Early Classification: A New Heuristic to Improve the Classification Step of K-Means

Joaquín Pérez[1], Carlos Eduardo Pires[2], Leandro Balby[2], Adriana Mexicano[1], Miguel Ángel Hidalgo[1]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)
[2] Universidade Federal de Campina Grande (UFCG)
jpo_cenidet@yahoo.com.mx, cesp@dsc.ufcg.edu.br, lbmarinho@dsc.ufcg.edu.br

Abstract. Cluster analysis is the study of algorithms and techniques for grouping objects according to their intrinsic characteristics and similarity. A widely studied and popular clustering algorithm is K-Means, which is characterized by its ease of implementation and high computational cost. Although various performance improvements have been proposed for K-Means, the algorithm is still considered an expensive alternative for clustering large scale datasets. This work proposes a new heuristic for reducing the number of calculations needed in the classification step of K-Means, without significant loss of quality reduction, by using statistical information about the displacement of centroids at each iteration. Our heuristic, called Early Classification (EC for short), identifies and excludes from future calculations those objects that, according to an equidistance threshold, have low likelihood of cluster change in subsequent iterations. To validate our proposal, a set of experiments is performed on synthetic and real-world datasets from the UCI Machine Learning repository. In addition, we compared our heuristic against two other state-of-the-art variations of K-Means. The Wilcoxon signed-rank test was used for calculating the statistical significance. The results are promising since the execution time of K-Means was reduced up to 82.49%, with a quality reduction of only 3.31%. Moreover, as the experiments will show, the superiority of our method is even more evident on large datasets.

Categories and Subject Descriptors: H.2 [**Database Management**]: Miscellaneous; H.3 [**Information Storage and Retrieval**]: Miscellaneous; I.7 [**Document and Text Processing**]: Miscellaneous

Keywords: Clustering, K-Means, Performance Optimization, Unsupervised Learning

1. INTRODUCTION

Clustering is a widely used and flexible method for grouping objects into clusters [Myatt and Johnson 2009]. The objects within a cluster are supposed to have high similarity to one another and high dissimilarity to objects in other clusters. Clustering has been successfully used in a wide variety of scientific and commercial applications, including medical diagnosis, insurance underwriting, financial portfolio management, organization of search results, marketing, pattern recognition, data analysis, and image processing [Jiawei and Micheline 2006].

Several clustering algorithms have been proposed in the literature [Ankerst et al. 1999; Dempster et al. 1977; Ester et al. 1996; Kaufman and Rousseeuw 1987]. In general, these algorithms partition the set of objects into a given number of clusters according to an optimization criterion. One of the most popular and widely studied clustering algorithms is K-Means [MacQueen 1967], also known as Lloyd's algorithm [Lloyd 1982]. The main steps of the standard K-Means are enumerated as follows[1]:

[1] A detailed description of the K-Means algorithm can be found in [MacQueen 1967]

We express our gratitude to the Consejo Nacional de Ciencia y Tecnología (CONACyT) and the Universidade Federal de Campina Grande (UFCG) for the facilities provided to realize this research work. Also, we would like to thank Alejandra Moreno Hernandez (student of CENIDET) for her assistance in the algorithm coding and experimentation.

1. *Initialization.* Consists in defining the objects to be partitioned, the number of clusters, and a centroid for each cluster. Several methods for defining the initial centroids have been developed [Agha and Ashour 2012; Zhanguo et al. 2012], although randomly selecting the centroids is still the most widely used approach;
2. *Classification.* For each object, its distance to the centroids is calculated, the closest centroid is determined, and the object is assigned to the cluster associated with this centroid;
3. *Centroid calculation.* The centroid is recalculated for each cluster generated in the previous step;
4. *Stopping criteria.* Several convergence conditions have been used, such as: stopping when reaching a given number of iterations, when there is no exchange of objects among clusters, or when the difference of the centroids at any two consecutive iterations is smaller than a given threshold. If the convergence condition is not satisfied, then steps 2, 3, and 4 are repeated.

Clearly, a factor that greatly affects the computational cost of K-Means is the number of iterations that the algorithm needs to carry out since, for each iteration, it calculates the distance of each object to each centroid. In this work, we propose a new heuristic, henceforth called *Early Classification* (EC), to reduce the number of calculations in the classification step of K-Means. The main idea is to use statistical information about the displacement of centroids by calculating the average of the two largest displacements of centroids at each iteration. This heuristic introduces the concepts of *equidistance index* and *equidistance threshold*, with the purpose of identifying and excluding from future calculations those objects that, according to the equidistance threshold, have low likelihood of cluster change in subsequent iterations. A formal definition of low/high probability of cluster change for an object i in iteration j is provided in Section 4.3. In order to evaluate the proposed heuristic, a set of experiments was performed using synthetic data and the *Iris*, *Concrete compressive strength*, and the *Skin segmentation* datasets, available at the UCI Machine Learning repository. The results show that the execution time of K-Means was reduced up to 82.49% with a quality reduction of only 3.31%.

This work is organized as follows: Section 2 presents the related work. Section 3 presents a motivating example. Section 4 describes the heuristic proposed to improve the classification step of K-Means. Section 5 presents the experimental results obtained by applying the proposed heuristic. The results are compared against related work. Finally, Section 6 concludes the article and points out directions for future work.

2. RELATED WORK

Several improvements have been proposed to minimize the number of calculations in the classification step of the K-Means algorithm. Lai and Liaw [2008] proposed an improvement for the Filtering Algorithm (FA), a variation of the K-Means algorithm [Kanungo et al. 2002]. FA considers that objects are stored in a kd-tree, i.e., a binary tree that divides the objects into cubes using perpendicular hyperplanes. Each node in the tree is associated with a set of data points called a cell. At each iteration, FA determines the nearest centroids of every cell by calculating all object centroid distances, and verifies whether each member of the centroid set should be pruned for each internal node. The improvement consists in identifying the centroids that, between the current and the previous iteration, were displaced. This allows the algorithm to determine the nearest centroid of the cell and check whether each centroid should be pruned using only the centroids that were displaced, eliminating the calculations involving objects in clusters in which the centroid was not displaced. Results show that the improvement reduces the execution time up to 33.6% in comparison to the FA algorithm.

The Pattern Reduction heuristic compresses and removes objects that are closed to a centroid [Tsai et al. 2007]. In this heuristic, an object is considered close to the centroid if the distance to its nearest centroid is smaller than the average distance of all the objects in the same cluster to their centroid. The heuristic is repeatedly applied after the second iteration until 80% of the objects are removed. Results show that this improvement reduces the execution time significantly, specially for high dimensional datasets. In the rest of the document we refer to this heuristic as K-Means+PR.

The improvement proposed by Fahim et al. [2006] consists in calculating and storing the shortest distance between each object and its nearest centroid at each iteration. For each object, the previous distance to the current one is compared. If the previous distance is less than or equal to the current one, the object remains in the cluster and there is no need to compute the distances to the other $k-1$ centroids; otherwise, it is necessary to determine the distance between the object and all cluster centroids as well as to identify the new nearest cluster. Results show that this improvement reduces the execution time without significantly decreasing cluster quality. Fahim called this heuristic Enhanced K-Means. In the rest of the article we refer to this as K-Means+E.

All the aforementioned work use information about the centroids displacements in order to reduce the complexity of the classification step of K-Means. However, none of them take into account the likelihood of cluster change for the objects that are in the borders of the clusters causing an early but less accurate classification than the one reached by our heuristic. For example Tsai et al. [2007] discard objects according to the current and past object centroid distances. However, the fact that the distance between an object and its centroid in the current iteration is less than the distance in the past iteration does not guarantee that the object remains close to the same centroid. On the other hand, besides using more calculations than our heuristic for discarding objects, they assume that only the objects which are far from their centroids can change in the following iterations [Tsai et al. 2007].

3. MOTIVATING EXAMPLE

Fig. 1 illustrates a clustering example with a dataset containing 36 uniformly distributed objects in 3 clusters. The top refers to the execution of the standard K-Means algorithm while the bottom refers to the execution of K-Means using the Early Classification heuristic (improved K-Means). The objects are represented by small dots and clustered in four iterations. At each iteration, the initial position of each centroid is represented by a large white dot, while the new position of the centroid (i.e., the position in the following iteration) is represented by a large grey dot. The filling of the objects is related to the filling of their nearest centroid, i.e., the dots with horizontal lines form a cluster whose centroid is represented by the large dot with horizontal lines. The dashed lines are equidistant to two centroids and represent the borders between the clusters. The shaded area refers to the borders separating the objects with low likelihood of cluster change from the objects with high likelihood of cluster change. We assume that the objects with low likelihood of cluster change are (i) near their centroid, (ii) not equidistant to their two nearest centroids, and (iii) not affected by the centroids' displacements. In Fig. 1 the shaded area contains the objects with high likelihood of cluster change.

The bottom of the Fig. 1 shows that during the execution of K-Means it is possible to identify and discard the objects with low likelihood of cluster change. For example, in Fig. 1e the objects in the white area have a low likelihood of cluster change, this is because the centroid displacements in the first iteration are large and the number of objects that can change cluster is high. In Figures 1f and 1h we can notice that the size of the border decreases since the displacements of the centroids are minimized at each iteration. Particularly in the case of Fig. 1f, it is possible to observe that 28 objects can be discarded from the calculations in the third iteration of the improved K-Means. Fig. 1g shows that for the third iteration the number of objects can be reduced to 31 leaving only 5 for the fourth iteration. Although both algorithms have the same clustering result (see Figures 1d and 1h), the improved algorithm allows us to minimize the number of calculations in the classification step of K-Means. In the following section, we present the proposed heuristic to improve the performance of the K-Means algorithm.

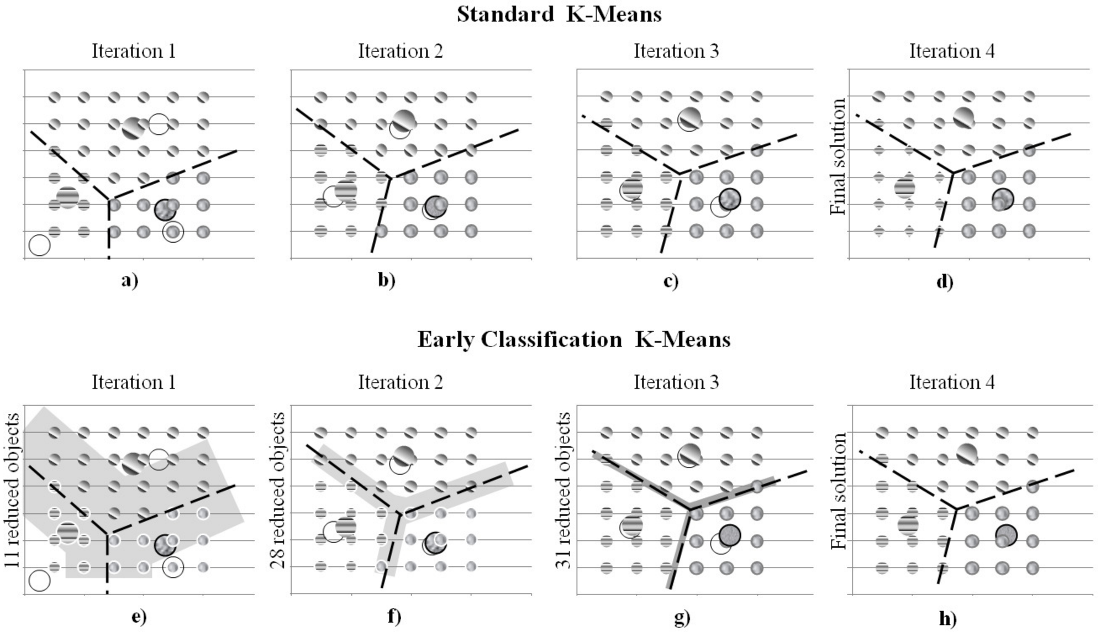

Fig. 1. Execution of the standard K-Means and the improved K-means using a dataset with 36 uniformly distributed objects

4. THE EARLY CLASSIFICATION HEURISTIC

The main goal of EC is to reduce the number of computations needed in the classification step of K-Means without significant loss of quality reduction. The reduction is performed by selecting objects that have been assigned to clusters in one iteration and are unlikely to change to other clusters in subsequent iterations. These objects are then marked and excluded from future calculations. To perform the selection process, we introduce two concepts named *equidistance index* and *equidistance threshold*, which are described in the following subsections.

The EC heuristic arose after observing the behavior of K-Means when clustering synthetic data with uniform distribution and different sample sizes. Some interesting observations are the following:

a) Objects close to the centroids are unlikely to change cluster in subsequent iterations;
b) Objects equidistant from their two nearest centroids can be assigned to any of the two clusters represented by these centroids;
c) Objects quasi equidistant from their two closest centroids have a high likelihood of cluster change in subsequent iterations;
d) A decisive factor for objects changing cluster is the displacement of the centroids at each iteration;
e) In general, at each iteration, centroids displacements decreases;
f) During the centroid displacement across different iterations, approximately half of the objects will be at a shorter distance from the new centroid position and the other half at a longer distance. The more distant objects are to the centroids' new position, the more likely is for the object to change cluster in subsequent iterations;
g) In one iteration, the centroids may or may not have suffered displacement. The amount of displacement between centroids in distinct iterations may vary;

h) Centroids can move in different directions across different iterations.

4.1 Equidistance Index

The equidistance index expresses the difference of the distances of an object i to its two closest centroids μ_1 and μ_2. Let $I = \{i_1, ..., i_n\}$ be a set of objects in a m-dimensional space to be partitioned, $C = \{C_1, \dots, C_k\}$ be the set of partitions of I into k sets $(2 \leq k < n)$. For each iteration of the classification step, the standard K-Means algorithm calculates $||i_p - \mu_l||^2$, being $||.||$ the ℓ^2 norm, for $p = 1, ..., n$ and $l = 1, ..., k$; where μ_l is the centroid of objects in $C_l \in C$, which represents the higher computational cost of the algorithm in terms of the number of calculations.

The equidistance index α_i is defined as follows: given an object i and its two nearest centroids μ_1 and μ_2, $\alpha_i = abs(||i - \mu_1||^2 - ||i - \mu_2||^2)$. The lower bound of α_i is 0, and the upper bound is $||\mu_1 - \mu_2||^2$. The lower bound indicates that object i is located at an equidistant position to the centroids μ_1 and μ_2, whereas the upper bound indicates that the object i is located at the same position of the centroid μ_1 or μ_2. In Fig. 2 the dashed line indicates the equidistant points to centroids μ_1 and μ_2; Fig. 2a shows that when the object i has a value of α_i close to 0, the object has a high likelihood of changing cluster in subsequent iterations. On the other hand, Fig. 2b shows that when the object i has a value of α_i that is close to its upper bound, there is a low likelihood that object i changes cluster in the following iterations.

4.2 Equidistance Threshold

The equidistance threshold β_j helps to identify the objects with high likelihood of cluster change. β_j is a reference value defined by the sum of the two largest displacements $\beta_j = m_1 + m_2$ of the centroids μ_x and μ_y in the iteration j $(j > 2)$; where $m_1 = ||\mu_{x,j-1} - \mu_{x,j}||^2$ and $m_2 = ||\mu_{y,j-1} - \mu_{y,j}||^2$ (see Fig. 3). The magnitude of the equidistance threshold varies between the last and the current iteration, since it is directly related to the centroid displacements. As we can see in Fig. 3, the center of the equidistance threshold β_j for an object i corresponds to the mean distance of the two nearest centroids μ_1 and μ_2.

4.3 Likelihood of Cluster Change

We say that an object i has high likelihood of cluster change if $\alpha_i \leq \beta_j$ (Fig. 3a), but has low likelihood of cluster change if $\alpha_i > \beta_j$ (Fig. 3b). Then, if the condition $\alpha_i > \beta_j$ is true and if μ_x is the nearest centroid of i, the object i can be early classified into the partition C_x at iteration j, and excluded from future calculations.

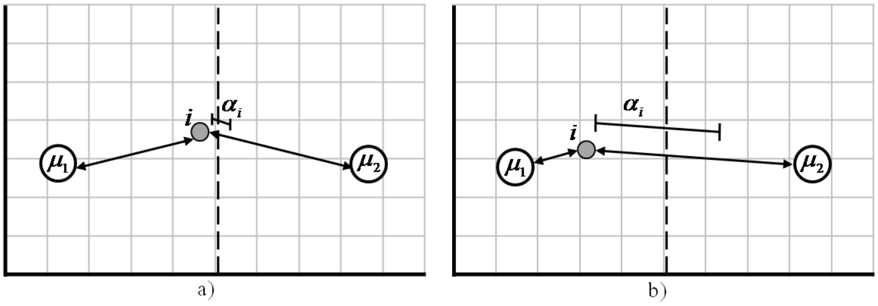

Fig. 2. Equidistance index: a) object i with high likelihood of cluster change, b) object i with low likelihood of cluster change

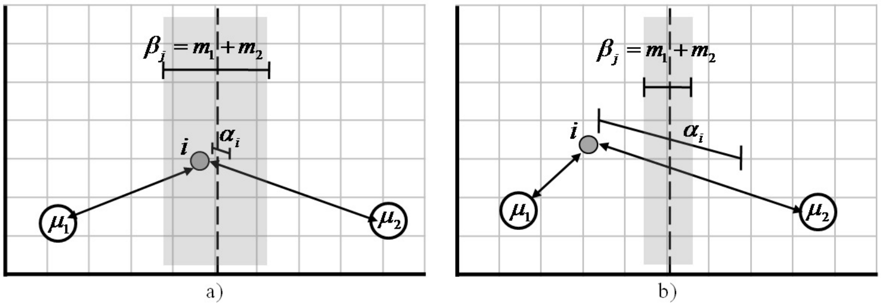

Fig. 3. Equidistance Threshold: object i has; a) high likelihood of cluster change, b) low likelihood of cluster change

5. EXPERIMENTAL RESULTS

This section presents the results of a set of experiments conducted to validate the proposed EC heuristic (K-Means+EC) to improve the K-Means algorithm. In addition, we compared our approach against the algorithms K-Means+E and K-Means+PR. All the algorithms were implemented in the C programming language. Experiments were conducted in a computer with the following configuration: AMD Athlon 64X2TK-57, 1.9 GHz processor, 4GB of RAM, 100GB of hard disk, and the Ubuntu 10.10 operating system.

We used three synthetic and three real datasets. The synthetic datasets were created using a uniform distribution, two dimensions, and with 2,500, 10,000 and 40,000 objects. For all datasets 100 clusters were generated. Uniform distribution was selected to analyze the behavior of the algorithms when the number of objects is changed. The real-world datasets used were: the well-known *Iris* with 150 objects and three dimensions, *Concrete compressive strength* with 1,030 objects and 8 dimensions, and *Skin segmentation* with 245,057 objects and 3 dimensions. The real-world datasets were extracted from the UCI Repository of Machine Learning Databases [Merz et al. 2012]. All the experiments described were repeated 30 times using the same datasets and number of clusters. The initial centroids were generated randomly each time.

The improvements of the K-Means+EC, the K-Means+E, and the K-Means+PR heuristics in comparison to the standard K-Means algorithm were measured in terms of execution time and quality of the clustering result. The quality of the clustering is expressed by the squared error function (eq. 1), which in optimization terms has to be minimized:

$$\mathcal{J} = \sum_{l=1}^{k} \sum_{i_j \epsilon C_l} ||i_j - \mu_l||^2 \tag{1}$$

where $\{i_1, \ldots, i_n\}$ is the set of objects, $C = \{C_1, \ldots, C_k\}$ is the set of clusters, and μ_l is the mean of elements in C_l.

In the following we present and discuss the results obtained by the chosen algorithms in terms of quality of the clustering. The algorithms were compared against the standard K-Means.

Table I presents the results on the large synthetic datasets with 2,500, 10,000 and 40,000 objects. The first column refers to the number of objects used in the experimentation. The columns 2 to 9 correspond to the average squared error $\mathcal{J}$ and standard deviation σ of the algorithms K-Means, K-Means+EC, K-Means+E, and K-Means+PR respectively. The columns 10, 11 and 12 show the percentage of the difference in quality of each algorithm when compared to the standard version. This is calculated through eq. 2 where s_i denotes the average squared error of an enhanced version of

Table I. Experimental results for large synthetic datasets

Num	K-Means		K-Means+EC		K-Means+E		K-Means+PR		% $\mathcal{E}$	% $\mathcal{E}$	% $\mathcal{E}$
	$\mathcal{J}$	σ	$\mathcal{J}$	σ	$\mathcal{J}$	σ	$\mathcal{J}$	σ	EC	E	PR
2500	4854.8	23.6	4916.8	38.3	5220.0	81.9	5154.4	73.4	-1.3	-7.5	-6.2
10000	38352.3	92.3	39377.0	355.8	41850.8	558.7	41322.1	507.9	-2.7	-9.1	-7.7
40000	305278.3	341.6	315379.2	2981.0	334925.2	7405.0	330594.3	4227.3	-3.3	-9.7	-8.3

Table II. Experimental results for *Iris* benchmark dataset

Cluster	K-Means		K-Means+EC		K-Means+E		K-Means+PR		% $\mathcal{E}$	% $\mathcal{E}$	% $\mathcal{E}$
Number	$\mathcal{J}$	σ	$\mathcal{J}$	σ	$\mathcal{J}$	σ	$\mathcal{J}$	σ	EC	E	PR
5	95.2	12.2	96.4	12.6	100.4	12.6	99.1	12.6	-1.2	-5.4	-4.1
20	66.6	6.9	67.5	7.1	74.2	6.6	72.2	6.9	-1.3	-11.4	-8.5
40	57.8	3.6	58.4	3.8	63.7	5.1	61.9	4.9	-1.0	-10.3	-7.1

Table III. Experimental results for large real datasets

Dataset	K-Means		K-Means+EC		K-Means+E		K-Means+PR		% $\mathcal{E}$	% $\mathcal{E}$	% $\mathcal{E}$
	$\mathcal{J}$	σ	$\mathcal{J}$	σ	$\mathcal{J}$	σ	$\mathcal{J}$	σ	EC	E	PR
Concrete	26023	3900	26197	3890	29602	3641	28792	3831	-0.7	-13.8	-10.6
Skin	1521153	48383	1625381	70031	1953108	136718	1884414	117309	-6.9	-28.4	-23.9

K-Means and s_s the average squared error of the standard K-Means.

$$\mathcal{E} = \frac{(s_s - s_i) * 100}{s_s} \tag{2}$$

The results show clearly that the algorithm with the smallest quality loss was EC with a reduction of 1.3% for the 2,500 dataset, 2.7% for the 10,000 dataset and 3.3% for the 40,000 dataset.

The results obtained using the *Iris* dataset are shown in Table II. Here too, EC presented the smallest percentage of reduction in clustering quality. The results shown in Table III are based on the large real datasets comprised of 100 clusters. For validating the results, we applied the Wilcoxon signed-rank test [Ron and Betsy 2012] by means of calculating the absolute minimum value of the sum of the positive and negative ranks w_s of data and compare it against the Wilcoxon signed-rank test critical value cv in order to accept or reject the null hypothesis H_0. In Wilcoxon signed-rank test if $w_s \leq cv$ then H_0 is rejected. The parameters used in the test were: i) a level of significance of $\alpha = 0.05$, ii) a null hypothesis H_0 : The differences in the squared error solutions obtained by the execution of two algorithms are due to randomness and iii) an alternative hypothesis H_a : The differences in the squared error solutions obtained by the execution of two algorithms are not due to randomness.

Table IV shows the obtained results. Column one refers to the synthetic instances with 2,500, 10,000, and 40,000 objects, the *Iris* dataset when grouped in 5, 20 and 40 clusters and the *Concrete compressive strength* and *Skin segmentation* datasets. The first row presents the results of applying the Wilcoxon signed-rank test to the results of K-Means+EC against K-Means, K-Means+E, and K-Means+PR. The column labelled n represents the signed rank which is the number of pairs of data for which the difference (in the squared error value) is positive or negative, w_s is the absolute minimum value of the sum of the positive and negative ranks, and cv corresponds to the critical value when $\alpha = 0.05$. We can see in Table IV that in all cases $w_s < cv$ and we conclude that there is enough evidence, at the 5% level of significance, to support that our results are not obtained due to randomness.

In the following we present the comparative results between the three enhanced versions of K-Means we considered against the standard K-Means, with respect to the execution time. The running time is presented in milliseconds, averaged over 30 runs, for each algorithm considered.

Table IV. Results of the Wilcoxon signed-rank test

Dataset	K-Means+EC and K-Means			K-Means+EC and K-Means+E			K-Means+EC and K-Means+PR		
	n	w_s	cv	n	w_s	cv	n	w_s	cv
Synthetic									
2500	30	0	152	30	0	152	30	0	152
10000	30	0	152	30	0	152	30	0	152
40000	30	0	152	30	0	152	30	0	152
Iris									
5	19	0	54	30	9	152	27	6	120
20	24	15	92	30	2	152	30	4	152
40	20	0	60	30	0	152	30	0	152
Concrete	26	7.5	110	30	0	152	30	126	152
Skin	30	0	152	30	0	152	30	0	152

Table V. Experimental results for large synthetic datasets

Dataset	K-Means		K-Means+EC		K-Means+E		K-Means+PR		% $\mathcal{T}$	% $\mathcal{T}$	% $\mathcal{T}$
	$\mathcal{T}$	σ	$\mathcal{T}$	σ	$\mathcal{T}$	σ	$\mathcal{T}$	σ	EC	E	PR
2500	641.6	134.2	397.0	32.1	56.1	2.2	218.5	9.1	38.1	91.3	65.9
10000	7577.9	2027.5	3011.6	91.2	236.8	13.2	2249.2	80.8	60.3	96.9	70.3
40000	71388.0	17133.4	12502.8	1539.4	926.5	35.2	30293.6	35.3	82.5	98.7	57.6

Table VI. Experimental results for *Iris* benchmark dataset

Dataset	K-Means		K-Means+EC		K-Means+E		K-Means+PR		% $\mathcal{T}$	% $\mathcal{T}$	% $\mathcal{T}$
	$\mathcal{T}$	σ	$\mathcal{T}$	σ	$\mathcal{T}$	σ	$\mathcal{T}$	σ	EC	E	PR
5	3.3	1.0	1.3	0.3	1.8	0.8	1.4	0.2	61.8	46.2	56.3
20	10.8	2.8	7.1	0.7	3.3	0.5	2.7	0.3	34.3	69.0	74.9
40	19.4	7.2	10.7	1.1	5.5	0.4	4.3	0.3	45.0	71.6	78.1

The results for the synthetic large datasets are shown in Table V. The first column shows the number of objects used in the experimentation. Columns 2 to 9 correspond to the average $\mathcal{T}$ and standard deviation σ of the running time of K-Means, K-Means+EC, K-Means+E and K-Means+PR algorithms respectively. Columns 10, 11 and 12 show the percentage of the difference in the running time between the three algorithms chosen against the standard K-Means, calculated using eq. 3, where t_i denotes the average running time of an enhanced version of K-Means and t_s the average running time of the standard K-Means.

$$\mathcal{T} = \frac{(t_s - t_i) * 100}{t_s} \tag{3}$$

In Table VI the results for the *Iris* dataset are presented. The results for the large real datasets with respect to the running time are displayed in Table VII.

Table VIII summarizes the results. The first column corresponds to the algorithm name, while the other columns refer to the dataset names. Each subcolumn of a dataset presents the average percentage of time ($\%\mathcal{T}$) reduced when compared to the standard algorithm and the percentage of reduction in the cluster quality ($\%\mathcal{E}$). Notice that EC presents, in all cases, the best trade-off between running time reduction and loss of clustering quality .

In general, the behavior of the K-Means+EC tends to be more effective when the relation between the number of clusters and objects is small, because the K-Means+EC algorithm requires calculating the displacements of the centroids at each iteration. This behavior of the K-Means+EC algorithm can be observed in Table V at column 10 where in the row of the 2,500 objects dataset, the K-Means+EC algorithm is 38.1% better than the K-means algorithm, whereas in the row of the 40,000 objects

Table VII. Experimental results for large real datasets

Dataset	K-Means		K-Means+EC		K-Means+E		K-Means+PR		% $\mathcal{T}$	% $\mathcal{T}$	% $\mathcal{T}$
	$\mathcal{T}$	σ	$\mathcal{T}$	σ	$\mathcal{T}$	σ	$\mathcal{T}$	σ	EC	E	PR
Concrete	89	26	62	26	23	1	59	5	30.7	74.6	33.6
Skin	222412	64826	115092	53149	5747	200	884051	17083	48.3	97.4	-297.5

Table VIII. Comparison between K-Means+EC, K-Means+E, and K-Means+PR

Algorithm	Iris (40)		Concrete		Skin		Synthetic (40,000)	
	% $\mathcal{T}$	% $\mathcal{E}$	% $\mathcal{T}$	% $\mathcal{E}$	% $\mathcal{T}$	% $\mathcal{E}$	% $\mathcal{T}$	% $\mathcal{E}$
Early Classification (K-Means+EC)	45.0	-1.0	30.7	-0.7	48.6	-6.9	82.5	-3.3
Enhanced K-Means (K-Means+E)	71.6	-10.3	74.6	-13.8	97.4	-28.4	98.7	-9.7
Pattern Recognition (K-Means+PR)	78.1	-7.1	33.6	-10.6	-297.5	-23.9	57.6	-8.3

dataset, the improvement is 82.5% better than the standard algorithm. In both experiments the number of clusters considered is 100.

When comparing the K-Means+EC algorithm with other similar algorithms, we could observe that, in general, its running time was greater than the others for the same instances, however, K-Means+EC showed the best relationship between the reduction in the quality of the solution and the improvement in running time. As shown in Tables III and VII, for *Skin segmentation* dataset with 245,057 objects, the K-Means+EC algorithm obtained a result where the quality reduction was of 6.9% and the running time reduction of 48.3%, while the K-Means+E algorithm obtained a quality reduction of 28.4% with a reduction in execution time of 97.4%.

6. CONCLUSIONS AND FUTURE WORK

One of the main drawbacks of K-Means is its high computational cost. This characteristic limits the processing of large and high dimensional datasets. This article shows that it is possible to improve the standard K-Means using a new heuristic in the classification step. A detailed analysis of the standard algorithm revealed that the application of the *Early Classification* heuristic allows the identification of objects with low likelihood of cluster change and their exclusion from subsequent iterations, thereby reducing the number of calculations at each iteration, without high loss of quality reduction. For assessing the proposed improvement, a set of synthetic data and the *Iris*, *Skin segmentation*, and *Concrete compressive strength* datasets taken from the UCI Machine Learning repository were used. Experiments with other two algorithms from the literature were conducted in order to better position our results. The Wilcoxon signed-rank test was used for calculating the statistical significance on the results. The test showed that the differences in the average results of the algorithms are statistically significant. The experimental results are promising. In the case of large synthetic datasets (the dataset with 40,000 objects and 100 clusters) the time was reduced up to 82.5% with a cluster quality reduction of only 3.3%. For the *Iris* dataset comprised of 40 clusters, we obtained a time reduction of 45.0% with a quality reduction in the clustering of only 1.2%. For the *Concrete compressive strength* dataset using $k = 100$, we obtained a time reduction of 30.7% with a quality reduction in the clustering of only 0.7%. At last, for the *Skin segmentation* dataset which has 245,057 objects and three dimensions generating 100 clusters, we obtained a time reduction of 48.3% with a quality reduction in the clustering of 6.9%. The comparative results showed that the Early Classification algorithm provides us a better accuracy than the algorithms available in the related work. Therefore, our heuristic improvement performs well with real and synthetic datasets. It is noteworthy to mention that as the number of objects increases, the heuristic achieves further reduction in the execution time.

In addition, the proposed heuristic is compatible with other optimization techniques for improving the K-Means algorithm. In other words, it can be combined with other variants of the K-Means algorithms, thus contributing to further improve their performance. Finally, we will continue the experimentation work with the aim of exploring other values for the equidistance threshold in other clustering datasets, including synthetic datasets with clear tendencies of cluster.

REFERENCES

Agha, M. E. and Ashour, W. M. Efficient and Fast Initialization Algorithm for K-means Clustering. *International Journal of Intelligent Systems and Applications* 1 (1): 21–31, 2012.

Ankerst, M., M., B. M., Kriegel, H.-P., and Sander, J. Optics: ordering points to identify the clustering structure. In *Proceedings of ACM SIGMOD International Conference on Management of Data*. Philadelphia, Pennsylvania, pp. 49–60, 1999.

Dempster, A., Laird, N., and Rubin, D. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society* 39 (1): 1–38, 1977.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Portland, Oregon, pp. 226–231, 1996.

Fahim, A. M., Salem, A. M., Torkey, F. A., and Ramadan, M. A. An Efficient Enhanced K-means Clustering Algorithm. *Journal of Zhejiang University SCIENCE A* 7 (10): 1626–1633, 2006.

Jiawei, H. and Micheline, K. *Data Mining Concepts and Techniques*. Elsevier Inc., 2006.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. An Efficient K-means Clustering Algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 24, pp. 881–892, 2002.

Kaufman, L. and Rousseeuw, P. Clustering by Means of Medoids. In D. Y. (Ed.), *Statistical Data Analysis Based on the L_1 Norm and Related Methods*. Delft University of Technology, North-Holland, pp. 405–416, 1987.

Lai, J. Z. C. and Liaw, Y. Improvement of the K-means Clustering Filtering Algorithm. *Pattern Recognition* 41 (12): 3677–3681, 2008.

Lloyd, S. P. Least Squares Quantization in PCM. *IEEE Transactions Information Theory* 28 (1): 129–137, 1982.

MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Berkeley Symposium on Mathematics, Statistics and Probability*. Berkeley, California, pp. 281–296, 1967.

Merz, C., Murphy, P., and Aha, D. UCI Repository of Machine Learning Databases. Department of Information and Computer Science, University of California. http://www.ics.uci.edu/ mlearn/MLRepository .html, 2012.

Myatt, G. N. and Johnson, W. P. *Making Sense of Data II: a practical guide to data visualization, advanced data mining methods, and applications*. JohnWiley & Sons, 2009.

Ron, L. and Betsy, F. *Elementary Statistics Picturing the World*. Pearson Education, Inc., 2012.

Tsai, C., Yang, C., and Chiang, M. A Time Efficient Pattern Reduction Algorithm for K-means Based Clustering. In *Proceedings of the Conference on Systems, Man and Cybernetics*. Montréal, Canada, pp. 504–509, 2007.

Zhanguo, X., Shiyu, C., and Wentao, Z. An Improved Semi-Supervised Clustering Algorithm Based on Initial Center Points. *Journal of Convergence Information Technology* 7 (5): 317–324, 2012.