ISSN 2007-9737

Comparative Analysis of K-Means Variants Implemented in R

Nelva Nely Almanza-Ortega[1], Joaquín Pérez-Ortega[2], José Crispín Zavala-Díaz[3],
José Solís-Romero[1]

[1] Tecnológico Nacional de México/IT de Tlalnepantla,
División de Estudios de Posgrado e Investigación,
Mexico

[2] Tecnológico Nacional de México/CENIDET,
Departamento de Ciencias Computacionales,
Mexico

[3] Universidad Autónoma del Estado de Morelos,
Facultad de Contaduría, Administración e Informática,
Mexico

nnaortega@outlook.com, jpo_cenidet@yahoo.com.mx, crispin_zavala@uaem.mx,
jose.sr1@tlalnepantla.tecnm.mx

Abstract. One of the ways of acquiring new knowledge or underlying patterns in data is by means of clustering algorithms or techniques for creating groups of objects or individuals with similar characteristics in each group and at the same time different from the other groups. There is a consensus in the scientific community that the most widely used clustering algorithm is K-means, mainly because its results are easy to interpret and there are different implementations. In this paper we present an exploratory analysis of the behavior of the main variants of the K-means algorithm (Hartigan-Wong, Lloyd, Forgy and MacQueen) when solving some of the difficult sets of instances from the Fundamental Clustering Problems Suite (FCPS) benchmark. These variants are implemented in the R language and allow finding the minimum and maximum intra-cluster distance of the final clustering. The different scenarios are shown with the results obtained**.**

Keywords. K-means, clustering, cluster analysis.

1 Introduction

Nowadays, huge amounts of data are produced, both by public and private institutions and in different fields of knowledge. Often, this data has underlying patterns that can be of great use for decision making in companies and institutions. One of the ways of acquiring new knowledge or underlying patterns in the data is by means of clustering algorithms or techniques for creating groups of objects or individuals with similar characteristics in each group and at the same time different from the other groups [1,2].

There are different clustering algorithms. A generic classification of these algorithms is partitional, hierarchical, fuzzy, density-based, grid-based, model-based, spatial, to name a few. There is a consensus in the scientific community that the most widely used clustering algorithm is K-means, mainly because its results are easy to interpret and there are different implementations [3].

Due to the wide use of the K-means algorithm, it has been extensively studied and numerous improvements have been proposed. It is worth mentioning that the problem K-means solves is NP-hard (i.e., it has a high computational complexity) so it remains an open research challenge to improve its efficiency when solving large instances [4].

There are data sets where K-means, by its very nature, does not work as desired. As we have experienced, for K-means to work properly, the centroids (i.e., the means of each cluster) have to be sufficiently far apart.

doi: 10.13053/CyS-26-1-4158

Therefore, in this research we conducted a set of experiments aiming at performing an exploratory analysis and observing the behavior of different variants of the K-means algorithm when solving data instances with a high level of complexity.

The R language allows us to run the four main variants of the K-means algorithm, which are included in the cluster package. The variants that the R language has implemented are Hartigan-Wong, Lloyd, Forgy, and MacQueen. The main differences between these variants are in the initialization and classification phase, in particular in the way the initial centroids are selected and the way each individual is assigned to the new cluster [5-6]. The datasets for the experimentation were obtained from R's CRAN package, Fundamental Clustering Problem Suite (FCPS) [7].

The document is organized as follows, Section 2 presents some of the most relevant works related to the article, from the origins of the algorithm to the current trend. Section 3 describes in general terms each of the datasets that will be used and the research in which those have been part of the experimentation to provide new knowledge in the area of clustering. The experimental process is shown in Section 4 and the results obtained in Section 5. Section 6 presents the cluster analysis and the discussion and Section 7 the conclusions.

2 Literature Review

When reviewing the specialized scientific literature, we found few articles that describe or address the analysis of variants of the K-means algorithm. In this work, we rely on experimentation to show the behavior of the variants in relation to the intra-cluster, minimum and maximum distance at which each variant converges, and subsequently, we analyze the final clustering.

Some of the relevant works found in the state-of-the-art study are presented below.

In 2006, in a summary of the algorithm variants and their results is presented in [8], half a century after its appearance. The interest in variants and their origins continued and in 2008, a review is presented in [9] of how the first variants originated, in their continuous and discrete version. Along the same lines, in 2010 the work described in [10] studied the main characteristics of clustering in general and described the key pieces for the design of new algorithms, marking a trend in research with respect to the K-means algorithm.

In 2013, the work described in [11] implemented in the Mathematica software the variants of the K-means algorithm proposed by Forgy/Lloyd, MacQueen, and Hartigan & Wong. In this work, they experimented with the combination of metrics to maximize distances or reduce differences based on the characteristics of the test data set. In 2015, hierarchical clustering is presented in [12] and comparisons are performed with specific measures of distance and linkage.

They mainly compare simple link, full link and average link using the SPSS statistical software. In 2017, in [13] the development of a hybrid algorithm is described to improve the variant proposed by Lloyd. In this work, they incorporate the advantages of clustering with the DBSCAN algorithm, which is a density-based algorithm to obtain the initial centroids, generating more suitable centroids from the input data set.

In 2019, the paper [14] provides information on the origins of the K-means algorithm. The main relevant improvements found using a systematic review of the literature and the trends and challenges of the algorithm. In 2020, 12 datasets are presented in [15], which constitute a challenge to be solved by the different clustering algorithms. Finally, in 2021, a benchmark is described in [16], which consists of the main clustering algorithms and dataset libraries of the main clustering programming languages.

As mentioned in the previous paragraphs, there are numerous variants of the K-Means algorithm, however, there is little work on comparative analysis of implementations using experimental methods. In this sense, the work carried out by [11] in 2013 stands out, where the Hartigan-Wong, Lloyd-Forgy and MacQueen variants, implemented in Mathematica, are analyzed and two instances are solved, a test instance of four dimensions and nine cases and the well-known iris flower dataset instance [17].

In contrast, in this research we selected the four variants provided by the R language, which are: Hartigan-Wong, Lloyd, Forgy and MacQueen. It should be noted that in this research, five synthetic instances of the FCPS repository [7] and the iris instance were solved. The average intra-cluster distance was used as the comparison metric.

2.1 Basic K-Means Algorithm

K-means is an iterative method that consists of partitioning a set of *n* objects into $k \geq 2$ clusters, such that the objects in one cluster are similar to each other and different from those in other clusters.

Formally, the problem solved by K-means is formalized as follows.

Let N = {x_1, ..., x_n} be the set of *n* objects to be partitioned by a similarity criterion, where $x_i \in \Re^d$ for *i* = 1, ..., *n* and $d \geq 1$ is the number of dimensions.

Also, let $k \geq 2$ be an integer and *K* = {1, ..., *k*}. For a *k*-partition *P* = {*G*(1), ..., *G*(*k*)} of *N*, let μ_j be the centroid of the group *G*(*j*), for $j \in K$.

The object x_i belongs to the group *G*(*j*) and $d(x_i, \mu_j)$ denotes the Euclidean distance between x_i and μ_j for *i* = 1,..., *n* and *j* =1,..., *k*.

In the following sections, a brief description of each of the K-means variants implemented in R is presented.

2.1.1 Lloyd

For Lloyd's algorithm (i.e., the basic K-means algorithm), let *K* be a set of *k* centroids, and for each centroid *μ* in *K*, let *G*(*μ*) denote its neighborhood (i.e., the set of data points for which *μ* is the nearest neighbor).

Each stage of Lloyd's algorithm moves each centroid *μ* to the centroid of *G*(*μ*) and then updates *G*(*μ*) by recalculating the distance from each object to its nearest centroid. These steps are repeated until convergence [18].

2.1.2 Forgy

This is essentially the basic K-means algorithm, except for the initialization of the centroids. This variant randomly selects *k* objects and uses these as the initial centroids [19].

2.1.3 MacQueen

This algorithm is fundamentally the same as the basic K-means. It adjusts all cluster centroids to the mean of their respective centroid *μ* each time an object x_i changes cluster membership *G*(*μ*) [20].

2.1.4 Hartigan-Wong

This algorithm assigns each object x_i to one of *K* groups or clusters to minimize the sum of squares within the cluster as shown by Eq. (1) [21]:

$$Sum(k) = \sum_{i=0}^{n} \sum_{j=o}^{d} \big(x(i,j) - x(k,j)\big)^2, \qquad (1)$$

where $x(k,\ j)$ is the average of the objects belonging to the cluster. The main difference with respect to the basic K-means consists in the objective function: in this algorithm the objective function is Eq. (1), while in K-means it is the Euclidean distance [22].

3 Description and Preparation of Datasets

The datasets used for the experimentation were obtained from the repository [7,15]. This repository can be found in the R language packages and includes a collection of more than 300 artificial datasets that were created expressly for the evaluation of clustering algorithms, heuristics and strategic improvements of these algorithms. This dataset is called the Fundamental Clustering Problems Suite (FCPS) [16].

Five datasets were taken from this repository in order to perform an exploratory clustering analysis with variants of the K-means algorithm, in particular, the variants that are implemented in the R language. In addition, the iris instance with four dimensions is included [17].

Table 1 presents the six selected datasets, the first column shows the instance Id, the second column the instance name and the last two columns the defined value of *k* and the number of objects in the instance.

The distribution of the objects, i.e., the clustering challenge for each of the instances described in Table 1, is presented below.

Table 1. Datasets for experiments

Id	Name	*k*	*N*
1	*diamond9*	9	3,000
2	*longsquare*	6	900
3	*spherical_4_3*	4	400
4	*spherical_5_2*	5	250
5	*triangle1*	4	1,000
6	*iris*	3	150

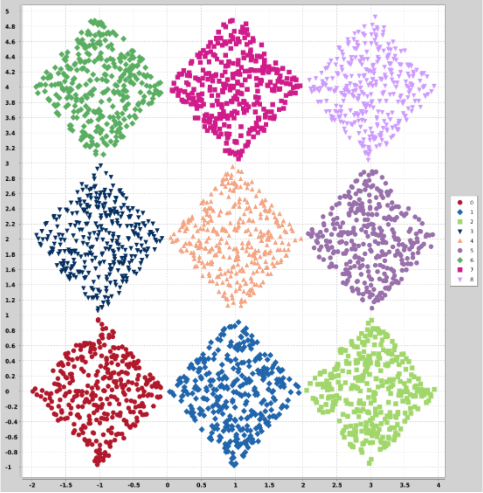


Fig. 1. Dataset *diamond9*

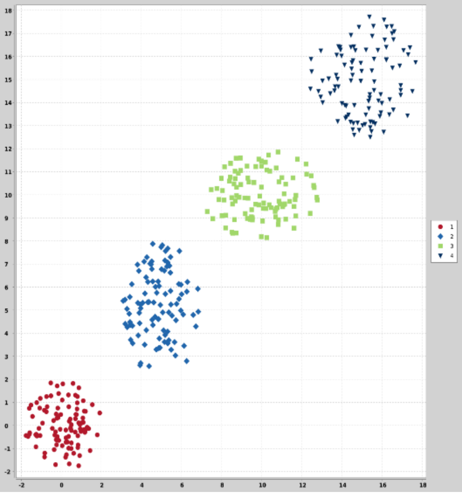


Fig. 3. Dataset *spherical_4_3*

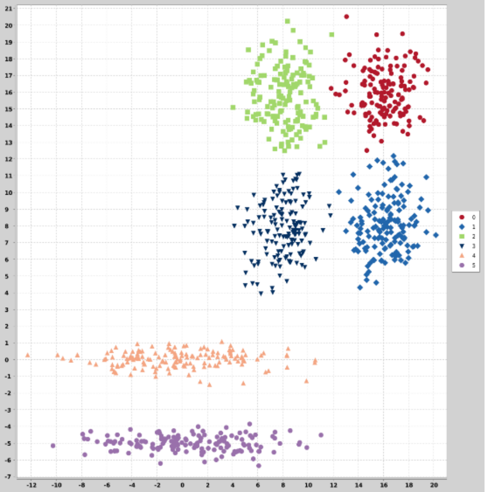


Fig. 2. Dataset *longsquare*

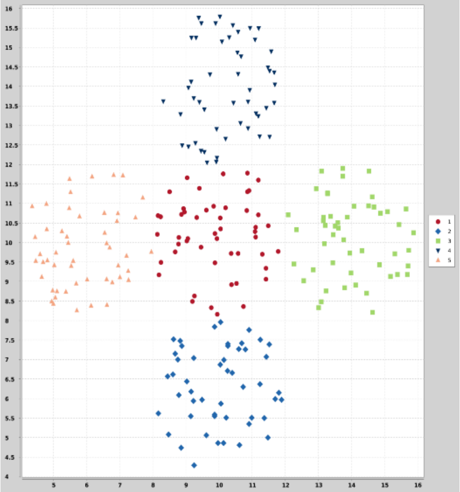


Fig. 4. Dataset *spherical_5_2*

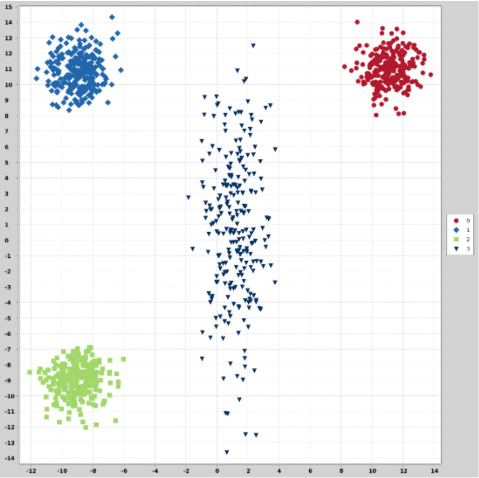


Fig. 5. Dataset *triangle1*

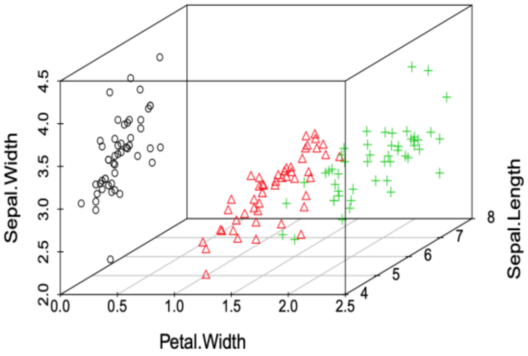


Fig. 6. Dataset *iris*

Table 2. Loading files into variables

Id	Read from a file
1	diamond9 <- read.csv("diamond9.csv")
2	longsquare <- read.csv("longsquare.csv")
3	spherical_4_3 <- read.csv("spherical_4_3.csv")
4	spherical_5_2 <- read.csv("spherical_5_2.csv")
5	triangle1 <- read.csv("triangle1.csv")
6	Iris <- datasets::iris[0:4]

Figure 1 shows the *diamond9* instance, which consists of nine diamonds or square groups that are connected to each other at the corners.

This instance has been used as part of the experiments described in [23] to automatically determine the number of clusters in an instance based on cluster evaluation metrics.

Figure 2 shows the *longsquare* instance, which consists of two different types of clustering: some based on cluster compactness and other based on connectivity. This instance has been used in the work presented in [24] as part of the optimization of multi-objective clustering.

Figure 3 shows the *spherical_4_3* instance, which originally contained four clusters in three dimensions, but for the purposes of this study, only the first two dimensions, *x* and *y*, were taken. This instance has been used the work described in [25] to evaluate the performance of a genetic algorithm, and subsequently, this algorithm is applied to image classification.

Figure 4 shows the *spherical_5_2* instance, which consists of five clusters where the objects are overlapping. This instance has been used in the experiments presented in [26] to evaluate the performance of various cluster validity indices: for example, the Davies-Bouldin (DB) index and the Dunn's index.

Figure 5 shows the *triangle1* instance, which, similar to Figure 2, consists of two different types of clustering: those based on cluster compactness and those based on connectivity.

Figure 6 shows the *iris* instance.

3.1 Data Preparation

Alternatively, if the data package FCPS cannot be installed in R, the dataset can be obtained from the benchmark [27]. In this case, the datasets are downloaded, and the two attributes to work with (the *x* and *y* coordinates) are selected and loaded into a variable in R with the instruction *read.csv*(“*path*”).

The *read.csv* command allows to load each instance from a file in table format and create a data frame from it. Table 2 presents the code required in R. The first column contains the identification of the dataset described in Table 1 and the next column contains the R code to read from a file.

In the case of the iris instance, we go through the exercise of taking it from the R dataset and load it into a variable.

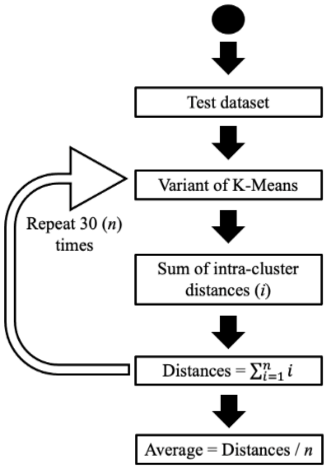
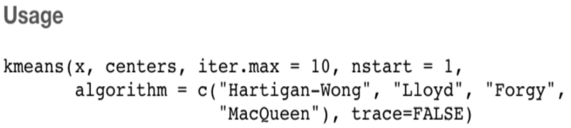


Fig. 7. Procedure for each variant of K-means



Usage

```
kmeans(x, centers, iter.max = 10, nstart = 1,
       algorithm = c("Hartigan-Wong", "Lloyd", "Forgy",
                     "MacQueen"), trace=FALSE)
```

Fig. 8. K-means function in R

Table 3. Average intra-cluster distances

Id	Average of intra-cluster distances			
	Hartigan	**Lloyd**	**Forgy**	**MacQueen**
1	15661	15684	15640	15630
2	41007	40927	40867	40504
3	34487	34439	34403	34185
4	29256	29216	29186	29004
5	56044	55624	55211	55075
6	589.73	598.25	593.99	585.47

4 Experimental Procedure

The K-means clustering algorithm is an unsupervised machine learning algorithm that allows to divide a given data set into a set of *k* clusters. The central idea of clustering is to define the clusters in such a way that the total intra-cluster variation is minimized.

To develop this experimental process with the variants of the K-means algorithm implemented in R, we follow the flow presented in Figure 7.

To this end, the next steps must be followed: 1) start with the test set (see Table 1) that will be processed by the K-means variant, 2) the variant is selected and the data set is processed, 3) when the algorithm converges, the sum of the intra-cluster distances is stored in a variable, 4) steps 1 to 3 are repeated 30 times and the intra-cluster distances are summed, and 5) the average intra-cluster distance for that instance with that variant is calculated.

4.1 Computing K-Means Clustering in R

The 4 variants of the K-means algorithm that are implemented in R are: Lloyd, Forgy, MacQueen and Hartigan-Wong. In this research, to analyze the behavior of each variant, the "intra-cluster distance", which is the sum of the distances between the centroids, will be used. In these tests, the variant of the algorithm that yields the highest "intra-cluster distance" will be the one that best separates the clusters.

The standard R function for K-means clustering is *kmeans*() and it is implemented as shown in the code in Figure 8.

Figure 8 shows the parameters of the function implemented in R [28], where *x* is the numerical matrix of the dataset to be clustered, *centers* is the number of partitions to be formed by the algorithm, *iter.max* is the maximum number of iterations allowed, and a*lgorithm* is the name of the variant to be used to cluster the dataset. When the variant is not specified in the algorithm parameter, the default function used is the Hartigan and Wong.

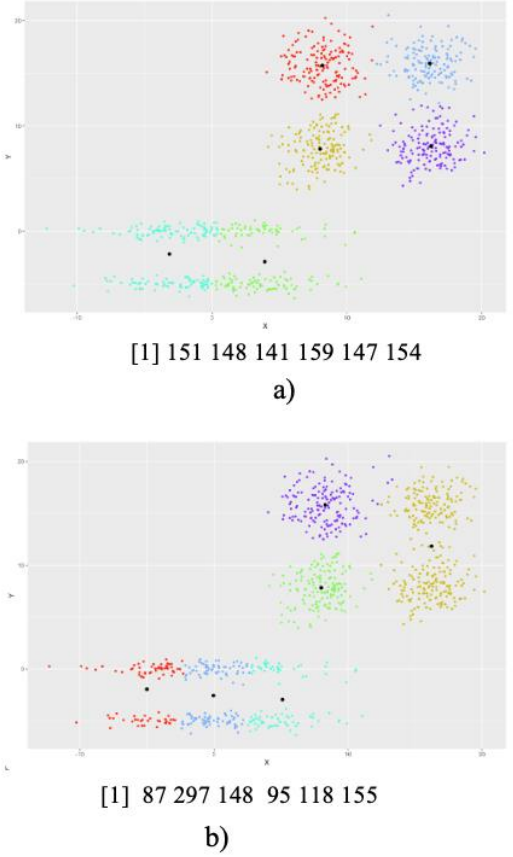
In the specialized literature, the vast majority of authors, when referring to the K-means algorithm, cite MacQueen (followed by Lloyd and Forgy) instead of the basic method. Additionally, in general, the Hartigan-Wong algorithm performs better than either of them.

5 Results of K-means Variants

Once the thirty runs of each variant were executed for each of the test datasets, the

Table 4. Minor and major intra-cluster distances

Id	Minor	Iterations	Major	Iterations
1	MacQueen	19	Lloyd	4
2	MacQueen	5	Hartigan	2
3	MacQueen	12	Hartigan	4
4	MacQueen	8	Hartigan	3
5	MacQueen	3	Hartigan	2
6	MacQueen	3	Lloyd	4



```
[1] 151 148 141 159 147 154
```

a)

```
[1]  87 297 148  95 118 155
```

b)

Fig. 9. Minor and major intra-cluster distance for *longsquare*

relevant results were summarized in Tables 3 and 4.

Table 3 shows the identifier of each dataset described in Table 1, and the following columns show the average distance between clusters obtained with each of the variants of the K-means algorithm: starting with Hartigan, Lloyd, Forgy, and MacQueen. The *minor* (minimum) and *major* (maximum) intra-cluster distance marked in blue and yellow, respectively.

Table 4 is an extension of Table 3, showing, in summary form, which of the variants obtained the *minor* (minimum) and *major* (maximum) intra-cluster distance and for each of them, the number of iterations needed to reach convergence.

It is important to mention that for all instances, the variant that in all cases reports the minor intra-cluster distances is the MacQueen variant. Additionally, except in two cases, the Lloyd variant reports the largest intra-cluster distance, for all other cases, it is the Hartigan variant.

Another important fact from the results shown in this table is the number of iterations, which is seen to increase by a third with the MacQueen variant to obtain the minor intra-cluster distances.

6 Clustering Analysis and Discussion

This section presents the cluster analysis for each of the instances and we will discuss the main observations found in the behavior of each of the instances resulting from the variants of the K-means algorithm, mainly concerning the intra-cluster distances, the clustering and the number of iterations.

Figure 9 shows how the variants obtained a different clustering for this instance. This is mainly due to the fact that the instance is composed of two different types of clustering, in the upper right part four clusters can be visually identified, which present a spherical shape and in the lower left part two clusters with an elongated shape can be seen.

The Figure 9 has two sections, a) and b). Section a) shows the clustering with the variant that obtained the minor intra-cluster distances, section b) shows the clustering with the variant that obtained the major intra-cluster distances, all referring to Table 3. In addition, below each section, the distribution of the objects by cluster is shown.

MacQueen's variant, Figure 9 a), generates one centroid for each of the spherical clusters in the upper part and two for the clusters in the lower part, separating them vertically, while in Figure 9 b), Hartigan's variant merges two of the spheres in the upper part and in the lower part, generates three clusters for the elongated regions, also separating them vertically.

It should be noted that the minor intra-cluster distance was always obtained for all instances with the MacQueen variant.

The major intra-cluster distance obtained in most instances corresponds to the Hartigan variant, except for the first and last instances, see Table 3.

The Lloyd variant obtains the major intra-cluster distance for the *diamond9* instance. This instance has the peculiarity that it is composed of nine clusters that are connected at their corners, usually used in experimentation to automatically identify the number of clusters that constitute an instance.

On the other hand, the final clustering obtained with the variants reporting the minor and major intra-cluster distance, maintains the same structure and distribution of objects for each cluster, except for the *longsquare* instance.

It is important to mention that the number of iterations required by each of the variants to obtain the final clustering also varies, see Table 3. This is best seen in case one, where the MacQueen variant executes 19 iterations while Lloyd performs only 4. Similarly, in case 3, the MacQueen variant executes 12 iterations and Hartigan 4.

What is surprising is that the Lloyd and Forgy variants rarely appear in these experiments. Forgy never got either the minor or major intra-cluster distance. Lloyd only achieved the major intra-cluster distance twice.

Lloyd and Forgy variants, in the general literature, are characterized by working well for large data sets and by randomly selecting the initial centroids.

However, when the algorithm is executed in the computer memory, it requires storing the results of the last two iterations, which is very expensive, and it also creates empty cluster sets. As for the input data, Lloyd's variant works with discrete data distributions and Forgy's with continuous data distributions.

On the other hand, the variant proposed by MacQueen is characterized by initializing the centroids with the first objects in the set, one for each cluster, and then recalculating the centroid of a cluster immediately after it is assigned an object, and not at the end of the iteration as in the other variants. In this sense, the MacQueen variant is said to be more efficient because it updates the centroids frequently and runs through all the clusters before convergence.

Finally, based on the behavior of these variants and the information obtained, it is possible to decide which variant to choose to solve an instance with the variants implemented in the R language.

7 Conclusion

In this paper we present an exploratory analysis of the behavior of the main variants of the K-means algorithm (Hartigan-Wong, Lloyd, Forgy and MacQueen) when solving some of the difficult sets of instances from the Fundamental Clustering Problems Suite (FCPS) benchmark.

The main results show that, of the four variants, MacQueen gives the best results when the variance of the clusters is intended to be the minor; however, the Hartigan variant is better for obtaining the major variance. In this sense, the default variant using R in the *kmeans*() function is the Hartigan variant.

It is important to note the time and number of iterations to reach the solution, since in all cases where the minor intra-cluster variance was obtained, the number of iterations was higher. In general, it is suggested to use the MacQueen variant when there are no restrictions on the solution time.

References

1. **Kambatla, K., Kollias, G., Kumar, V., Grama, A. (2014).** Trends in big data analytics. Journal of parallel and distributed computing, Vol. 74, No. 7, pp. 2561–2573.

2. **Tsai, C. W., Lai, C. F., Chao, H. C., Vasilakos, A. V. (2015).** Big data analytics: A survey. Journal of Big data, Vol. 2, No. 1, pp. 1–32.
3. **Wu, X., Kumar, V. (2009).** The top ten algorithms in data mining. CRC press.
4. **Mahajan, M., Nimbhorkar, P., Varadarajan, K. (2012).** The planar k-means problem is NP-hard. Theoretical Computer Science, Vol. 442, pp. 13–21.
5. **Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K. (2012).** Cluster: Cluster analysis basics and extensions. R package v. 1.14.2.
6. **Hornik, K., Feinerer, I., Kober, M., Buchta, C. (2012).** Spherical k-means clustering. Journal of statistical software, Vol. 50, No. 1, pp. 1–22.
7. **FCPS (2008).** https://cran.r-project.org/package=FCPS.
8. **Steinley, D. (2006).** K-means clustering: A half-century synthesis. British Journal of Mathematical and Statistical Psychology, Vol. 59, No. 1, pp. 1–34.
9. **Hans-Hermann, B.O.C.K. (2008).** Origins and extensions of the k-means algorithm in cluster analysis. Journal Electronique d'Histoire des Probabilités et de la Statistique Electronic Journal for History of Probability and Statistics, Vol. 4, No. 2.
10. **Jain, A.K. (2010).** Data clustering: 50 years beyond K-means. Pattern recognition letters, Vol. 31, No. 8, pp. 651–666.
11. **Morissette, L., Chartier, S. (2013).** The k-means clustering technique: General considerations and implementation in Mathematica. Tutorials in Quantitative Methods for Psychology, Vol. 9, No. 1, pp. 15–24.
12. **Yim, O., Ramdeen, K.T. (2015).** Hierarchical cluster analysis: Comparison of three linkage measures and application to psychological data. The quantitative methods for psychology, Vol. 11, No. 1, pp. 8–21.
13. **Yim, O., Ramdeen, K.T. (2015).** Hierarchical cluster analysis: Comparison of three linkage measures and application to psychological data. The quantitative methods for psychology, Vol. 11, No. 1, pp. 8–21.
14. **Pérez-Ortega, J., Almanza-Ortega, N.N., Vega-Villalobos, A., Pazos-Rangel, R., Zavala-Díaz, C., Martínez-Rebollar, A. (2019).** The k-means algorithm evolution. Introduction to Data Science and Machine Learning. IntechOpen.
15. **Thrun, M.C., Ultsch, A. (2020).** Clustering benchmark datasets exploiting the fundamental clustering problems. Data in brief, Vol. 30, 105501.
16. **Thrun, M. C., Stier, Q. (2021).** Fundamental clustering algorithms suite. SoftwareX, Vol. 13, 100642.
17. **Fisher, R.A. (1936)** The use of multiple measurements in taxonomic problems. Annals Eugenics, Vol. 7, No. II, pp. 179–188.
18. **Lloyd, S. (1982).** Least squares quantization in PCM. IEEE transactions on information theory, Vol. 28, No. 2, pp. 129–137.
19. **Forgy, E. (1965).** Cluster analysis of multivariate data: Efficiency vs. interpretability of classification, Biometrics, Vol. 21, pp. 768.
20. **MacQueen, J. (1967).** Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probabilities, Vol. 1, pp. 281–296.
21. **Data Mining Algorithms in R: Clustering/K-Means (2012)**. https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/K-Means.
22. **Hartigan, J.A., Wong, M.A. (1979).** Algorithm AS 136: A K- Means Clustering Algorithm. Applied Statistics, Vol. 28, No. 1, pp. 100–108.
23. **Salvador, S., Chan, P. (2004).** Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. 16th IEEE international conference on tools with artificial intelligence, pp. 576–584.
24. **Handl, J., Knowles, J. (2007).** An evolutionary approach to multiobjective clustering. IEEE transactions on Evolutionary Computation, Vol. 11, No. 1, pp. 56–76.
25. **Bandyopadhyay, S., Maulik, U. (2002).** Genetic clustering for automatic evolution of clusters and application to image classification. Pattern recognition, Vol. 35, No. 6, pp. 1197–1208.
26. **Bandyopadhyay, S., Maulik, U. (2001).** Nonparametric genetic clustering: comparison of validity indices. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 31, No. 1, pp. 120–125.
27. **Clustering Benchmark. (2019).** https://github.com/deric/clustering-benchmark.
28. **R Documentation. (2019).** https://www.r-project.org/other-docs.html.

Article received on 28/07/2021; accepted on 30/09/2021.
Corresponding author is Joaquín Pérez-Ortega.