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 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 in the classification step of K-Means by using statistical information about the displacement of centroids at each iteration. Our heuristic, denoted Early Classification (EC), 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 execution time of K-Means was reduced up to 85.01%. Moreover, as the experiments will show, the superiority of our method is even more evident on large datasets.

Resumo. A análise de agrupamento (cluster) é o estudo de algoritmos e técnicas para agrupar objetos de acordo com as características intrínsecas de valores semelhantes. Um algoritmo de agrupamento popular e amplamente estudado é K-Means, que se caracteriza por sua facilidade de implementação. No entanto, tem um alto custo computacional. Embora vários melhoramentos tenham sido propostos para k-means, o algoritmo ainda é considerado uma alternativa custosa para o agrupamento de dados em grande escala. Este artigo propõe uma nova heurística para reduzir o número de cálculos na fase de classificação do k-means, utilizando a informação estatística do deslocamento dos centróides de cada iteração. Nossa heurística, chamada de classificação antecipada, introduz os conceitos de índice de equidistância e limite de equidistância, com o objetivo de identificar e excluir dos cálculos futuros aqueles objetos que, de acordo com o limite de equidistância, têm baixa probabilidade de mudança de cluster para as iterações subsequentes. A fim de validar a nossa heurística, um conjunto de experimentos foi realizado com instâncias sintéticas e do repositório UCI Machine Learning. Os resultados dos testes são promissores porque o tempo de execução é reduzido em até 85,01%. A superioridade do nosso método é mais notável ao utilizar conjuntos de dados grandes.

Categories and Subject Descriptors: Knowledge Discovery from Databases [**H. m. Miscellaneous**]: Databases

General Terms: Algorithms and Techniques for Data Mining

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

1. INTRODUCTION

Clustering is a widely used and flexible method of grouping objects into clusters without relying on labeled training instances [Myatt and Johnson 2009]. The objects within a cluster are supposed to have high similarity between each other 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 the following[1]:

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], and the random method is 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 related to 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 two any 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 the clusters' centroids. 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 dataset, available at the UCI Machine Learning repository. The results show that the execution time of K-Means was reduced up to 85.01%.

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

2. MOTIVATING EXAMPLE

Fig. 1 illustrates a clustering example with a dataset containing 36 uniformly distributed objects in 3 clusters. The left side refers to the execution of the standard K-Means algorithm while the right side refers to the execution of K-Means including 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 dot. The color of the objects is related to the color 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 to 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.

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

Figures 1a, 1b, 1c, and 1d depict the formation of clusters at each iteration. The right side of 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 are 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 version allows 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.

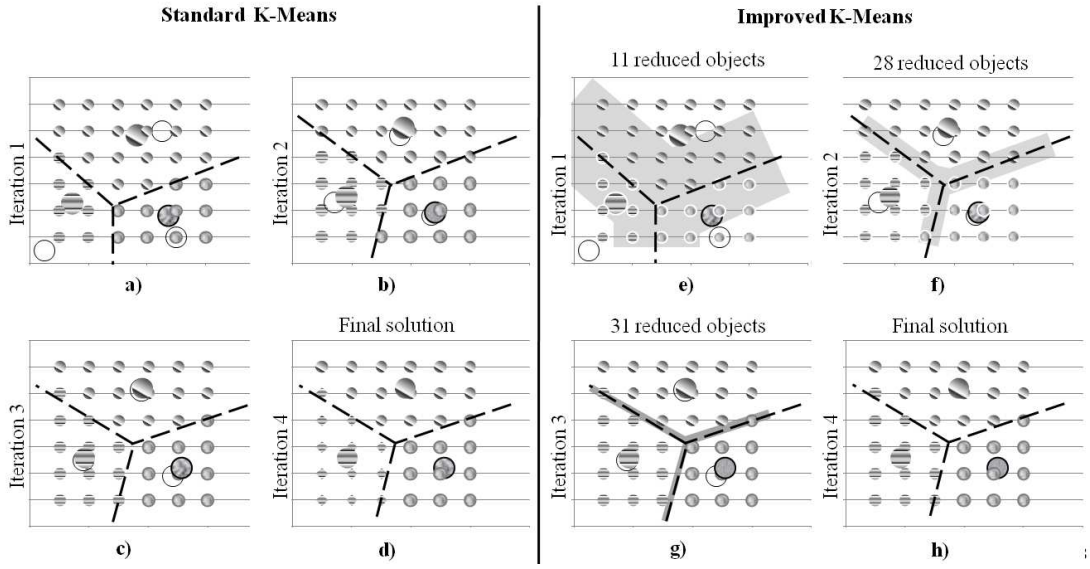

Fig. 1. Execution of the standard K-Means and the improved K-means using an instance with 36 uniformly distributed objects.

3. THE EARLY CLASSIFICATION HEURISTIC

The *Early Classification* (EC) heuristic is simple and its objective is to reduce the number of objects that participate in the distant calculations between the objects to the centroids at each iteration. 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 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 solving instances of synthetic data with uniform distribution and different instance sizes. Some of the interesting observations were the following:

a) Objects close to the centroids are unlikely to change cluster in subsequent iterations;

b) Objects equidistant from its two nearest centroids can be assigned to any of the two clusters;
c) Objects quasi equidistant from its 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 smaller distance from the new centroid position and the other half at a larger 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 displacement. The amount of displacement between centroids and between iterations may vary;
h) Centroids can move in different directions across different iterations.

3.1 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 m-dimensional space to be partitioned, $C = \{C_1, \ldots, C_k\}$ 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 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. 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.

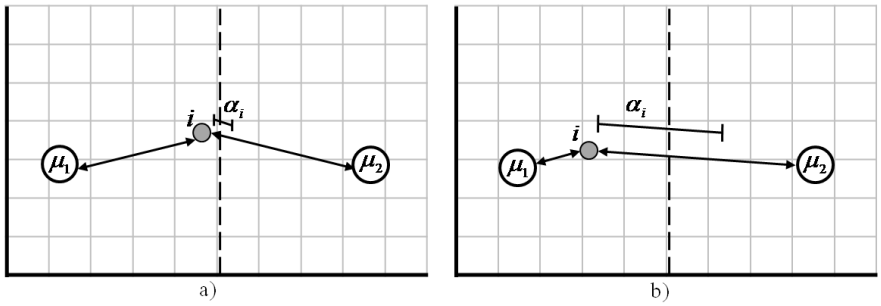

Fig. 2. Equidistance index; a) object i with high likelihood of cluster change, b) object i with low likelihood of cluster change.

3.2 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.

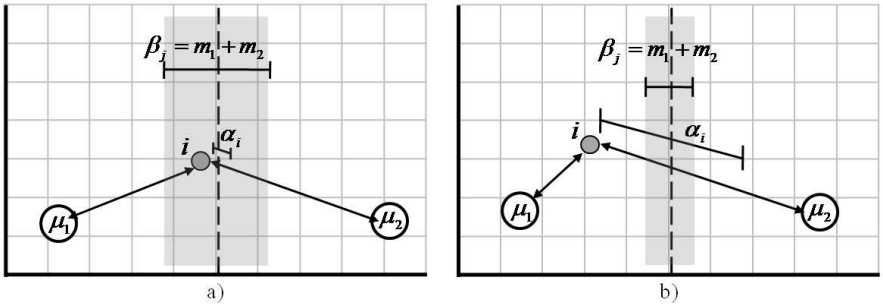

Fig. 3. Threshold equidistance; a) high likelihood of cluster change, b) low likelihood of cluster change.

4. EXPERIMENTAL RESULTS

This section presents the results of a set of experiments conducted to validate the proposed EC heuristic to improve the K-Means algorithm. The standard K-Means and the EC heuristic were implemented in the language “C". Experiments were conducted in a computer with the following configuration: Intel(core) i5, 2.66GHz processor, and 8GB of RAM, 300GB of hard disk, and Ubuntu 12.04 operating system.

We used three synthetic and three real datasets. The synthetic instances were created using a uniform distribution, two dimensions, and containing 2 500, 10 000, and 40 000 objects generating 100 clusters. The real datasets used were *iris* with 150 data and three dimensions, *concrete compressive strength* with 1 030 data and 8 dimensions, and *skin segmentation* with 245 057 data and 3 dimensions. The real 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 improvement of the EC in comparison to the standard K-Means algorithm was measured in terms of execution time and quality of the clustering. The quality of the clustering is expressed by the squared error function (equation 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.

Table I shows the algorithm behavior using large instances containing 2 500, 10 000 and 40 000 objects. The column *Execution time* shows in the sub columns K-Means+EC and K-Means, the

average time of 30 executions for the improved t_i and the standard t_s algorithms, expressed in units of milliseconds. The column *Squared error* $\mathcal{J}$ shows in the sub columns K-Means+EC and K-Means, the average quality of 30 executions for the improved s_i and the standard s_s algorithms expressed as the value of the sum of squared error. The column $\%\ \mathcal{T}$ shows the percentage of the difference in the execution time between the improved algorithm and the standard one, calculated with equation 2. The column $\%\ \mathcal{E}$ shows the percentage of the difference in quality for the improved and standard algorithms, calculated using the equation 3.

$$\mathcal{T} = \frac{(t_s - t_i) * 100}{t_s} \tag{2}$$

$$\mathcal{E} = \frac{(s_s - s_i) * 100}{s_s} \tag{3}$$

Results show that the clustering quality was not affected significantly, but time is considerably reduced. It is remarkable that in the case of the dataset with 40 000 objects and 100 clusters, the difference in execution time between both algorithms was 85.01%, while in quality it was only 3.3%.

Table I. Experimental results for large synthetic instances

Number of objects	Execution time (ms)		Squared error $\mathcal{J}$		$\%\ \mathcal{T}$	$\%\ \mathcal{E}$
	K-Means+EC	K-Means	K-Means+EC	K-Means		
2 500	1 328.14	2 488.43	4 906.03	4 853.43	46.63	-1.08
10 000	6 817.35	28 767.83	39 321.60	38 342.93	76.30	-2.55
40 000	42 354.54	282 516.42	315 521.79	305 432.04	**85.01**	**-3.30**

The results obtained using the *iris* dataset are shown in Table II. According to the results, the quality of the cluster was decreased in 0.67%, and the execution time in 44.61%. It is noteworthy that the dataset with 5 clusters obtained a time reduction of 63.5% reducing only 0.5% of the quality solution.

Table II. Experimental results for iris benchmark instances

Number of clusters	Execution time (ms)		Squared error $\mathcal{J}$		$\%\ \mathcal{T}$	$\%\ \mathcal{E}$
	K-Means+EC	K-Means	K-Means+EC	K-Means		
3	16.73	36.75	123.40	123.28	54.48	-0.10
5	23.66	64.82	95.85	95.37	**63.50**	**-0.50**
10	35.15	73.61	81.44	80.42	52.25	-1.27
20	53.46	86.28	65.42	64.74	38.04	-1.05
30	57.78	89.85	61.03	60.38	35.69	-1.08
40	47.41	82.41	57.38	57.16	42.47	-0.38
50	62.45	84.19	55.51	55.32	25.82	-0.34
Average values					**44.61**	**-0.67**

Regarding Table III results are based on the large real datasets generating 100 clusters; for *concrete compressive strength* dataset, we obtained a time reduction of 20.57% with only a quality reduction in the clustering of 1.57% and for the *skin segmentation* dataset we obtained a time reduction of 50.16% with only a quality reduction in the clustering of 6.94.

Table III. Experimental results for large real instances

Dataset name	Number of objects	Execution time (ms)		Squared error $\mathcal{J}$		$\%\ \mathcal{T}$	$\%\ \mathcal{E}$
		K-Means+EC	K-Means	K-Means+EC	K-Means		
Concrete	1030	1 600.42	2 014.81	93 834.63	92 381.74	20.57	-1.57
Skin	245 057	695 678.39	1 395 858.42	4 277 627.50	4 000 178.75	50.16	-6.94

5. 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 execution time up to 33.6% in comparison to the FA algorithm.

[Tsai et al. 2007] proposed a heuristic 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 repeatedly until 80% of the objects are removed. Results show that this heuristic reduces execution time up to 79% especially for high dimensional data sets, and the cluster quality up to 14.08% in comparison to the standard K-Means algorithm.

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; 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.

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.

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 for subsequent iterations, thereby reducing the number of calculations at each iteration. 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. Regarding large synthetic instances, the instance with 40 000 objects and 100 clusters, the time was reduced up to 85.01% with only a cluster quality reduction of 3.3%. For the iris dataset which has 150 objects generating 5 clusters, we obtained a time reduction of 63.5% with only a quality reduction

in the clustering of 0.5%. For the *concrete compressive strength* dataset, which has 1 030 objects and 8 dimensions using $k = 100$, we obtained a time reduction of 20.57% with only a quality reduction in the clustering of 1.57% and for the *skin segmentation* dataset which has 245 057 objects and three dimensions generating 100 clusters,we obtained a time reduction of 50.16% with a quality reduction in the clustering of 6.94%. Therefore, our heuristic improvement performs well with real and synthetic instances. It is noteworthy to mention that as the number of objects increases, the heuristic achieves a further reduction of the percentage of 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 instances. We also plan to introduce this heuristic with other variants of the algorithm.

Acknowledgments. We express our gratitude to the Universidade Federal de Campina Grande for the facilities provided in the realization of this research work. Also, we would like to thank Alejandra Moreno and Emanuel Sotelo (students of the MSc program at the National Center for Research and Technological Development, CENIDET) for their assistance in the algorithm coding and experimentation.

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*. 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*. 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*. 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 aplications*. 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*. 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.