchy* algorithm to establish the quasi-hierarchy.

We will now discuss each stage of the algorithm in detail.

Computation of a Family of Elementary Sets or Seeds

The objective is to identify elementary subsets of size d , commonly known as seeds, utilizing the $seedFunc(.)$ function. This function is tasked with locating the necessary d neighbors. The process involves iterating over all the points within the set U , associated with the pretopological space p . The pseudocode for the resultant algorithm, denoted as *ElemQuasiclosures*, is presented in Algorithm 2.

Algorithm 2 Set_Seeds: Construction of the seeds of size d by applying the function $seed_Func(.)$ on all the elements of the set U .

Require: $((U, a(.)), d, seed_Func(.))$
Ensure: $seed_List$
 $seed_List \leftarrow list()$
for all $x \in U$ **do**
 $seed \leftarrow seed_Func(x, d)$
 $seed_List.append(seed)$
end for

Algorithm 2 utilizes the function $FindNeighbors$, for which the pseudocode is outlined in Algorithm 3. This function accepts an element from U , the desired number of neighbors d , and the function that determines the nearest neighbors, denoted as $seed_Func(.)$. The $seed_Func(.)$ function typically assumes one of the following two forms:

- This function identifies the nearest nodes to an element. It is employed when a distance can be calculated, such as when the studied relations are quantifiable.

Algorithm 3 **FindNeighbors**: Determine the d neighbors of $firstNode$ using the $seedFunc()$ function.

Require: $(first_Node, d, seed_Func())$

Ensure: $path$

```

 $path \leftarrow list()$ 
 $last\_Treated\_Node \leftarrow first\_Node$ 
for all  $i \in range(d)$  do
     $new\_Node \leftarrow seed\_Func(last\_Treated\_Node)$ 
     $path.append(new\_Node)$ 
     $last\_Treated\_Node \leftarrow new\_Node$ 
end for

```

- This function randomly navigates through neighboring nodes. Its usage is preferred when the relations are not quantifiable, for instance, when values describe categories.

Creation of Subsets through Iterative Pseudoclosure Applications

The $seed_Func()$ function generates the subsets that will be organized by the quasi-hierarchy algorithm, using the seed list $seedList$ computed earlier by *QuasistructuralAnalysis*. For each seed in $seedList$, the membership function is applied iteratively until the pseudoclosure no longer produces larger sets.

The resulting subsets are stored in a list of sets named QF_{tmp} which organizes the subsets based on the number of elements they contain. The subsets of size s are positioned in the s -th slot of QF_{tmp} . As the pseudoclosure function $a()$ only yields a set that is larger or equal in size, applying the pseudoclosure function to sets in ascending order of size ensures that all elements are processed once and only once.

The list $Sets_{ipc}$ is then constructed from the lists in QF_{tmp} and returned. The corresponding pseudocode is provided in Algorithm 4.

Creation of the Attraction Matrix

The iterative application of a pseudoclosure to two seeds can generate distinct sets that have non-empty intersections. Traditional hierarchies of sets typically handle sets that either have no intersection or are contained within one another (i.e., subsets and supersets). Therefore, a different type of relationship needs to be defined, referred to as a quasi-hierarchy.

To establish this quasi-hierarchy, an attraction matrix is constructed first, as outlined in Algorithm 5. This matrix represents the "attraction" that sets exhibit for each other. Here, the term "attraction" denotes a non-symmetrical relationship between two intersecting sets, determined by the sizes of the sets and the size of their intersection. The underlying principles include:

- Two subsets should only be attracted to each other if their intersection is non-empty (i.e., $A \cap B \neq \emptyset$).
- The larger the cardinality of the intersection $A \cap B$ relative to that of A , the stronger the attraction between A and B ,

Algorithm 4 Iterative_Pseudoclosure: Calculation of subsets by iterative application of the pseudo-closure function.

Require: $((U, a(.)), seed_List)$

Ensure: $Sets_{ipc}$

QF_{tmp} a list of $Size(U)$ of empty sets

for all $seed \in seed_List$ **do**

$QF_{tmp}[Size(seed)].append(seed)$

end for

for all $i \in range(1, Size(U) + 1)$ **do**

for all $s \in QF_{tmp}[i]$ **do**

$a_s \leftarrow a(s)$

if a_s **not in** lists of QF_{tmp} **then**

$QF_{tmp}[Size(a_s)].append(a_s)$

end if

end for

end for

$Sets_{ipc} \leftarrow list()$

for all $i \in range(Size(QF_{tmp}))$ **do**

$Sets_{ipc}.extend(QF_{tmp}[i])$

end for

- The larger the cardinality of the subset B relative to that of A , the less critical it is for $A \cap B$ to be large for the relation between A and B to be strong. In other words, a very large set will attract smaller sets even if their intersection is not very large.

Algorithm 5 Attraction_Matrix: Construction of the attraction matrix for the quasihierarchy.

Require: $(Sets_{ipc})$

Ensure: $Attr$

$Attr \leftarrow Squared_Matrix_Zeros(size(Sets_{ipc}))$

for all $A, B \in Sets_{ipc}$ **do**

$A_has_B \leftarrow Size(A \cap B)/Size(B)$

$B_has_A \leftarrow Size(A \cap B)/Size(A)$

$A_bigger_B \leftarrow Size(A)/Size(B)$

$B_bigger_A \leftarrow Size(B)/Size(A)$

$Attr[B_index, A_index] = B_bigger_A * B_has_A$

$Attr[A_index, B_index] = A_bigger_B * A_has_B$

end for

Creation of the Quasi-Hierarchy

The quasi-hierarchy is characterized by a list of sets and an adjacency matrix. The adjacency matrix is derived from the attraction matrix by assessing whether the attraction values in the attraction matrix surpass the threshold th_{qh} . The establishment of the quasi-hierarchy involves applying the following rules to the values of Atr :

- A link (as defined by graph theory) between two subsets is formed in the quasi-hierarchy if their attraction surpasses the threshold th_{qh} .
- Two subsets exhibiting strong mutual attraction (i.e., surpassing the threshold th_{qh}) are considered equivalent, and only one of them is retained. If the sets are of equal size, one of them is randomly selected; otherwise, the smaller set is removed.
- The updated list of sets, along with their adjacency matrix, dictates the structure of the quasi-hierarchy.

Algorithm 6 Quasi_Hierarchy: Ensures QuasiHierarchy

```

Require:  $Sets_{ipc}$ ,  $Atr$ ,  $th_{qh}$ 
Ensure:  $Sets_{qh}$ ,  $Adj_{qh}$ 
 $Adj_{qh} \leftarrow \text{Squared\_Matrix\_Zeros}(\text{size}(Sets_{ipc}))$ 
 $Adj_{qh}[Atr > th_{qh}] \leftarrow 1$ 
for all  $i, j \in \text{Range}(\text{size}(Sets_{ipc}))$  do
    if  $Adj_{qh}[i, j] = 1 \& Adj_{qh}[j, i] = 1$  then
        if  $\text{size\_of\_set}(i) \geq \text{size\_of\_set}(j)$  then
            remove set  $j$  from  $Adj_{qh}$  and  $Sets_{ipc}$ 
        else
            remove set  $i$  from  $Adj_{qh}$  and  $Sets_{ipc}$ 
        end if
    end if
end for

```

3.4 Hyperparameters

The definition of the pretopological space significantly influences cluster formation. For example, considering all n numeric features together, with their Euclidean distances calculated in one graph of G , and separately calculating Hamming distances for all categorical values in another graph of G , results in a straightforward pretopological space. The DNF could be a logical "AND" or "OR" combination of the Euclidean and Hamming distances. Alternatively, features can be considered individually, each with its own graph, similarity measure, and threshold, making the DNF more comprehensive and specific.

Thresholds are automatically calculated to adapt to the number of points, the number of close neighbors each point has, and the dispersion in the dataset. Alternatively, they can be set manually. The parameters in the threshold calculation function can be adjusted to obtain either high thresholds, resulting in small clusters with low

inner dispersion and a high number of outliers, or lower thresholds, leading to larger clusters with fewer outliers.

The threshold th_{qh} used in constructing the quasi-hierarchy is usually fixed at 0.5.

The DNF function defines the logical rules governing cluster formation. Using a logical AND (i.e., G_i AND G_j) creates a more constrained clustering, where clusters exhibit similar values for characteristics i and j . Conversely, a logical OR (i.e., G_i OR G_j) results in less constrained clusters, where clusters show similar values for either characteristic i or j .

Several functions and values are employed to define the networks specific to each feature and the subsequent pretopological space. They are designed to adapt to the dataset's characteristics, such as the number of elements, the nature of the features, and the dispersion of the values. These functions are also tailored to user needs, such as the approximate size of desired clusters or the extent to which outliers are accepted into clusters. For instance, a higher square length results in fewer clusters, and the higher the threshold, the more outliers are found. Users can adjust these parameters by changing the way they are calculated or by directly modifying the coefficients (or powers).

Here is a list of the hyperparameters:

- $\text{thresholds} = \left(\frac{\text{real_points}}{\text{size}} \right)^{\text{threshold_power}}$;
- $\text{real_points} = \text{nb_elements} - \sum(\text{inverse})$;
- $\text{inverse}_i = \frac{\text{closest}_i - 1}{\text{closest}_i}$;
- $\text{closest}_i = \text{count}(j \text{ with } \text{dist}(i, j) < \text{square_lgth} \times \text{closest_coeff})$;
- $\text{square_length} = \sqrt{\frac{\text{area}}{\text{size}}} \times \text{square_lgth_coeff}$;
- $\text{area} = \text{area_method}(\text{dm}, \text{df_data})$;
- $\text{dm} = \text{matrix of distances using the distance function}$.

3.5 Explainability

The proposed clustering method aligns with the principles of eXplainable Artificial Intelligence by offering a clear, customizable, and traceable path from input data to clustered groups. This facilitates intuitive understanding and validation of clusters by domain experts, highlighting the following key features:

- **Customizable Hyperparameters:** Hyperparameters are easily customizable, either through fixed values or adaptive functions that respond to data size, dispersion, and desired outlier count. This tunability empowers experts to exert precise control over the clustering process, enhancing the transparency and interpretability of the formed clusters.
- **Hierarchical Structure:** The clustering method adopts a hierarchical structure, presenting a tree-like representation of data groups. This structure facilitates a step-by-step comprehension of how clusters are formed and their interrelationships, contributing to the transparency of the clustering process.
- **Disjunctive Normal Form:** The DNF explicitly defines the role of each parameter in clustering, ensuring transparency in the logic behind grouping. This formalism contributes to a clear articulation of the reasoning behind the formation of clusters.

- **Adaptation to Mixed Data Types:** The method accommodates mixed data types, providing transparency as each data characteristic and its role in clustering are explicitly defined. This stands in contrast to approaches requiring Dimensionality Reduction, enhancing interpretability.
- **Threshold Management:** Transparent threshold management ensures clarity in boundary decisions within the clustering process. This transparency aids in understanding why specific data points are considered similar or dissimilar, enhancing the overall interpretability of the results.

4 Results and Discussions

4.1 Evaluation and comparison of clusters

This section begins by showcasing the functionality of our algorithm through small-scale examples. Subsequently, we delve into an analysis of the algorithm's outcomes when applied to a comprehensive, publicly available mixed dataset. Additionally, we present the results obtained using various state-of-the-art algorithms that we introduced. Finally, we engage in a discussion of the findings and propose potential enhancements. Supplementary results can be accessed on Github².

In Figure 6, we present a generated dataset comprising elements characterized by three attributes: their positions in a two-dimensional space, their size, and their shape. For each attribute, a weighted graph, termed a prenetwork, is created, where node weights indicate the similarity between elements based on that specific attribute. In this instance, Euclidean distance computes the similarity in the 2D space, absolute differences quantify size similarity, and Hamming distance evaluates shape similarity. Various calculation methods for these attributes can be applied, as the chosen method is a parameter of the prenetworks. Subsequently, the pretopological space is defined by establishing the DNF that links the prenetworks.

Figure 6 visually represents clustering results using the DNF: Position AND Size AND Shape. This DNF generates multiple clusters of elements that exhibit similarity across all attributes. In Figure 7, we showcase the clustering outcome on another generated dataset using the DNF: (Position AND Size) OR (Position AND Shape). This DNF forms three large clusters of closely positioned elements in the 2D space. An analysis of the hierarchy of subclusters identified by our algorithm reveals that these subclusters consist of elements closely located in the 2D space and sharing similar shapes and/or sizes (Figure 8). This underscores that by defining a specific pretopological prenetwork, a hierarchical structure of clusters tailored to specific requirements can be established. In this context, the three final clusters are obtained by selecting the sets with "-1" as a parent in the dendrogram (i.e., sets with no parent or whose parent is the entire set of elements).

4.2 Evaluation of the clustering methods

The assessment of various clustering algorithms is conducted utilizing the clustering quality measures outlined in the article. The Calinski-Harabasz, Davies-Bouldin, and

²The link will be provided after the review (double-blind)

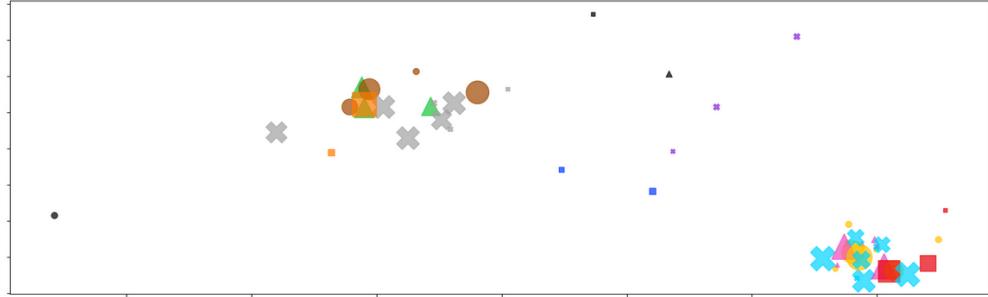


Fig. 6 Clusters identified by our algorithm by taking into account the position, size and shape using the DNF of Position **and** size **and** shape.

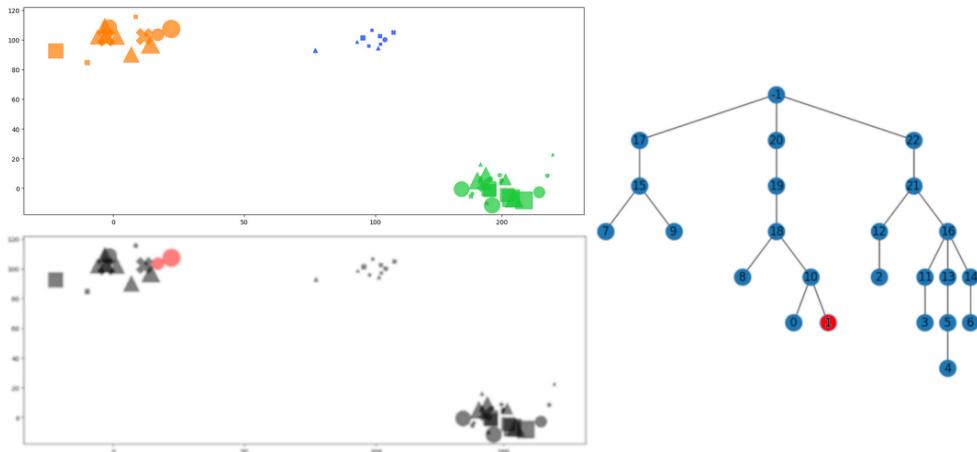


Fig. 7 Clusters identified by our algorithm by taking into account the position, size and shape using the DNF of (Position **and** size) **or** (Position **and** shape). The subcluster "1" and its position in the hierarchy are colored in red.

Silhouette scores are computed by transforming datasets into Euclidean spaces through FAMD. The output space maintains the same dimensionality as the initial space, and the selection of FAMD is based on the following considerations:

- It is a factorial method, the inertia of the model is known.
- It is deterministic.
- It does not rely heavily on hyper-parameters.

The Silhouette score, unique in its ability to accept a pairwise distance matrix as input, is computed using the Gower matrix. This approach is adopted to mitigate potential bias towards FAMD and to contribute to the analysis, particularly in cases where FAMD yields low inertia.

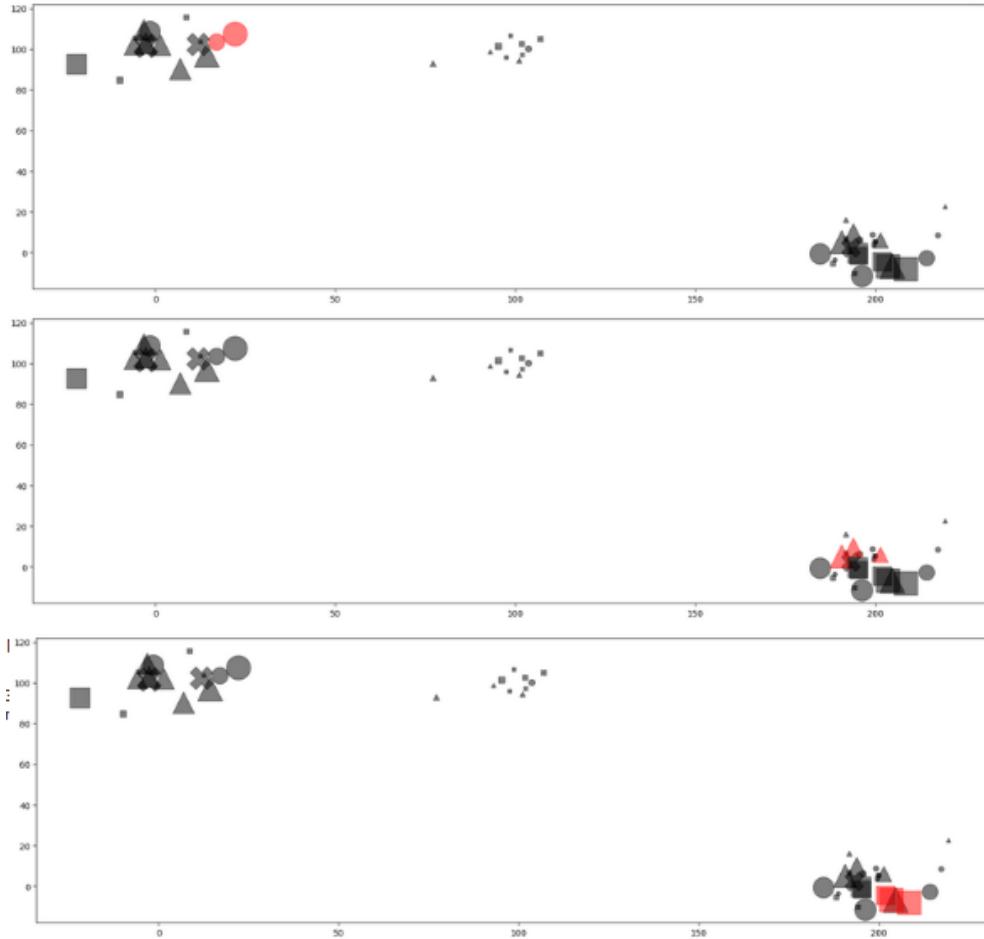


Fig. 8 Different subsets of the clusters identified in Figure 7 are identified, we can see how the DNF has influenced the way the clusters are defined.

It is noteworthy that certain algorithms may yield either a single cluster or solely identify outliers. In such instances, the results table indicates “-” to signify these scenarios.

4.3 Palmer Penguins

The Palmer Penguins dataset, compiled from physical measurements of 344 penguins in the Palmer Archipelago, Antarctica [30], serves as a foundational case study. Featuring four numerical and four categorical features, it is a commonly employed dataset in literature due to its typical structure and pronounced clustering tendencies across various dimension reductions.

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
K-Prototypes	176.08	0.34	0.44	1.23
Modha Spangler	176.08	0.34	0.44	1.23
KAMILA	176.08	0.34	0.44	1.23
ClustMD	176.08	0.34	0.44	1.23
MixtComp	176.08	0.34	0.44	1.23
Phillip & Ottaway	176.08	0.34	0.44	1.23
DenseClus	176.08	0.34	0.44	1.23
Pretopo FAMD	182.22	0.66	0.65	0.70
Pretopo UMAP	163.89	0.36	0.44	1.24
Pretopo PaCMAF	61.80	0.34	0.29	1.12
PretopoMD	105.17	0.239	0.263	1.71

Table 2 Results of the selected Algorithms on the Palmer Penguins dataset.

Several clustering algorithms, such as DenseClus and those leveraging the Elbow Method, consistently partition the Palmer Penguins dataset into two clusters. The Elbow Method, by design, dictates a fixed number of clusters—in this case, two—resulting in identical outcomes for algorithms adhering to its constraints. The PretopoMD algorithm also identifies two clusters, albeit with a more equitable distribution; however, these clusters yield lower scores in the chosen evaluation metrics.

Contrastingly, the pretopological algorithm, implemented in three iterations with dimensionality reduction, presents diverse outcomes. The UMAP variant identifies three distinct clusters, PaCMAF discerns eleven clusters along with 112 outliers, and the FAMD iteration subdivides the data into twenty-six clusters. Intriguingly, the FAMD iteration, which produces the most clusters, showcases superior performance indices, suggesting its delineation captures a rich set of intrinsic dataset information. Notably, the FAMD inertia for this dataset peaks at an impressive 98.2%, indicating a high representation of data variance. Refer to Table 2 for specific evaluation metric scores.

4.4 Sponge

The Sponge dataset, characterized by numerous categorical features and a relatively modest sample size, presents distinct characteristics when subjected to clustering algorithms. When applying FAMD to this dataset, a lower inertia of 86.13% is observed compared to the Penguin dataset. The Hopkins statistic, indicative of the dataset's clustering tendency, is low at 0.63, as supported by the iVAT visualization, which reveals no clear evidence of inherent cluster structures.

In contrast, PaCMAF yields a significantly higher Hopkins statistic (0.88) and a more discernible iVAT, implying a comparatively simpler task for subsequent clustering algorithms, though not necessarily leading to superior clusters.

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
K-Prototypes	16.83	0.170	0.421	2.01
Modha Spangler	16.72	0.168	0.404	2.03
KAMILA	16.83	0.170	0.421	2.01
ClustMD	-	-	-	-
MixtComp	-	-	-	-
Phillip & Ottaway	15.66	0.161	0.437	2.09
DenseClus	-	-	-	-
Pretopo FAMD	1.65	-0.061	-0.168	1.46
Pretopo UMAP	7.53	0.084	0.142	2.79
Pretopo PaCMAP	16.76	0.169	0.431	2.00
PretopoMD	6.36	0.484	0.012	0.383

Table 3 Results of the selected Algorithms on the Sponge dataset.

The evaluation metrics for the Sponge dataset, including Calinski-Harabasz scores, are approximately an order of magnitude lower than those observed for the Penguin dataset, indicating the presence of less well-defined clusters. Notably, K-prototype and KAMILA achieve the highest CH scores, followed by Pretopo-PaCMAP and Modha-Spangler. The Silhouette Gower score is highest for Phillip & Ottaway.

Pretopological FAMD clustering exhibits a notably low Davies-Bouldin score, identifying 63 outliers and forming 6 clusters, each consisting of 2 to 3 elements. While the clusters are distinct, the other indices are significantly low, emphasizing that the clusters do not demarcate a clear partition of the dataset. This highlights the importance of using multiple indicators to gauge clustering quality.

Conversely, PretopoMD identifies a single, dominant cluster comprising 74 elements and only one outlier. Given the dataset's weak clustering propensity, this outcome seems relevant, with PretopoMD boasting the highest FAMD Silhouette score and the lowest DB score. For specific metric scores, refer to the detailed results in the respective evaluation Table 3.

4.5 Dataset generator

To evaluate the different algorithms over every desired configurations, we use a dataset generator. The most common way to generate datasets to benchmark and evaluate clustering algorithms is to generate isotropic gaussian blobs. This method is natively present in the widely used scikit-learn for Python by [31], MixSim for R by [32] and Linfa for Rust³.

First, we generated cluster centers, with an average pairwise distance of 1. Then we generate samples from a gaussian mixture model with the density described by:

³<https://rust-ml.github.io/linfa/>

$$p(x) = \frac{1}{k} \sum_{i=1}^k \mathcal{N}(\mu_i, \Sigma_i) \quad (5)$$

where:

- k is the number of clusters; - μ_i are the cluster centers; - Σ_i refers to the cluster covariances. Here, it is a diagonal matrix of the clusters variance.

Inspired by [33], we split features upon quantiles to transform them into categorical features. Thus, we get a mixed dataset. With this method, the different parameters we can tune to obtain different configurations are:

- The number of samples to generate (the number of individuals);
- The number of clusters k ;
- The number of numerical features;
- The number of categorical features;
- The number of unique values taken by categorical variables;
- The standard deviation of clusters.

4.6 Generated dataset with 3 clusters

In the case of the Base Generated dataset with 500 individuals, 5 features (numerical/categorical), 3 clusters, 3 categorical unique values, and a standard deviation of 0.1, the Elbow Method suggests $k = 3$ as the optimal number of clusters. This alignment with the intended number of clusters provides an advantage to algorithms utilizing the Elbow Method.

As a result, algorithms employing the Elbow Method produce very similar partitions of the dataset, with closely aligned results across all four evaluation indices. The two algorithms utilizing UMAP, DenseClus, and Pretopo-UMAP, also yield similar results even without relying on the Elbow Method.

On the other hand, Pretopo-FAMD reports 444 outliers, and Pretopo-PaCMAP identifies six clusters with 124 outliers. Meanwhile, PretopoMD detects three sizable clusters and 60 outliers. As observed in other datasets, adjusting the hyperparameters for PretopoMD might lead to either improvements or deteriorations in the obtained results. For detailed metric scores, refer to the specific evaluation table showcasing the results for the Base Generated Case dataset (Table 4).

4.7 Generated Dataset with 10 clusters

For the generated dataset with 10 clusters, created with the same parameters as the previous case but with the distinction of having 10 clusters, the Elbow Method deviates from the intended number of clusters, identifying $k = 2$ as optimal.

Pretopo FAMD identifies 470 outliers and pinpoints 14 small clusters, each containing fewer than 10 individuals. Despite this seemingly suboptimal clustering, it achieves the best Davies-Bouldin score. DenseClus and PretopoMD both identify approximately 300 outliers among the 500 data points, even though the generated datasets are not designed to contain noise.

Lastly, Pretopo-UMAP divides the dataset into 8 distinct clusters and achieves the highest scores across both Silhouette versions and the Calinski-Harabasz index. It

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	296.330	0.394	0.521	1.071
DenseClus	295.224	0.393	0.519	1.074
Phillip & Ottaway	290.556	0.389	0.514	1.083
Kamila	296.330	0.394	0.521	1.071
K-Prototypes	296.330	0.394	0.521	1.071
MixtComp	294.906	0.393	0.518	1.075
Modha-Spangler	296.330	0.394	0.521	1.071
Pretopo-FAMD	34.160	0.067	0.061	1.453
Pretopo-UMAP	293.835	0.392	0.516	1.078
Pretopo-PaCMAP	66.563	0.049	0.013	1.845
PretopoMD	127.338	0.230	0.308	1.508

Table 4 Results of the selected Algorithms on the Base Generated Case.

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	95.934	0.175	0.197	2.216
DenseClus	81.313	0.183	0.200	1.914
Phillip & Ottaway	70.949	0.194	0.213	1.190
Kamila	86.348	0.160	0.157	2.120
K-Prototypes	92.096	0.168	0.185	2.263
MixtComp	83.316	0.163	0.205	2.163
Modha-Spangler	79.214	0.153	0.206	2.421
Pretopo-FAMD	14.482	-0.015	-0.018	2.219
Pretopo-UMAP	127.624	0.353	0.359	1.122
Pretopo-PaCMAP	66.870	0.174	0.154	1.557
PretopoMD	48.178	0.108	0.164	2.197

Table 5 Results of the selected Algorithms on a generated dataset with 10 clusters.

also holds the second-best position for the Davies-Bouldin score. For detailed metric scores, refer to the specific evaluation Table 5 showcasing the results for the Generated Dataset with 10 Clusters.

4.8 Generated dataset of high dimension

Then, we analyze how the different algorithms perform in a high dimension context (15 quantitative features and 15 categorical features). To do so, we generate a dataset with 15 clusters of each size.

There, the Elbow Method finds $k = 2$ clusters. ClustMD, PretopoMD, pretopo UMAP and MixtComp don't converge on such a dataset, and only produce noise. Pretopo FAMD finds 498 outliers out of the 500 individuals. Pretopo UMAP produces 1 cluster of 332 individuals and 168 outliers (that might be merged into another cluster). DenseClus and Pretopo PaCMAP both find 3 balanced clusters, and the latter obtains a slightly better score on the 4 indices.

Kamila, K-prototype and Phillip and Ottaway have merged two clusters into one and therefore find a cluster of approximatively 333 elements and another one of around

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	0.000	-1.000	-1.000	-1.000
DenseClus	118.048	0.218	0.102	1.723
Phillip & Ottaway	119.284	0.195	0.087	1.831
Kamila	120.934	0.196	0.088	1.817
K-Prototypes	118.048	0.191	0.085	1.862
MixtComp	0.000	-1.000	-1.000	-1.000
Modha-Spangler	120.934	0.196	0.088	1.817
Pretopo-FAMD	0.631	-0.030	-0.001	3.079
Pretopo-UMAP	0.000	-1.000	-1.000	-1.000
Pretopo-PaCMAP	122.833	0.227	0.107	1.681
PretopoMD	0.000	-1.000	-1.000	-1.000

Table 6 Results of the selected Algorithms on a generated dataset with 15 quantitative features and 15 categorical features.

	Calinski Harabasz	Silhouette FAMD	Silhouette Gower	Davies Bouldin
ClustMD	184.929	0.150	0.132	2.109
DenseClus	133.900	0.144	0.142	2.987
Phillip & Ottaway	5.673	0.100	0.157	2.558
Kamila	184.963	0.150	0.132	2.117
K-Prototypes	181.246	0.148	0.128	2.145
MixtComp	169.897	0.142	0.144	2.204
Modha-Spangler	169.793	0.143	0.145	2.202
Pretopo-FAMD	1.199	0.017	0.026	2.160
Pretopo-UMAP	162.758	0.136	0.135	2.261
Pretopo-PaCMAP	202.114	0.191	0.190	1.848
PretopoMD	17.020	-0.032	-0.055	2.800

Table 7 Results of the selected Algorithms on a generated dataset with 1000 individuals, 10 dimensions of each type, and a deviation of 0.15

137 elements. For a detailed breakdown of metric scores, please refer to the dedicated evaluation Table 6.

4.9 Generated Dataset with sparse clusters

The performance of various algorithms was studied on datasets with sparser clusters, specifically a deviation of 0.15, in contrast to the base case's deviation of 0.10. The dataset comprises 1000 individuals, with 10 dimensions of each type. The Elbow method suggests $k = 2$ as the optimal number of clusters, while the dataset is intended to contain 3 clusters.

The algorithms that obtain the optimal scores are the algorithms that use UMAP and PaCMAP. As those reduction move the neighbors closer to each other, it is not surprising to see them perform well on datasets with a higher clusters deviation. Among those 3, Pretopo PaCMAP is the only one that produces no outlier, therefore it obtains the best Silhouette scores. For a detailed breakdown of metric scores, please refer to the dedicated evaluation Table 7.

4.10 Measures and comparison of performances

To assess the performance of various clustering methods, we conducted experiments on several generated datasets with diverse specifications. The execution statistics presented here were obtained on a system equipped with an AMD Ryzen 7 5800H CPU, operating at a frequency of 3.20GHz, featuring 512KB of L1 cache and 32GB of DDR4 RAM. The detailed results are accessible on GitHub, and a more comprehensive discussion will be provided in an upcoming paper.

Memory Usage and Computational Time Comparison (Dataset: 500 Elements, 5 Numerical Features, 5 Categorical Features) Figure 9 offers a visual representation comparing the memory usage and computational time of different clustering methods. Notable observations include:

- PretopoMD Algorithm: Demonstrates the lowest memory usage but requires substantial computation time. Analyzing datasets of varying sizes indicates that, for this implementation of the algorithm, computation time scales linearly with the number of individuals and remains invariant with respect to the number of features (refer to Figure 10).
- ClustMD: Exhibits the highest computation time among the algorithms considered.

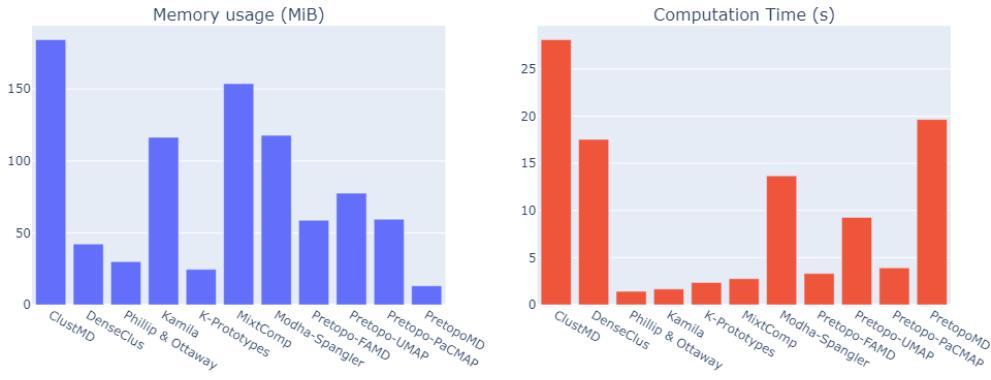


Fig. 9 Memory usage and computation time on a 500 elements generated dataset. With 5 numerical features and 5 categorical features.

4.11 Comparison with SOTA methods

new subsection

PretopoMD distinguishes itself by directly processing mixed data without requiring preliminary dimensionality reduction, thereby mitigating the risks of information loss and projection bias commonly encountered in high-dimensional transformations. Its capability to operate on heterogeneous data types ensures that the intrinsic variability of numerical and categorical features is preserved. Furthermore, PretopoMD yields a hierarchical clustering structure that produces an interpretable dendrogram, enabling

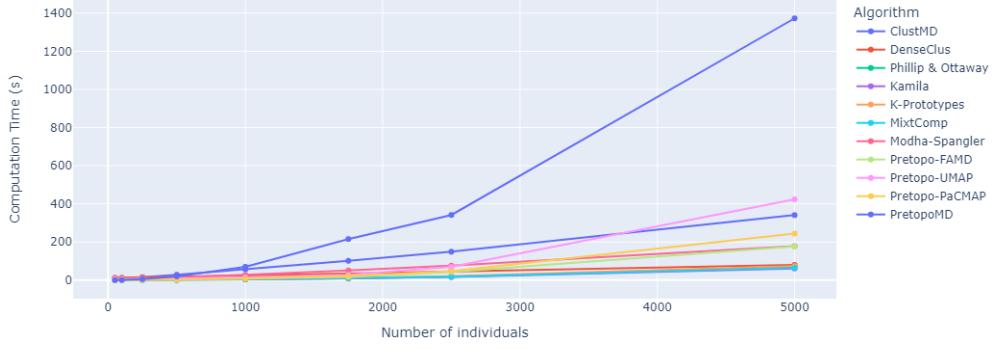


Fig. 10 Computation time per number of elements.

the exploration of nested, multi-level clusters without the necessity to predefine the number of clusters. The integration of customizable logical rules further enhances the method’s interpretability, allowing researchers to incorporate domain-specific knowledge into the clustering process and derive more meaningful insights.

In contrast, methods such as K-Prototypes and Convex K-Means, although computationally efficient, require pre-specification of cluster counts. For instance, K-Prototypes combines Euclidean and matching distances to jointly handle numerical and categorical data but relies on a simplistic categorical distance measure and necessitates careful tuning of balancing parameters, potentially limiting its effectiveness. Similarly, Convex K-Means improves convergence through convex-hull-based refinement of centroid positions; however, its additional computational overhead and the inherent requirement for a predetermined number of clusters restrict its applicability in more complex scenarios.

Model-based approaches, including MixtComp, Kamila, and ClustMD, offer a probabilistic framework through latent variable modeling that can provide deep insights into the underlying data structure. Despite their rigorous statistical foundations and potential for rich interpretability, these methods are often hampered by high computational costs and convergence issues related to Bayesian estimation or parameter initialization. In applications demanding rapid and intuitive clustering outcomes, such complexities may render these approaches less practical.

Overall, PretopoMD’s strengths lie in its flexibility and enhanced explainability. By eliminating the need for a preset number of clusters and delivering a dendrogram that integrates logical rules, it fosters a transparent clustering process especially advantageous for domains where interpretability is paramount. However, this benefit is counterbalanced by its significant computational complexity—stemming from the iterative pseudoclosure procedure and adjacency matrix construction—which becomes more pronounced when dealing with large or high-dimensional datasets. Moreover, the method’s performance is highly sensitive to hyperparameter settings, necessitating meticulous tuning and substantial domain expertise.

In summary, while PretopoMD provides a robust and explainable framework for clustering mixed data, its adoption requires careful consideration of the trade-offs

between computational efficiency and interpretability. In environments where scalability and rapid computation are essential, simpler techniques like K-Prototypes or Convex K-Means may be preferable. Conversely, in scenarios that prioritize the interpretability of clusters and the ability to capture subtle, domain-specific patterns, PretopoMD represents a compelling alternative despite its higher computational demands.

4.12 Discussion

Experimental results indicate that clustering performance often improves when applied after dimensionality reduction. This improvement can be partially attributed to the Curse of Dimensionality—a phenomenon that arises in high-dimensional spaces, particularly in clustering and machine learning tasks, where the exponential growth of search spaces hinders algorithm efficiency [34, 35]. Moreover, distance metrics that perform well in lower-dimensional spaces may lose their effectiveness in higher dimensions [36], making dimensionality reduction a valuable tool to alleviate these issues.

A detailed analysis of clustering outcomes using PretopoMD, in comparison with other methods across diverse datasets, reveals that the algorithm is sensitive to dataset characteristics such as dimensionality, noise level, and cluster dispersion. For example, PretopoMD demonstrates strong performance on balanced datasets with well-defined cluster boundaries—evidenced by competitive evaluation metrics on the Base Generated dataset—but shows reduced performance on high-dimensional or noisy datasets, as observed with the Sponge dataset. This variability is largely attributable to the algorithm’s dependency on carefully tuned hyperparameters, including threshold values, logical rules embedded in DNF, and seed sizes.

A comprehensive sensitivity analysis is underway to quantify the impact of these hyperparameters on cluster quality. Preliminary results indicate that even subtle variations in key parameters can result in differences in both the number of detected clusters and the proportion of outliers. Additionally, the hierarchical structure of PretopoMD enables the construction of dendograms, which offer intuitive visualizations of cluster relationships.

These findings underscore the critical need for hyperparameter tuning and dataset-specific customization when applying PretopoMD. Although the algorithm offers significant advantages—such as handling mixed data without prior dimensionality reduction and enhancing explainability through customizable parameters—its optimal performance relies on human expertise. Elements such as threshold values, seed selection, and logical rules require careful adjustment to balance computational efficiency with clustering quality, particularly in high-dimensional or noisy contexts.

Additionally, we are developing a DNF-based clustering approach in collaboration with energy experts. This approach aims to cluster buildings based on features like construction material and construction date, which are presumed to correlate with latent characteristics affecting energy consumption. The use of DNF enables us to explicitly formulate logical rules (e.g., “feature A above a certain threshold OR feature B within a specific range”), thereby improving interpretability. However, it is important to acknowledge that while DNF promotes clarity, its simplicity may restrict

expressiveness in datasets with complex interdependencies, potentially reducing cluster granularity.

To balance explainability with clustering quality, several strategies can be employed:

- **Multi-Stage Clustering Approach:** Initially conduct a fine-grained clustering that emphasizes the fidelity of the data. Subsequently, abstract the results into interpretable logical rules, preserving subtle patterns while ultimately presenting a clear summary of the clusters.
- **Automated Hyperparameter Tuning:** Utilize optimization techniques combined with internal (e.g., Silhouette score, Calinski-Harabasz index) and external metrics (e.g., ARI, MI) to determine the most appropriate hyperparameter settings. This approach helps in achieving an optimal balance between clustering fidelity and explainability.
- **Iterative Expert Feedback:** Engage domain experts in iterative evaluation cycles. Expert insights can guide further tuning of hyperparameters and the refinement of logical rules, ensuring that the clustering outcomes are both high in quality and consistent with domain-specific expectations.

By adopting these strategies, it becomes possible to harness the strengths of PretopoMD—namely its performance and explainability—while mitigating trade-offs between clarity and cluster granularity. This balanced approach is crucial, as high scores on quality indicators such as the Calinski-Harabasz index, Silhouette coefficient (computed with FAMD or Gower distances), or the Davies-Bouldin index do not necessarily coincide with the expectations of domain experts. In many real-world applications, experts may prioritize clusters that resonate with domain-specific insights—for instance, grouping buildings with similar operational constraints—even if such groupings yield comparatively lower scores on generic quality metrics. This observation highlights the necessity for human-centered evaluation criteria, underscoring that clustering should be approached not only as an optimization problem but also as a tool for deriving meaningful, domain-relevant insights.

5 Conclusion

This study addresses the intricate challenges of clustering mixed data, with particular emphasis on applications in the energy sector and other diverse fields. We provide a comprehensive overview of existing clustering methods and dimensionality reduction techniques for mixed data, critically discussing their strengths and limitations. A central theoretical contribution of this work is the introduction of the pretopology-based algorithm, pretopoMD, which offers a novel perspective on how mixed data can be effectively clustered.

PretopoMD presents several key innovations. It allows for Customizable Logical Rules, enabling the definition of tailored rules for cluster construction that provide flexibility to meet specific application requirements. It also features Adjustable Hyperparameters, which facilitate fine-tuning for clustering and division conditions and thus offer precise control over the clustering process. Additionally, the algorithm supports

Explainable Cluster Construction by enabling hierarchical dendrogram analysis that enhances the transparency and interpretability of clustering outcomes.

Our experimental results underscore that, although pretopoMD may involve relatively higher computational times, it exhibits low memory usage—a crucial advantage when handling large-scale datasets. While these outcomes validate its potential as a robust and interpretative solution for mixed data clustering, the study also acknowledges limitations, such as the increased computational burden and the need for further hyperparameter optimization. Future research should therefore focus on improving the algorithm’s efficiency, exploring its applicability to richer energy datasets, and extending its use to other domains including biology, medicine, marketing, and economics.

In summary, this work contributes new theoretical insights and practical tools to the domain of mixed data clustering by integrating customization, fine-tuning, and transparency into its framework. These advances not only deepen our understanding of clustering heterogeneous data but also pave the way for future innovations in managing complex datasets across various fields.

Declarations

Funding

This paper is the result of research conducted at the energy data management company *Energisme*. We thank *Energisme* for the resources that have been made available to us and Julio Laborde for his assistance with the conception of our pretopological hierarchical algorithm library.

Conflict of interest/Competing interests

Not applicable.

Ethics approval and consent to participate

Not applicable.

Consent for publication

All authors consent for publication.

Data availability

All datasets are given in the github <https://github.com/ClementCornet/Benchmark-Mixed-Clustering>.

Materials availability

All methods with a README are given in the github <https://github.com/ClementCornet/Benchmark-Mixed-Clustering>.

Code availability

All codes are given in the github <https://github.com/ClementCornet/Benchmark-Mixed-Clustering>.

Author contribution

Authors equally contribute to this paper.

References

- [1] Abumaloh, R., Alrahal, M., El-Haggar, N., Alsumayt, A., Alfawaer, Z., Aljameel, S.: Exploring individuals' experiences with security attacks: A text mining and qualitative study. Emerging Science Journal **8**(1) (2024)
- [2] Ahmad, A., Khan, S.S.: Survey of state-of-the-art mixed data clustering algorithms. IEEE Access **7**, 31883–31902 (2019)
- [3] Caruso, G., Gattone, S., Fortuna, F., Di Battista, T.: Cluster analysis for mixed data: An application to credit risk evaluation. Socio-Economic Planning Sciences **73**, 100850 (2021)
- [4] Han, L., Shen, P., Yan, J., Huang, Y., Ba, X., Lin, W., Wang, H., Huang, Y., Qin, K., Wang, Y., *et al.*: Exploring the clinical characteristics of covid-19 clusters identified using factor analysis of mixed data-based cluster analysis. Frontiers in medicine **8**, 644724 (2021)
- [5] McParland, D., Phillips, C.M., Brennan, L., Roche, H.M., Gormley, I.C.: Clustering high-dimensional mixed data to uncover sub-phenotypes: joint analysis of phenotypic and genotypic data. Statistics in medicine **36**(28), 4548–4569 (2017)
- [6] Yu, W., Qiang, G., Xiao-li, L.: A kernel aggregate clustering approach for mixed data set and its application in customer segmentation. In: 2006 International Conference on Management Science and Engineering, pp. 121–124 (2006). IEEE
- [7] Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data mining and knowledge discovery **2**(3), 283–304 (1998)
- [8] Huang, Z.: Clustering large data sets with mixed numeric and categorical values. In: Proceedings of the 1st Pacific-asia Conference on Knowledge Discovery and Data mining,(PAKDD), pp. 21–34 (1997). Citeseer
- [9] Modha, D.S., Spangler, W.S.: Feature weighting in k-means clustering. Machine learning **52**(3), 217–237 (2003)
- [10] Biernacki, C.: Bigstat for big data: Big data clustering through the bigstat saas platform. In: Journée Scientifique Big Data & Data Science (2016)

- [11] Foss, A., Markatou, M., Ray, B., Heching, A.: A semiparametric method for clustering mixed data. *Machine Learning* **105**(3), 419–458 (2016)
- [12] McParland, D., Gormley, I.C.: Model based clustering for mixed data: clustmd. *Advances in Data Analysis and Classification* **10**(2), 155–169 (2016)
- [13] Philip, G., Ottaway, B.: Mixed data cluster analysis: an illustration using cypriot hooked-tang weapons. *Archaeometry* **25**(2), 119–133 (1983)
- [14] Liu, S., Cao, W., Fu, R., Yang, K., Yu, Z.: Rpsc: robust pseudo-labeling for semantic clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 14008–14016 (2024)
- [15] Yu, Z., Dong, Z., Yu, C., Yang, K., Fan, Z., Chen, C.P.: A review on multi-view learning. *Frontiers of Computer Science* **19**(7), 197334 (2025)
- [16] Iqbal, I., Ullah, I., Peng, T., Wang, W., Ma, N.: An end-to-end deep convolutional neural network-based data-driven fusion framework for identification of human induced pluripotent stem cell-derived endothelial cells in photomicrographs. *Engineering Applications of Artificial Intelligence* **139**, 109573 (2025)
- [17] Yu, Z., Zhong, Z., Yang, K., Cao, W., Chen, C.P.: Broad learning autoencoder with graph structure for data clustering. *IEEE Transactions on Knowledge and Data Engineering* **36**(1), 49–61 (2023)
- [18] Chenghu, C., Thammano, A.: A novel classification model based on hybrid k-means and neural network for classification problems. *HighTech and Innovation Journal* **5**(3), 716–729 (2024)
- [19] Shi, Y., Yang, K., Yu, Z., Chen, C.P., Zeng, H.: Adaptive ensemble clustering with boosting bls-based autoencoder. *IEEE Transactions on Knowledge and Data Engineering* **35**(12), 12369–12383 (2023)
- [20] Yaro, A.S., Maly, F., Prazak, P., Malý, K.: Improved fingerprint-based localization based on sequential hybridization of clustering algorithms. *Emerging Science Journal* **8**(2), 394–406 (2024)
- [21] Escofier, B.: Traitement simultané de variables qualitatives et quantitatives en analyse factorielle. *Cahiers de l'Analyse des Données* **4**(2), 137–146 (1979)
- [22] Wang, Y., Huang, H., Rudin, C., Shaposhnik, Y.: Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *Journal of Machine Learning Research* **22**(201), 1–73 (2021)
- [23] Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* **3**(1), 1–27 (1974)

- [24] Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* **20**, 53–65 (1987)
- [25] Davies, D.L., Bouldin, D.W.: A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* (2), 224–227 (1979)
- [26] Auray, J.-P., Bonnevay, S., Bui, M., Duru, G., Lamure, M.: Prétopologie et applications : un état de l’art. *Studia Informatica Universalis* (Hermann) **7**, 27–44 (2009)
- [27] Le, T.V.: Classification prétopologique des données: application à l’analyse des trajectoires patients. PhD thesis, Lyon 1 (2007)
- [28] Laborde, J.: Pretopology, a mathematical tool for structuring complex systems: methods, algorithms and applications. PhD thesis, Paris Sciences et Lettres (ComUE) (2019)
- [29] Belmandt, Z., Fortet, R.M.: Manuel de prétopologie et ses applications: sciences humaines et sociales, réseaux, jeux, reconnaissance des formes, processus et modèles, classification, imagerie, mathématiques (1993)
- [30] Gorman, K.B., Williams, T.D., Fraser, W.R.: Ecological sexual dimorphism and environmental variability within a community of antarctic penguins (genus pygoscelis). *PloS one* **9**(3), 90081 (2014)
- [31] Kramer, O., Kramer, O.: Scikit-learn. Machine learning for evolution strategies, 45–53 (2016)
- [32] Melnykov, V., Chen, W.-C., Maitra, R.: Mixsim: An r package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software* **51**, 1–25 (2012)
- [33] Costa, E., Papatsouma, I., Markos, A.: Benchmarking distance-based partitioning methods for mixed-type data. *Advances in Data Analysis and Classification*, 1–24 (2022)
- [34] Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is “nearest neighbor” meaningful? In: Database Theory—ICDT’99: 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7, pp. 217–235 (1999). Springer
- [35] Verleysen, M., François, D.: The curse of dimensionality in data mining and time series prediction. In: Computational Intelligence and Bioinspired Systems: 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8–10, 2005. Proceedings 8, pp. 758–770 (2005). Springer

- [36] Steinbach, M., Ertoz, L., Kumar, V.: The challenges of clustering high dimensional data. *New directions in statistical physics: econophysics, bioinformatics, and pattern recognition*, 273–309 (2004)