JIRSS (2021)
Vol. 20, No. 02, pp 29-42
DOI:10.52547/jirss.20.2.29

A Comparative Study of Some Clustering Algorithms on Shape Data

Sahar Asili [1], Adel Mohammadpour [1], Omid Naghshineh Arjmand [1], and Mousa Golalizadeh [2].

[1] Department of Statistics, Amirkabir University of Technology (Tehran Polytechnic).
[2] Department of Statistics, Tarbiat Modares University.

Received: 06/11/2020, Revision received: 22/10/2021, Published online: 09/11/2021

Abstract. Recently, some statistical studies have been done using the shape data. One of these studies is clustering shape data, which is the main topic of this paper. We are going to study some clustering algorithms on shape data and then introduce the best algorithm based on accuracy, speed, and scalability criteria. In addition, we propose a method for representing the shape data that facilitates and speeds up the shape clustering algorithms. Although the mentioned method is not very accurate, it is fast; therefore, it is useful for datasets with a high number of landmarks or observations, which take a long time to be clustered by means of other algorithms. It should be noted that this method is not new, but in this article we apply it in shape data analysis.

Keywords. Shape Data Clustering, Hierarchical, Partitioning, Fuzzy Clustering.

MSC: 62H35, 91C20.

Sahar Asili (s.asili@aut.ac.ir)
Corresponding Author: Adel Mohammadpour (adel@aut.ac.ir)
Omid Naghshineh Arjmand (naghshineh@aut.ac.ir)
Mousa Golalizadeh (golalizadeh@modares.ac.ir)

[DOI: 10.52547/jirss.20.2.29]

1 Introduction

Statistical shape analysis is a branch of science that deals with the behavior of the shapes of various objects and the relationships between the shape and size of the shapes with the features of the related objects. On the other hand, clustering is an unsupervised method for grouping objects in such a way that the objects in one cluster have the most similarities compared to the objects of other clusters. This method helps the researchers find the patterns within the objects. Clustering shape data is a useful method for analyzing shape datasets and discovering the relations between shape and size and the features of the objects. For example, to discover the relationship between the age and the size and shape of the bones in a kind of animal, shape clustering is used. Until now, many studies have been done on clustering shape data. Some of these studies are listed in continuation. The distance matrix and its usage in hierarchical clustering were studied by Lele and Richtsmeier (2001). Usage of hierarchical clustering in imaged objects and finding the probability models for clusters of shapes were studied by Srivastava et al. (2005). The application of K-means in shape data was made by Amaral et al. (2010). The application of hierarchical clustering was considered by Ishihara et al. (2011), and the agglomerative hierarchical clustering with different shape distances was studied in Nabil and Golalizadeh (2016).

Our goal in this article is to examine four kinds of clustering methods, including agglomerative hierarchical, divisive hierarchical, partitioning, and fuzzy clustering on shape data to achieve the best clustering method in terms of accuracy, speed, and scalability. In addition, a vectorized representation of shape data is proposed. Although this representation leads to a simple and fast clustering algorithms, it loses some of the accuracies.

In the next section, we briefly explain some basic concepts of statistical shape analysis that are used in shape clustering. Then, in Section 3, a simulation study is done. Section 4 contains the results of our study on five benchmark data and then in the last section, we summarize the results and introduce a simpler and faster clustering method for shape datasets.

We mention that all calculations in this article are performed with a computer with the following features: CPU: P4/3.4 GHz, H.D.D: 320 G, RAM: 1 GB.

[DOI: 10.52547/jirss.20.2.29]

2 The Basic Concepts of Clustering Shape Data

In this section, we define some basic concepts about clustering shape data, which are useful in the subsequent parts. It is necessary to mention that most of the notations and concepts of this section are based on Dryden and Mardia (2016).

A “shape" is all the geometrical information that remains when location, scale, and rotational effects are filtered out from an object, Kendall (1977). To represent a shape in mathematical form, landmarks are used. The landmarks are some points that are placed on the border or the image of an object to represent the shape of the object approximately. Suppose there is an object and its shape is desired to be investigated. To represent the shape of an object, a matrix with k rows and m columns will be used, which named configuration matrix and is represented by X. Here, k is the number of landmarks, which is located on the border of an object or its image and m is the dimension of the mentioned landmarks. Therefore, when we have n objects, the shape dataset will be a $k \times m \times n$ array. It is worth mentioning that in all configuration matrices of this paper, there is no similar correspondent labels.

Another useful concept is a pre-shape of X, which is represented by Z and is the result of some mathematical changes in X such that it is invariant under the translation and scaling of the original configuration. Now, we are going to find the best clustering algorithm for shape data in terms of speed and accuracy. We will compare clustering algorithms on simulated and benchmark shape datasets.

As mentioned in Izenman (2008) and Rencher (2002), the agglomerative hierarchical clustering is suitable for the situations that a large number of clusters is desired, and the divisive is useful for clustering with a few number of clusters. On the other hand, a partitioning clustering algorithm is faster than the hierarchical, and the fuzzy clustering has more accuracy. Because of the mentioned properties of each of these algorithms, we are interested in examining their impact on shape datasets.

Clustering shape data has many applications in the real world. Some examples are explained in Dryden and Mardia (2016) and Lele and Richtsmeier (2001). For example, determining the effect of schizophrenia on the shape of the human brain or determining the effect of age to shape and size of a certain bone in a kind of animal.

Clustering methods group objects in such a way that objects within each group are more similar in comparison with those of other groups. An essential tool for measuring similarities or dissimilarities is the distance measure. For clustering shape data, one could use the shape distances. There are some specific shape distances, like partial

[DOI: 10.52547/jirss.20.2.29]

Procrustes, full Procrustes, Riemannian, and size-and-shape distances. As pointed out in Nabil and Golalizadeh (2016), size-and-shape distance is the best measure of distance for clustering shape data when the size is of interest; otherwise, full Procrustes is the best distance to use. Therefore, we will use size-and-shape and full Procrustes distances for clustering datasets. Based on Dryden and Mardia (2016), these two distances are defined as (2.1) and (2.2):

$$d_F(X_1, X_2) = \inf_{\Gamma \in SO(m), \beta \in R} \| Z_2 - \beta Z_1 \Gamma \|, \tag{2.1}$$

$$d_S^2(X_1, X_2) = \inf_{\Gamma \in SO(m)} \| HX_1 - HX_2\Gamma \|^2 . \tag{2.2}$$

Here, X_1 and X_2 are two configuration matrices and $Z_i = HX/ \| HX_i \|$ for $i = 1, 2$ is the related pre-shape of i-th object. Besides, $\| HX_i \|$ stands for the norm of HX_i, where H is Helmert sub-matrix with $k - 1$ rows and k columns. The first row of Helmert matrix is removed in H. Furthermore, $SO(m)$ which is a special orthogonal group, is the set of all $m \times m$ rotation matrices and Γ is a rotation matrix.

One other concept is the definition of a mean shape. As mentioned in Dryden and Mardia (2016) and Nabil and Golalizadeh (2016), a mean shape can be inferred as the minimizer of any shape distance. For N configuration matrices $X_1, \ldots, X_N$, it is given by

$$\hat{\mu} = \arg \inf_{a \in R^{km}} \sum_{i=1}^{N} d^2(a, X_i),$$

where d denotes the distance used in the shape space.

There are several criteria for identifying the accuracy of clustering algorithms, which can be classified in two general categories of internal and external criteria. We will use three internal criteria: Silhouette, Dunn, and tau, as well as the four external criteria: Rand, Jaccard, Russel-Rao, and Fowlkes-Mallows to compare clustering algorithms. The Silhouette is a criterion that shows the similarity of an object with its cluster members compared with the members of other clusters. Its value could be in $(-1, 1)$. The higher the amount of Silhouette, the greater the adaptation to the desired cluster. Dunn's interpretation is like Silhouette. The tau criterion measures the relationship between two vectors. The higher value means the higher similarity. The Rand index is a measure of the similarity between two data clustering methods: real classes and the result of clustering algorithms. Its value is between 0 and 1. Jaccard's output is a value between 0 and 100 %, which higher values indicate higher similarity. Fowlkes-Mallows criterion can be used to determine the similarity between two clustering methods. The

[DOI: 10.52547/jirss.20.2.29]

higher value of this index, the greater similarity between the clusters. Russel-Rao criterion is like Fowlkes-Mallows. According to Gan et al. (2007), we consider two clustering structure $C = \{C_1, \ldots, C_m\}$ and $P = \{P_1, \ldots, P_s\}$ for defining the above criteria. Besides, we define a, b, c, and d as follows:

- a is the number of pairs of data points that are in the same cluster of C and in the same cluster of P.
- b is the number of pairs of data points that are in the same cluster of C but in different clusters of P.
- c is the number of pairs of data points that are in different clusters of C but in the same cluster of P.
- d is the number of pairs of data points that are in different clusters of C and in different clusters of P.

Now we have four indexes including Rand index (R), Jaccard (J), Russel-Rao (RR), and Fowlkes-Mallows (FM) as follows.

$$R = \frac{a+b}{a+b+c+d}, J = \frac{a}{a+b+c}, RR = \frac{a}{a+b+c+d}, FM = \sqrt{\frac{a}{a+b} \times \frac{a}{a+c}}.$$

3 Simulation Study

We are going to generate a shape dataset involving four hypothetical types of flower images in two dimensions. Then we will perform four clustering algorithms on our simulated data to achieve the best algorithm in terms of accuracy, speed, and scalability. In the end, we will examine a new method to represent our simulated shape data and compare the results of clustering by this new method with previous algorithms.

In order to generate shape data, we use the linear model (3.1). As mentioned in Dryden and Mardia (2016), equation (3.1) is the configuration matrix definition.

$$X_i = \beta_i \mu \Gamma_i + 1_k \gamma_i^T, i = 1, \ldots, n, \tag{3.1}$$

where X_i is the i-th element of our desired shape dataset array, β_i is a scale parameter with positive values, μ is the mean shape, Γ_i is an $m \times m$ rotation matrix, 1_k is a vector involving k numbers of ones and γ_i is a location vector with $m \times 1$ dimensions.

Suppose we are going to study the shape and size of four kinds of flowers, including three-leaf, four-leaf, five-leaf and violet flower, and so we have four mean shapes, as

[DOI: 10.52547/jirss.20.2.29]

in Figure 1. We put 31 landmarks on each mean shape and simulate 10, 15, 20 and 25 configuration matrices for each kind of flower, respectively, see Figure 2. Therefore, we are going to cluster a $31 \times 2 \times 70$ dataset to four clusters. In the simulation study, we use a fixed μ for each of these four mean shapes. Then with some changes in scale, rotation, and location, we generate new configurations. In the following algorithm, we explain that μ has a fixed value in step 3.

- Step 1. Create a function named Gamma which generate n stochastic orientation matrix as a $2 \times 2 \times n$ array which elements are the 2×2 orientation matrices as $\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$. The parameter θ can be randomly chosen from a uniform distribution on $(-\pi, \pi)$. Here we choose $(0, -\pi/6)$ as we wish. There is no different in final results if we choose θ in other distances from $-\pi$ to π.

- Step 2. Define a random vector of uniform distribution as a scale vector. We called it Beta. As we explain earlier, β belongs to R^+. Here we choose it on $(1, 2)$.

- Step 3. Input mean shape as a matrix that has n rows and m column. It's name is Mu.

- Step 4. Define a random array of location vectors of standard normal distribution, We called it gamma.

- Step 5. Create a linear function which gives Beta, Mu, Gamma, and gamma and gets the simulated dataset according to equation (3.1).

For clustering this dataset, we consider four clustering algorithms, including `hclust`, `diana`, `pam` and `fanny` in R software. These algorithms are based on the agglomerative, divisive, partitioning and fuzzy clustering method, respectively. As it is mentioned in the help of R software, `hclust` is a function for hierarchical cluster analysis on a set of dissimilarities and the methods for analyzing it. It is necessary to know that, we use the ward. D2 method in all over of this paper. `diana` computes a divisive hierarchical clustering of the dataset returning an object of class `diana`. `pam` partitions (clusters) data into k clusters "around medoids", which is a more robust version of K-means. `fanny` computes a fuzzy clustering of the data into k clusters. It is necessary to mention that the shape-and-size distance is used to clustering this dataset because both shape and size is important in this study.

[DOI: 10.52547/jirss.20.2.29]

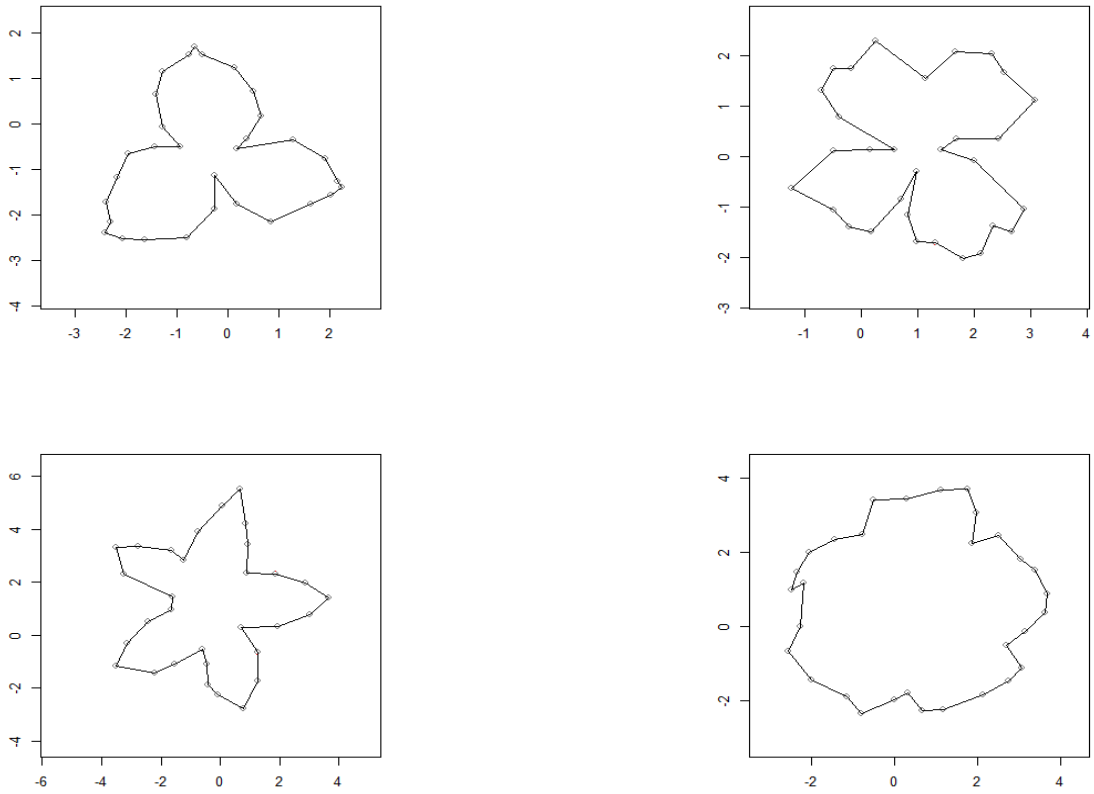

Figure 1: Simulated mean shapes of four kinds of flower, including three-leaf, four-leaf, five-leaf and violet.

To compare the accuracy of these clustering algorithms, we will use some internal and external indexes. To compare the speed of the algorithms, the average running time in 100 repetitions is calculated and the one with the lowest average will be chosen as a better algorithm. It should be noted that the time used here is the user time and is in seconds. The scalability of different methods is investigated by a set of simulated data with 100, 150, 200, and 250 configuration matrices (n = 700) of each kind of the mentioned flowers, respectively.

We summarize the results of 100 repetitions of algorithms in Table 1. In each column of this table, we have bolded cases with the best results. Therefore, it is clear that `pam` algorithm has higher accuracy and speed than the other algorithms and then

[DOI: 10.52547/jirss.20.2.29]

diana takes the second place. In addition, the results of scalability for the previous algorithms except fanny are gathered in Table 2. The fanny algorithm is omitted from this comparison because by Table 1 it is not useful even in the small dataset. As it is shown in Table 2, for large number of observations the diana algorithm is most accurate in comparison to the hclust and pam. In this situation, the hclust algorithm is faster than the others.

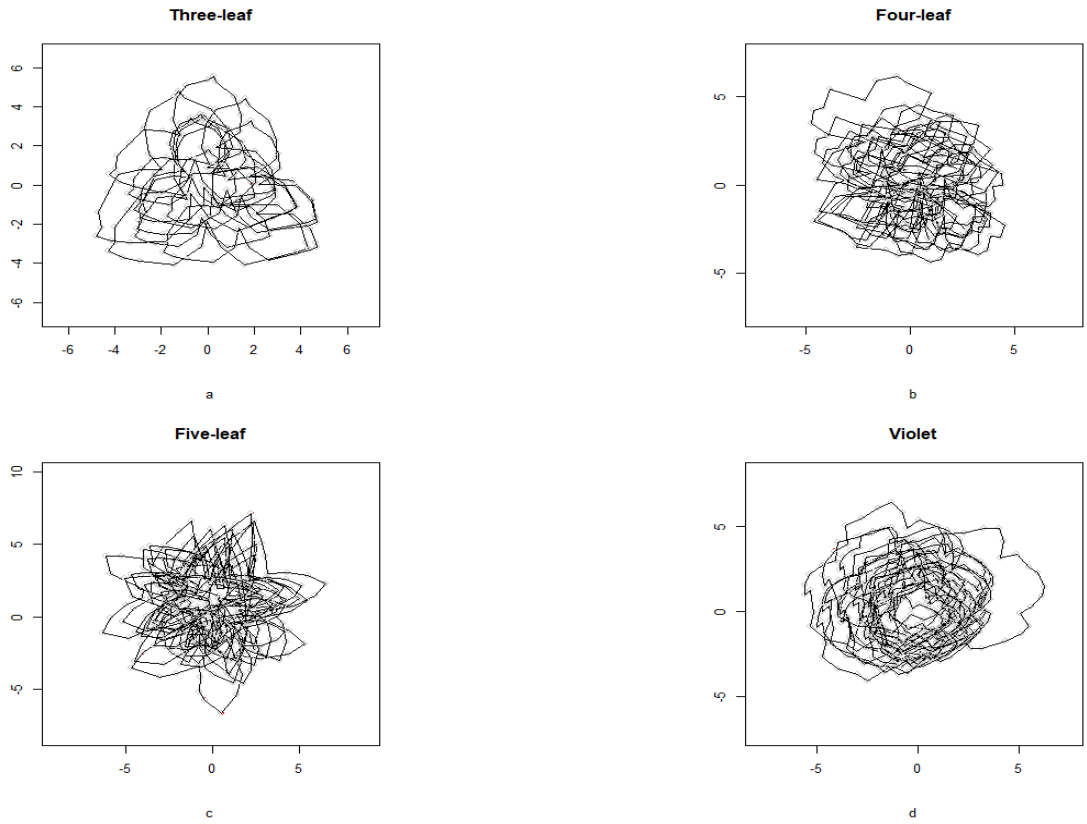

Figure 2: The four kinds of flower landmarks, with 31 landmarks per Flower image:(a) 10 three-leaf; (b) 15 four-leaf; (c) 20 five-leaf and (d) 25 violet flower.

[DOI: 10.52547/jirss.20.2.29]

Table 1: The results of 100 repetitions of the algorithms on a simulated dataset with n=70. The best values are bolded. The time average is reported in seconds.

	Desired criteria							
Algorithm	Silhouette	Dunn	Tau	Rand	Jaccard	Russel-Rao	Fowlkes-Mallows	Time
`hclust`	0.513658	0.189075	0.783638	0.853834	0.657008	0.215954	0.731606	3.602400
`diana`	0.512121	**0.247597**	0.803604	0.932447	0.800520	0.237863	0.882373	**3.572100**
`pam`	**0.523196**	0.201804	**0.904391**	**0.941723**	**0.813739**	**0.242605**	**0.884712**	**3.573400**
`fanny`	0.505655	0.095020	0.832288	0.903317	0.626673	0.203130	0.787413	3.593600

Table 2: The results of 100 repetitions of the algorithms on a simulated dataset with n=700. The best values are bolded. The time average is reported in seconds.

	Desired criteria							
Algorithm	Silhouette	Dunn	Tau	Rand	Jaccard	Russel-Rao	Fowlkes-Mallows	Time
`hclust`	0.495745	0.077016	0.780923	0.854842	0.655617	0.224789	0.768154	**351.7919**
`diana`	**0.529793**	**0.295104**	**0.998913**	**0.997577**	**0.994356**	**0.272549**	**0.996532**	352.1557
`pam`	0.522502	0.125237	0.944293	0.964526	0.887448	0.261003	0.946582	352.0493

3.1 Matrix Based Shape Clustering (MBSC)

Now we draw your attention to a small trick for clustering shape data. Previously, this idea was used in Mohammadpour et al. (2004) and Gonzalez and Woods (2002). Our purpose is to consider each $k \times m$ configuration matrix of the shape dataset as a $1 \times km$ vector to achieve an $n \times km$ matrix instead of the $k \times m \times n$ array. With this approach, the results of clustering algorithms will be impressive. Although the accuracy of the algorithms is somewhat diminished, the speed and simplicity are so dramatic. In addition, when one uses this method, it will be so easy for him/her to apply some clustering algorithms that don't accept a distance matrix as input, like K-means. It is necessary to mention that we will still have the impact of all the data in these clustering processes. We named this common vectorized method "MBSC", the abbreviation of the Matrix Based Shape Clustering, for shape data. The results of performing this approach for clustering the simulated data for $n = 70$ are gathered in Table 3. It is possible to somewhat rely on this method because it does not lose much accuracy in comparison with the other algorithm in Table 1. The results of the scalability of "MBSC" method on simulated data are gathered in Table 4. From this table, we conclude that "MBSC" method is useful for a large number of observations.

[DOI: 10.52547/jirss.20.2.29]

Table 3: The results of 100 repetitions of the algorithms on simulated dataset by MBSC approach for $n = 70$.

	Desired criteria	
Algorithm	Rand index	Time
`kmeans`	0.690667	**0.011740**
`diana`	0.674990	0.012450
`pam`	**0.709892**	0.012730

Table 4: The results of 100 repetitions of the algorithms on simulated dataset by MBSC approach for $n = 700$.

	Desired criteria	
Algorithm	Rand index	Time
`kmeans`	0.684412	**0.9557**
`diana`	0.674467	2.1604
`pam`	**0.707982**	1.1397

4 Application

Now we are going to investigate the accuracy of the results of the simulation study with some benchmark data. The data were studied by Dryden and Mardia (2016) and are available in R package `shapes`. At first, in Subsection 4.1, we introduce the benchmark datasets that will be used in this section and then, in Subsection 4.2, we will check the results of the simulation study with the given benchmark datasets.

4.1 Benchmark Datasets

We use 5 datasets named mice, macaques, schizophrenia, sand, and rats. In the following, we will introduce each of them briefly, for more information, see Dryden and Mardia (2016).

- Mice dataset: It is a $6 \times 2 \times 46$ array, including the configuration matrix of the image of the second thoracic vertebra T2 in mice. Here, 23 large and 23 small bones are studied. The aim is to recognize the effect of mice's size in the shape and size of the mentioned bone.

- Macaques dataset: This dataset is a $7 \times 3 \times 18$ array, including the configuration matrix of the shape of the skulls of 9 male and 9 female macaques. The aim is to recognize the effect of gender of macaques in the shape and size of the skulls.

[DOI: 10.52547/jirss.20.2.29]

- Schizophrenia dataset: In this dataset, the aim is to cluster a $13 \times 2 \times 28$ array to two clusters, including 14 volunteers and 14 schizophrenia patients with respect to the image of their brains for detecting the effect of schizophrenia on brain shape.
- Sand dataset: 25 river sands and 24 sea sands are targeted to describe the differences in shape variability of the sands by means of a $50 \times 2 \times 49$ array.
- Rats dataset: It is a $8 \times 2 \times 144$ array, including the configuration matrix of the image of the skulls of rats at ages 7,14, 21, 30, 40, 60, 90 and 150 days. There are 18 rats with complete sets of eight landmarks in two dimensions at each age. The aim is to find the effect of age on the size and shape of skulls.
  It should be noted that the shape data may not necessarily be labeled in clustering analysis and the clustering algorithms are efficient for unlabeled shape data, too.

4.2 Evaluation of Clustering Algorithms

In this subsection, we put the results of the four clustering algorithms outlined in Section 3 on each of the introduced benchmark datasets. It should be noted that we have used the appropriate shape distance for each dataset. It means that for the mice, rats, and macaques datasets the shape-and-size distance is used and for the schizophrenia and sand datasets the full Procrustes distance is applied. The results of four clustering algorithms on these datasets are summarized in Table 5 and 6. It is necessary to mention that in the sand and schizophrenia datasets we consider just the shape and in the other ones both shape and size are important so the fuzziness parameter is 2 for all datasets except the sand and schizophrenia, which is 1.5. That is because when the fuzzy parameter increased the fuzziness in clustering will increase too. For more information see the help of R software about `fanny` function. In Table 5, we put the average user time of each clustering algorithm for each dataset and in Table 6, the comparison of the accuracy of the four algorithms for the above-mentioned datasets is summarized.

Table 5: The average time of running clustering algorithms on five shape datasets.

	Dataset's name				
Algorithm's name	Mice	Rats	Macaques	Schizophrenia	Sand
`hclust`	0.4683	5.1621	0.2643	0.1493	**2.3129**
`diana`	**0.4655**	**5.1042**	**0.2623**	0.1484	2.3397
`pam`	0.4775	5.2500	0.2634	**0.1477**	2.3152
`fanny`	0.4703	5.3484	0.2634	0.1502	2.3286

[DOI: 10.52547/jirss.20.2.29]

It is clear from Table 5 that divisive hierarchical clustering is faster than the other algorithms and in the second rank, the partitioning algorithm is suitable in terms of speed.

Table 6: The results of accuracy indexes of clustering algorithms on benchmark datasets. The best values are bolded.

	Silhouette	Dunn	Tau	Rand	Jaccard	Russel-Rao	Fowlkes-Mallows
				Mice			
hclust	0.414389	0.337954	**0.957427**	**0.956522**	**0.914934**	**0.467633**	**0.955578**
diana	**0.421518**	**0.348350**	0.877058	0.875362	0.775652	0.430918	0.873687
pam	0.419584	0.323346	0.916515	0.914976	0.840580	0.448309	0.913393
fanny	0.419584	0.323346	0.916515	0.914976	0.840580	0.448309	0.913393
				Rats			
hclust	0.333330	0.253835	**0.076438**	0.778749	0.048852	0.011364	0.093183
diana	**0.342476**	**0.260198**	0.066309	0.776127	**0.051440**	**0.012141**	**0.097933**
pam	0.325581	0.180406	0.063231	**0.781080**	0.045320	0.010392	0.086713
fanny	0.311732	0.180406	0.061467	0.780206	0.049958	0.011558	0.095183
				Macaques			
hclust	0.328196	0.425589	0.777778	**0.790850**	0.636364	0.366013	0.777778
diana	**0.345807**	**0.465953**	**0.797724**	**0.790850**	**0.644444**	**0.379085**	**0.784070**
pam	0.345714	0.434053	0.620174	0.633987	0.481482	0.339869	0.653276
fanny	0.328196	0.425589	0.777778	**0.790850**	0.636363	0.366013	0.777778
				Sand			
hclust	0.193411	**0.222455**	-0.285307	0.525510	0.368063	0.276361	0.538659
diana	**0.220038**	0.154416	**-0.428333**	**0.583333**	**0.403167**	**0.281463**	**0.574653**
pam	0.214734	0.178345	-0.271058	0.525510	0.351916	0.257653	0.520646
fanny	**0.220038**	0.154416	**-0.428333**	**0.583333**	**0.403167**	**0.281463**	**0.574653**
				Schizophrenia			
hclust	0.109056	0.421800	0	0.481481	0.304965	0.227513	0.467419
diana	0.097007	**0.444431**	**0.382360**	**0.547619**	**0.389286**	**0.288360**	**0.561572**
pam	0.125513	0.430909	0.144338	0.492064	0.314286	0.232804	0.478289
fanny	**0.130492**	0.428562	0.144338	0.492064	0.314286	0.232804	0.478289

Table 7: Rand index of MBSC for benchmark datasets. The best values are bolded.

	Desired criterion and algorithm		
Dataset's name	Rand index of kmeans	Rand index of pam	Rand index of diana
Mice	0.671981	**0.837681**	0.736232
Rats	0.776363	**0.783217**	0.761267
Macaques	**0.790850**	0.575163	**0.790850**
Sand	0.750000	0.750000	**0.780612**
Schizophrenia	0.484127	**0.492064**	0.484127

[DOI: 10.52547/jirss.20.2.29]

Table 8: The average time of 1000 times running MBSC for benchmark datasets. The best values are bolded.

	Algorithm		
Dataset	`kmeans`	`pam`	`diana`
Mice	**0.00044**	0.00079	0.00075
Rats	**0.00074**	0.01009	0.0067
Macaques	**0.00033**	0.00035	**0.00033**
Sand	**0.00076**	0.0012	0.00113
Schizophrenia	**0.00039**	0.00054	0.00041

At the first glance in Table 6, it is concluded that divisive hierarchical clustering is more accurate almost everywhere. However, a shape dataset with a large number of landmarks has higher speed under the agglomerative hierarchical clustering.

In Tables 7 and 8, the results of MBSC for the benchmark data are presented. As it is clear from the results, the MBSC has more accuracy under partitioning and divisive hierarchical clustering in comparison with the K-means. In other hands, it is very fast under the K-means and very time-consuming under the partitioning one. It is clear from Tables 6 and 7 that although the MBSC method will lose some of the accuracies, it will save a lot of time.

5 Conclusion

In this article, we examined the agglomerative hierarchical, divisive hierarchical, partitioning and fuzzy clustering algorithms on a collection of shape data to achieve the best clustering algorithm in terms of accuracy, speed, and scalability. According to the obtained results, the fastest algorithm is the divisive hierarchical clustering in ordinary datasets. When the number of landmarks increases, agglomerative hierarchical clustering is better in terms of speed. On the other hand, the divisive hierarchical and partitioning clustering methods are, respectively, more accurate compared with other methods. In addition, we used the simple and fast vectorized method for shape clustering, which is suitable for time-consuming clustering datasets. Although it is not as accurate as of the regular approach, it is really fast and simple.

The codes are available from Github of corresponding author.

[DOI: 10.52547/jirss.20.2.29]

References

Amaral, G. J. A., Dore, L. H., Lessa, R. P., Stosic, B. (2010), K-means algorithm in statistical shape analysis. *Communications in Statistics-Simulation and Computation*, **39**(5), 1016-1026.

Dryden, I. L. and Mardia, K. V. (2016), *Statistical Shape Analysis: with Applications in R. 2nd ed.*. Wiley, Chichester.

Gan, G, Ma, C, and Wu, J. (2007), *Data Clustering: Theory, Algorithms, and Applications*. ASA-SIAM Series on Statistics and Applied Probability.

Gonzalez R. C. and Woods R. E. (2002), *Digital Image Processing*. Prentice Hall.

Ishihara, S., Ishihara, K., and Nagamachi, M. (2011), Statistical shape of Head-lights. *IEEE International Conference on Biometrics and Kansei Engineering*, 27-32.

Izenman, A. J. (2008), *Modern Multivariate Statistical Techniques*. Springer, New York.

Kendall, D. G. (1977), The diffusion of shape. *Advances in Applied Probability*, **9**, 428-430.

Lele, S. R. and Richtsmeier, J. T. (2001), *An Invariant Approach to Statistical Analysis of Shapes*. Chapman and Hall/CRC, Boca Raton.

Mohammadpour, A., Féron, O., and Mohammad-Djafari A. (2004), Bayesian segmentation of hyperspectral images. *Bayesian Inference and Maximum Entropy Methods, Maxent Workshop*, Jul. pp. 541-548.

Nabil, M. and Golalizadeh, M. (2016), On clustering shape data. *Journal of Statistical Computation and Simulation*, **86**, 2995-3008.

Rencher, A. C. (2002), *Methods of Multivariate Analysis*. Wiley.

Srivastava, A., Joshi, S. H., Mio, W., and Liu, X. (2005), Statistical shape analysis: Clustering, learning, and testing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(4): 590-602.

[DOI: 10.52547/jirss.20.2.29]