



Using Bagging to improve clustering methods in the context of three-dimensional shapes

Inácio Nascimento¹ · Raydonal Ospina^{1,2}  · Getúlio Amorim¹

Received: 10 January 2023 / Revised: 7 June 2024 / Accepted: 6 August 2024
© Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

Cluster Analysis techniques are a common approach to classifying objects within a dataset into distinct clusters. The clustering of geometric shapes of objects holds significant importance in various fields of study. To analyze the geometric shapes of objects, researchers often employ Statistical Shape Analysis methods, which retain crucial information after accounting for scaling, locating, and rotating an object. Consequently, several researchers have focused on adapting clustering algorithms for shape analysis. Recently, three-dimensional (3D) shape clustering has become crucial for analyzing, interpreting, and effectively utilizing 3D data across diverse industries, including medicine, robotics, civil engineering, and paleontology. In this study, we adapt the *K-means*, *CLARANS* and *Hill Climbing* methods using an approach based on the *Bagging* procedure to achieve enhanced clustering accuracy. We conduct simulation experiments for both isotropy and anisotropy scenarios, considering various dispersion variations. Furthermore, we apply the proposed approach to real datasets from relevant literature. We evaluate the obtained clusters using cluster validation measures, specifically the Rand Index and the Fowlkes-Mallows Index. Our results demonstrate substantial improvements in clustering quality when implementing the *Bagging* approach in conjunction with the *K-means*, *CLARANS* and *Hill Climbing* methods. The combination of the Bagging method and clustering algorithms provided substantial gains in the quality of the clusters.

✉ Raydonal Ospina
raydonalmartinez@ufba.br

Inácio Nascimento
inacio.nascimento@ufpe.br

Getúlio Amorim
gjaa@de.ufpe.br

¹ Departamento de Estatística, CASTLab, Universidade Federal de Pernambuco, 50.740–540, Recife, Pernambuco, Brazil

² Departamento de Estatística, LInCa, Universidade Federal de Bahia, 40.170–110, Salvador, Bahia, Brazil

Keywords Cluster analysis · Statistical shape analysis · Three-dimensional shapes · Bagging · Clustering validation

Mathematics Subject Classification 62H10 · 62H30 · 62T10 · 68T09 · 68Q87

1 Introduction

In many everyday situations, it is relevant to classify individuals from a dataset into groups, either to help understand the phenomenon studied or to organize data. Cluster Analysis is a set of techniques that aims to create clusters so that each element is similar, but the clusters are different. Clustering methods differ in several ways, Everitt et al. (2011) comment that the two main methods are hierarchical and partitional. The methods of interest in the present work are the partitioning ones, which form groups from an initial partition and the pre-defined number K of clusters. Initially, there is an initial partition, and, according to the algorithm process, the elements change from cluster to cluster until their formats reach a final version (final clustering) (Friedman and Rubin 1967).

The K -means method, proposed over 50 years ago, is based on the sum of squares, also known as the “Within-cluster Sum of Squares” (WSS). It aims to minimize the sum of the squared distances between data points and the centroids of their respective clusters (Hastie et al. 2009; Jain 2010). Another well-known method is *CLARANS* (*Clustering Algorithm based on Randomized Search*), introduced by Ng and Han (2002), which utilizes a representative object called the *medoid*, positioned closer to the center of the cluster. The medoid reduces the search space, enhances robustness against outliers, and allows for identifying complex cluster structures. Additionally, the *Hill Climbing* method, proposed by Friedman and Rubin (1967), aims to iteratively find the cluster with the optimal value of a given clustering criterion, such as WSS or the “Between-cluster Sum of Squares” (BSS).

Geometric representations and images of objects play a vital role in studies and research across various fields and applications, including Biology, Medicine, Neuroscience, Archaeology, and, with the advancement of technological resources, Logic, Computer Vision, and Pattern Recognition (Adams and Otárola-Castillo 2013; Srivastava et al. 2005; Baxter 2015; Srivastava and Klassen 2016; Guo et al. 2023; King and Eckersley 2019). Statistical Analysis of Shapes is a branch of statistics used to work with geometric representations and shapes of objects. In the geometric approach to shape analysis, the central idea is to utilize the representation of the object itself. Certain mathematical operations remove the effects of location, scale, and rotation, encapsulating all the information in the shape. Some related practical applications involve comparing the shapes of the brain cortex in patients with and without schizophrenia in a neuroscientific study based on brain magnetic resonance (Brignell et al. 2010).

Studying summary measures and comparisons between shapes is necessary in several areas of knowledge to gain a deeper and more comprehensive understanding of objects and phenomena in various research and application fields. Developing different statistical techniques in the context of shapes provides a systematic and quantitative approach for applications and theoretical development. Bookstein et al. (1986) and

Kendall (1984) proposed the fundamental concepts of statistical analysis of shapes. In several situations, it is necessary to classify data sets of shapes that belong to non-Euclidean spaces into clusters. In this context, several authors have adapted algorithms for this purpose. For two-dimensional shapes (2D), there is the work by Amaral et al. (2010), who adapted the *K-means* algorithm proposed by Hartigan and Wong (1979) for an application in Oceanography. In their study, they needed to classify species of fish based on their shapes.

One of the main advantages of working with three-dimensional data is the ability to analyze the surface area and volume of objects of interest and their shapes. Although there are many applications of three-dimensional data in science and technology, there are relatively few studies related to 3D shapes in the literature. Moving from 2D shapes to 3D is not simply the addition of a new dimension, as the 3D shape space is part of a stratified space that contains some singularities¹ These singularities can make the analysis more complex and computationally costly, unlike the 2D shape space, a Riemannian manifold (Bhattacharya and Bhattacharya 2012).

Few works involving adaptations of clustering methods address three-dimensional (3D) shapes. One of the most significant works in this area, developed by Vinué et al. (2014), presents adaptations of the *K-means* (Lloyd 1982) and *trimmed K-means* (García-Escudero and Gordaliza 1999) versions for clustering three-dimensional shapes. Despite the scarcity of these methods, the present work also aims to adapt the *CLARANS* (Ng and Han 2002) and *Hill Climbing* (Friedman and Rubin 1967) methods to the context of three-dimensional shapes.

Clustering methods can benefit from leveraging the strengths of other algorithms. The *Bagging* method has been applied in conjunction with clustering algorithms to enhance their performance by incorporating new training sets formed through *bootstrap* samples into the cluster analysis framework. An example of clustering ensembles is the *bagged clustering* method introduced by Leisch (1999), which combines hierarchical and partitioning methods in cluster analysis to improve stability. Additionally, to improve the clustering performance of partitioning algorithms based on medoids, such as *PAM* (Rousseeuw and Kaufman 1990), Dudoit and Fridlyand (2003) utilized the *Bagging* method for data analysis and stabilize the results. The method provides robustness against outliers, consistency, and lower sensitivity to the dimensionality of the data.

Regarding the use of clustering ensemble techniques in the context of shape analysis, the work by Assis et al. (2021) introduces the *Bagging* method, proposed by Breiman (1996), in conjunction with the *K-means* algorithm, proposed by MacQueen (1967), to enhance the quality of clustering results for 2D shapes. The method is more resistant to random fluctuations in the data.

Motivated by all these works, this paper aims to leverage the *Bagging* method, specifically *BagClust1* proposed by Dudoit and Fridlyand (2003), to enhance the quality of clustering results for three-dimensional shapes obtained using the adapted versions of *K-means*, *CLARANS* and *Hill Climbing* algorithms in the context of 3D

¹ Here, singularities refer to points or regions in the space of shapes where abrupt changes or non-smooth behaviors occur in the geometric properties of objects. For example, Peaks and Valleys where the surface curvature is very high or low, Confluence Points where multiple parts of the shape come together, and Fold Points where folds or discontinuities occur.

shapes. By incorporating the *Bagging* method, we seek to improve the clustering outcomes' robustness and resistance to fluctuations. To evaluate the clustering quality of these methods, the Rand Index (Rand 1971) and the Fowlkes-Mallows Index (Fowlkes and Mallows 1983) are used as performance measures. To validate the results of the metrics obtained via simulations, we used the paired Wilcoxon test (Wilcoxon 1992).

This paper is organized as follows. Section 2 deals with the basic concepts of statistical analysis of shapes. In Sect. 3, we present how the algorithms under study can be adapted for statistical shape analysis in three-dimensional space. A simulation study and applications on real datasets are presented in Sect. 4. Finally, the conclusions are discussed in Sect. 5.

2 Background

According to Kendall (1977), the geometric information of an object that remains when the effects of location, scale, and rotation are removed is referred to as *shape*. Additionally, the book by Dryden and Mardia (2016) synthesizes the main concepts of shape analysis and provides a overview of the necessary definitions for statistical shape analysis to establish the notation used throughout this paper. One way to characterize the shape of an object is to detect a finite set of points around its silhouette, which are called landmarks.

A configuration is a collection of *landmarks* in a given object, mathematically represented by a matrix \mathbf{X} of dimension $k \times m$, where k is the number of landmarks and m is the number of dimensions (Cartesian coordinates) for each landmark. The space of all possible coordinates of the landmarks is called the *configuration space*. In this work, we will study the shapes of objects in three dimensions, i.e., the cases where $m = 3$. Next, we present some definitions for statistical shape analysis according to Dryden and Mardia (2016).

Definition 1 The **shape space** is the space of all shapes. It is the set of equivalence classes of the configuration matrices under the action of Euclidean similarity transformations (location, rotation, and scale).

The shape space admits a Riemannian manifold structure, meaning standard statistical methods cannot be directly applied. The complexity of working with this structure depends on the values of k and m . When $m = 2$, the shape space is a complex projective space (Goodall and Mardia 1999). However, for dimensions $m \geq 3$, the shape space is not a Riemannian manifold but a stratified space (Dryden and Mardia 2016). This space contains singularities and is less familiar than a Riemannian manifold. Nevertheless, in practical applications, it is generally assumed that we are far from these singularities and confined to a restricted variation within the shape space.

However, the configuration matrix \mathbf{X} does not adequately describe an object, as it is not invariant under Euclidean similarity transformations, i.e., location, scale, and rotation. The location and rotation effects of \mathbf{X} are removed one at a time. Initially, the Kendall coordinates (Kendall 1984) will be used to remove the location effect, employing the Helmert submatrix. The location effect is removed by multiplying \mathbf{X} by the Helmert submatrix \mathbf{H} (see Dryden and Mardia (2016), p. 49). The Helmert

matrix \mathbf{H}^F is an orthogonal matrix of dimension $k \times k$, where all the elements of the first row are equal to $1/\sqrt{k}$, and the j -th row has $(j - 1)$ elements equal to $-j(j + 1)^{\frac{1}{2}}$, followed by an element equal to $(j - 1) * (j(j + 1))^{\frac{1}{2}}$, and $(k - j)$ zeros.

In order to eliminate the scaling effect, the Helmertized configuration must be divided by its norm, as follows:

$$\mathbf{Z} = \frac{\mathbf{X}_H}{\|\mathbf{X}_H\|} = \frac{\mathbf{HX}}{\|\mathbf{HX}\|} = \frac{\mathbf{HX}}{\sqrt{\text{tr}((\mathbf{HX})^T(\mathbf{HX}))}}, \tag{1}$$

where \mathbf{Z} is called the *pre-shape* of the configuration matrix \mathbf{X} . It is important to realize that, in the pre-shape, the rotation effects remain.

Definition 2 The pre-shape space S_m^k is the set of all possible pre-shapes.

It can be mathematically represented as:

$$S_m^k = \left\{ \mathbf{Z} = \frac{\mathbf{HX}}{\|\mathbf{HX}\|} \mid \mathbf{X} \in \mathbb{R}^{k \times m} \right\}$$

where \mathbf{Z} is the pre-shape of the configuration matrix \mathbf{X} , $\mathbf{X} \in \mathbb{R}^{k \times m}$ represents the matrix of Cartesian coordinates of k landmarks in m dimensions. The space S_m^k contains all possible configurations after removing the scaling effect from the original shape space.

Definition 3 The shape of a configuration matrix is all the geometric information that remains after removing the effects of location, scale, and rotation. The shape can be represented by:

$$[\mathbf{Z}] = \{\mathbf{Z}\mathbf{\Gamma} : \mathbf{\Gamma} \in SO(m)\}, \tag{2}$$

where $\mathbf{\Gamma}$ is a rotation matrix, \mathbf{Z} is the pre-shape, and $SO(m)$ is the orthogonal group of rotations.

According to Vinué et al. (2014), S_m^k is a hypersphere of unit radius in $\mathbb{R}^{(k-1)m}$, a Riemannian submanifold that is widely studied and known. However, for $m > 2$, Σ_m^k is not a usual space. As Σ_m^k is considered a quotient space of S_m^k under rotations, it is easier and more intuitive to work in this space, given that it is a Riemannian submanifold.

In practice, comparing and measuring objects is of constant interest to understand their variations, similarities, and dissimilarities. Procrustes analysis is a widely used technique for statistical shape analysis. It deals with the comparison and alignment of shapes by removing the effects of translation, rotation, and scaling. The method aims to

fit shapes onto a standard template by minimizing the distance between corresponding points on the shapes through rigid transformations.

In this context, defining distance concepts aimed at shape analysis is essential. Consider two configuration matrices of k landmarks in m dimensions, denoted by \mathbf{X}_1 and \mathbf{X}_2 , with pre-shapes equal to \mathbf{Z}_1 and \mathbf{Z}_2 , respectively. Next, a distance measure focused on the analysis of shapes is defined.

Definition 4 The Riemannian distance $\rho(\mathbf{X}_1, \mathbf{X}_2)$ is the nearest great circle distance (over rotations) between \mathbf{Z}_1 and \mathbf{Z}_2 in the pre-shape hypersphere S_m^k .

The Riemannian distance is intrinsic, as it is defined in the space of the form Σ_m^k . For more comprehensive details, see Dryden and Mardia (2016). Yet,

Definition 5 The full Procrustes distance between \mathbf{X}_1 and \mathbf{X}_2 is:

$$d_F(\mathbf{X}_1, \mathbf{X}_2) = \inf_{\Gamma \in SO(m), \beta \in \mathbb{R}^+} \|\mathbf{Z}_2 - \beta \mathbf{Z}_1 \Gamma\|,$$

where β is a scalar.

The full Procrustes distance and the Riemannian distance have the following relationship: $d_F = \sin(\rho)$. Another important concept that needs to be defined is the mean shape. In statistical shape analysis, the definition of the mean shape plays a crucial role in data analysis. However, in non-Euclidean spaces, no concept of mean is equivalent to the commonly known arithmetic mean. To address this, we will employ a Fréchet-type mean (Fréchet 1948), a type of mean or average that minimizes the sum of squared distances to any shape.

Consider $\mathbf{X}_1, \dots, \mathbf{X}_n$ as a set of configuration matrices representing the coordinates of landmark points on shapes.

Definition 6 The full Procrustes mean shape in shape space is given by

$$[\hat{\mu}] = \arg \inf_{\mu} \sum_{i=1}^n d_F^2(\mathbf{X}_i, \mu), \quad (3)$$

where d_F represents the full Procrustes distance of Definition 5.

For two-dimensional data, where $m = 2$, Kent (1994) presents an eigenvector solution for the optimization problem in Definition 6 to find a mean shape. However, an iterative process becomes necessary for dimensions where $m \geq 3$ since the matching procedure cannot be expressed as a linear expression. The iterative process is outlined in Algorithm 1 [see pp. 138 in (Dryden and Mardia 2016) for more details].

Algorithm 1 Algorithm for calculating the mean shape of Procrustes $[\hat{\mu}]$.

- 1 **Translations:** Center settings to remove location effects. Let \mathbf{X} be a configuration matrix. Initially, consider:

$$\mathbf{X}_i^P = \mathbf{C}\mathbf{X}_i,$$

where $i = 1, \dots, n$ and $\mathbf{C} = \mathbf{H}^T\mathbf{H}$.

- 2 **Rotations:** Calculate $G = \frac{1}{n} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \|\mathbf{X}_i^P - \mathbf{X}_j^P\|^2$. For the i -th iteration, do:

$$\tilde{\mathbf{X}}_{(i)} = \frac{1}{n-1} \sum_{j \neq i} \mathbf{X}_j^P.$$

Optimize $\|\tilde{\mathbf{X}}_{(i)} - \mathbf{X}_i^P\Gamma\|^2$ by rotations. Let $\mathbf{X}_i^P = \mathbf{X}_i^P\hat{\Gamma}$, where $\hat{\Gamma}$ is the optimal rotation matrix. Repeat for all i . Calculate the new value of G . The process is repeated until G can no longer be reduced.

- 3 **Scaling:** For the i -th configuration, calculate:

$$\hat{\beta}_i = \left(\frac{\sum_{k=1}^n \|\mathbf{X}_k^P\|^2}{\|\mathbf{X}_i^P\|^2} \right)^{\frac{1}{2}} \phi_i,$$

where ϕ_i is the i -th eigenvector component ϕ , corresponding to the highest eigenvalue of the correlation matrix Φ of $\text{vec}\{\mathbf{X}_i^P\}$.

Consider $\mathbf{X}_i^P = \hat{\beta}_i\mathbf{X}_i^P$. Repeat for all i . Calculate the new value of G .

- 4 **Repetition:** Repeat steps 2 and 3 until G can no longer be reduced.

- 5 Therefore, $[\hat{\mu}] = \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i^P$ is the mean shape.

3 Clustering three-dimensional shapes

The classification problem in shape analysis is to separate the objects in a dataset into groups based on their shapes. In this paper, the clustering methods discussed in the context of 3D shapes are the version of *K-means* adapted by Vinu  et al. (2014), and our adaptations of the *CLARANS* and *Hill Climbing* methods. In principle, we consider $\mathbf{X}^* = \{\mathbf{X}_i, i = 1, \dots, n\}$ as a set of n objects, where each of them is a configuration matrix, as defined in Sect. 2. The algorithms will form these objects into a set $G = (G_r, r = 1, \dots, K)$ of K clusters.

3.1 K-means clustering algorithm for three-dimensional shapes

The *K-means* clustering algorithm, introduced by Lloyd (1982), was adopted and built a new version Vinu  et al. (2014) to work with three-dimensional shape data. The Riemannian distance, presented in Definition 4, replaced the Euclidean distance used in the original algorithm.

The *K-means* optimization problem, in the context of shape analysis, aims to minimize the following criterion:

$$W = \sum_{r=1}^K \sum_{i \in G_r} \text{Dist}^2(\mathbf{X}_i, \boldsymbol{\mu}_r), \tag{4}$$

here, G_r , $r = 1, \dots, K$, represents the formed groups that contain each of the \mathbf{X}_i , $i = 1, \dots, n$, observations in the dataset. Note that a configuration matrix is a collection of *landmarks* of a given geometric object. The *Dist* function is a measure of distance. In this case, it is the Riemannian distance as defined in Definition 4, and μ_r represents the mean shape (*centroid*) obtained using Algorithm 1. The *K-means* clustering algorithm for three-dimensional shapes follows the steps below.

Algorithm 2 *K-means* algorithm for three-dimensional shapes.

- 1 **Input:** A set \mathbf{X}^* of configuration matrices and the K number of clusters.
 - 2 **Output:** Clusters formed ($G = G_r, r = 1, \dots, K$).
 - 3 **Assignment:** Arbitrarily select K configuration matrices (observations) as the initial centroids (initial partition) of the K clusters.
 - 4 **Partition:** For each observation, calculate the Riemannian distance from Definition 4 between observations and K -centroids. Then, assign each observation to the group with the closest centroid.
 - 5 **Update Centroids:** For each of K clusters, calculate the mean shape of Definition 6 by Algorithm 1. The mean shapes of each cluster will be considered the new centroids.
 - 6 **Repetition:** Repeat steps 4 and 5 until the observations no longer switch groups, that is, until a minimum value of the objective function in Equation (4).
-

The clusters formed by Algorithm 2 depend on an initial partition (randomly defined) representing the centroids, which can lead to convergence to a local optimum. It is necessary to run the algorithm multiple times with different initial partitions and then select the execution that yields the best value of the clustering criterion to mitigate this issue. The *K-means* method is primarily limited by its sensitivity to outliers in the dataset because it uses the sample mean to calculate the centroids of the clusters, allowing outliers to influence the resulting clusters significantly. Outliers can pull the centroids away from the central cluster, leading to suboptimal cluster assignments. Algorithm 2 updates centroids (means of groups) only after assigning all elements to a cluster. The algorithm performs a two-step process. First, it assigns all elements to the closest cluster centroid based on the distance criterion, and then it recalculates the centroids based on the newly formed clusters.

3.2 CLARANS clustering algorithm for three-dimensional shapes

Ng and Han (2002) proposed the *CLARANS* (Clustering Algorithm based on Randomized Search) algorithm as an extension of the *PAM* (Partitioning Around Medoids) algorithm. In *PAM*, the cluster representatives are called *medoids*. A medoid is the data point within a cluster with the lowest average dissimilarity (distance) to all other points in the cluster, which makes *PAM* less sensitive to outliers compared to *K-means*, as the medoids are actual data points, whereas *K-means* uses the mean of the cluster to represent its center.

The *CLARANS* forms clusters based on a random search approach, and the groups are generated by searches over the entire dataset, which is treated as a graph. Let $G_{n, K}$ be the graph, where *nodes* are defined as sets of medoids, and each node results in different clusterings. Each of these nodes has $K(n - K)$ neighbors, where K represents the number of clusters, and n is the number of individuals in the dataset. Two nodes

are considered neighbors if they differ by only one object. Consequently, it is possible to have different groups depending on the selected medoid set. A cost is assigned to each node, defined as the total dissimilarity between every object and the medoid of its group. Overall, the *CLARANS* searches for the optimal node in the complete graph, applying the *PAM* for a sample of neighbors.

The *CLARANS* algorithm has two main parameters: *maxneighbor*, which represents the maximum number of neighbors to be examined, and *numlocal*, which represents the number of search iterations for a minimal node. If the value of *maxneighbor* is very close to or equal to $K(n - K)$, the quality of the groups formed by *CLARANS* will be closer to those generated by *PAM*, and the search for a minimal node will take longer (Ng and Han 2002). Algorithm 3 presents the steps for adapting *CLARANS* to the context of 3D shapes, which is one of the proposals of the present work.

Algorithm 3 *CLARANS* algorithm for three-dimensional shapes.

- 1 **Input:** A set \mathbf{X}^* of configuration matrices, K number of clusters, *numlocal*, and *maxneighbor*.
 - 2 **Output:** Clusters formed ($G = G_r, r = 1, \dots, K$).
 - 3 **Initialization:** Initialize $i = 1$ (first iteration of fetch) and *mincost* as a large number.
 - 4 **Local Search:** Define *current* as a node (set of medoids) randomly selected from $G_{n,K}$.
 - 5 Initialize $j = 1$ (first neighbor analyzed).
 - 6 Let S be a random neighbor of *current*. Calculate the cost of current using the Riemannian distance 4.
 - 7 **Update:** If S have a lower cost, then S is defined as *current*; go to step 5.
 - 8 Otherwise, increment j in 1. If $j \leq \text{maxneighbor}$, go to step 6.
 - 9 If $j > \text{maxneighbor}$, compare the cost of *current* with *mincost*. If the cost of *current* is less than *mincost*, then *mincost* is defined as the cost of *current* and *current* is defined as the best node.
 - 10 **Repeat:** Increment i in 1. If $i > \text{numlocal}$, the cluster formed by the best node is returned as a result. If not, return to step 4.
-

In general, *CLARANS* searches for the optimal node in the entire graph by applying *PAM* in a sample of neighbors. These algorithms examine the notions of proximity between the partitions examined during the iterative process. By examining multiple neighboring partitions and considering the quality of their clustering solutions, *CLARANS* tries to overcome the limitations of traditional methods like *PAM* or *K-means*, mainly when dealing with noisy or large datasets. The iterative exploration of the search space through proximity-based evaluations enhances the effectiveness of *CLARANS* in forming meaningful and robust clusters (Ng and Han 2002).

3.3 Hill Climbing clustering algorithm for three-dimensional shapes

The *Hill Climbing* algorithm was initially proposed by Friedman and Rubin (1967). Such an algorithm motivated the development of algorithms designed to seek the optimal value of a clustering criterion, restructuring existing partitions and maintaining the new partition only if it provides improvements. This algorithm can also be called Hill Descending in cases where criteria minimization is required (Everitt et al. 2011). The *Hill Climbing* method is considered a search algorithm and aims to find the solution to a given problem by exploring a series of possible paths. The *Hill Climbing* algorithm is a loop that repeats itself continuously in search of an optimal value of the

clustering criterion. In this paper, this method was adapted to group three-dimensional shapes.

The clustering criterion that we used for the *Hill Climbing* method, according to Rousseeuw and Kaufman (1990), has the following equation, which can be used as a criterion to measure the quality of the groups:

$$TD = \sum_{r=1}^K \sum_{i \in G_r} d(\mathbf{X}_i, m_r). \quad (5)$$

Here, G_r , $r = 1, \dots, K$, represents the groups formed containing each of the observations \mathbf{X}_i , $i = 1, \dots, n$, from the data set. The TD criterion represents total dissimilarity, a quality measure used to evaluate how well the data points within each cluster are represented by their respective m_r medoids. The lower the value of Eq. (5), the better the clustering. The steps of the *Hill Climbing* method for three-dimensional shapes are presented in Algorithm 4. Note that the update of m_r takes place in steps 4 and 5.

Algorithm 4 *Hill Climbing* algorithm for three-dimensional shapes.

Algorithm 4: *Hill Climbing* algorithm for three-dimensional shapes.

- 1 **Input:** A set \mathbf{X}^* of configuration matrices, the K number of clusters and maximum number of iterations.
 - 2 **Output:** Clusters formed ($G = G_r, r = 1, \dots, K$).
 - 3 **Evaluation:** Arbitrarily select K observations as the initial partition. Then assign each observation to the cluster with the closest medoid, according to the Riemannian distance value from Definition 4. Calculate the value of the criterion in Equation (5).
 - 4 **Move:** Move an object that was not selected as medoid by a medoid object, producing a change in the clustering criterion of Equation (5).
 - 5 **Update:** Select the change that leads to the best improvement in the clustering criterion value.
 - 6 **Repeat:** Repeat step 4 and 5 until the maximum number of iterations is met.
-

In his book, Everitt et al. (2011) described the *Hill Climbing* algorithm and the clustering criteria used. *Hill Climbing* represents a simple and effective method for optimization problems with small search spaces. However, it can be susceptible to getting stuck in local optima and only sometimes finding the global optimum for complex problems. Various enhancements and variations, such as Simulated Annealing and Genetic Algorithms, can be adopted to overcome these limitations and improve the search capabilities of the algorithm in future research. However, it is a motivation to improve the algorithm via the *Bagging* procedure, as proposed in the following sections.

3.4 Bagging algorithm

Combining clustering algorithms with ensemble techniques is known as “clustering ensembles”. The algorithms used in clustering ensembles have two stages and aim to improve the performance of clustering algorithms by focusing on finding a global optimal result. In the first stage, the results generated by individual clustering algorithms are stored. In the second stage, a consensus function is applied to combine the stored results and define a final clustering solution. In summary, ensemble techniques are used to solve problems by combining several models or algorithms, whether for classification or regression tasks, through adapted versions of the data and unifying their outputs (Flach 2012). Clustering ensembles leverage this approach to enhance the clustering process and achieve more robust and accurate results. Ensemble techniques are not limited to clustering and can be applied in various machine learning domains to improve model performance and generalization. Ensemble techniques can mitigate overfitting and enhance overall predictive power by harnessing the collective knowledge of multiple algorithms.

We used ensemble techniques to improve the quality of the clusters obtained by the presented algorithm, specifically the *Bagging* method. The reason for applying this method in cluster analysis is to reduce the variability of the results obtained from partitioning algorithms. For example, it can stabilize the results of the *CLARANS* method. The *Bagging* method, which stands for Bootstrap Aggregating (Breiman 1996), involves training multiple instances of the same algorithm on different subsets of the data and then combining their outputs to create a more robust and stable final result. By reducing the variance and mitigating the impact of random fluctuations, *Bagging* can improve clustering performance and enhance the reliability of the obtained clusters (Bühlmann 2012).

The *Bagging* method used in the present work, initially proposed by Dudoit and Fridlyand (2003) to reduce the possible variability of the clusters formed by the *PAM*, is called *BagClust1*. In short, this algorithm applies the *PAM* multiple times for each *bootstrap* sample obtained from the data set, and the final cluster is then formed based on the labels that obtained the most votes for each observation. Assis et al. (2021) used the *Bagging* method, inspired by the work of Breiman (1996), to improve statistical analysis data clustering in two-dimensional shapes partitioning different subsets of the original data and using the voting approach as a consensus function to find the best partition generated by clustering algorithms.

In this paper, we have used the *Bagging* method to improve the clustering of data-oriented methods for statistical analysis of three-dimensional shapes. In this article, the *K-means*, *CLARANS* or *Hill Climbing* methods are applied multiple times for each *bootstrap* sample obtained from the data set of three-dimensional shapes, and the final cluster is then formed based on the labels that receive the most votes for each observation. The steps of the *Bagging* method are presented in Algorithm 5.

Algorithm 5 Bagging algorithm adapted for the analysis of three-dimensional shapes.

- 1 **Input:** A set \mathbf{X}^* of configuration matrices, K number of clusters and number of *bootstrap* B samples.
- 2 **Output:** Clusters formed by the winning labels of each object ($G = G_r, r = 1, \dots, K$).
- 3 **Clustering:** Apply one of the Clustering Algorithms 2, 3, or 4, i.e. partitioning clustering procedure \mathcal{P} , to the original data set to get the cluster labels for each observation.
- 4 **Bootstrapping:** Form the b -th *bootstrap* samples of configuration matrices $\mathcal{L}^b = (\mathbf{X}_1^b, \dots, \mathbf{X}_n^b)$.
- 5 **Training:** Apply Algorithm 2, 3, or 4 of clustering \mathcal{P} to the *bootstrap* learning set \mathcal{L}^b , and obtain the cluster labels for each observation in \mathcal{L}^b .
- 6 **Aggregation:** Swap the cluster labels assigned to the *bootstrap* learning set so that there is maximum overlap with the original clustering of these observations. Being S_K the set of all permutations of the integers $1, \dots, K$. Find the permutation $\tau^b \in S_K$ that maximizes

$$\sum_{i=1}^n \mathbb{I} \left\{ \tau^b \left(\mathcal{P}(\mathbf{X}_i^b; \mathcal{L}^b) \right) = \mathcal{P}(\mathbf{X}_i^b; \mathcal{L}) \right\},$$

where $\mathbb{I}(\cdot)$ is the indicator function, equaling 1 if the condition between the braces is true and 0 otherwise.

- 7 **Repeat:** Repeat steps 4 – 6 a number B of times and assign the final cluster label to each observation i by majority vote.

Note that τ represents the permutations between the clustering labels of the set *bootstrap* and τ^b represents the permutation that maximizes the condition from step 6. Algorithm 5 uses b -th *bootstrap* samples as input data for the clustering algorithms. The process is then repeated B times to obtain different sets of cluster labels for each sample used as three-dimensional shape data. At the end of the repetitions, it is verified which cluster label was most attributed to each object, and the label that received the most votes is then chosen as the final one. Therefore, the final clustering is formed by the winning labels for each object. In case of a tie, the label is chosen at random. For this algorithm class, $B = 20$ was originally used, which can substantially improve clustering accuracy when applying *Bagging* for cluster analysis, as demonstrated in the experiment conducted by Dudoit and Fridlyand (2003). However, in our experiments, we chose to use $B = 100$.

Applying *Bagging* to cluster analysis aims to reduce variability in dataset partitioning results and improve the robustness of clustering results. It achieves this by averaging the effects of different *bootstrap* samples, which helps stabilize the clustering process and reduce the impact of random fluctuations Dudoit and Fridlyand (2003).

However, it is essential to acknowledge that the *Bagging* method may have a higher computational cost, especially when dealing with large data sets, when training multiple instances of the same clustering algorithm on different *bootstrap* samples. To address this issue in future research, one can use parallel computing or distributed processing techniques (Lazarevic and Obradovic 2002) to speed up the execution of the *Bagging* method or explore ensemble techniques that may offer similar advantages with potentially lower computational costs, such as Random Subspace Method (García-Pedrajas and Ortiz-Boyer 2008) or Random Patches Method (Louppe and Geurts 2012).

4 Numerical evaluation

Next, the methods *K-means*, *CLARANS*, and *Hill Climbing* for three-dimensional shape clustering were compared to their *Bagging* versions. We conducted experiments on simulated datasets. Additionally, two real datasets were analyzed. The Riemannian distance given in Definition 4 was used as an appropriate dissimilarity measure for the shape space. The analyzed algorithms were implemented in the R programming language (R Core Team 2024). The experiments were conducted on an Aspire A315-41 laptop, with a Ryzen 3 2200U processor, 2.50 GHz, 18GB of RAM, 64-bit system, and Windows 10 Home platform, using R version 4.4.0.

As for the values of the parameters used by the adapted algorithms, we decided to use the values proposed/used by the authors in their original applications for each application. We set the stop criterion for *K-means* to 0.0001, meaning that observations will no longer switch groups once the criterion in Eq. (4) reaches the value 0.0001. We also fixed the number of initializations and steps per initialization at 10. The number of steps per initialization (iteration steps) searches for the best value of the objective function, while number of initializations represents the number of random initializations in each of these iteration steps. For *CLARANS*, *numlocal* was considered equal to 2, and *maxneighbor* was set to 1.25% of $K(N - K)$, where N is the size of the dataset, and K is the number of groups. The algorithm's search process, Hill Climbing, ends when it reaches the loop size. We chose these parameter settings to ensure consistency and comparability with the original algorithms' applications. In our studies, the number of groups K is known priori for both the simulated and real data sets. We use K as preprocessing in the context of partitional clustering. However, when we do not know the predefined number K of groups, we can use hierarchical clustering to find K (Everitt et al. 2001).

To compare the efficiency of the algorithms proposed in this article, we used the Rand Index (RI) (Rand 1971) and Fowlkes-Mallows Index (FMI) (Fowlkes and Mallows 1983) validation measures. These indexes measure the similarity between a cluster provided by a clustering method and true clusters known a priori. They assume values in the $[0, 1]$ range, where 1 indicates perfect agreement between the clusters generated by the algorithm and the known true clusters, while values close to 0 correspond to an agreement found by chance. We opted not to explore other more complex procedures, such as the Adjusted Rand Index (Hubert and Arabie 1985) or the Silhouette Index (Rousseeuw 1987), as they produce measurements with negative output. This limitation could pose a significant disadvantage to the procedures we adopted for analyzing the data in our work.

Based on the RI and FMI results, we use the paired Wilcoxon test for simulation experiments to evaluate whether the methods improve with the *Bagging* approach. The null hypothesis was that there is no difference between the results of the indexes without and with *Bagging*, and the alternative hypothesis was that the results of the indexes without *Bagging* are inferior to the results with *Bagging*. In short, we want to evaluate whether there was an improvement in the clusterings generated by the methods combined with *Bagging*. A p-value less than or equal to the 5% significance level is statistically significant. We chose $B = 100$ because the test results were consistent across all algorithms for this value. The value of $B = 100$ is also commonly used in

the literature and in experiments that use resampling *bootstrap*, as is the case with the work of Assis et al. (2021) which also uses the *Bagging* approach in clustering context and form analysis.

We calculate the Rand Index and the Fowlkes-Mallows Index for the best result of each clustering algorithm, that is, for the best partition found by each clustering algorithm. For each algorithm, $B = 100$ *bootstrap* replicas were generated and combined to form the labels. These labels went through the voting process to generate the best partition. The objective is to demonstrate the efficiency of using the *Bagging* method in combination with clustering algorithms compared to their standalone versions without it. Applying the *Bagging* method can reduce the variability in the labeling processes' clustering results and make the final clustering more robust and stable (Dudoit and Fridlyand 2003).

In experiments with simulated data, the methods were evaluated through different artificial data configurations using Monte Carlo experiments; more specifically, 50 replications were performed. Each method was run for each replication with different random partitions until convergence, and the best result according to the objective function was selected. We chose to work with 50 Monte Carlo replicas to balance time and precision because, for our experiments, we considered this number of replicas sufficient to have adequate variability in the generated data.

We carried out a single execution of the methods to analyze the algorithms on real data sets and subsequently calculated the results of interest based on this execution. We also use a Relative Gain measure to evaluate the results of applying the methods to real data sets and the validation indexes. We use this measure to compare the effectiveness of clustering methods with and without *Bagging*, based on the values of the clusters validation measures. The gain relative to the use of *Bagging* is defined by:

$$Relative\ Gain = 100 * \frac{Validation\ Index_{Bagging} - Validation\ Index}{Validation\ Index},$$

where $Validation\ Index_{Bagging}$ is the value of the RI or FMI when the algorithm uses the *Bagging* method and $Validation\ Index$ is the value for the RI or FMI when the algorithm acts alone. This measure aims to verify whether there was a gain when using the proposed algorithms combined with the *Bagging* method on real data sets.

Our comparison of the performance of the *Bagging*-based clustering algorithms with their standalone counterparts will not only reveal any improvements in clustering accuracy and quality but also underscore the effectiveness of the *Bagging* technique in enhancing clustering performance. This analysis will further highlight the benefits of combining *Bagging* with various clustering algorithms, instilling confidence in our approach.

4.1 Simulation study results

Experiments were conducted in the landmark space based on the data simulated by Vinué et al. (2014) to illustrate the performance of the algorithms. To represent the average configurations, the experiments utilized two predefined geometric figures, a cube, and a parallelepiped. The number of landmarks for each object was set to $k = 8$.

Then, n_1 objects corresponding to one cluster and n_2 objects corresponding to another cluster were simulated. The clusters 1 and 2 were defined by a multivariate normal distribution with a three-dimensional mean vector of dimension $3k$, represented by the predefined cube for the cluster 1 and predefined parallelepiped for the cluster 2. And additionally, a covariance matrix Σ_i , $i = 1, 2$ of dimension $3k \times 3k$.

The orientation of the cubes and parallelepipeds was defined arbitrarily. A rotation was applied about the axis z according to a random angle generated by the function `rvm` from the R package `CircStats` (Agostinelli and Agostinelli 2018). This function generates pseudo-random numbers from a von Mises distribution. The Von Mises distribution probability function density has the form

$$f(\theta; \mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} \exp\{\kappa \cos(\theta - \mu)\},$$

where: $I_0(\kappa)$ is the modified Bessel function of order 0, μ is the mean direction and κ measures the concentration of the angles around the mean direction (Best and Fisher 1979). In Fig. 1, the predefined cube and parallelepiped shapes are shown for the landmark size $k = 8$. In Fig. 2, the parallelepiped is displayed after being rotated according to a random angle generated by the `rvm` function. Different sample sizes and dispersion scenarios were considered, including isotropy (similar dispersion in all directions) and anisotropy (different dispersion in different directions).

In the isotropy scenario, the landmarks have approximately the same variability. The covariance matrix Σ is a multiple of the identity matrix \mathbf{I} , i.e., $\Sigma_i = \sigma_i^2 \mathbf{I}_{3k \times 3k}$ with different values for σ_i , $i = 1, 2$. The values for σ_1 and σ_2 were chosen so that the data have low ($\sigma_i = 3$), average ($\sigma_i = 6$), and high ($\sigma_i = 9$) dispersion, respectively, in each case. In the anisotropy scenario, the landmarks do not have approximately the same variability. The covariance matrix in this case is represented by the result of the Kronecker product between two operations involving the identity matrix, as follows:

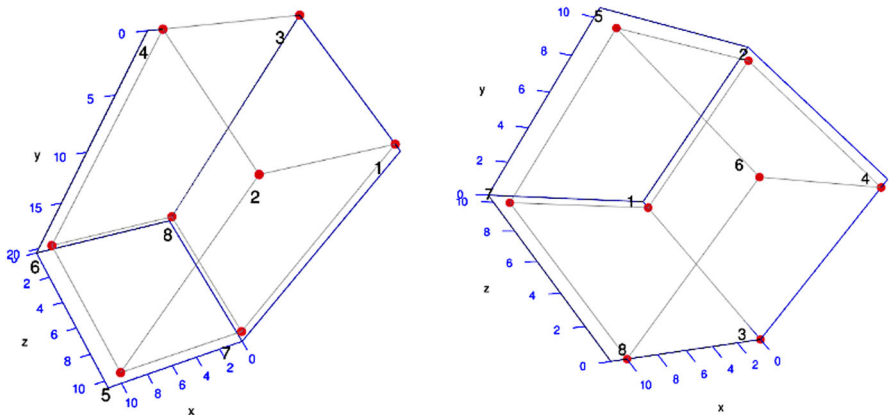
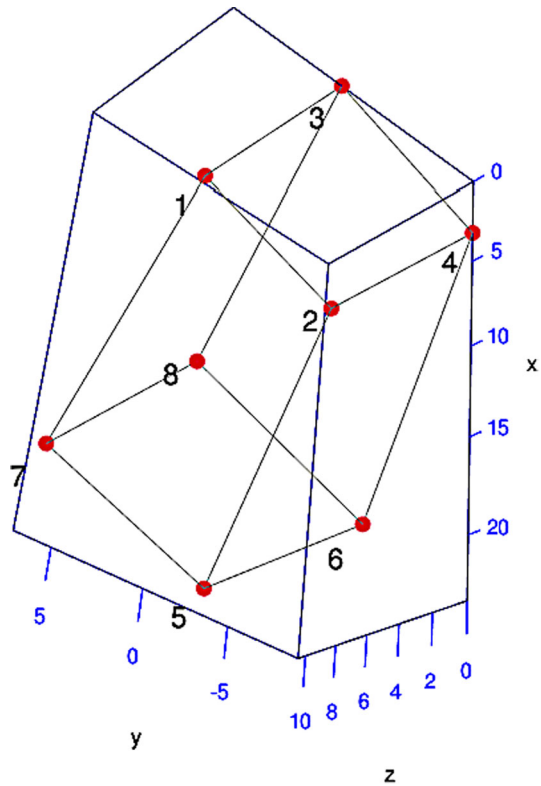


Fig. 1 Parallelepiped and cube formed by 8 landmarks

Fig. 2 Parallelepiped formed by 8 landmarks rotated by a random angle



$$\Sigma_i = \frac{\sigma_i^2}{\gamma^2} (\mathbf{1}_k \mathbf{1}_k^T + (\gamma - 1) \mathbf{I}_{k \times k}) \otimes (\mathbf{1}_m \mathbf{1}_m^T + (\gamma - 1) \mathbf{I}_{m \times m}), \quad i = 1, 2$$

where \mathbf{m} represents the dimension. The values for σ_1 and σ_2 are the same as in the isotropy scenario, and γ equals 4.

Each simulated dataset consists of N objects divided into two groups: one containing cubes and the other containing parallelepipeds. We considered sample sizes of $N = 100$. The values of n_1 and n_2 were $n_1 = 50$ and $n_2 = 50$, respectively. We choose these sample sizes to manage computational costs when applying *Bagging* on large samples of 3D shapes. The landmarks were generated following a multivariate normal distribution and transformed into configuration matrices to form the objects of each cluster. Tables 1, 2 and 3 show the RI and FMI values obtained for the simulated isotropy data, while Tables 4, 5 and 6 present the results of the simulated anisotropy data. In all Tables, the symbols \bar{x} and sd represent the mean and standard deviations, respectively, calculated for applying the algorithms to 50 Monte Carlo replicas, where each replica represents a set of distinct 3D shape data containing $N = 100$ objects.

4.2 Results in the isotropy scenario

In Tables 1, 2, and 3, the results of applying the methods to data sets simulated under isotropy for $N = 100$ are presented. Table 1 shows the results obtained for the *K-means* and *K-means Bagging* methods. The combined use of the *Bagging* method with the *K-means* method resulted in an improvement in the estimates of the results measured by the indexes, according to the significance of the paired Wilcoxon test for the significance level of 5%, only for the Fowlkes-Mallows in cases with high dispersion. Furthermore, the results show no significance of *Bagging* for low and medium dispersion cases.

Table 2 presents the results obtained for the *CLARANS* and *CLARANS Bagging* methods. It is observed that the application of the *Bagging* method in the *CLARANS* algorithm led to an improvement in the method clusterings according to the estimates of both validation indexes and in all dispersion scenarios, according to the significance of the paired Wilcoxon test at the 5% significance level. Table 3 presents the results obtained for the *Hill Climbing* and *Hill Climbing Bagging* methods. According to the Wilcoxon test, it is evident that there was an improvement in the quality of the clusters with the *Bagging* approach, according to the estimates of both indexes, in all dispersion scenarios considered.

Figure 3 presents the violin plots of the pairwise differences of the results between *Bagging* and the original algorithm for the simulated isotropy data. The Fig 3a, c, and d show the results for the Rand Index. As we can see in item (a), *Bagging* did not provide major improvements to the *K-means* method. However, we can see improvements in the results of items (b) *CLARANS* and (c) *Hill Climbing*, especially under medium or high dispersion. We can see a similar interpretation in the results for the Fowlkes-Mallows Index presented in the Fig. 3b, e and f. Furthermore, note that there were cases in which we obtained $sd=0$ for both validation measures, which seems to be a characteristic of these simulated data. We can observe similar situations in the experiments of the work Vinué et al. (2014), especially when σ is small.

As expected, in cases of isotropy, the *Bagging* approach showed significant improvements in clustering, except for some scenarios with low dispersion. In the context of isotropic landmarks, the algorithms that benefited most from the use of *Bagging* were *CLARANS* and *Hill Climbing*. Furthermore, as expected, as the dispersion between the landmarks grew, the IR and FMI values decreased, given that the landmarks show a large variability between them. This behavior was observed consistently across all methods and scenarios analyzed in the experiments.

4.3 Results in the anisotropy scenario

In Tables 4, 5 and 6, we present the results obtained for the methods *K-means*, *CLARANS* and *Hill Climbing*, respectively, along with their *Bagging* versions, applied to the simulated anisotropy data. Table 4 shows the results obtained for the *K-means* and *K-means Bagging* methods. Based on the significance of the paired Wilcoxon test at the 5% significance level, it is evident that the combination of *Bagging* did not improve the clusters generated.

Table 1 Results for the algorithms *K-means* and *K-means Bagging* applied to simulated data of cubes and parallelepipeds represented by $k = 8$ landmarks, under isotropy

Isotropic variation	Rand index				Fowlkes-mallows index				p value
	Without bagging		B=100		Without bagging		B=100		
	\bar{x}	sd	\bar{x}	sd	\bar{x}	sd	\bar{x}	sd	
$\sigma_1 = \sigma_2 = 3$	1	0	1	0	1	0	1	0	1.0000
$\sigma_1 = \sigma_2 = 6$	0.9175	0.0320	0.9148	0.0309	0.9169	0.0321	0.9139	0.0309	0.9290
$\sigma_1 = \sigma_2 = 9$	0.6656	0.0667	0.6720	0.0646	0.6657	0.0664	0.6744	0.0633	0.0254

Table 2 Results for the algorithms *CLARANS* and *CLARANS Bagging* applied to simulated data of cubes and parallelepipeds represented by $k = 8$ landmarks, under isotropy

Isotropic variation	Rand index			Fowlkes-Mallows Index						
	Without bagging			Without bagging			B=100			<i>p</i> value
	\bar{x}	sd	<i>p</i> value	\bar{x}	sd	<i>p</i> value	\bar{x}	sd	<i>p</i> value	
$\sigma_1 = \sigma_2 = 3$	0.9948	0.0118	1	0	0.0017	0.9948	0.0119	0.9996	0.0028	0.0030
$\sigma_1 = \sigma_2 = 6$	0.7027	0.0926	0.7937	0.0645	<0.0001	0.7093	0.0885	0.7979	0.0594	<0.0001
$\sigma_1 = \sigma_2 = 9$	0.5322	0.0361	0.5499	0.0452	0.0002	0.5396	0.0397	0.5666	0.0530	<0.0001

Table 3 Results for the algorithms *Hill Climbing* and *Hill Climbing Bagging* applied to simulated data of cubes and parallelepipeds represented by $k = 8$ landmarks, under isotropy

Isotropic variation	Rand index				Fowlkes-mallows index				
	Without bagging		B=100		Without bagging		B=100		p value
	\bar{x}	sd	\bar{x}	sd	\bar{x}	sd	\bar{x}	sd	
$\sigma_1 = \sigma_2 = 3$	0.9980	0.0061	1	0	0.9979	0.0061	1	0	0.0184
$\sigma_1 = \sigma_2 = 6$	0.7365	0.1044	0.8617	0.0559	0.7432	0.0973	0.8611	0.0545	<0.0001
$\sigma_1 = \sigma_2 = 9$	0.5316	0.0359	0.5741	0.0585	0.5452	0.0392	0.5960	0.00607	<0.0001

For the case of anisotropy, Table 5 presents the results obtained for the methods *CLARANS* and *CLARANS Bagging* and Table 6 presents the results obtained for the *Hill Climbing* and *Hill Climbing Bagging* methods. Both tables show that applying the *Bagging* method along with the algorithms improves the quality of the clusters generated in most dispersion scenarios, based on the significance of the Wilcoxon test at a 5% significance level. However, we observed that the results did not show significance for cases with low dispersion.

Figure 4 presents violin plots of the pairwise differences of the results between *Bagging* and the original algorithm for the simulated anisotropy data. The Fig. 4a, c and e show the Rand Index results. As we can see in item (a), *Bagging* did not provide improvements to the *K-means* method, as occurred in the isotropy scenario. However, we can observe improvements in the results of items (b) *CLARANS* and (c) *Hill Climbing*, especially under medium or high dispersion. We can see a similar interpretation in the results of the Fowlkes-Mallows Index presented in Fig. 4b, d, and f.

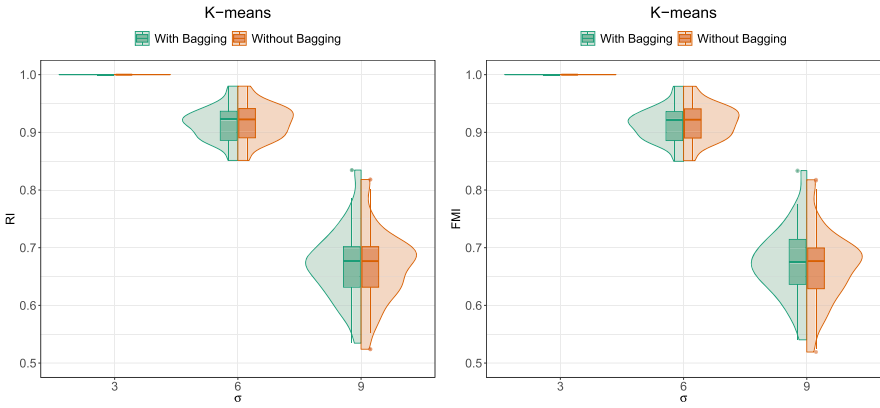
Furthermore, note that in this scenario, there are also cases where we obtain $sd=0$ for both validation measures. Note that, as in the case of isotropy, the methods did not benefit from using *Bagging* in low-dispersion scenarios. The *Bagging* method did not benefit *K-means*, while the other two algorithms benefited from the *Bagging* approach.

The supplemental material includes hypothesis tests concerning the results from simulations under various isotropy and anisotropy dispersion scenarios. These experiments consider different values of B .

4.4 Macaques skull

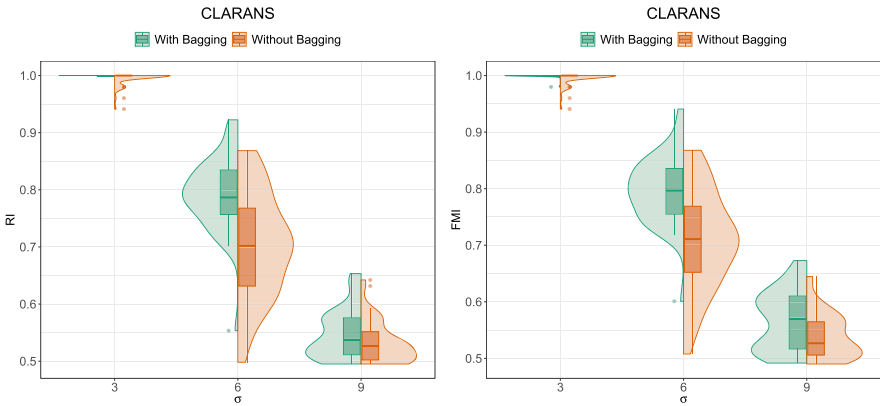
In a study to evaluate the existence of differences in size and shape between the skulls of male and female *Macaca fascicularis*, a sample of 18 individuals was obtained by Paul O'Higgins (Hall York Medical School). This data set is composed of $K = 2$ groups, one group being made up of 9 male macaques and the other group being made up of 9 females. A subset of $k = 7$ anatomical landmarks were selected from a total of $k = 26$ representing each skull (Dryden and Mardia 2016). The selected landmark names are 1-prosthion, 2-nasion, 3-bregma, 4-opisthion, 5-asterion, 6-interfrontomale, and 7-midpoint. An artist's impression of a 3D representation of a skull with projections of anatomical landmarks can be seen in Fig. 5. More details about the data are given by Dryden and Mardia (1993).

Table 7 presents the results for clustering algorithms applied to the Macaques Skull dataset. From these values, we can observe that using the *Bagging* method improves the clustering quality for the *Hill Climbing* algorithm, considering both validation indexes. The *CLARANS* algorithm shows an improvement in clustering according to the FMI. However, the *K-means* algorithm does not exhibit performance gains, even when the *Bagging* method is used for this dataset. This data set has only $N = 18$ objects; in this sense, it is easy for the clustering algorithms to find the best formation of groups according to the dissimilarities between the objects, even when the *Bagging* method is not used. The relative gains presented show that the joint operation of the algorithms with the *Bagging* method can provide more efficiency in the quality of the



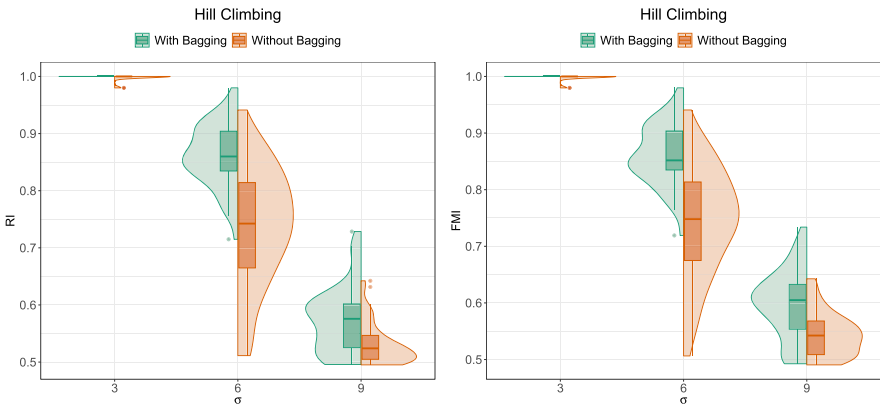
(a) Results of the Rand Index of the *K-means* algorithm in isotropy

(b) Results of the Fowlkes–Mallows Index of the *K-means* algorithm in isotropy



(c) Results of the Rand Index of the *CLARANS* algorithm in isotropy

(d) Results of the Fowlkes–Mallows Index of the *CLARANS* algorithm in isotropy



(e) Results of the Rand Index of the *Hill Climbing* algorithm in isotropy

(f) Results of the Fowlkes–Mallows Index of the *Hill Climbing* algorithm in isotropy

Fig. 3 Paired violin plots of the results comparing *bagging* and the original algorithm for the simulated isotropy data

Table 4 Results for the algorithms *K-means* and *K-means Bagging* applied to simulated data of cubes and parallelepipeds represented by $k = 8$ landmarks, under anisotropy

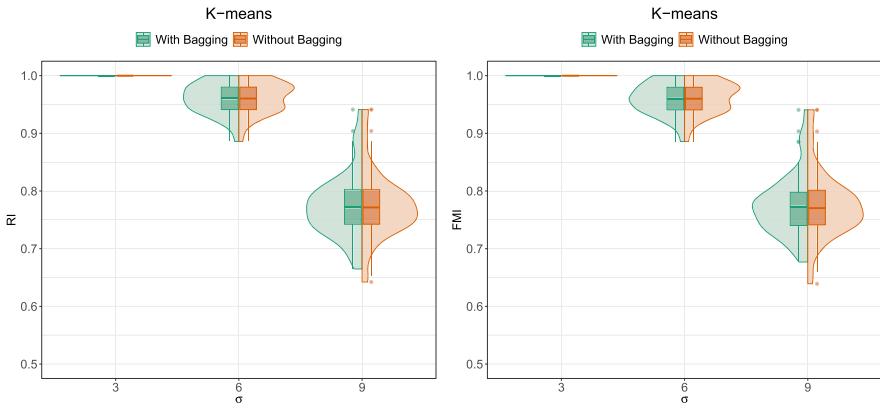
Anisotropic variation	Rand index				Fowlkes-mallows index				
	Without bagging		B=100		Without bagging		B=100		p value
	\bar{x}	sd	\bar{x}	sd	\bar{x}	sd	\bar{x}	sd	
$\sigma_1 = \sigma_2 = 3$	1	0	1	0	1	0	1	0	1.0000
$\sigma_1 = \sigma_2 = 6$	0.9623	0.0275	0.9608	0.0275	0.9620	0.0276	0.9600	0.0275	0.9882
$\sigma_1 = \sigma_2 = 9$	0.7761	0.0601	0.7745	0.0561	0.7761	0.0597	0.7755	0.0542	0.6347

Table 5 Results for the algorithms *CLARANS* and *CLARANS Bagging* applied to simulated data of cubes and parallelepipeds represented by $k = 8$ landmarks, under anisotropy

Anisotropic variation	Rand index				Fowlkes-mallows index				p value
	Without bagging		B=100		Without bagging		B=100		
	\bar{x}	sd	\bar{x}	sd	\bar{x}	sd	\bar{x}	sd	
$\sigma_1 = \sigma_2 = 3$	0.9992	0.0039	1	0	0.9992	0.0040	1	0	0.1728
$\sigma_1 = \sigma_2 = 6$	0.7909	0.1004	0.8911	0.0426	0.7937	0.0962	0.8933	0.0392	<0.0001
$\sigma_1 = \sigma_2 = 9$	0.5719	0.0651	0.6179	0.0765	0.5792	0.0665	0.6358	0.0757	<0.0001

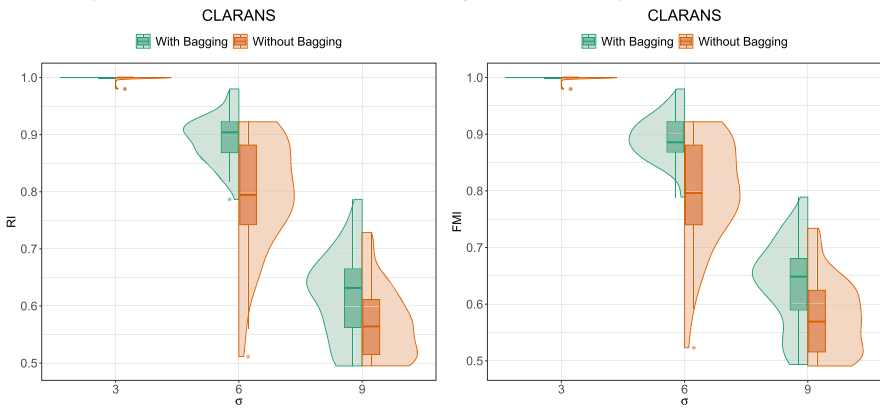
Table 6 Results for the algorithms *Hill Climbing* and *Hill Climbing Bagging* applied to simulated data of cubes and parallelepipeds represented by $k = 8$ landmarks, under anisotropy

Anisotropic variation	Rand index				Fowlkes-Mallows Index				
	Without bagging		B=100		Without bagging		B=100		p value
	\bar{x}	sd	\bar{x}	sd	\bar{x}	sd	\bar{x}	sd	
$\sigma_1 = \sigma_2 = 3$	1	0	1	0	1	0	1	0	1.0000
$\sigma_1 = \sigma_2 = 6$	0.8525	0.0884	0.9358	0.0342	0.8548	0.0833	0.9346	0.0345	<0.0001
$\sigma_1 = \sigma_2 = 9$	0.5705	0.0546	0.6596	0.0662	0.5842	0.0511	0.6768	0.0609	<0.0001



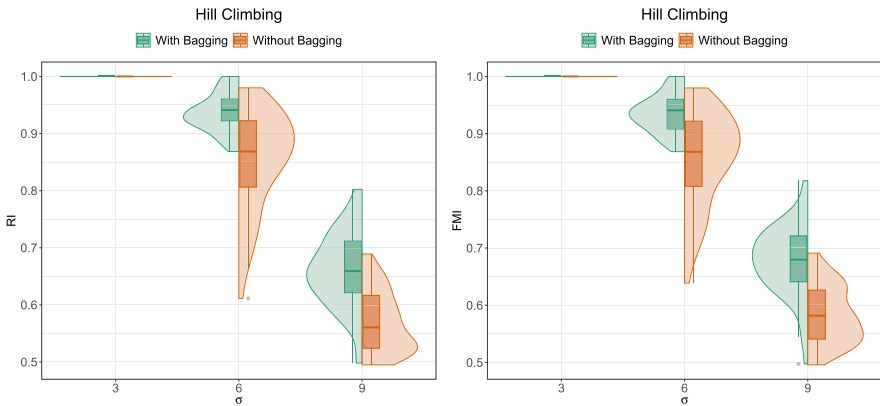
(a) Results of the Rand Index of the *K-means* algorithm in anisotropy

(b) Results of the Fowlkes–Mallows Index of the *K-means* algorithm in anisotropy



(c) Results of the Rand Index of the *CLARANS* algorithm in anisotropy

(d) Results of the Fowlkes–Mallows Index of the *CLARANS* algorithm in anisotropy



(e) Results of the Rand Index of the *Hill Climbing* algorithm in anisotropy

(f) Results of the Fowlkes–Mallows Index of the *Hill Climbing* algorithm in anisotropy

Fig. 4 Paired violin plots of the results comparing *bagging* and the original algorithm for the simulated anisotropy data

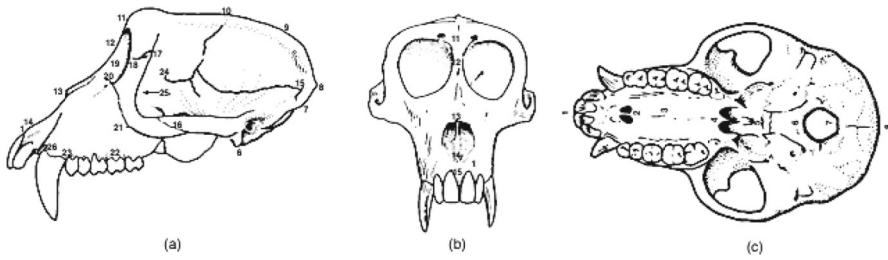


Fig. 5 3D representation of a macaque skull of the species *Macaca fascicularis*: **a** side view; **b** front view; and **c** bottom view

Table 7 Results for the Macaque Skull dataset

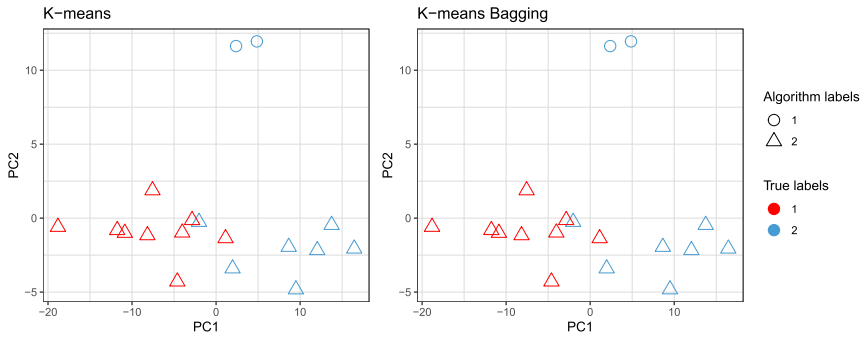
Methods	Rand index		RI relative gain	Fowlkes-mallows index		FMI relative gain
	Without bagging	B=100		Without bagging	B=100	
K-means	0.4967	0.4967	0.00%	0.6214	0.6214	0.00%
CLARANS	0.5294	0.5294	0.00%	0.5527	0.6124	10.78%
Hill Climbing	0.4771	0.4967	4.11%	0.5025	0.6124	21.86%

clusters, except for the *K-means* method. Figure 6 displays the labels clustering the Macaques Skull dataset into two clusters using the proposed methods. The axes of the plot are the Principal Components (PCs). Each point on the plot represents the shape of a single analysis object. According to Vinué et al. (2014), the closer two objects are, the more similar they are in shape.

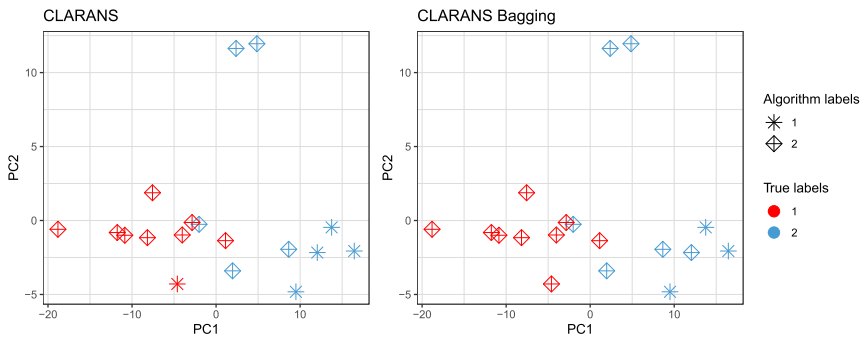
4.5 Brains

In an investigation to verify the difference between the shapes of adult human brains, anatomical landmarks distributed across the surface of the cerebral cortex of healthy adults were collected. The data set can be divided into $K = 2$ distinct groups: 43 right-handed adults and 15 left-handed adults. $k = 12$ anatomical landmarks were identified in each brain hemisphere, accounting for $k = 24$ landmarks per individual. Figure 7 presents three views of an individual's left hemisphere indicating the approximate locations of anatomical landmarks, totaling 12. More details about the dataset can be found in Free et al. (2001).

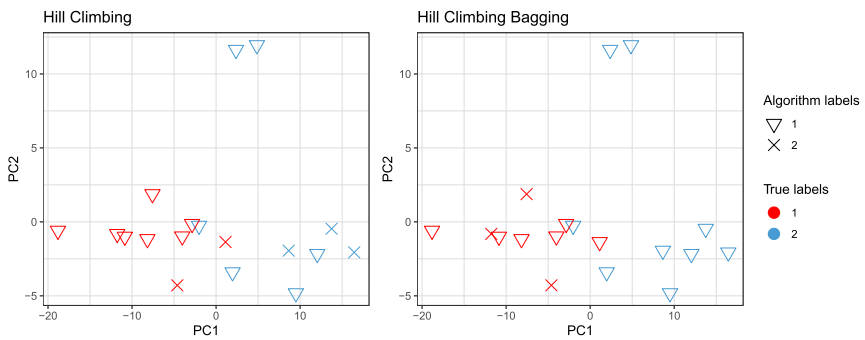
Continuing with the applications of our approaches to different datasets already present in the literature, we evaluate the effectiveness of our methods by applying them directly to the Brains dataset and calculating the results of the Rand and Fowlkes-Mallows indexes and the Relative Gain. Table 8 presents the results for the Brains dataset. From these values, we can observe that using the *Bagging* method improves the clustering quality for all algorithms. In all cases, the relative gain was greater than



(a) The resulting labels for the Macaques skull dataset according to the *K-means* x *K-means Bagging* methods



(b) The resulting labels for the Macaques skull dataset according to the *CLARANS* x *CLARANS Bagging* methods



(c) The resulting labels for the Macaques skull dataset according to the *Hill Climbing* x *Hill Climbing Bagging* methods

Fig. 6 Clusterings scatter plot for Macaques Skull dataset

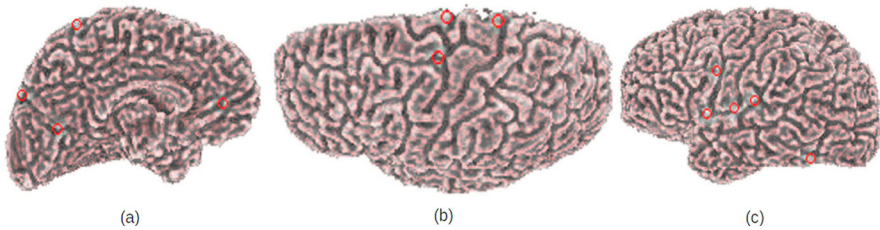


Fig. 7 Visualization of the locations of $k = 12$ anatomical landmarks of an individual's left hemisphere

Table 8 Results for the Brains dataset

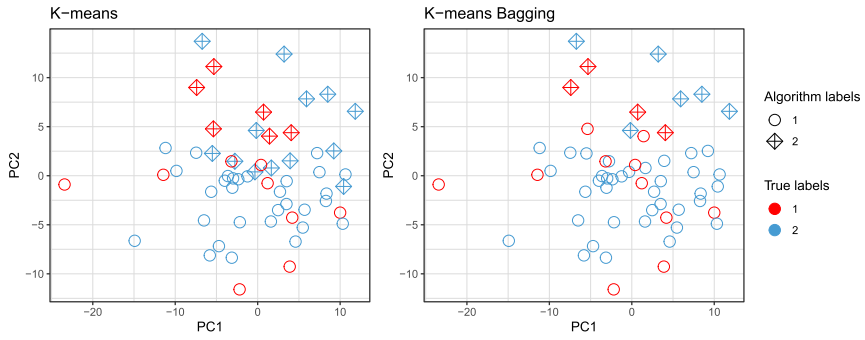
Methods	Rand index		RI relative gain	Fowlkes-mallows index		FMI relative gain
	Without bagging	B=100		Without bagging	B=100	
K-means	0.5208	0.5783	11.03%	0.5882	0.6824	16.00%
CLARANS	0.4967	0.5783	16.44%	0.5538	0.7282	31.13%
Hill Climbing	0.4937	0.5783	17.15%	0.5452	0.7139	30.94%

10%. In this sense, there is a benefit to using the *Bagging* approach. Figure 8 displays the labels clustering the Brains dataset into two clusters using the proposed methods. The plot axes are the principal components (PCs), and each point on the plot represents the shape of a single object. The use of PCs to generate this type of plot is widespread in cluster analysis.

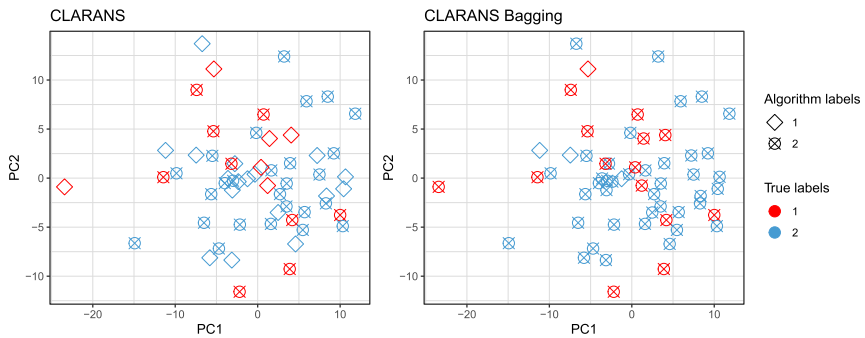
5 Conclusions

The discussion about adapting new methods in Statistical Shape Analysis is essential for the scientific community. This paper introduced the *K-means*, *CLARANS*, and *Hill Climbing* methods for clustering three-dimensional shapes and applied a *Bagging* procedure to improve their performances. Experimental results with different datasets demonstrated the effectiveness of the *Bagging* approach in this context, in some cases. Unlike typical cluster analysis studies that only show the best result when running the algorithm, we perform repeated applications of the algorithms with Monte Carlo simulations. We calculated the mean and standard deviation for two cluster validation measures: Rand Index (RI) and Fowlkes-Mallows Index (FMI). We also used the paired Wilcoxon test to validate the effectiveness of the methods using the *Bagging* approach in simulation experiments.

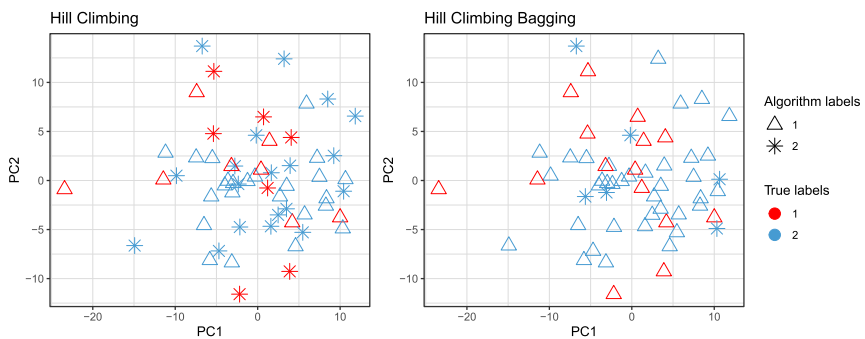
The findings indicate that the *Bagging* approach consistently improves clustering quality, with the RI estimates exhibiting values and variability close to the FMI estimates. We evaluate the algorithms with and without *Bagging*, considering $B = 100$ replicas of *bootstrap*.



(a) The resulting labels for the Brains dataset according to the *K-means* x *K-means Bagging* methods



(b) The resulting labels for the Brains dataset according to the *CLARANS* x *CLARANS Bagging* methods



(c) The resulting labels for the Brains dataset according to the *Hill Climbing* x *Hill Climbing Bagging* methods

Fig. 8 Clusterings scatter plot for Brains dataset

By applying *Bagging* to improve the clustering of algorithms on simulated datasets, we found different scenarios for each algorithm. Based on the results obtained by the paired Wilcoxon test, we observed that the use of *Bagging* did not significantly impact the metric estimates for small values of σ_i in our simulated datasets.

For *K-means*, the use of *Bagging* resulted in group quality improvements only for some cases, indicating that the benefits of *Bagging* for *K-means* can be limited and may not provide consistent improvements across all data sets.

On the other hand, for *CLARANS*, using *Bagging* was generally advantageous as it led to improved validation measure estimate values in most cases. This improvement is probably due to the small value used for the *maxneighbor* parameter, which, although recommended by the authors, may limit the algorithm's ability to explore the solution space effectively. However, high values for *maxneighbor* would significantly increase computational costs.

For *Hill Climbing*, *Bagging* also improved the metric estimates for the cases under medium and high dispersion, suggesting that *Hill Climbing* can also benefit from the diversity introduced by *Bagging*.

The results for the real data set depended on each data set's specific characteristics. Based on RI and FMI measurements, the real datasets' clustering results suggest that both algorithms have similar clustering effectiveness in some cases. The interpretation of the results based on the Relative Gain measure suggests that the proposed methods improved the quality of the clusters generated, especially for the Brains dataset.

Our work includes a comparative study of three different clustering methods in the context of three-dimensional shapes, as only some studies of this type are present in the literature. In summary, the impact of *Bagging* varies between algorithms and datasets. While *CLARANS* and *Hill Climbing* tend to benefit from *Bagging*, *K-means* may show limited improvements. The *BagClust1* method was initially proposed to improve the clustering of the *PAM* method. For this reason, we think that the *CLARANS* method, precisely because it is a variation of *PAM*, was the one that benefited most from this *Bagging* approach as well as *Hill Climbing*, which in the approach of the present work focused on the clusters classifications formed by the same clustering criteria used in the *PAM* algorithm.

In conclusion, the *Bagging* method applied to the proposed clustering algorithms showed significant improvements in the precision and quality of the generated clusters, particularly in cases of medium to high dispersion between landmarks. As a future direction for this research, we suggest exploring other ensemble clustering methods, such as *Boosting*, and comparing their performance with the results obtained using *Bagging* when applied with clustering methods. Ultimately, we believe our paper can serve as a valuable guide for further use of *Bagging* methods in shape clustering. Applying ensemble techniques to shape analysis can pave the way for more accurate and robust clustering results, benefiting various fields of study that rely on shape data.

Acknowledgements This work was supported in part by the following Brazilian agencies: Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), No. 303192/2022-4 and 402519/2023-0 (RO), and Fundação de Amparo a Ciência e Tecnologia do Estado de Pernambuco (FACEPE) in Brazil.

Author Contributions IN, RO and GA: Conceptualization, Methodology, Formal Analysis, Investigation, Software, Visualization, Writing, drafting & editing, Resources and Funding acquisition. All authors have contributed to interpretation of the results and manuscript revision.

Declarations

Conflict of interest The authors declare no Conflict of interest. The funders had no role in the design of the study; in the collection, analyzes, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

- Adams DC, Otárola-Castillo E (2013) geomorph: an R package for the collection and analysis of geometric morphometric shape data. *Methods Ecol Evol* 4(4):393–399
- Agostinelli C, Agostinelli MC (2018) Package ‘circstats’. See [https://cran-r-project.org/web/packages/CircStats/CircStats.pdf](https://cran.r-project.org/web/packages/CircStats/CircStats.pdf)
- Amaral GJA, Dore LH, Lessa RP, Stosic B (2010) K-means algorithm in statistical shape analysis. *Commun Stat Simul Comput* 39(5):1016–1026
- Assis ECD, Souza RMCRD, Amaral GJAD (2021) Using bagging to enhance clustering procedures for planar shapes. *Int J Bus Intell Data Min* 18(1):30–48
- Baxter MJ (2015) Exploratory multivariate analysis in archaeology. ISD LLC
- Best D, Fisher NI (1979) Efficient simulation of the von mises distribution. *J Roy Stat Soc Ser C (Appl Stat)* 28(2):152–157
- Bhattacharya A, Bhattacharya R (2012) Nonparametric inference on manifolds: with applications to shape spaces, vol 2. Cambridge University Press, Cambridge
- Bookstein FL et al (1986) Size and shape spaces for landmark data in two dimensions. *Stat Sci* 1(2):181–222
- Breiman L (1996) Bagging predictors. *Mach Learn* 24(2):123–140
- Brignell CJ, Dryden IL, Gattone SA, Park B, Leask S, Browne WJ, Flynn S (2010) Surface shape analysis with an application to brain surface asymmetry in schizophrenia. *Biostatistics* 11(4):609–630
- Bühlmann P (2012) Bagging, boosting and ensemble methods. Concepts and methods, *Handbook of computational statistics*, pp 985–1022
- Dryden IL, Mardia KV (1993) Multivariate shape analysis. *Sankhyā The Indian J. Stat. Ser. A (1961–2002)* 95(3):460–480
- Dryden IL, Mardia KV (2016) *Statistical shape analysis: with applications in R*, 2nd edn. Wiley, New Jersey
- Dudoit S, Fridlyand J (2003) Bagging to improve the accuracy of a clustering procedure. *Bioinformatics* 19(9):1090–1099
- Everitt B, Dunn G et al (2001) *Applied multivariate data analysis*, vol 2. Wiley, New Jersey
- Everitt BS, Landau S, Leese M, Stahl D (2011) *Cluster Analysis*, 5th edn. Wiley, New Jersey
- Flach P (2012) *Machine Learning: the art and science of algorithms that make sense of data*. Cambridge University Press, Cambridge
- Fowlkes EB, Mallows CL (1983) A method for comparing two hierarchical clusterings. *J Am Stat Assoc* 78(383):553–569
- Fréchet M (1948) Les éléments aléatoires de nature quelconque dans un espace distancié. *Annales de l’institut Henri Poincaré* 4(10):215–310
- Free SL, O’Higgins P, Maudgil DD, Dryden IL, Lemieux L, Fish DR, Shorvon SD (2001) Landmark-based morphometrics of the normal adult brain using mri. *Neuroimage* 13(5):801–813
- Friedman HP, Rubin J (1967) On some invariant criteria for grouping data. *J Am Stat Assoc* 62(320):1159–1178
- García-Escudero LÁ, Gordaliza A (1999) Robustness properties of k-means and trimmed k-means. *J Am Stat Assoc* 94(447):956–969
- García-Pedrajas N, Ortiz-Boyer D (2008) Boosting random subspace method. *Neural Netw* 21(9):1344–1362
- Goodall CR, Mardia KV (1999) Projective shape analysis. *J Comput Graph Stat* 8(2):143–168
- Guo R, Lee H, Patrangenaru V (2023) Test for homogeneity of random objects on manifolds with applications to biological shape analysis. *Sankhya A* pp 1–27

- Hartigan JA, Wong MA (1979) Algorithm as 136: a k-means clustering algorithm. *J Roy Stat Soc Ser C (Appl Stat)* 28(1):100–108
- Hastie T, Tibshirani R, Friedman J (2009) *The elements of statistical learning*, 2nd edn. Springer, Berlin
- Hubert L, Arabie P (1985) Comparing partitions. *J Classif* 2(1):193–218
- Jain AK (2010) Data clustering: 50 years beyond k-means. *Pattern Recogn Lett* 31(8):651–666
- Kendall DG (1977) The diffusion of shape. *Adv Appl Probab* 9(3):428–430
- Kendall DG (1984) Shape manifolds, procrustean metrics, and complex projective spaces. *Bull Lond Math Soc* 16(2):81–121
- Kent JT (1994) The complex bingham distribution and shape analysis. *J Roy Stat Soc Ser B (Methodol)* 56(2):285–299
- King AP, Eckersley R (2019) *Statistics for biomedical engineers and scientists: How to visualize and analyze data*. Academic Press, London
- Lazarevic A, Obradovic Z (2002) Boosting algorithms for parallel and distributed learning. *Distrib Parallel Databases* 11:203–229
- Leisch F (1999) Bagged clustering (working paper no. 51). WU Vienna University of Economics and Business: SFB Adaptive Information Systems and Modelling in Economics and Management Science
- Lloyd S (1982) Least squares quantization in PCM. *IEEE Trans Inf Theory* 28(2):129–137
- Loupe G, Geurts P (2012) Ensembles on random patches. In: *machine learning and knowledge discovery in databases: European conference, ECML PKDD 2012, Bristol, UK, September 24–28, 2012. Proceedings, Part I* 23, Springer, pp 346–361
- MacQueen J (1967) Some methods for classification and analysis of multivariate observations. *Proc Fifth Berkeley Symp Math Stat Probab* 1(14):281–297
- Ng RT, Han J (2002) CLARANS: a method for clustering objects for spatial data mining. *IEEE Trans Knowl Data Eng* 14(5):1003–1016
- R Core Team (2024) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, <http://www.R-project.org/>
- Rand WM (1971) Objective criteria for the evaluation of clustering methods. *J Am Stat Assoc* 66(336):846–850
- Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 20:53–65
- Rousseeuw PJ, Kaufman L (1990) *Finding Groups in Data*. Wiley, New Jersey
- Srivastava A, Klassen EP (2016) *Functional and shape data analysis*, vol 1. Springer, Berlin
- Srivastava A, Joshi SH, Mio W, Liu X (2005) Statistical shape analysis: clustering, learning, and testing. *IEEE Trans Pattern Anal Mach Intell* 27(4):590–602
- Vinué G, Simó A, Alemany S (2014) The K-means algorithm for 3D shapes with an application to apparel design. *Adv Data Anal Classif* 10(1):103–132
- Wilcoxon F (1992) Individual comparisons by ranking methods. *Breakthroughs in statistics: methodology and distribution*. Springer, Berlin, pp 196–202

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.