2023 26th International Conference on Computer and Information Technology (ICCIT)
13-15 December, Cox's Bazar, Bangladesh

Data Segmentation with Improved K-Means Clustering Algorithm

Emam Hasan, Md. Abdur Rahman, MD. Shojib Talukder, Md Farnas Utsho, Md. Shakhan, and Dewan Md. Farid
Department of Computer Science and Engineering, United International University,
United City, Madani Avenue, Badda, Dhaka 1212, Bangladesh
Email : {ehasan201302;mrahman202260;stalukder201345;mutsho201176;mshakhan201301}
@bscse.uiu.ac.bd, dewanfarid@cse.uiu.ac.bd

***Abstract*—Unsupervised learning is also known as learning by observation in machine learning which groups the data instances based on their similarities. k-Means clustering technique is one of the most commonly used partition-based clustering methods that continuously relocate data instances from one cluster to another cluster to ameliorate the cluster validation. In this paper, we have introduced a new approach to improve the data clustering performance of the k-Means clustering algorithm. The proposed approach significantly reduces the number of iterations. Initially, we need to set the value of k, the number of clusters, and randomly select k number of instances from data as initial cluster centers. Then rest of the instances are assigned to the clusters based on the minimum Euclidean value. In the traditional k-means clustering method, each data instance is compared with each cluster center. But, in this proposed method we assign an instance into a cluster based on the average value of all instances that are already assigned to the cluster instead of the cluster center. The primary innovation lies in this modification of the assignment of instances into a cluster, which diverges significantly from conventional methodologies. By harnessing the in-place-mean of cluster instances calculation during assignments, the proposed approach significantly curtails the number of iterations required for convergence.**

Keywords: ***Leaning by Observation, Partition-based Clustering; Unsupervised Learning*.**

I. INTRODUCTION

A branch of artificial intelligence known as "machine learning" is concerned with creating algorithms and models that allow computers to learn from data and make predictions or judgments without being explicitly programmed. Unsupervised learning is one of the fundamental paradigms used in machine learning. It is used to categorize unlabeled data (when we only have input data, X, and no corresponding class values) to discover patterns, similarities, and intriguing data structures. Since there are no known outputs in the training set of data, the goal of the machine learning method is to derive useful information from the input values [1]. The main tasks in unsupervised learning are density estimation, mining of association rules, clustering, dimension reduction, and anomaly detection.

Clustering is the technique of dividing a collection of instances into clusters (subsets or groups) so that instances within a cluster have a high degree of resemblance to one another but differ greatly from instances in other clusters [2]. It plays a vital role in data analysis by partitioning an input space into K regions denoted as $G_1, G_2, ..., G_k$, utilizing a defined distance measure. This technique is particularly useful in the context of a multidimensional space containing a set of objects $X = x_1, x_2, ..., x_n$ with a size of n. The main goal of clustering algorithms is to construct a Kxn partition matrix $U(X)$, represented as $U = [u_kj]$, where k ranges from 1 to K, and j ranges from 1 to n. The membership status of the object x_k in cluster G_k is represented by u_kj, which takes a value of 1 if x_j belongs to G_k, and $u_kj = 0$ otherwise [3]. In a variety of fields, including artificial intelligence, robotics, and medical science, cluster analysis has been successfully used to address data clustering issues [4].

The k-Means clustering algorithm [5] is widely used due to its simplicity. However, its computational complexity for large datasets is notable. To address this, In this paper, we introduce an advanced k-means approach that efficiently identifies cluster centers in significantly fewer iterations compared to the traditional method.

II. RELATED WORKS

Sinaga et al. [5] presented an unsupervised clustering method (U-k-Means) that determines optimal clusters without initialization or parameter choices. Yang et al. [6] introduced a multi-view k-Means variant that computes feature weights to eliminate irrelevant features, termed Feature Reduction Multi-view k-Means. The research of Ahmed et al. [7] highlighted two unavoidable problems with the k-Means algorithm (i.e., centroids selection, some clusters, and the ability to handle different types of data) and their possible solutions. Each of the existing algorithms is application or data-specific. Fränti et al. [8] addressed k-Means performance issues, suggesting improved initialization and algorithm repetition as solutions, with the latter outperforming when dealing with cluster overlap. Alguliyev et al. [9] introduced a parallel batch k-means approach, partitioning the data for separate clustering before merging for centroid calculation, effectively reducing computation time. Javidan et al. [10] introduced an innovative method for grape leaf disease diagnosis using image processing, k-means clustering, and SVM, achieving up to 98.97% classification accuracy and surpassing deep learning models in

979-8-3503-5901-5/23/$31.00 ©2023 IEEE

accuracy and speed. In the research of Peng et al. [11] a Mini Batch K-means, which is combined with principal components analysis (PCA), is proposed for the intrusion detection system (IDS). After preprocessing the dataset, the PCA method is used to reduce dimension and improve clustering efficiency. Feldman et al. [12] introduced a technique to reduce high-dimensional data to a condensed, weighted point set while maintaining comparable results to the original dataset.

III. Methodology

In this section, we initiate the discussion with an in-depth exploration of the proposed method followed by our algorithm's intricacies, and merits.

A. kMeans Clustering

The k-Means algorithm begins by choosing k initial centroids randomly. Data points are then assigned to the cluster with the closest centroid, based on distance matrices such as Euclidean or Manhattan distances [13]. The algorithm recalculates cluster centroids by computing the mean of points within each cluster, potentially causing data points to shift to different clusters based on proximity. However, it exhibits various limitations, particularly in terms of the number of iterations required for convergence [14]. Moreover, despite getting the optimal k value [13], the problem remains in the complexity due to calculating the within-cluster sum of squares (WSS) for each value of k. Also taking the random centroids initially might cause sub-optimal results [15] and convergence to local minima [16]. In our proposed methodology, we introduce a novel approach to address the issue of iteration time in the k-Means algorithm.

Algorithm 1 k-Means Clustering

Require: $data$: Input data points
Require: k: Number of clusters
Initialize centroids randomly
while not converged **do**
  for each data point **do**
    Assign data point to the closest centroid
  end for
  for each cluster **do**
    Calculate new centroid as the mean of points in the cluster
  end for
  if centroids don't change much **then**
    break
  end if
end while
return clusters and centroids

B. Proposed kMeans Clustering

In our proposed methodology, we introduce a novel approach to address the issue of iteration in the k-Means algorithm by introducing improvements that expedite convergence. Unlike the traditional k-Means approaches [13]–[16], our proposed method involves $instance\ sorting$ for optimal k determination and utilizes the Elbow method to identify the most suitable k value [17] through WCSS analysis (Eq:1). This fusion of preprocessing and data-driven insights enhances the subsequent steps. Our proposed approach addresses k-Means limitations by introducing improvements that expedite convergence. It involves $instance\ sorting$ for optimal k determination and utilizes the Elbow method to identify the most suitable k value [17] through WCSS analysis (Eq:1). This fusion of preprocessing and data-driven insights enhances the subsequent steps.

$$\text{WCSS} = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{1}$$

$\text{WCSS} = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$

where

$$i = \text{index of the cluster} \tag{2}$$
$$x = \text{data point} \tag{3}$$
$$C_i = \text{cluster that contains } x \tag{4}$$
$$\mu_i = \text{mean of cluster } C_i \tag{5}$$

In the Eq:1, C_i is the mean of the cluster, x is the data point, i is an index of the cluster μ_i is the mean of the cluster of C_i, and C_i is a cluster that contains x. The next pivotal aspect of our method is the selection of centroids. Leveraging the sorted instance list, we introduce the function $\text{select\_centroids}(sorted\_instances, k)$ in Eq:6 that strategically pick instances as centroids. This function ensures that centroids are spaced out effectively along the sorted list, aiming to enhance the diversity and quality of the initial cluster centers

for selecting centroids we have used the following equation:

$$\begin{aligned} &\text{select\_centroids}(sorted\_instances, k) = \\ &\{X_i \mid i = \frac{ik}{k-1}, \text{for } i = 0, 1, 2, ..., k-1\} \end{aligned} \tag{6}$$

$\text{select\_centroids}(sorted\_instances, k) = \{X_i \mid i = \frac{ik}{k-1}, \text{for } i = 0, 1, 2, ..., k-1\}$

where $\frac{ik}{k-1}$ is used to determine the position of the instances in the sorted list based on the value of k, which is the number of clusters; X_i is the i^{th} instance that will be chosen as centroid, and k is the number of clusters to be created.

With centroids selected, we proceed to our enhanced clustering method shown in the algorithm 8. The core innovation here lies in the cluster assignment process. First, instances are sorted, then instead of comparing instances with centroids, we calculate the in-place mean, shown in Eq:7 of each cluster and assign instances based on their proximity to this mean. This unique approach capitalizes on the spatial distribution of instances in the sorted list and aligns with the objective of optimizing iterations.

The formula for calculating the mean of each cluster is given by:

Algorithm 2 Mean-kMeans Clustering

Require: $data$: Input data points
Require: k: Number of clusters
Sort the data points
if k is not given **then**
  apply elbow method to determine optimal k
end if
initialize centroids randomly
while not converged **do**
  for each data point **do**
    for each cluster **do**
      calculates the new centroid as mean of the current number of points in the cluster
    end for
    assigns data point to the closest centroid
  end for
  if old centroids match new centroids **then**
    break
  end if
end while
return clusters and centroids

$$\text{Mean of Cluster, } \mu_{Ci} = \frac{1}{N(C_i)} \sum_{x \in C_i} x \tag{7}$$

Where

$N(C_i) =$ number of instances in the cluster C_i

$\sum_{x \in C_i} x =$ instances' sum x_i within the cluster C_i

In a nutshell, we're calculating the mean value 7 of all the instances in a specific cluster C_i by adding up the values of each instance within the cluster and then dividing that sum by the total number of instances in the cluster. This calculation gives us the center point or centroid of the cluster. As instances are assigned to clusters, the in-place-mean (Eq:7) of each cluster is continually updated. This dynamic process accounts for the changes in cluster composition and ensures that clusters adapt to the data distribution. By minimizing the number of iterations while maintaining clustering quality, we anticipate significant improvements in convergence speed compared to traditional k-Means.

IV. Results and Discussion

This section introduces the study's dataset and experimental setup. It presents an overview of evaluation metrics, followed by concise results, discussion, and analysis presentation.

A. Dataset Description

This experiment employs a selection of 4 valid real-world datasets. The datasets derive from well-known numeric sources accessible through the UC Irvine Machine Learning Repository https://archive.ics.uci.edu/ml/. Specifically, the selected authentic datasets encompass Iris, Glass, Wine, and Yeast data. The selection of these diverse datasets is conducted with the intention of comprehensively evaluating the internal measures across various scenarios. Elaborated information pertaining to the datasets is succinctly encapsulated in Table II.

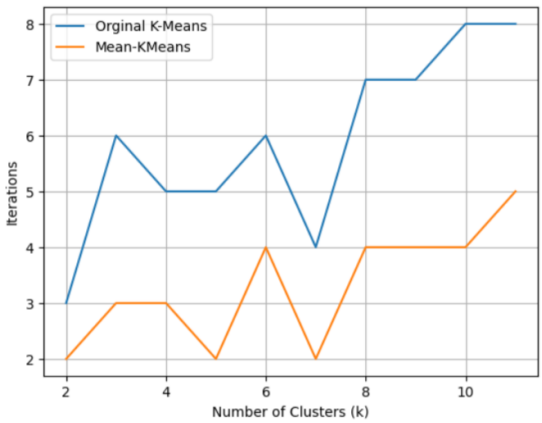

Fig. 1. Iterations comparison on the Iris dataset.

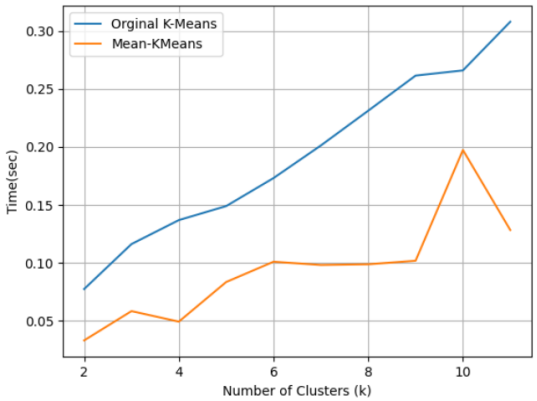

Fig. 2. Runtime comparison on the Iris dataset.

B. Experimental Setup

In our experiment, we begin by using the elbow method [18] to determine the optimal value of k, a crucial parameter in clustering analysis. We then apply our proposed Mean-kMeans shown in algorithm 8 and the traditional k-Means to different datasets from Table II. The determined k value is used for both algorithms. The Silhouette score is used to assess cluster quality by considering data point coherence and distinctiveness within clusters [19]. It combines intra-cluster and nearest neighboring cluster distances, providing a comprehensive measure of cluster effectiveness. These metrics are crucial for a thorough evaluation of clustering outcomes. Additionally, we visualize the resulting clusters (Figure:4)using Principal Component Analysis (PCA) [20]. For each dataset, we apply both Mean-kMeans and k-Means algorithms with k values ranging from 2 to 11. Initial centroids are randomly selected from the datasets. Detailed records of clustering labels and centroids are maintained throughout iterations. To measure optimization benefits, we use two key metrics: the average per-

TABLE I
ITERATION AND SILHOUETTE SCORE COMPARISON AND RESULT ANALYSIS BETWEEN THE ORIGINAL K-MEANS AND MEAN-KMEANS.

Dataset	k	Orig. kmeans Ite.	Mean-Kmeans Ite.	Ite. Imp. (%)	Pct. imp. (%)	Avg. Imp. (%)	sil. scores of Org. Kmeans	sil. scores Mean-Kmeans	Score Inc.	Pct. Imp. (%)	Avg. Imp. (%)
Iris	2	3	2	1	33.333	43.988	0.681	0.687	0.006	0.835	-7.3
	3	6	3	3	50.000		0.553	0.503	-0.050	-9.024	
	.....						.....				
	11	8	5	3	37.500		0.309	0.283	-0.026	-8.352	
Wine	2	5	2	3	60.000	30.158	0.657	0.602	-0.054	-8.287	-2.4
	.....						.....				
	10	3	4	-1	-33.333		0.521	0.510	-0.011	-2.150	
	11	3	2	1	33.330		0.521	0.498	-0.023	-4.422	
Glass	2	3	4	-1	-33.330	20.558	0.594	0.577	-0.017	-2.894	-23
	.....		.....		.....		.....				
	10	4	5	-1	-25.000		0.358	0.264	-0.094	-26.158	
	11	5	4	1	20.000		0.366	0.277	-0.088	-24.2	
Yeast	2	12	3	9	75	73.434	0.263	0.561	0.298	113.149	5.0
	.....						.....				
	11	24	3	21	87.500		0.181	0.153	-0.027	-15.1	
	Overall improvement of iteration across all datasets					42.035					
	Overall improvement of shilhouette score across all datasets										-6.9

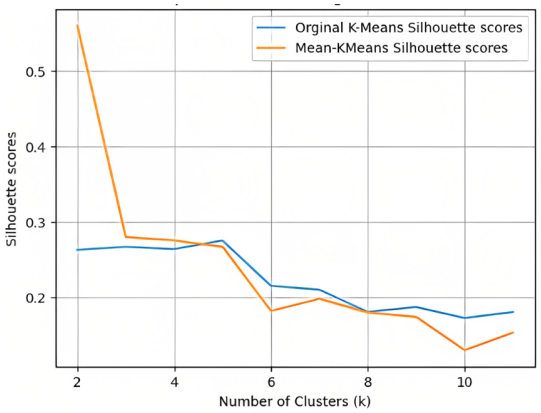

Fig. 3. Silhouette scores comparison on yeast dataset.

centage improvement in the iteration count, and the silhouette score.

TABLE II
EXPERIMENT DATASETS

Dataset	#Instances	#Dimensions
Iris	150	4
Glass	214	9
Wine	178	13
Yeast	1484	8

We calculate the individual improvement percentage (Eq:8), average percentage improvement (Eq:10), and the overall average improvement percentage across all datasets (Eq:11), providing insights into how optimization affects convergence and cluster quality. Our presentation concludes with a Comparative graph similar to Figures 1 and 3. This graph visually highlights the differences between the two approaches, aiding in understanding performance disparities between k-Means and mean-kMeans algorithms across varying k values.

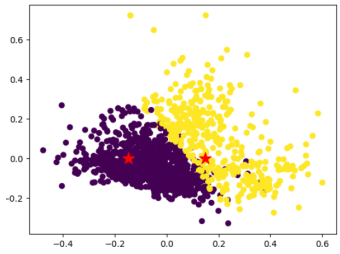

Fig. 4. Mean-kMean cluster Visualization on yeast dataset

$$P_i = \frac{O_i - U_i}{O_i} \times 100 \tag{8}$$

$$\text{T.I.P} = \sum_{i=1}^{n} P_j \tag{9}$$

$$P_{\text{avg}} = \frac{\text{T.I.P}}{n} \tag{10}$$

$$P_{\text{overall}} = \frac{\sum_{j=1}^{N} P_{\text{avg}}^{(j)}}{N} \tag{11}$$

In the equations 8, 9, 10, and 11 - n is the total number of k values. - N is the total number of datasets. - P_i is the Individual improvement percentage of each k value of j^{th} dataset. - $P_{\text{avg}}^{(i)}$ is the average improvement percentage for the j^{th} dataset. - O_i is the original iteration count or silhouette score for the i^{th} k value of j^{th} dataset. - U_i is the updated iteration count or silhouette score for the i^{th} k value of j^{th} dataset. - T.I.P is Total Improvement Percentage. - P_{avg} is the average improvement percentage across all the k values of j^{th} dataset. - P_{overall} is the overall average improvement percentage across all datasets.

C. Experimental Results

The experiment's results (Table I) highlight the k-Means clustering approach across Glass, Iris, Wine, and Yeast datasets. Notably, the Yeast dataset shows remarkable performance improvement of 73.43% (Table I). This positions it as the top performer. A positive silhouette score increase of 5.078% accompanies this progress (Figure 43). In contrast, the Iris dataset demonstrates a 43.988% increase in iteration efficiency (Figure 1), but a negative 7.284% silhouette score deviation raises concerns. Despite a positive increase of 20.55% in the iteration for the Glass dataset, it still experienced a significant negative impact of 23.032% in silhouette score, pointing to compromised overall performance. Across various k values, a conspicuous decline in cluster quality is evident, yet this drawback is offset by reduced coverage due to the method's computational efficiency compared to the original k-Means. Conversely, the wine dataset demonstrates more favorable results with acceptable iteration performance and silhouette scores, showing no alarming deficits. The proposed approach, when combining results across datasets, demonstrates a significant 42.035% increase in iteration efficiency. Additionally, the comparisons of time, as shown in Fig.2, further support the claim of improvement in justifying interactions, indicating its efficacy. However, the simultaneous -6.92% decrease in silhouette scores, as shown in Table:I, raises concerns about clustering quality.In the presented table I, it is observed that, for each k value ranging from 2 to 11, the number of iterations consistently remains below the maximum limit, as the maximum iterations are initialized to 100. This empirical evidence substantiates the reliable convergence of our proposed Mean-kMeans algorithm within the prescribed iteration limit. These findings emphasize the method's nuanced effectiveness, influenced by diverse datasets and their interplay with silhouette scores. The trade-off is evident: quicker iterations might come at the expense of clustering accuracy. Balancing iteration speed with cluster quality becomes crucial, demanding careful consideration by practitioners and researchers depending on the context.

V. Conclusion

The proposed K-Means clustering algorithm demonstrates a promising approach for enhancing the efficiency of traditional K-Means clustering. It achieves a notable average improvement of 42.035% in iteration convergence across diverse datasets. However, this efficiency gain is accompanied by a trade-off, as silhouette scores exhibit a decrease of -6.92%, suggesting a potential compromise in clustering quality. Future research directions include investigating the algorithm's adaptability across various datasets, refining the proposed K-Means balance between speed and clustering quality, and exploring hybrid approaches by combining proposed K-Means with other clustering techniques to achieve improved results.

References

[1] A. Glielmo, B. E. Husic, A. Rodriguez, C. Clementi, F. Noé, and A. Laio, "Unsupervised learning methods for molecular simulation data," *Chemical Reviews*, vol. 121, no. 16, pp. 9722–9758, 2021.

[2] D. M. Farid, A. Nowe, and B. Manderick, "A feature grouping method for ensemble clustering of high-dimensional genomic big data," in *2016 Future Technologies Conference (FTC)*. IEEE, 2016, pp. 260–268.

[3] J. Parraga-Alava, R. A. Caicedo, J. M. Gómez, and M. Inostroza-Ponta, "An unsupervised learning approach for automatically to categorize potential suicide messages in social media," in *2019 38th International Conference of the Chilean Computer Science Society (SCCC)*. IEEE, 2019, pp. 1–8.

[4] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, 2022.

[5] K. P. Sinaga and M.-S. Yang, "Unsupervised k-means clustering algorithm," *IEEE access*, vol. 8, pp. 80 716–80 727, 2020.

[6] M.-S. Yang and K. P. Sinaga, "A feature-reduction multi-view k-means clustering algorithm," *IEEE Access*, vol. 7, pp. 114 472–114 486, 2019.

[7] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020.

[8] P. Fränti and S. Sieranoja, "How much can k-means be improved by using better initialization and repeats?" *Pattern Recognition*, vol. 93, pp. 95–112, 2019.

[9] R. M. Alguliyev, R. M. Aliguliyev, and L. V. Sukhostat, "Parallel batch k-means for big data clustering," *Computers & Industrial Engineering*, vol. 152, p. 107023, 2021.

[10] S. M. Javidan, A. Banakar, K. A. Vakilian, and Y. Ampatzidis, "Diagnosis of grape leaf diseases using automatic k-means clustering and machine learning," *Smart Agricultural Technology*, vol. 3, p. 100081, 2023.

[11] K. Peng, V. C. Leung, and Q. Huang, "Clustering approach based on mini batch kmeans for intrusion detection system over big data," *IEEE Access*, vol. 6, pp. 11 897–11 906, 2018.

[12] D. Feldman, M. Schmidt, and C. Sohler, "Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering," *SIAM Journal on Computing*, vol. 49, no. 3, pp. 601–657, 2020.

[13] D. M. Saputra, D. Saputra, and L. D. Oswari, "Effect of distance metrics in determining k-value in k-means clustering using elbow and silhouette method," pp. 341–346, 2020.

[14] T. M. Ghazal, "Performances of k-means clustering algorithm with different distance metrics," *Intelligent Automation & Soft Computing*, vol. 30, no. 2, pp. 735–742, 2021.

[15] M. Rezaei, "Improving a centroid-based clustering by using suitable centroids from another clustering," *Journal of classification*, vol. 37, pp. 352–365, 2020.

[16] A. Fahim, "K and starting means for k-means algorithm," *Journal of Computational Science*, vol. 55, p. 101445, 2021.

[17] A. Kuraria, N. Jharbade, and M. Soni, "Centroid selection process using wcss and elbow method for k-mean clustering algorithm in data mining," *International Journal of Scientific Research in Science, Engineering and Technology*, pp. 190–195, 2018.

[18] H. Humaira and R. Rasyidah, "Determining the appropiate cluster number using elbow method for k-means algorithm," in *Proceedings of the 2nd Workshop on Multidisciplinary and Applications (WMA) 2018, 24-25 January 2018, Padang, Indonesia*, 2020.

[19] K. R. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," in *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*. IEEE, 2020, pp. 747–748.

[20] N. F. Jansson, R. L. Allen, G. Skogsmo, and S. Tavakoli, "Principal component analysis and k-means clustering as tools during exploration for zn skarn deposits and industrial carbonates, sala area, sweden," *Journal of Geochemical Exploration*, vol. 233, p. 106909, 2022.