Chapter

# The *K*-Means Algorithm Evolution

*Joaquín Pérez-Ortega, Nelva Nely Almanza-Ortega, Andrea Vega-Villalobos, Rodolfo Pazos-Rangel, Crispín Zavala-Díaz and Alicia Martínez-Rebollar*

## Abstract

Clustering is one of the main methods for getting insight on the underlying nature and structure of data. The purpose of clustering is organizing a set of data into clusters, such that the elements in each cluster are similar and different from those in other clusters. One of the most used clustering algorithms presently is *K*-means, because of its easiness for interpreting its results and implementation. The solution to the *K*-means clustering problem is NP-hard, which justifies the use of heuristic methods for its solution. To date, a large number of improvements to the algorithm have been proposed, of which the most relevant were selected using systematic review methodology. As a result, 1125 documents on improvements were retrieved, and 79 were left after applying inclusion and exclusion criteria. The improvements selected were classified and summarized according to the algorithm steps: initialization, classification, centroid calculation, and convergence. It is remarkable that some of the most successful algorithm variants were found. Some articles on trends in recent years were included, concerning *K*-means improvements and its use in other areas. Finally, it is considered that the main improvements may inspire the development of new heuristics for *K*-means or other clustering algorithms.

**Keywords:** clustering, *K*-means, systematic review, historical developments, perspectives on clustering

## 1. Introduction

The accelerated progress of technology in recent time is fostering an important increase in the amount of generated and stored data [1–4] in fields such as engineering, finance, education, medicine, and commerce, among others. Therefore, there is justified interest in obtaining useful knowledge that can be extracted from those huge amounts of data, in order to help making better decisions and understanding the nature of data. Clustering is one of the fundamental techniques for getting insight on the underlying nature and structure of data. The purpose of clustering is organizing a set of data into clusters whose elements are similar to each other and different from those in other clusters.

One of the clustering algorithms more widely used to date is *K*-means [5], because of its easiness for interpreting its results and implementation. Another factor that has contributed to its use is the existence of versions implemented in the Weka and SPSS platforms and open-source programming languages such as Python and R, among others.

It is convenient to point out that *K*-means is a family of algorithms that were developed in the 1950s as a result of independent investigations. These algorithms have in common four processing steps, with some differences in each step. It was in an article by MacQueen [6] where the name *K*-means was coined.

The solution to the *K*-means clustering problem is hard, and it has been proven that it is NP-hard, which justifies the use of heuristic methods for its solution. According to the *no free lunch* theorem, there is no algorithm that is superior to other algorithms for all types of instances of an NP-complete problem. This has limited the design of a general algorithm for the clustering problem. For more than 60 years, a large number of variants of the algorithm have been proposed. There exist some surveys on *K*-means and its improvements. Classical surveys are [7] that synthesize *K*-means variants and their results, and in [8] a historical review is presented of continuous and discrete variants of the algorithm. In [9] several clustering methods and key aspects on clustering algorithm design are summarized, and a remarkable list of challenges and research directions on *K*-means was proposed. In [10] a review of theoretical aspects on *K*-means and scalability for Big Data is presented. Unlike these surveys, this documentary research focuses on classifying the *K*-means improvements according to the algorithm steps. This classification is particularly useful for designing versions customized of *K*-means for solving certain types of problem instances. This is also a contribution to the knowledge on the most important improvements for each step and, in general, to the behavior of the algorithm.

For selecting the most relevant articles, systematic review methodology was used. The filters used and the analysis of results allowed finding some of the most successful and referenced algorithm variants for each step of the algorithm.

The chapter is organized as follows: Section 2 summarizes the pioneering works that originated the family of *K*-means-type algorithms; additionally, it describes the standard algorithm and the formulation of the clustering problem. Section 3 describes the application of systematic review methodology for retrieving the most important articles on *K*-means; it also includes the step or steps to which the improvements apply and tables that summarize the number of articles; lastly, it includes a subsection on the new trends on the use of *K*-means. Finally, Section 4 includes the conclusions, highlighting the most successful and referenced algorithm variants.

## 2. Origins of the family of *K*-means-type algorithms

During the decades of the 1950s and 1960s, several *K*-means-type algorithms were proposed. These proposals were developed independently by researchers from different disciplines [8]. These algorithms had in common a process that originated what is currently known as the *K*-means algorithm.

According to the specialized literature [6, 11–20], four algorithms gave origin to this family. The following subsections describe the articles related to these algorithms and their authors.

### 2.1 Steinhaus (1956)

Mathematician Hugo Steinhaus, from the Mathematics Institute of the Polish Academy of Sciences, published an article titled "Sur la Division des Corps Matériels en Parties" [11], in which he presented the problem of partitioning a heterogeneous solid by the adequate selection of partitions. He also mentioned applications in the

fields of anthropology and industry. Steinhaus was the first researcher that proposed explicitly an algorithm for multidimensional instances.

### 2.2 Lloyd (1957)

Stuart Lloyd, from Bell Laboratories, in the article titled "Least Squares Quantization in PCM" [12] approached the problem of transmitting a random signal $X$ in a multidimensional space. Lloyd worked in the communications and electronics fields, and its algorithm is presented as a technique for pulse-code modulation.

### 2.3 MacQueen (1967)

James MacQueen, from Department of Statistics of the University of California, in his article titled "Some Methods for Classification and Analysis of Multivariate Observations" [6], proposed an algorithm for partitioning an instance into a set of clusters whose variance was small for each cluster. The term $K$-means was coined by him; it was known by different names: dynamic clustering method [13–15], iterative minimum-distance clustering [16], nearest centroid sorting [17], and h-means [18], among others.

### 2.4 Jancey (1966)

Jancey, from the Department of Botany, School of Biological Sciences, University of Sydney, in one of his articles titled "Multidimensional Group Analysis" [19], presented a clustering method for characterizing species *Phyllota phylicoides*. Jancey conducted his research in the field of taxonomy. There exists a variant of this method with similar characteristics, which was introduced by Forgy in the article "Cluster Analysis of Multivariate Data: Efficiency Versus Interpretability of Classification" [20]. The fundamental difference with respect to Jancey's work lies in the way in which the initial centroids are selected.

Because the results from Jancey's research will be used as reference for this chapter, his algorithm will be described in detail. The author stated that the similarity measures are based on the results published by the following authors: (a) Pearson in his article titled "On the Coefficient of Racial Likeness [21] published in 1926," (b) Rao in the article named "The Use of Multiple Measurements in Problems of Biological Classification" [22] published in 1948, and (c) Sokal in his article titled "Distance as a Measure of Taxonomic Similarity" [23] in 1961.

Pearson [21] in his article "On the Coefficient of Racial Likeness," when studying craniology and physical anthropology, confronted the difficulty of comparing two types of races, in order to determine the membership of a limited number of individuals to one race or another or both. As a result, Pearson proposed a coefficient of racial likeness (CRL). For calculating this coefficient, it is necessary to obtain first the means and variances of each characteristic in each sample, since it is assumed that there is variability for each of the characteristics considered. This coefficient is used to measure the dispersion around the mean and the degree of association between two variables.

The article published by Radhakrishna Rao [22] in the *Journal of the Royal Statistical Society*, titled "The Utilization of Multiple Measurements in Problems of Biological Classification," aimed at presenting a statistical approach for two types of problems that arise in biological research. The first deals with the determination of an individual as member of one of the many groups to which he/she possibly might belong. The second problem deals with the classification of groups into a system based on the configuration of their different characteristics.

Sokal [23] published his article titled "Distance as Measure of Taxonomic Similarity," which is based on the methods for quantifying the taxonomy classification process, and he points out the importance of having fast processing and data calculation methods. The purpose of his work is to evaluate the similarities among taxa that have observed characteristic values, instead of phylogenetic speculations and interpretations.

The similarity among objects is evaluated based on many attributes, and all the attributes are considered as equal taxonomic values; therefore, an attribute is not weighted more or less than any other.

For performing the weighting of attributes, three types of coefficients are used: association, correlation, and distance, where the last one is of interest for this study. This distance coefficient is employed for determining the similarity between two objects by using a distance function in an $n$-dimensional space, in which the coordinates represent the attributes.

A measure of similarity between the objects 1 and 2 based on two attributes would be the distance in a two-dimensional space (i.e., a Cartesian plane) between the two objects. This distance $\delta_{1,2}$ can be easily calculated through the well-known formula from analytic geometry, Eq. (1):

$$\delta_{1,2} = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \tag{1}$$

where $X_1$ and $Y_1$ are the object 1 coordinates and $X_2$ and $Y_2$ are the object 2 coordinates.

Similarly, when three attributes are needed for two different objects, it is now necessary to carry out the distance calculation in a three-dimensional space so that the exact position of the two objects can be represented regarding the three attributes. For calculating the distance between these two objects, an extension to the three-dimensional space of the formula for $\delta_{1,2}$ can be applied. When more than three dimensions are needed for the objects, it is not possible to represent their positions using conventional geometry; therefore, it is necessary to resort to algebraic calculation of data. However, the formula for distance calculation from analytic geometry is equally valid for an $n$-dimensional space.

The general formula for calculating the distance for two objects with $n$ attributes is shown in Eq. (2):

$$\delta_{1,2}^2 = \sum_{i=1}^{n} (X_{i1} - X_{i2})^2 \tag{2}$$

where $X_{ij}$ is the value of attribute $i$ for object $j$ ($j$ = 1, 2).

Once the object classification process is completed, then the matrix of similarity coefficients obtained (based on object distances) can be used in the usual methods for clustering analysis.

Finally, it is important to emphasize the feasibility of calculating distance as the summation of the squared differences of the attribute values of objects of different kinds.

The clustering method proposed by Jancey consists of the following four steps:

1. Initialization. First, $k$ points are randomly generated in the space, which are used as initial centroids.

2. Classification. The distances from all the objects to all the centroids are calculated, and each object is assigned to its closest centroid.

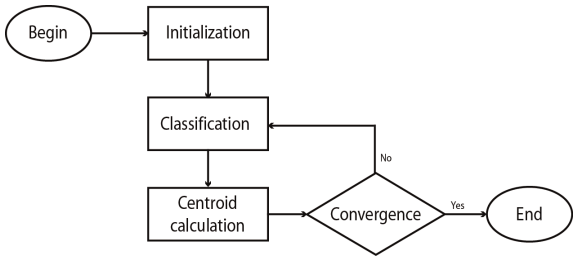


**Figure 1.**
*Standard K-means algorithm.*

3. Centroid calculation. New centroids are calculated using the mean value of the objects that belong to each cluster.

4. Convergence. The algorithm stops when equilibrium is reached, i.e., when there are no object migrations from one cluster to another. While no equilibrium is reached, the process is repeated from step 2.

### 2.5 *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 a cluster are similar to each other and are different from those in other clusters. In the following paragraphs, the clustering problem related to *K*-means is formalized.

Let $N = \{x_1, ..., x_n\}$ be the set of $n$ objects to be clustered by a similarity criterion, where $x_i \in \Re^d$ for $i = 1, ..., n$ and $d \geq 1$ is the number of dimensions. Additionally, let $k \geq 2$ be an integer and $K = \{1, ..., k\}$. For a $k$-partition, $P = \{G(1), ..., G(k)\}$ of $N$, let $\mu_j$ denote the centroid of cluster $G(j)$, for $j \in K$, and let $M = \{\mu_1, ..., \mu_k\}$ and $W = \{w_{11}, ..., w_{ij}\}$.

Therefore, the clustering problem can be formulated as an optimization problem [24], which is described by Eq. (3):

$$\boldsymbol{P}: \text{minimize } z(W, M) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij} d\left(x_i, \mu_j\right) \tag{3}$$

$$\text{subject to } \sum_{j=1}^{k} w_{ij} = 1, \text{ for } i = 1, ..., n,$$

$$w_{ij} = 0 \text{ or } 1, \text{ for } i = 1, ..., n, \text{ and } j = 1, ..., k,$$

where $w_{ij} = 1$ implies object $x_i$ belongs to cluster $G(j)$ and $d(x_i, \mu_j)$ denotes the Euclidean distance between $x_i$ and $\mu_j$ for $i = 1,..., n$ and $j = 1,..., k$.

The standard version of the *K*-means algorithm consists of four steps, as shown in **Figure 1**.

The pseudocode of the standard *K*-means algorithm is shown in Algorithm 1.

---

**Algorithm 1**. Standard *K*-means algorithm

```
# Initialization:
    N: = {x_1, ..., x_n};
    M: = {μ_1, ..., μ_k};
# Classification:
    For x_i ∈ N and μ_k ∈ M
        Calculate the Euclidean distance from each x_i to the k
        centroids;
```

```
7:            Assign object x_i to the closest centroid μ_k;
8:        # Centroid calculation:
9:            Calculate centroid μ_k;
10:       # Convergence:
11:           If M: = {μ_1, ..., μ_k} remains unchanged in two consecutive
                iterations
              then:
12:           Stop the algorithm;
13:           else:
14:           Go to Classification
15:       End
```

Since the pioneering studies conducted by Steinhaus [11], Lloyd [12], MacQueen [6], and Jancey [19], many investigations have been aimed at finding a $k$-partition of $N$ that solves problem $\boldsymbol{P}$, defined by Eq. (3).

It has been shown that the clustering problem belongs to the NP-hard class for $k \geq 2$ or $d \geq 2$ [25, 26]. Therefore, obtaining an optimal solution for an instance of moderate size is generally an intractable problem. Consequently, a variety of heuristic algorithms have been proposed for obtaining the closest possible solution to the optimum of $\boldsymbol{P}$, being the most important of those designed as $K$-means-type algorithms [6].

It is important to emphasize that the establishment of useful gaps between the optimal solution of the problem $\boldsymbol{P}$ and the solution achieved by $K$-means remains an open research problem.

The computational complexity of $K$-means is $O(nkdr)$, where $r$ represents the number of iterations [8, 9], which restricts its use for large instances, because each iteration involves the calculation of all the object-centroid distances. For reducing the complexity of $K$-means, numerous investigations have been carried out using different strategies for reducing the computational cost and minimizing the objective function.

## 3. Classification of articles on $K$-means improvements according to the algorithm steps

This section presents a classification of the most relevant articles on improvements to $K$-means regarding the algorithm steps. The articles were selected applying the systematic review methodology.

### 3.1 Systematic review process

The search for articles was carried out using four highly prestigious databases: Springer Link, ACM, IEEE Xplore, and ScienceDirect. Queries were issued to these databases using the following search string:

((({k means} OR {kmeans} OR {Lloyd algorithm} OR {k+means} OR {“k means”} OR {“algoritmo de lloyd”}) AND ({improvement} OR {enhancement} OR {mejora})) AND ({Initialization} OR {inicializacion} OR {beginning} OR {inicio} OR {partition} OR {particion} OR {first step} OR {primer paso} OR {centroide}) AND ({classification} OR {clasificacion} OR {sorting} OR {assignment} OR {asignacion} OR {range search} OR {neighnour search} OR {búsqueda en vecindario}) AND (OR {centroide calculation} OR {calculo de centroide}) AND ({Convergence} OR {Convergencia} OR {Stop criteria} OR {criterio de paro} OR {Stop condition} OR {Condicion de parada} OR {convergence condition} OR {Condicion de convergencia} OR {final step} OR {Paso final})).

As a result of the queries, 1125 articles were retrieved related to the *K*-means algorithm and its improvements. Next, inclusion and exclusion criteria were applied, which reduced the number of articles to 79. The remaining articles were classified according to the algorithm steps as shown in the following subsections.

The flow chart in **Figure 2** shows the steps of the process carried out for selecting the articles. In step "a," the database queries were issued, and a document list was generated, and in step "b," duplicate articles were identified and eliminated. In step "c," based on article titles, those irrelevant to this research were identified and discarded. In step "d," article abstracts were analyzed, and those with little affinity to the subject of study were excluded. In step "e," those documents written in languages different from English or Spanish were eliminated. In step "f," those articles that did not describe an improvement process were discarded. In step "g," the text of the articles was reviewed, and those with little affinity to the subject of study were excluded. In step "h," four articles were eliminated because of possible plagiarism. Finally, in step "i," articles with a small number of citations were discarded; specifically, those with citations below a threshold adjusted by year and category.

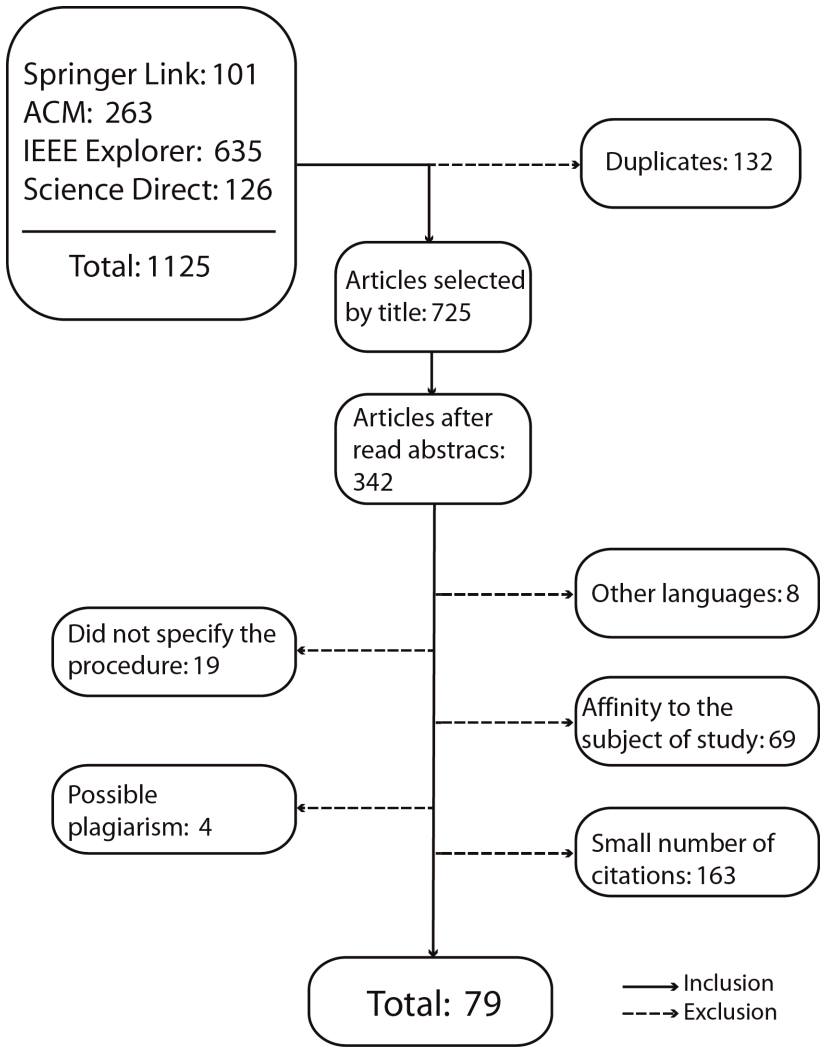


**Figure 2.**
*Process for selecting articles.*

### 3.2 Article classification

As a result of the analysis of the articles, those addressing an improvement to one of the algorithm steps were identified. However, several works were found that involved improvements to more than one step; therefore, the following groups or categories were defined: (a) initialization, (b) classification and centroid calculation, (c) convergence, (d) convergence and initialization, and (e) convergence and classification.

In **Figure 3**, the number of articles for each of the aforementioned groups is shown. Notice that the step with the most articles is initialization and the step with the least attention by researchers is convergence. In the following subsections, the most important articles in each group are briefly described.

### 3.3 Initialization

The initialization step has received the most attention by researchers, because the algorithm is sensitive to the initial position of the centroids; i. e., different initial centroids may yield different resulting clusters. Consequently, a good initial selection might find a better solution and a reduction in the number of iterations needed by the algorithm to converge.

For this step 38 documents were found about improvements proposed for generating better initial centroids. **Table 1** summarizes information on the articles for this step. Column 1 shows the articles for this step. Columns 2 through 5 indicate the different strategies that researchers have used for obtaining improvements for this step. Finally, column 6 shows the number of articles for each of the strategies.

The second row shows articles on approaches that perform a preprocessing for generating the initial centroids by using particular algorithms or methods. The third row includes articles on methods based on information on data set. The fourth row shows articles on techniques that involve more effective data structures. Finally, the fifth row includes articles where the improvements use other strategies.

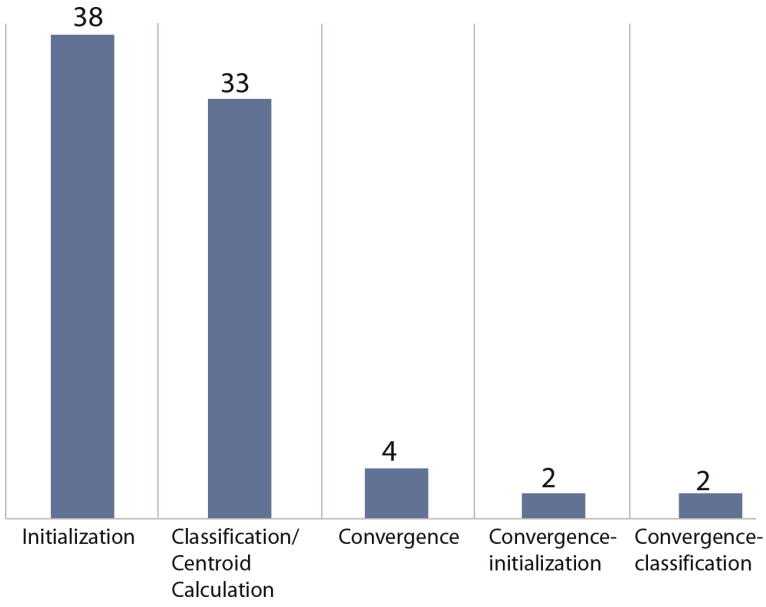


**Figure 3.**
*Number of articles for each of the aforementioned groups.*

| Articles | Strategy | | | | Number of articles |
|---|---|---|---|---|---|
| | Algorithm/ method | Instance information | Data structure | Other | |
| [24, 27–40] | ● | | | | 15 (39.47%) |
| [41–55] | | ● | | | 15 (39.47%) |
| [56–59] | | | ● | | 4 (10.53%) |
| [60–63] | | | | ● | 4 (10.53%) |

**Table 1.**
*Summary of information on the articles for initialization.*

In the rest of this subsection, some of the most important works on the initialization step are summarized. Several of these works mentioned can be used in other algorithms similar to *K*-means for selecting the initial centroids.

In [27] a clustering method is proposed, where the centroids of the final clusters are used as initial centroids for *K*-means. The main idea is to randomly select an object $x$, which is used as a first initial centroid; from this object, the following $k-1$ centroids are selected considering a distance threshold set by the user.

In [28] a modification to Lloyd's work [12] is developed in the field of quantization. The main idea is that objects that are farther from each other have a larger probability of belonging to different clusters; therefore, the strategy proposed consists in choosing an object with the largest Euclidean distance to the rest of the objects for being the first centroid. The following $i$-th centroids ($i = 2, 3, \ldots, k$) will be selected in decreasing order with respect to the largest distance to the first centroid.

In [29] two initialization methods are developed, which are aimed at being applied to large data sets. The proposed methods are based on the densities of the data space; specifically, they need first to divide uniformly the data space into $M$ disjoint hypercubes and to randomly select $kN_m/N$ objects in hypercube $m$ ($m = 1, 2, \ldots, M$) for obtaining a total of $k$ centroids.

In [30] a preprocessing is performed called *refining*. This method consists in using *K*-means for solving $M$ samples of the original data set. The results of SSQ (sum of squared distances) are compared for each of the $M$ solutions, and from the solution with the smallest value, the set of final centroids is extracted, which are used as the initial centroids for solving the entire instance using *K*-means.

In [31] a preprocessing method is proposed which uses a selection model based on statistics. In particular, it uses the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) for selecting the set of objects that will be used as initial centroids.

In [42] an algorithm is presented based on two main observations, which state that the more two objects are similar to each other, the largest the possibility that they end up assigned to the same cluster, so they are discarded from the selection of initial centroids. This method is based on density-based multiscale data condensation (DBMSDC) and allows identifying regions with large data densities, and afterwards a list is generated sorting the values by density. Next, select an object according to the sorted list as the first initial centroid, and all the objects that have a ratio inversely proportional to the density of the selected object are discarded. Afterwards, the second centroid is selected as the next object in the list that has not been eliminated, and its surrounding objects are excluded. This process is repeated until all the initial centroids needed are obtained.

A variance-based method for finding the initial centroids is proposed in [44]. First, the method calculates the variances of the values for each dimension, and it

selects the dimension with the largest variance. Next, it sorts the objects by the values on the dimension with the largest variance. Finally, it creates $k$ groups with the same number of objects each, and for each group it calculates the median. The medians constitute the initial centroids.

Other researchers have focused their works on using information on data set, such as the distribution of objects and statistical information of them, among others.

In [45] a method is proposed for randomly generating the initial centroids as described next: the first centroid is randomly generated using a uniform probability distribution for the objects; subsequent centroids ($i$ = 2, 3, ..., $k$) are generated calculating a probability that is proportional to the square of their minimal distances to the set of previously selected centroids (1, ..., $i$–1).

In [50] a method is proposed, which is based on a sample of the data set for which an average is calculated. Next, the objects whose distance is larger than the average are identified, and a distance-between-objects criterion is applied for selecting the objects that will constitute the initial objects. The authors claim that this method obtains good results regarding time and solution quality when solving large data sets.

In [55] a method is proposed for eliminating those objects that may cause noise, as well as *outliers*. The method determines the most dense region of the data space, from which it locates the best initial centroids.

By using particular data structures, in [59] a method is presented for estimating the data density in different locations of the space by using kd-tree type structures.

Other researchers [58, 60] have used a combination of genetic algorithms and $K$-means for the initialization step; however, this method has high computational complexity, since it is necessary to execute the $K$-means algorithm on the entire set of objects of the population, for each of the generations of the genetic algorithm.

### 3.4 Classification

Of the four steps of the algorithm, classification is the most time-consuming, because for each object it is necessary to calculate the distance from each object to each centroid.

The 33 articles that are the best proposals for this step are classified in **Table 2**. Column 1 shows the articles related to this step. Columns 2 through 6 indicate the different strategies that researchers have used for achieving improvements for this step. Finally, column 7 shows the number of articles for each of these strategies aiming at reducing the number of calculations of object-centroid distances.

The second row shows articles on approaches that use compression thresholds. The third row includes articles on methods that use information from the initialization step. The fourth row shows articles on techniques that involve more efficient data structures. The fifth row includes articles that present mathematical/statistical processes. Finally, the sixth row shows articles where the improvements use other strategies.

In [64] an improvement is proposed, which reduces the number of calculations of object-centroid distances. For this purpose, an exclusion criterion is defined based on the information of object-centroid distances in two successive iterations: $i$ and $i$ + 1. This criterion allows to exclude an object $x$ from the distance calculations to the rest of the centroids, if it is satisfied that the distance to the centroid in iteration $i$ + 1 is smaller than that of iteration $i$.

In [69] a heuristic is proposed, which reduces the number of objects considered in the calculations of object-centroid distances; i.e., the objects with small probability of changing cluster membership are excluded. The rationale behind this heuristic derives from the observation that objects closest to a centroid have a small probability

| Articles | Strategy | | | | | Number of articles |
|---|---|---|---|---|---|---|
| | Compression thresholds | Information of previous steps | Data structure | Mathematical/ statistical process | Other | |
| [64–72] | ● | | | | | 9 (27.27%) |
| [73–80] | | ● | | | | 8 (24.24%) |
| [81–87] | | | ● | | | 7 (21.21%) |
| [88–92] | | | | ● | | 5 (15.15%) |
| [93–96] | | | | | ● | 4 (12.12%) |

**Table 2.**
*Summary of information on the articles for classification.*

of changing cluster membership, whereas those closer to the cluster border have higher probability of migrating to another cluster. The heuristics determine a threshold for deciding which objects should be excluded. The calculation of the threshold is defined as the sum of the two largest centroid shifts with respect to the previous iteration. Another work with a similar strategy is presented in [66].

In [72] an improvement is presented, which allows excluding from the calculation of object-centroid distances those objects in clusters that have not had object migrations in two successive iterations. This type of clusters is called *stable*, and the objects in such clusters keep their membership unchanged for the rest of the iterations of the algorithm. In the article [73], a similar strategy is presented.

In [74] an improvement to *fast global K-means* algorithm is proposed, which is based on the cluster membership and the geometrical information of the objects. This work also includes a set of inequalities that are used to determine the initial centroids.

In [79] a heuristic is presented, which reduces the number of calculations of object-centroid distances. Specifically, it calculates the distances for each object only to those centroids of clusters that are neighbors of the cluster where the object belongs. This heuristic is based on the observation that objects can only migrate to neighboring clusters.

One of the most representative works for this step is presented in [82], where an improvement is proposed, called *filtering algorithm (FA)*, which uses data structures of binary tree type, called *kd-tree*. Each node of the tree is associated to a set of objects called *cell*. An improvement of this work is described in [85], where the authors claim that it reduces execution time by 33.6% with respect to algorithm FA. Another remarkable improvement to the work in [82] is presented in [83].

### 3.5 Centroid calculation

Centroid calculation was defined as another step in this analysis, because there exist two variants for this step that differentiate two types of the *K*-means algorithm. In one of the types, centroid calculations are performed once all the objects have been assigned to one cluster. This type of calculation method is called *batch* and is used by [12, 19], among others. The second type of calculation is performed each time an object changes cluster membership. This type of calculation is called *linear K-means* and was proposed by MacQueen. No documents were retrieved related to this step.

### 3.6 Convergence

The convergence step of the algorithm has received little attention by researchers, which is manifested by the small number of papers on this subject. It is

worth mentioning that in recent years, research on this step has produced very promising results concerning the reduction of algorithm complexity at the expense of a minimal reduction of solution quality.

A pioneering work for this step was presented in [97]. The main contribution consisted in associating the values of the squared errors to the stop criterion of the algorithm. In particular, the proposed condition for stopping is when, in two successive iterations, the value of the squared error in one iteration is larger than in the previous one, which guarantees that the algorithm stops at the first local optimum.

Other articles for this step are [98, 99], from the point of view of mathematical analysis, aiming at proving when does the solution obtained reach a global optimum.

### 3.7 Convergence and initialization

This subsection summarizes two works on improvements for the convergence and initialization steps.

In [100] the stop criterion is associated to the number $k$ of clusters; i.e., in each iteration a new initial centroid is generated for creating a new cluster. This stop criterion is called *incremental*.

In [101] convergence is reached in two ways. In the first condition, the algorithm stops when it reaches a predefined number of iterations. In the second, the algorithm stops when there is no region with a density value larger than a predefined threshold. It is important to mention that in each iteration, the algorithm creates a new cluster guided by the density value in a region.

### 3.8 Convergence and classification

In this subsection two works are summarized, which present improvements for the convergence and classification steps.

In [102] a stop criterion is proposed, which stops the algorithm when, in ten consecutive iterations, the difference of the squared errors, between iterations $i$ and $i + 1$, does not exceed a predefined threshold.

The work presented in [103] proposes an optimization by integrating the core (classification) of the $K$-means algorithm and multiple kernel learning using support-vector machines (LS-SVM). By using the Rayleigh coefficient, it optimizes the separation among each group. This algorithm reaches local convergence by obtaining the maximal separation among each of the centroids.

## 4. Trends

Preceding sections include articles published from the origins of the algorithm up to 2016. This section includes three types of articles: recently published articles on important improvements to $K$-means, articles that propose improvements to the algorithm implementation using parallel and distributed computing, and articles for new applications of the algorithm.

Regarding improvements to the steps of $K$-means, several of recently published articles are summarized next. In [104] two algorithm improvements are proposed: one deals with the *outliers* and *noise data* problems, and the other deals with the selection of initial centroids. In [105] two problems are dealt with: the selection of initial centroids and the determination of the number $k$ of clusters. In [106] a new measure of distance or similarity between objects is proposed. In [107] an

improvement to the work in [69] is proposed, by defining a new criterion for reducing the processing time of the assignment of objects to clusters. This approach is particularly useful for large instances. In [108] a new stop criterion is proposed, which reduces the number of iterations. In [109] an improvement is proposed for the convergence step of the algorithm aimed at solving large instances. The improvement consists of a new criterion that balances processing time and the quality of the solution. The main idea is to stop the algorithm when the number of objects that change membership is smaller than a threshold.

Recently, in the specialized literature, the parallelization of *K*-means has been proposed by using MapReduce paradigm [110, 111], which makes possible to process efficiently large instances. In [112] a method is proposed for parallelizing the *K-means++* algorithm [45], which has shown good results for obtaining the initial centroids.

In recent years, a trend has been observed for modifying *K*-means oriented to new applications, in particular, its application to natural language and text processing. In this regard, one of the remarkable works is presented in [113], in which a modification to *K*-means is proposed for grouping bibliographic citations. Later, in [114] an improvement is proposed to the algorithm in [113], in order to solve recommendation problems. In [115, 116] an improvement to *K*-means is proposed for the field of natural language.

## 5. Conclusions

This chapter presents three aspects of the *K*-means algorithm: (a) the works that originated the family of *K*-means algorithms, (b) a systematic review on the algorithm improvements up to 2016, and (c) some of the most recent publications that describe the prospective uses of the algorithm.

Regarding the origin of *K*-means, it is worth mentioning that it is not only an algorithm but a family of algorithms with the same purpose, which were developed independently in the decades of the 1950s and 1960s.

The systematic review process involved accessing four large databases, from which 1125 documents were retrieved. After applying inclusion and exclusion criteria, 79 documents remained.

Next, we will mention the most important observations organized by subjects:

1. Initialization. Of the four steps of the algorithm, initialization is the step on which the largest number of investigations has focused. The reason for this interest is that the algorithm is highly sensitive to the initial positions of the centroids. Some of the most cited publications are [45, 59].

2. Classification. Most of the works related to this step aim at reducing the number of calculations of object-centroid distances by applying heuristic methods. Some of the most cited works are [73, 82]. Some promising and recently published articles are [72, 107].

3. Centroid calculation. No documents were retrieved related to this step.

4. Convergence. It is remarkable that for this step the number of articles is very small. However, some articles recently published present very promising results by reducing the algorithm complexity without decreasing significantly the solution quality.

Regarding the most recent publications on *K*-means, improvements have been proposed for solving large instances, as well as parallel and distributed implementations and applications of *K*-means to new fields such as natural language and text processing, among others.

Finally, because the nature of data clustering is exploratory and applicable to data from many disciplines, it is foreseeable that *K*-means will continue to be widely used, mainly because of the easiness for interpreting its results.

## Author details

Joaquín Pérez-Ortega[1*], Nelva Nely Almanza-Ortega[1], Andrea Vega-Villalobos[1], Rodolfo Pazos-Rangel[2], Crispín Zavala-Díaz[3] and Alicia Martínez-Rebollar[1]

1 National Institute of Technology of Mexico/CENIDET, Cuernavaca, Mexico

2 National Institute of Technology of Mexico/ITCM, Ciudad Madero, Mexico

3 Autonomous University of the State of Morelos, Cuernavaca, Mexico

*Address all correspondence to: jpo_cenidet@yahoo.com.mx

## References

[1] Kambalta K, Kollias G, Kumar V, Grama A. Trends in big data analytics. Journal of Parallel and Distributed Computing. 2014;**74**(7):2561-2573. DOI: 10.1016/j.jpdc.2014.01.003

[2] Laney D. 3D Data Management: Controlling Data Volume, Velocity And Variety [Internet]. 2001. Available from: http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf [Accessed: August 24, 2017]

[3] Raykov YP, Boukouvalas A, Baig F, Little MA. What to do when k-means clustering fails: A simple yet principled alternative algorithm. PLoS One. 2016; **11**(9):e0162259. DOI: 10.1371/journal.pone.0162259

[4] Chun-wei T, Ching-Feng L, Han-Chieh C, Vasilakos AV. Big data analytics: A survey. Journal of Big Data. 2015;**2**(1):21. DOI: 10.1186/s40537-015-0030-3

[5] Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, et al. Top 10 algorithms in data mining. Knowledge and Information Systems. 2008;**14**:1-37. DOI: 10.1007/s10115-007-0114-2

[6] MacQueen J. Some methods for classification and analysis of multivariate observations. In: Proc. 5th Berkeley Symp. Math. Statistics and Probability. Vol. 1. 1967. pp. 281-297

[7] Steinley D. K-means clustering: A half-century synthesis. The British Journal of Mathematical and Statistical Psychology. 2006;**59**:1-34

[8] Bock H-H. Origins and extensions of the K-means algorithm in cluster analysis. Journal Electronique d'Histoire des Probabilités et de la Statistique. 2008;**4**(2):1-18

[9] Jain AK. Data clustering: 50 years beyond k-means. Pattern Recognition Letters. 2010;**31**:651-666. DOI: 10.1016/j.patrec.2009.09.011

[10] Blomer J, Lammersen C, Schmidt M, Sohler C. Theoretical analysis of the k-means algorithm—A survey. Algorithm Engineering. 2016;**9220**:81-116. DOI: 10.1007/978-3-319-49487-6_3

[11] Steinhaus H. Sur la division des corps matériels en parties. Bulletin de l'Académie Polonaise des Sciences. Classe 3. 1956;**12**:801-804

[12] Lloyd SP. Least squares quantization in PCM. In: Bell Telephone Labs Memorandum, Murray Hill NJ. Reprinted in: IEEE Trans. Information Theory IT-28. Vol. 2. 1982. pp. 129-137

[13] Diday E. Une nouvelle méthode de classification automatique et reconnaissance des formes: la méthode des nuées dynamiques. Revue de Statistique Appliquée. 1971;**XIX**:19-33

[14] Diday E. The dynamic clusters method in nonhierarchical clustering. International Journal of Computing and Information Sciences. 1973;**2**:61-88

[15] Diday E, Govaert G. Classification avec distance adaptive. Comptes Rendus de l'Académie des Sciences. 1974;**278A**: 993-995

[16] Bock H-H. Automatische Klassifikation: Theoretische und Praktische Methoden zur Strukturierung von Daten (Clusteranalyse). Göttingen: Vandenhoeck & Ruprecht; 1974

[17] Anderberg MR. Cluster Analysis for Applications. New York: Academic Press; 1973

[18] Späth H. Cluster Analyse Algorithmen zur Objektklassifizierung

und Datenreduktion. Oldenbourg Verlag, München K Wien. English Translation: Cluster Analysis Algorithms for Data Reduction and Classification of Objects. Chichester, UK: Ellis Horwood Ltd; 1980

[19] Jancey RC. Multidimensional group analysis. Australian Journal of Botany. 1966;**14**:127-130

[20] Forgy EW. Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. In: Biometric Society Meeting, Riverside, California. Abstract in Biometrics. Vol. 21. 1965. p. 768

[21] Pearson K. On the coefficient of racial likeness. Biometvika. 1926;**18**: 105-117

[22] Rao CR. The use of multiple measurements in problems of biological classification. Journal of the Royal Statistical Society: Series B: Methodological. 1948;**10**(2):159-203

[23] Sokal RR. Distance as a measure of taxonomic similarity. Systematic Zoology. 1961;**10**(2):70-79

[24] Selim SZ, Ismail MA. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1984;**1**:81-87

[25] Aloise D, Deshpande A, Hansen P, Popat P. NP-hardness of Euclidean sum-of-squares clustering. Machine Learning. 2009;**75**(2):245-248

[26] Mahajan M, Nimbhorkar P, Varadarajan K. The planar k-means problem is NP-hard. Theoretical Computer Science. 2012;**442**:13-21

[27] Tou JT, Gonzalez RC. Pattern Recognition Principles. USA: Addison-Wesley; 1974

[28] Katsavounidis I, Kou J, Zhang Z. A new initialization technique for generalized Lloyd iteration. IEEE Signal Processing Letters. 1994;**1**:144-146

[29] Moh'd B, Roberts SA. New methods for the initialization of clusters. Pattern Recognition Letters. 1996;**17**:451-455. DOI: 10.1016/0167-8655(95)00119-0

[30] Bradley PS, Fayyad UM. Refining initial points for K-means clustering. In: Proceeding of the 15th International Conference on Machine Learning (ICML98). San Francisco: Morgan Kaufmann; 1998. pp. 91-99

[31] Pelleg D, Moore A. X-means: Extending K-means with efficient estimation of the number of clusters. In: Proceedings of the Seventieth International Conference on Machine Learning (ICML); July 2000; Palo Alto, CA

[32] Su T, Dy JG. In search of deterministic methods for initializing k-means and gaussian mixture clustering. Intelligent Data Analysis. 2007;**11**: 319-338. DOI: 10.3233/IDA-2007-11402

[33] Zalik KR. An efficient k'-means clustering algorithm. Pattern Recognition Letters. 2008;**29**:1385-1391. DOI: 10.1016/j.patrec.2008.02.014

[34] Nazeer KA, Sebastian MP. Improving the accuracy and efficiency of the k-means clustering algorithm. In: Proceedings of the World Congress on Engineering; 1-3 July 2009; London, UK. 2009. pp. 1-3

[35] Lee D, Baek S, Sung K. Modified k-means algorithm for vector quantizer design. IEEE Signal Processing Letters. 1997;**4**(1):2-4. DOI: 10.1109/97.551685

[36] Ahmed AH, Ashour W. An initialization method for the k-means algorithm using RNN and coupling degree. International Journal of Computer Applications. 2011;**25**:1-6

[37] Nazeer KA, Kumar SD, Sebastian MP. Enhancing the k-means clustering algorithm by using a O(n logn) heuristic method for finding better initial centroids. In: International Conference on Emerging Applications of Information Technology; 19-20 February 2011; Kolkata, India. 2011. pp. 261-264

[38] Salaman R, Kecman V, Li Q, Strack R, Test E. Two stage clustering with k-means algorithm. Recent Trends in Wireless and Mobile Networks. 2011; **162**:110-122. DOI: 10.1007/978-3-642-21937-5_11

[39] Celebi ME. Improving the performance of k-means for color quantization. Image and Vision Computing. 2011;**29**:260-271. DOI: 10.1016/j.imavis.2010.10.002

[40] Zhanguo X, Shiyu C, Wentao Z. An improved semi-supervised clustering algorithm based on initial center points. Journal of Convergence Information Technology. 2012;**7**:317-324

[41] Yuan F, Meng ZH, Zhang HX, Dong CR. A new algorithm to get the initial centroids. In: Proceeding of 2004 International Conference on Machine Learning and Cybernetics; 26-29 August 2004; Shanghai. China: IEEE; 2004. pp. 1191-1193

[42] Khan SS, Ahmad A. Cluster center initialization algorithm for K-means clustering. Pattern Recognition Letters. 2004;**25**:1293-1302. DOI: 10.1016/j.patrec.2004.04.007

[43] Pham DT, Dimov SS, Nguyen CD. Selection of k in k-means clustering. Proceedings of the Institution of Mechanical Engineers-Part C: Journal of Mechanical Engineering Science. 2005; **219**:103-119. DOI: 10.1243/095440605X8298

[44] Al-Daoud. A new algorithm for cluster initialization. In: The Second World of Enformatika Conference. 2005

[45] Arthur D, Vassilvitskii S. K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. 2007. pp. 1027-1035

[46] Zhang Z, Zhang J, Xue H. Improved k-means clustering algorithm. Signal & Image Processing. 2008;**4**:169-172

[47] Vanisri D, Loganathan D. An efficient fuzzy clustering algorithm based on modified k-means. International Journal of Engineering, Science and Technology. 2010;**2**: 5949-5958

[48] Yedla M, Pathakota SR, Srinivasa TM. Enhancing k-means clustering algorithm with improved initial center. International Journal of Computer Science & Information Technology. 2010;**2**:121-125

[49] Xie J, Jiang S. A simple and fast algorithm for global k-means clustering. In: 2010 Second International Workshop on Education Technology and Computer Science; 6-7 March 2010. Wuhan, China: IEEE; 2010. pp. 36-40

[50] Eltibi M, Ashour W. Initializing k-means algorithm using statistical information. International Journal of Computer Applications. 2011;**29**:51-55

[51] Li CS. Cluster center initialization method for k-means algorithm over data sets with two clusters. Procedia Engineering. 2011;**24**:324-328. DOI: 10.1016/j.proeng.2011.11.2650

[52] Xie J, Jiang S, Xie W, Gao X. An efficient global k-means clustering algorithm. Journal of Computers. 2011; **6**:271-279

[53] Elagha M, Ashour WM. Efficient and fast initialization algorithm for k-means clustering. International Journal of Intelligent Systems and Applications. 2012;**4**:21-31

[54] Zhang Y, Cheng E. An optimized method for selection of the initial centers of K-means clustering. Lecture Notes in Computer Science. 2013;**8032**: 149-156. DOI: 10.1007/978-3-642-39515-4_13

[55] Abudaker M, Ashour W. Efficient data clustering algorithms: Improvements over K-means. International Journal of Intelligent Systems and Applications. 2013;**3**:37-49. DOI: 10.5815/ijisa.2013.03.04

[56] Alsabti K, Ranka S, Singh V. An efficient k-means clustering algorithm. In: Electrical Engineering and Computer Science. 1997. p. 43

[57] Li M, Ng MK, Cheung YM, Huang JZ. Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters. IEEE Transactions on Knowledge and Data Engineering. 2008;**20**(11):1519-1534. DOI: 10.1109/TKDE.2008.88

[58] Laszlo M, Mukherjee S. A genetic algorithm using hyper-quadtrees for low dimensional k-means clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2006;**28**:533-543. DOI: 10.1109/TPAMI.2006.66

[59] Redmond SJ, Heneghan C. A method for initializing the K-means clustering algorithm using kd-trees. Pattern Recognition Letters. 2007;**28**: 965-973. DOI: 10.1016/j.patrec.2007.01.001

[60] Babu GP, Murty MN. A near-optimal initial seed value selection in k-means algorithm using a genetic algorithm. Pattern Recognition Letters. 1993;**14**:763-769. DOI: 10.1016/0167-8655(93)90058-L

[61] Steinley D. Local optima in K-means clustering: What you don't know may hurt you. Psychological Methods. 2003; **8**(3):294-304. DOI: 10.1037/1082-989X.8.3.294

[62] Tian J, Lin Z, Suqin Z, Lu L. Improvement and parallelism of k-means clustering algorithm. Tsinghua Science and Technology. 2005;**10**: 277-281. DOI: 10.1016/S1007-0214(05)70069-9

[63] Hand JD, Krzanowski WJ. Optimising k-means clustering results with standard software packages. Computational Statistics and Data Analysis. 2005;**49**:969-973. DOI: 10.1016/j.csda.2004.06.017

[64] Fahim AM, Salem AM, Torkey FA, Ramadan MA. An efficient enhanced k-means clustering algorithm. Journal of Zhejiang University. 2006;**7**:1626-1633. DOI: 10.1631/jzus.2006.A1626

[65] Tsai C, Yang C, Chiang M. A time efficient pattern reduction algorithm for k-means based clustering. In: IEEE International Conference on Systems, Man and Cybernetics; 1-10 October 2007. Montreal, Quebec, Canada: IEEE; 2008. pp. 504-509

[66] Chiang M, Tsai C, Yang C. A time-efficient pattern reduction algorithm for k-means clustering. Information Sciences. 2011;**181**:716-731. DOI: 10.1016/j.ins.2010.10.008

[67] Singh RV, Bhatia MP. Data clustering with modified k-means algorithm. In: 2011 International Conference on Recent Trends in Information Technology (ICRTIT). 3-5 June 2011. Chennai, India: IEEE; 2011. pp. 717-721

[68] Lee S, Lin J. An accelerated k-means clustering algorithm using selection and erasure rules. Journal of Zhejiang University. Science. 2012;**13**:761-768. DOI: 10.1631/jzus.C1200078

[69] Perez J, Pires CE, Balby L, Mexicano A, Hidalgo M. Early classification: A new heuristic to improve the classification step of k-means. Journal of

Information and Data Management. 2013;**4**:94-103

[70] Yusoff IA, Isa NAM, Hasikin K. Automated two-dimensional k-means clustering algorithm for unsupervised image segmentation. Computers and Electrical Engineering. 2013;**39**:907-917. DOI: 10.1016/j.compeleceng.2012.11.013

[71] Mexicano A, Rodriguez R, Cervantes S, Ponce R, Bernal W. Fast means: Enhancing the k-means algorithm by accelerating its early classification version. AIP Conference Proceedings. 2015;**1648**:820004-1-820004-4. DOI: 10.1063/1.4913023

[72] Perez J, Pazos R, Hidalgo M, Almanza N, Diaz-Parra O, Santaolaya R, et al. An improvement to the k-means algorithm oriented to big data. AIP Conference Proceedings. 2015;1648: 820002-1-820002-4. DOI: 10.1063/1.4913021

[73] Lai JZC, Huang T, Liaw Y. A fast k-means clustering algorithm using cluster center displacement. Pattern Recognition. 2009;**42**:2551-2556. DOI: 10.1016/j.patcog.2009.02.014

[74] Lai JZC, Huang T. Fast global k-means clustering using cluster membership and inequality. Pattern Recognition. 2010;**43**:1954-1963. DOI: 10.1016/j.patcog.2009.11.021

[75] Al-Zoubi M, Hudaib A, Hammo B. New efficient strategy to accelerate k-means clustering algorithm. American Journal of Applied Sciences. 2008;**5**: 1247-1250

[76] Chang C, Lai JZC, Jeng M. A fuzzy k-means clustering algorithm using cluster center displacement. Journal of Information Science and Engineering. 2011;**27**:995-1009

[77] Bagirov AM, Ugon J, Webb D. Fast modified global k-means algorithm for incremental cluster construction. Pattern Recognition. 2011;**44**:866-876. DOI: 10.1016/j.patcog.2010.10.018

[78] Osamor VC, Adebiyi EF, Oyelade JO, Doumbia S. Reducing the time requirement of k-means algorithm. PLoS One. 2012;**7**:1-10. DOI: 10.1371/journal.pone.0049946

[79] Perez J, Mexicano A, Santaolaya R, Hidalgo M, Moreno A, Pazos R. Improvement to the K-means algorithm through a heuristic based on a bee honeycomb structure. In: Fourth World Congress on Nature and Biologically Inspired Computing; 5-9 November 2012; Mexico. Mexico: IEEE; 2013. pp. 175-180

[80] Bai L, Liang J, Siu C, Dang C. Fast global k-means clustering based on local geometrical information. Information Sciences. 2013;**245**:168-180. DOI: 10.1016/j.ins.2013.05.023

[81] Phillips SJ. Acceleration of k-means and related clustering algorithms. Lecture Notes in Computer Science. 2002;**2409**:166-177. DOI: 10.1007/3-540-45643-0_13

[82] Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002;**24**:881-892

[83] Vrahatis MN, Boutsinas B, Alevizos P, Pavlides G. The new K-windows algorithm for improving the K-means clustering algorithm. Journal of Complexity. 2002;**18**:375-391. DOI: 10.1006/jcom.2001.0633

[84] Napoleon D, Lakshmi PG. An efficient k-means clustering algorithm for reducing time complexity using uniform distribution data points. In: Trendz in Information Sciences and Computing (TISC2010), 17-19

December 2010. Chennai, India: IEEE; 2011. pp. 42-45

[85] Lai JZC, Liaw Y. Improvement of the k-means clustering filtering algorithm. Pattern Recognition. 2008; **41**:3677-3681. DOI: 10.1016/j.patcog.2008.06.005

[86] Hamerly G, Drake J. Accelerating Lloyd's algorithm for k-means clustering. In: Cebeli M, editor. Partitional Clustering Algorithms. Springer: Cham; 2015. pp. 41-78. DOI: 10.1007/978-3-319-09259-1_2

[87] Wang J, Wang J, Ke Q, Zeng G, Shipeng L. Fast approximate k-means via cluster closures. In: Multimedia Data Mining and Analytics. 2015. Springer International Publishing AG. Cham. pp. 373-395. DOI: 10.1007/978-3-319-14998-1_17

[88] Cofarelli C, Nieddu L, Seref O, Pardalos PM. K-T.R.A.C.E: A kernel k-means procedure for classification. Computers and Operations Research. 2007;**34**:3154-3161. DOI: 10.1016/j.cor.2005.11.023

[89] Salaman R, Kecman V, Li Q, Strack R, Test E. Fast k-means algorithm clustering. International Journal of Computer Networks and Communications. 2011;**3**. DOI: 10.5121/ijcnc.2011.3402

[90] Kaur N, Sahiwal JK, Kaur N. Efficient k-means clustering algorithm using ranking method in data mining. International Journal of Advanced Research in Computer Engineering & Technology. 2012;**1**: 85-91

[91] Scitovski R, Sabo K. Analysis of the K-means algorithm in the case of data points occurring on the border of two or more clusters. Knowledge-Based Systems. 2014;**57**:1-7. DOI: 10.1016/j.knosys.2013.11.010

[92] Xu L, Hu Q, Hung E, Szeto C. A heuristic approach to effective an efficient clustering on uncertain objects. Knowledge-Based Systems. 2014;**66**: 112-125. DOI: 10.1016/j.knosys.2014.04.027

[93] Elkan C. Using the triangle inequality to accelerate k-means. In: Proceedings of the Twentieth International Conference on Machine Learning (ICML-2013); Washington, DC. 2003

[94] Fahim AM, Saake G, Salem AM, Torkey FA, Ramadan MA. K-means for spherical clusters with large variance in sizes. International Journal of Scientific Research and Innovative Technology. 2008;**2**:2923-2928

[95] Sarma TH, Viswanath P, Reddy BE. A hybrid approach to speed-up the k-means clustering method. International Journal of Machine Learning and Cybernetics. 2013;**4**(2):107-117. DOI: 10.1007/s13042-012-0079-7

[96] Pakhira MK. A modified k-means algorithm to avoid empty clusters. International Journal of Recent Trends in Engineering. 2009;**1**:220-226

[97] Perez J, Pazos R, Cruz L, Reyes G, Basave R, Fraire H. Improving the efficiency and efficacy of the K-means clustering algorithm through a new convergence condition. In: International Conference on Computational Science and its Applications (ICCSA 2007). 2007

[98] Samma A, Salam R. Adaptation of k-means algorithm for image segmentation. World Academy of Science, Engineering and Technology. 2009;**50**:58-62

[99] Bottou L, Bengio Y. Convergence properties of the K-means algorithms. In: Advances in Neural Information Processing Systems 7, Tesauro G,

Touretzky D, editors. Cambridge, MA: The MIT Press; 1995:586-592

[100] Pham DT, Dimov SS, Nguyen CD. An incremental K-means algorithm. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science. 2004; **218**:783-795. DOI: 10.1243/0954406041319509

[101] Likas A, Vlassis N, Verbeek JJ. The global K-means clustering algorithm. Pattern Recognition. 2003;**36**:451-461. DOI: 10.1016/S0031-3203(02)00060-2

[102] Lam YK, Tsang PWM. eXploratory K-means: A new simple and efficient algorithm for gene clustering. Applied Soft Computing. 2012;**12**:1149-1157. DOI: 10.1016/j.asoc.2011.11.008

[103] Yu S, Tranchevent L, Liu X, Glanzel W, Suykens JAK, Moor B, et al. Optimized data fusion for kernel k-means clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012;**34**:1031-1039. DOI: 10.1109/TPAMI.2011.255

[104] Yu S, Chu S, Wang C, Chan Y, Chang T. Two improved k-means algorithms. Applied Soft Computing. 2018;**68**:747-755. DOI: 10.1016/j.asoc.2017.08.032

[105] Zhang G, Zhang C, Zhang H. Improved K-means algorithm based on density canopy. Knowledge-Based Systems. 2018;**145**:289-297. DOI: 10.1016/j.knosys.2018.01.031

[106] Wang R, Li H, Chen M, Dai Z, Zhu M. MIC-KMeans: A maximum information coefficient based high-dimensional clustering algorithm. Advances in Intelligent Systems and Computing. 2019;**764**:208-218. DOI: 10.1007/978-3-319-91189-2_21

[107] Perez J, Almanza N, Ruiz J, Pazos R, Saenz S, Lelis J, et al. A-means: Improving the cluster assignment phase of k-means for big data. International Journal of Combinatorial Optimization Problems and Informatics. 2018;**9**(2):3-10

[108] Mexicano A, Rodriguez R, Cervantes S, Montes P, Jimenez M, Almanza N, et al. The early stop heuristic: A new convergence criterion for k-means. In: AIP Conference Proceedings 2016. AIP Publishing; 2016. p. 310003

[109] Perez J, Almanza N, Romero D. Balancing effort and benefit of K-means clustering algorithms in big data realms. PLoS One. 2018;**13**(9):1-19. DOI: 10.1371/journal.pone.0201874

[110] Zhao W, Ma H, He Q. Parallel k-means clustering based on MapReduce. Lecture Notes in Computer Science. 2009;**5931**:674-679. DOI: 10.1007/978-3-642-10665-1_71

[111] Moertini VS, Venica L. Enhancing parallel k-means using MapReduce for discovering knowledge from big data. In: IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA); 5-7 July 2016. Chengdu, China: IEEE; 2016. pp. 81-87

[112] Bahmani B, Moseley B, Vattani A, Kumar R, Vassilvitskii S. Scalable K-means ++. Proceedings of the VLDB Endowment. 2012;**5**:622-633. DOI: 10.14778/2180912.2180915

[113] McCallum A, Nigam K, Ungar L. Efficient clustering on high-dimensional data sets with application of reference matching. In: International Conference on Knowledge Discovery and Data Mining; 20-23 August 2000. Boston, Massachusetts: ACM; 2000. pp. 169-178

[114] Li J, Zhang K, Yang X, Wei P, Wang J, Mitra K, et al. Category preferred canopy-k-means based collaborative filtering algorithm. Future Generation Computer Systems. 2019;**93**:1046-1054. DOI: 10.1016/j.future.2018.04.025

[115] Hussain S, Haris M. A k-means based co-clustering (KCC) algorithm for sparse, high dimensional data. Expert Systems with Applications. 2019;**118**: 20-34. DOI: 10.1016/j.eswa.2018.09.006

[116] Naeem S, Wumaier A. Study and implementing K-mean clustering algorithm on English text and techniques to find the optimal value of K. International Journal of Computer Applications. 2018;**182**(31):7-14. DOI: 10.5120/ijca2018918234