

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

Joaquín Pérez¹, Carlos Eduardo Pires², Leandro Balby², Adriana Mexicano^{1,3}, Miguel Ángel Hidalgo¹

¹ Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)

² Universidade Federal de Campina Grande (UFCG)

³ Universidad Politécnica del Estado de Morelos (UPEMOR)

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 high 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. The results are promising since the running 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 of 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 follows¹:

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

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

[Agha and Ashour 2012; Zhanguo et al. 2012], although randomly selecting the centroids is still the most widely used;

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. In order to evaluate the proposed heuristic, a set of experiments was performed using synthetic data and the well-known *Iris*, *Concrete compressive strength*, and the *Skin segmentation* datasets, available at the UCI Machine Learning repository. The results show that the running 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 and results are compared with related work. Finally, Section 6 concludes the article and points out directions for future work.

2. RELATED WORK

Several improvements were 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]. The 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 running time up to 33.6% in comparison to the FA algorithm.

[Tsai et al. 2007] proposed a heuristic called Enhanced K-Means which compresses and removes objects that are close to the centroid. 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 applied after the second iteration and repeatedly until 80% of the objects are removed. Results show that this improvement reduces the running time significantly specially for high dimensional datasets. In the rest of the document we refer to this heuristic as K-Means+E.

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 is discarded for subsequent calculations in the current iteration; 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 running time without significantly decreasing cluster quality. Fahim called the heuristic as Patter Recognition, in the rest of the document we refer to this as K-Means+PR.

All the aforementioned works use information about centroids displacement 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, [Tsai et al. 2007] besides using more calculations than our heuristic for discarding objects, assume that only the objects which are far from their centroids can change in the following iterations.

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.

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 high 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 cluster in subsequent iterations. These objects are then marked and excluded from future calculations. To perform the

selection process, we introduced 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 dataset sizes. Some interesting observations were 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 iteration 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 $I = \{i_1, \dots, i_n\}$ be a set of objects in a m -dimensional space to be partitioned,

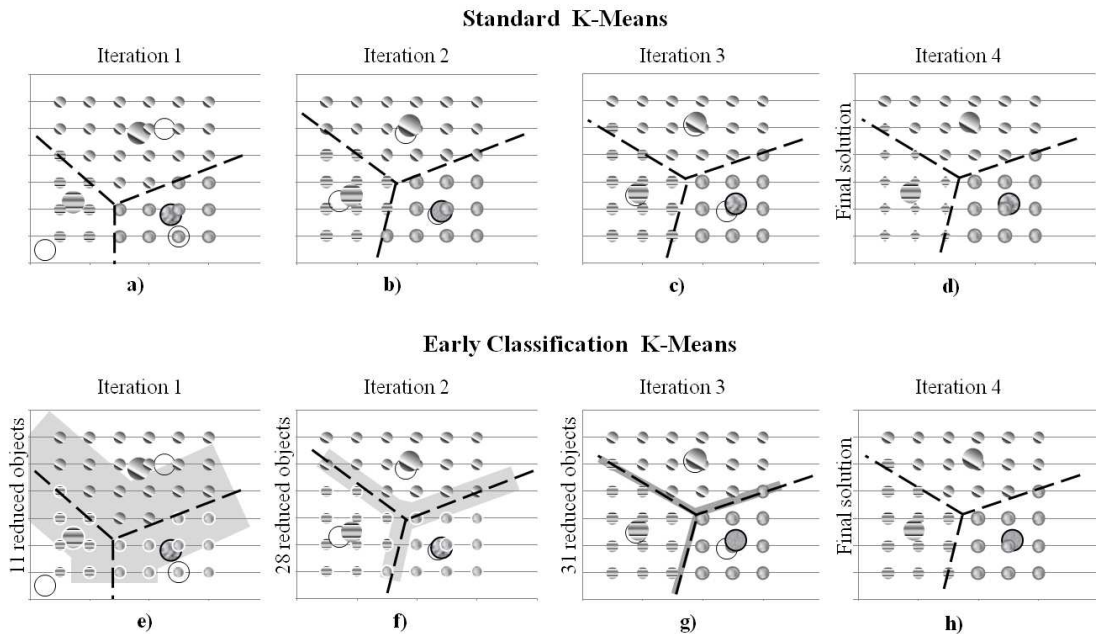


Fig. 1. Execution of the standard K-Means and the improved K-means using a dataset with 36 uniformly distributed objects

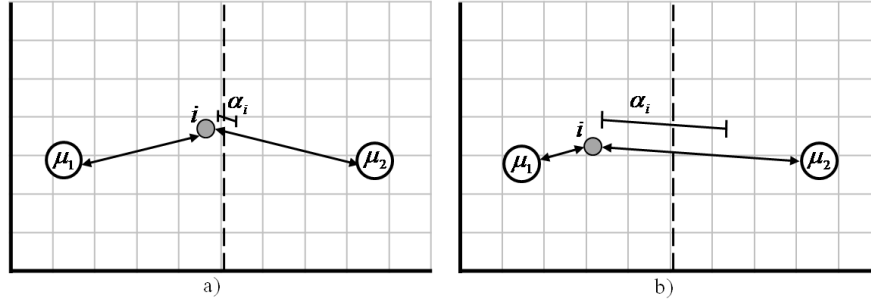


Fig. 2. Equidistance index; a) object i with high likelihood of cluster change, b) object i with low likelihood of cluster change

$C = \{C_1, \dots, C_k\}$ 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 $\|\cdot\|$ the ℓ^2 norm, for $p = 1, \dots, n$ and $l = 1, \dots, k$; 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 , $\alpha_i = \text{abs}(\|i - \mu_1\|^2 - \|i - \mu_2\|^2)$. 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 .

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, given that μ_x is the nearest centroid of i , the object i can be early classified into the partition C_x at iteration j if the condition $\alpha_i > \beta_j$ is true.

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. Additionally experiments with the K-Means+E and the K-Means+PR algorithms were conducted. All the algorithms were implemented in “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

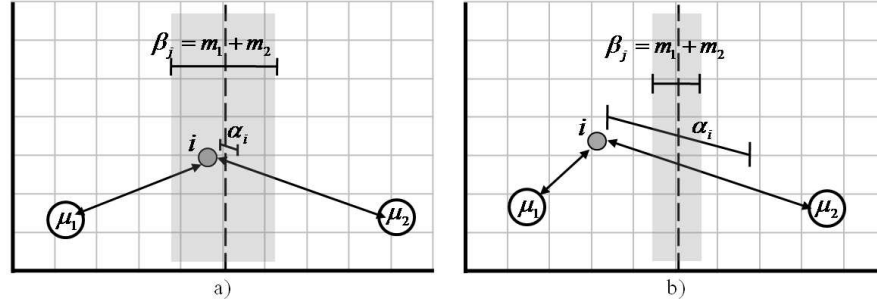


Fig. 3. Threshold equidistance; a) high likelihood of cluster change, b) low likelihood of cluster change

uniform distribution, two dimensions, and with 2,500, 10,000 and 40,000 objects. For all datasets 100 clusters were generated. 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 [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 running 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:

$$\mathcal{J} = \sum_{l=1}^k \sum_{i_j \in C_l} \|i_j - \mu_l\|^2 \quad (1)$$

where $\{i_1, \dots, i_n\}$ is the set of objects, $C = \{C_1, \dots, C_k\}$ 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 three improvements in terms of quality of the clustering. The three improvements were compared against the standard K-Means.

Table I shows the algorithm behavior using large synthetic datasets with 2,500, 10,000 and 40,000 objects comprised in 100 clusters. Table I shows in the first column, the number of objects used in the experimentation. Columns 2 to 5 correspond to the K-Means, Early Classification, Enhanced and Patter Reduction algorithms. Respectively, columns 6, 7 and 8 show the percentage of the difference in quality for each improved algorithm when compared to the standard version, calculated using eq. 2 where s_i denotes the squared error of the corresponding improved algorithm and s_s the squared error of standard K-Means.

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

Results show that in the three cases the algorithm which obtained less reduction in cluster quality was the EC with a reduction of 1.28% for the 2,500 dataset, 2.67% for the 10,000 dataset and 3.31% for the 40,000 dataset. On the other hand the algorithm that obtained the higher reduction in the quality was the K-Means+E algorithm with 7.52%, 9.12%, and 9.71% for the 2,500, 10,000, and 40,000 datasets respectively.

The results obtained using the *Iris* dataset are shown in Table II. According to the results, which generated 40 clusters, the quality of the clustering was decreased in 1.02% using the K-Means+EC algorithm, in the case of K-Means+E algorithm the reduction was of 10.30%, and using the K-

Table I. Experimental results for large synthetic datasets

N. objects	K-Means	K-Means+EC	K-Means+E	K-Means+PR	% \mathcal{E}_{EC}	% \mathcal{E}_E	% \mathcal{E}_{PR}
2,500	4,854.73	4,916.80	5,219.92	5,154.36	-1.28	-7.52	-6.17
10,000	38,352.27	39,376.96	41,850.83	41,322.13	-2.67	-9.12	-7.74
40,000	305,278.44	315,379.19	334,925.16	330,594.28	-3.31	-9.71	-8.29

Table II. Experimental results for *Iris* benchmark dataset

N. clusters	K-Means	K-Means+EC	K-Means+E	K-Means+PR	% \mathcal{E}_{EC}	% \mathcal{E}_E	% \mathcal{E}_{PR}
5	95.21	96.35	100.39	99.09	-1.20	-5.44	-4.08
20	66.58	67.46	74.18	72.21	-1.32	-11.41	-8.46
40	57.76	58.35	63.71	61.85	-1.02	-10.30	-7.08

Table III. Experimental results for large real datasets

Dataset	K-Means	K-Means+EC	K-Means+E	K-Means+PR	% \mathcal{E}_{EC}	% \mathcal{E}_E	% \mathcal{E}_{PR}
Concrete	26,023.23	26,197.25	29,602.34	28,792.4	-0.67	-13.75	-10.64
Skin	1,521,153.38	1,625,380.88	1,953,108.00	1,884,414.25	-6.85	-28.40	-23.88

Means+PR algorithm the reduction was of 7.08%. It is noteworthy that our version had the less percentage of reduction in clustering quality.

The results shown in Table III are based on the large real datasets comprised of 100 clusters. For the *Concrete compressive strength* dataset, the K-Means+EC algorithm obtained a quality reduction in the clustering of 0.67%. For the *Skin segmentation* dataset a quality reduction of 6.85% was obtained. The K-Means+E algorithm obtained for the *Concrete compressive strength* and the *Skin segmentation* datasets a quality reduction of 13.75% and 28.40%, respectively. Finally the K-Means+PR algorithm had a quality reduction of 10.64% and 23.88% for the *Concrete compressive strength* and the *Skin segmentation* datasets, respectively.

In the following are the comparative results between the three improvement algorithms against the standard K-Means, with respect to the running time are presented. The running time is presented in milliseconds, averaged over 30 executions, for each algorithm version.

The results for synthetic large datasets are shown in Table IV. The first column shows the number of objects used in the experimentation. Columns 2 to 5 correspond to the average running time consumed for K-Means, Early Classification, Enhanced and Patter Reduction algorithms respectively. Columns 6, 7 and 8 show the percentage of the difference in the running time between the three improved algorithms against the standard one, calculated using eq. 3 where t_i denotes the running time of the improvement and t_s the running time of standard K-Means.

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

Results show that the time is considerably reduced, mainly by the K-Means+E algorithm. We can observe that, in general, the largest running time reduction was obtained by the K-Means+E algorithm. In comparison to the standard K-Means, the K-Means+E algorithm reached for the dataset with 40,000 a reduction on running time of 98.70%; the K-Means+EC, a reduction of 82.49%; and finally the K-Means+PR, a reduction of 57.56%. The K-Means+PR algorithm obtained the shortest reduction in running time for the largest set.

In Table V the results for *Iris* dataset are presented. Note that the average time consumed with 5 clusters was reduced in 61.7% by the K-Means+EC algorithm in comparison to the standard version. The K-Means+PR obtained a reduction of 56.27% and the K-Means+E a reduction of 46.18%. When the experimentation was conducted generating 20 and 40 clusters the K-Means+PR algorithm obtained the greatest reduction in running time of 74.86% and 78.08% respectively. The results for

Table IV. Experimental results for large synthetic datasets

N. objects	K-Means	K-Means+EC	K-Means+E	K-Means+PR	% \mathcal{T}_{EC}	% \mathcal{T}_E	% \mathcal{T}_{PR}
2500	641.57	397.04	56.09	218.46	38.11	91.26	65.95
10000	7,577.85	3,011.55	236.77	2,249.16	60.26	96.88	70.32
40000	71,387.99	12,502.82	926.45	30,293.64	82.49	98.70	57.56

Table V. Experimental results for *Iris* benchmark dataset

N. clusters	K-Means	K-Means+EC	K-Means+E	K-Means+PR	% \mathcal{T}_{EC}	% \mathcal{T}_E	% \mathcal{T}_{PR}
5	3.27	1.25	1.76	1.43	61.77	46.18	56.27
20	10.78	7.08	3.34	2.71	34.32	69.02	74.86
40	19.43	10.68	5.52	4.26	45.03	71.59	78.08

Table VI. Experimental results for large real datasets

Dataset	K-Means	K-Means+EC	K-Means+E	K-Means+PR	% \mathcal{T}_{EC}	% \mathcal{T}_E	% \mathcal{T}_{PR}
Concrete	89.03	61.67	22.61	59.13	30.73	74.60	33.58
Skin	222,411.95	115,092.12	5,746.75	884,051.38	48.25	97.42	-297.48

the large real datasets with respect to the running time are displayed in Table VI. For the *Concrete compressive strength* and *Skin segmentation* datasets, the K-Means+E obtained a time reduction of 74.60% and 97.42%, respectively. The algorithm K-Means+EC obtained a running time reduction of 30.73% and 48.25%, respectively. Finally the K-Means+PR algorithm obtained a running time reduction of 33.58% for the *Concrete compressive strength* dataset; however, the running time was increased in 297.48% when the algorithm was applied over the *Skin segmentation* dataset.

Table VII highlights the most important results. The first column corresponds to the algorithm name, while the other columns refer to the dataset names. Each sub column of a dataset name 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}). Table VII shows that in all cases the K-Means+EC algorithm obtained the lowest reduction in the solution quality. In the case of *Iris* dataset with 40 clusters, the K-Means+PR algorithm obtained the highest time reduction. For the *Concrete*, *Skin* and synthetic (40,000 objects) datasets the K-Means+E algorithm obtained the highest reduction time, but also K-Means+E was the algorithm that obtained the highest reduction in cluster quality. It is noteworthy that the K-Means+EC algorithm showed a good performance without a high loss of cluster quality.

Finally, Figures 4 and 5 show the results graphically. Fig. 4 shows the results of three algorithms and the six datasets, the graphic compares the reduction in quality for each improved algorithm with respect to the standard K-Means. The Y axis shows the percentage of reduction in the cluster quality for each improved algorithms compared with the cluster quality of the standard K-Means algorithm. The X axis shows three sets of bars, they correspond to the results of each dataset for each improved algorithm. We can see in the graphic that the K-Means+EC algorithm has a smaller decrease in the cluster quality, whereas the K-Means+E algorithm has the greatest one.

Fig. 5 shows a comparative between the reduction in running time of each improved algorithm with respect to the standard K-Means solving the six datasets. The Y axis is a measure of the percentage of the difference of running time for each improved algorithm compared with the K-means running time. The X axis shows three sets of bars that correspond to the results of each dataset for a given algorithm. In general, the algorithm that showed a greater reduction in the running time was the K-Means+E.

Table VII. Comparison between the three improvement algorithms

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)	45.03	-1.02	30.73	-0.67	48.25	-6.85	82.49	-3.31
Enhanced K-Means (K-Means+E)	71.59	-10.30	74.60	-13.75	97.42	-28.40	98.70	-9.71
Pattern Recognition (K-Means+PR)	78.08	-7.08	33.58	-10.64	-297.48	-23.88	57.56	-8.29

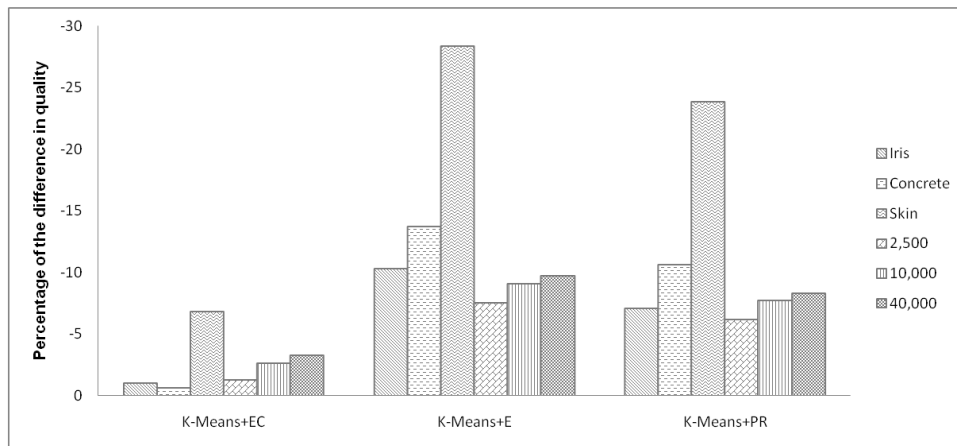


Fig. 4. Comparison of cluster quality for the three improved algorithms solving the six datasets

6. CONCLUSIONS AND FUTURE WORK

One of the main drawback of K-Means is its high computational cost. This limitation restricts the processing of large and high dimensional datasets. This work 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. The experimental results were promising. In the case of large synthetic datasets, the dataset with

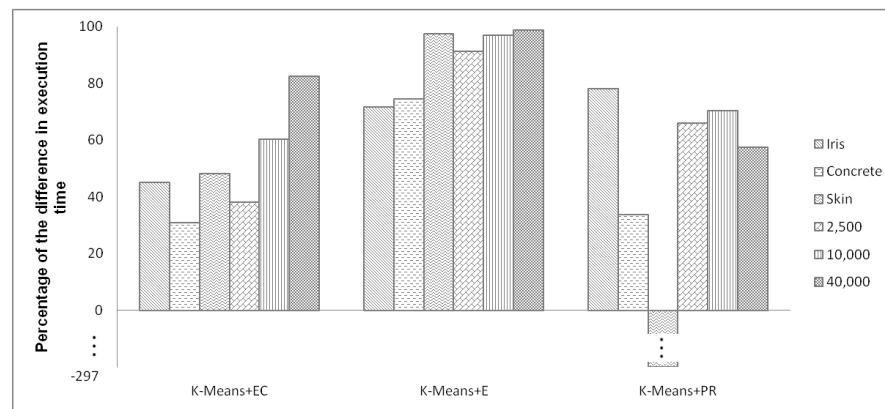


Fig. 5. Comparison of running time for the three improved algorithms solving the six datasets

40,000 objects and 100 clusters, the time was reduced up to 82.49% with a cluster quality reduction of only 3.31%. For the *Iris* dataset comprised of 40 clusters, we obtained a time reduction of 45.03% with a quality reduction in the clustering of only 1.20%. For the *Concrete compressive strength* dataset using $k = 100$, we obtained a time reduction of 30.73% with a quality reduction in the clustering of only 0.67%. At last, for the *Skin segmentation* dataset which has 245,057 objects and three dimensions generating 100 clusters, we obtained a time reduction of 48.25% with a quality reduction in the clustering of 6.85%. The comparative results showed that the Early Classification algorithm provides us a better accuracy than the heuristics 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 a further reduction in the running 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 for other clustering datasets. We also plan to introduce this heuristic with other variants of the algorithm.

ACKNOWLEDGMENT

We express our gratitude to CONACYT and the UFCG for the facilities provided in the realization of this research work. Also, we would like to thank A. Moreno (student of CENIDET) for her assistance.

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 *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. *J Zhejiang Univ 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 Trans. Information Theory* 28 (1): 129–137, 1982.
- MACQUEEN, J. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematics, Statistics and Probability*. 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.
- TSAI, C., YANG, C., AND CHIANG, M. A Time Efficient Pattern Reduction Algorithm for k-means Based Clustering. In *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.