A Lightweight Deterministic Initialization Method for K-Means Clustering

Dong Hung Ha[1,2*] and *Thanh Hien* Lam[1**]

[1]Lac Hong University, Dong Nai, Vietnam
[2]Faculty of Information Technology, Van Lang School of Technology, Van Lang University, Ho Chi Minh City, Vietnam

Abstract. The K-Means clustering algorithm is a fundamental tool in machine learning, but its performance is often strongly affected by the instability of traditional random initialization methods, which can lead to convergence to poor local optima. Although many studies have proposed deterministic models to address this issue, they often involve high computational cost with $O(n^2)$ complexity. This paper introduces a new engineering approach that is lightweight and efficient. Specifically, the method uses the L2 norm to directly map each multi-dimensional data point into a one-dimensional scalar value. This value serves as a sorting criterion and is stored in a dictionary data structure to deterministically extract K initial centroids. Experimental evaluation shows that the proposed method completely eliminates variability, fixing the Adjusted Rand Index (ARI) at stable values of 0.6345 for the Digits dataset and 0.7302 for the Iris dataset across all runs. These results demonstrate that a practical data structure-based approach can achieve high reliability with very low computational cost.

1 Introduction

Data clustering is one of the most fundamental and important problems in unsupervised machine learning and data mining. Among existing clustering algorithms, the K-Means algorithm [1, 2] remains the most widely used method due to its simplicity, high computational efficiency, and strong scalability [3]. The core mathematical objective of K-Means is to partition a multi-dimensional dataset into K distinct clusters by minimizing the Sum of Squared Errors (SSE).

Despite its popularity and practical efficiency, the traditional K-Means algorithm is highly dependent on the initialization of cluster centroids [4, 5]. If the initial centroids are placed in sub-optimal positions, the algorithm can easily become trapped in poor local optima. Consequently, we obtain not only unstable clustering results, but also empty clusters, increasing the number of iterations required for convergence, and computational resources [6]. Therefore, the problem of finding an optimal centroid initialization method has become a topic of great interest in the research community for several decades.

One of the most well-known methods for centroid initialization is K-Means++ [7]. This algorithm uses a probabilistic sampling mechanism proportional to the squared distance (D^2 sampling). K-Means++ helps the initial points to be more widely spread in the data space and provides a theoretical guarantee. However, it is still probabilistic in nature and requires a sequential selection process, leading to high computational cost when dealing with very large-scale datasets. To completely eliminate randomness, researchers have shifted toward deterministic initialization methods [8]. These techniques often rely on the data structure. For example, the PCA-Part algorithm partitions data along the direction of the principal component with the largest variance [5, 9]. On the other hand, at a more complex level, some studies apply highly sophisticated approaches, such as graph-based density estimation [3] , or metaheuristic algorithms [10, 11]. These complex models provide very high accuracy and an excellent ability to avoid local optima.

The studies mentioned above indicate a gap between practical computational efficiency and complex theoretical foundations. Most recent academic research tends to rely heavily on metaheuristic algorithms or space-based optimization models (e.g., density estimation or pairwise distance computations with $O(n^2)$ complexity), which consume a large amount of computational resources. In contrast, fast linear methods such as PCA-Part are often too simple, as they only aggregate attributes without providing strong geometric reasoning in a multi-dimensional space. In practical software engineering and system development, engineers seek algorithms that achieve high stability while maintaining lightweight code, simple structure, and fast execution.

To fill this gap, this study proposes a new initialization method for K-Means centroids. The novelty of this method does not come from introducing a new mathematical foundation, but from an engineering approach. Specifically, it directly transforms multi-dimensional data into one-dimensional L2 keys through a dictionary structure to extract representative center points. The proposed method provides a practical, lightweight, and fully deterministic initialization technique.

*e-mail: hung.hd@vlu.edu.vn
**e-mail: lthien@lhu.edu.vn

Instead of computing the pairwise distance matrix between all data points, our technical solution directly maps each multi-dimensional data point to a scalar value using the L2 norm (Euclidean distance from the origin). This representative value serves as a unique key to classify and store each data point in a dictionary-based data structure. This process creates a well-indexed list of records, ordered deterministically according to their geometric magnitude. Finally, by traversing this ordered data space to extract representative points, the algorithm can accurately determine the initial centroid positions, providing strong stability from the very first iteration.

The remainder of this paper is organized as follows. Section 2 presents the background; whereas Section 3 describes the proposed method. Section 4 provides numerical examples. Finally, the conclusion is given.

2 Background and Preliminaries

2.1 Similarity of two data points

In machine learning and data clustering, determining the similarity between observations is a key factor for grouping. This similarity is usually evaluated through the L2 distance as follows.

L2 Distance (Euclidean Distance): This is the most common measure to determine the relative distance between two data points $\mathbf{x}$ and $\mathbf{y}$ in a d-dimensional space. Computing the L2 distance for all pairs of points requires a distance matrix with complexity $O(n^2)$. The L2 distance between two points is defined as follows.

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{\sum_{j=1}^{d} (x_j - y_j)^2}. \tag{1}$$

In traditional clustering algorithms, this measure is typically used in the step of assigning data points to the nearest centroid.

L2 Norm: In contrast to the relative distance between two points, the L2 norm of a data point $\mathbf{x}$ represents its geometric magnitude (Euclidean distance) from the origin $(0, 0, \ldots, 0)$. The L2 norm is defined as follows.

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{j=1}^{d} x_j^2}. \tag{2}$$

The L2 norm plays a central role in the proposed engineering approach. Instead of computing the expensive $O(n^2)$ distance matrix, the L2 norm allows each multi-dimensional data point to be independently mapped to a single scalar value with only $O(n)$ complexity. This scalar value is then used as a geometric key for classification and extraction of representative center points.

2.2 K-Means Algorithm

K-Means is the most widely used partition-based clustering algorithm, with the objective of dividing a dataset $\mathbf{X}$ into K clusters such that the Within-Cluster Sum of Squared Errors (SSE) is minimized. The objective function is defined as Eq. (3).

$$\text{SSE} = \sum_{i=1}^{K} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2. \tag{3}$$

The standard K-Means algorithm operates based on a greedy iterative strategy as follows.

Step 1 (Initialization):
Randomly select K data points from the dataset as the initial centroids (e.g., Forgy/Random method).

Step 2 (Cluster Assignment):
For each data point $\mathbf{x} \in \mathbf{X}$, compute the L2 distance to all K current centroids. Assign $\mathbf{x}$ to the cluster with the nearest centroid.

Step 3 (Centroid Update):
Recompute the centroid $\boldsymbol{\mu}_i$ of each cluster by taking the mean of all data points assigned to that cluster in Step 2.

Step 4 (Convergence Check):
Repeat Step 2 and Step 3 iteratively. The algorithm stops when the centroids no longer change between two consecutive iterations, or when a predefined maximum number of iterations is reached.

3 The proposed algorithm

The proposed algorithm is carried out step-by-step as follows.

Step 1: Space Mapping via L2 Norm
The algorithm scans the dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ only once. For each data point $\mathbf{x}_i$ in a d-dimensional space, we computes its L2 norm to produce a scalar value k_i as Eq. (4).

$$k_i = \|\mathbf{x}_i\|_2 = \sqrt{\sum_{j=1}^{d} x_{i,j}^2}. \tag{4}$$

Here, k_i serves as a unique geometric signature representing the spatial characteristics of $\mathbf{x}_i$. This operation is fully independent for each point, resulting in a lightweight computational complexity of $O(n)$.

Step 2: Dictionary Indexing
The scalar value k_i is then used as a key to store the corresponding multi-dimensional data point $\mathbf{x}_i$ into a dictionary data structure. This structure systematically organizes and orders the entire dataset according to a one-dimensional geometric magnitude in a fully deterministic manner.

Step 3: Deterministic Centroid Extraction
Based on the structured key space $\{k_i\}$ in the dictionary, the algorithm divides this ordered list into K representative partitions. By traversing these partitions, it directly extracts K center data points to serve as initialization seeds. Due to the clear separation provided by the L2 keys, the positions of these K centroids are fixed and consistent across different runs.

Step 4: Clustering Optimization
Using the high-quality initialized centroids, the algorithm proceeds with the standard K-Means process. In this step, each data point is assigned to the nearest centroid $\boldsymbol{\mu}_c$, and

the centroids are iteratively updated until the SSE converges. Since the initial positions are geometrically well-distributed through the L2-based dictionary structure, the algorithm avoids many unnecessary iterations, leading to significant savings in overall clustering time (Clustering Time and Total Time).

4 Numerical result

To evaluate the performance of our proposed method, experiments are conducted on two standard datasets: Iris (simple, low-dimensional) and Digits (complex, high-dimensional). The main evaluation metrics include the Adjusted Rand Index (ARI) to measure clustering accuracy, along with Clustering Time and Total Time to assess computational efficiency. The algorithm is executed 10 independent runs to evaluate its stability.

4.1 Analysis on the Iris Dataset

The Iris dataset consists of 150 samples with 3 natural clusters. Due to its relatively simple structure, the traditional K-Means algorithm achieves reasonably good results but still shows instability. Based on the statistical results in Table 1, the original random K-Means records ARI values ranging from 0.7163 to 0.7302 over 10 runs, with an average of 0.7233. This variation indicates that random initialization may lead the algorithm to different local optima. In contrast, the proposed algorithm demonstrates perfect stability. The ARI consistently reaches the value of 0.7302 in 100% of the runs. In terms of computational time, the average total execution time of the proposed method (0.1443s) is comparable, and slightly better, than the random method (0.1454s). This result shows that the L2 norm mapping technique is highly efficient for low-dimensional data.

4.2 Analysis on the Digits Dataset

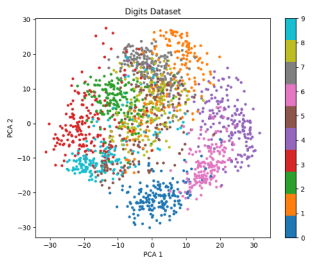

Figure 1. Ground truth of original digits after PCA

The difference in stability becomes most evident on the Digits dataset, which consists of high-dimensional handwritten digit images divided into 10 clusters. For the random K-Means algorithm, the ARI in Table 2 shows a very large variation. The minimum drops to 0.5633 while the maximum reaches 0.7314 (with an average of 0.6474).

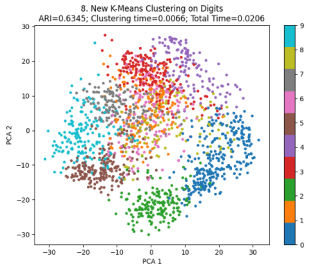

Figure 2. Clustering visualization of Digits using the proposed method

This high variability forces practitioners to run the algorithm multiple times in order to select the result with the lowest SSE. By applying the proposed engineering approach, the algorithm once again achieves full determinism, fixing the ARI at 0.6345 across all runs. Although this value is lower than the best random case (0.7314), it is obtained consistently from the very first execution. In terms of computational cost, the clustering time of the proposed method (approximately 0.0049s–0.0066s) remains competitive with random K-Means. The average total time is slightly higher (0.1612s compared to 0.1490s). This difference of about 0.012 seconds corresponds to the cost of the L2 distance computation and dictionary storage operations. However, this trade-off is reasonable and more beneficial than running the random algorithm many times to obtain a stable result.

5 Conclusion

This study has demonstrated the feasibility of the engineering approach based on L2 distance mapping for K-Means centroid initialization. By transforming multi-dimensional data into scalar values used as keys in a dictionary structure, the algorithm organizes the spatial order of the data in a fully deterministic way with only $O(n)$ time complexity. With a negligible additional cost of a few milliseconds, the method completely removes randomness and achieves stable ARI values on both the Iris and Digits datasets from the first iteration. However, the proposed method still has some limitations. From a mathematical perspective, the L2 norm does not capture the direction of vectors, which may lead to the issue of "distance duplication" for data points located on the same hypersphere. As a result, although the ARI is stable, it may be lower than the best possible values. For future work, this approach has strong potential for further evaluation on noisy datasets, large-scale data (Big Data), and integration as an initialization step for more advanced clustering algorithms.

References

[1] S. Lloyd, Least squares quantization in pcm, IEEE transactions on information theory **28**, 129 (1982).
[2] J.B. McQueen, Some methods of classification and analysis of multivariate observations, in *Proc. of*

Table 1. Statistical results of ARI and execution time on the Iris dataset

Algorithm	Avg ARI	Max ARI	Min ARI	Avg Time (s)	Iters to Conv
K-Means Random	0.7233	0.7302	0.7163	0.1455	9
Proposed algorithm	0.7302	0.7302	0.7302	0.1443	3

Table 2. Statistical results of ARI and execution time on the Digits dataset

Algorithm	Avg ARI	Max ARI	Min ARI	Avg Time (s)
K-Means Random	0.6474	0.7314	0.5633	0.1490
Proposed Method	0.6345	0.6345	0.6345	0.1612

5th Berkeley Symposium on Math. Stat. and Prob. (1967), pp. 281–297

[3] J. Yang, Y.K. Wang, X. Yao, C.T. Lin, Adaptive initialization method for k-means algorithm, Frontiers in Artificial Intelligence **4**, 740817 (2021).

[4] A. Kumar, S. Kumar, Density based initialization method for k-means clustering algorithm, I.J. Intelligent Systems and Applications pp. 40–48 (2017). 10.5815/ijisa.2017.10.05

[5] T. Su, J. Dy, A deterministic method for initializing k-means clustering, in *16th IEEE International Conference on Tools with Artificial Intelligence* (IEEE, 2004), pp. 784–786

[6] M.E. Celebi, H.A. Kingravi, P.A. Vela, A comparative study of efficient initialization methods for the k-means clustering algorithm, Expert Systems with Applications **40**, 200 (2013).

[7] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA)* (2007), Vol. 7, pp. 1027–1035

[8] L. Chen, D.R. Roe, M. Kochert, C. Simmerling, R.A. Miranda-Quintana, k-means nani: an improved clustering algorithm for molecular dynamics simulations, Journal of Chemical Theory and Computation **20**, 5583 (2024).

[9] A.Q. Obaid, M. Alabbas, A comparative evaluation of initialization strategies for k-means clustering with swarm intelligence algorithms, Iraqi Journal for Electrical and Electronic Engineering **20**, 271 (2024).

[10] T. Nguyen-Trang, T. Nguyen-Thoi, T. Truong-Khac, A.T. Pham-Chau, H.L. Ao, An efficient hybrid optimization approach using adaptive elitist differential evolution and spherical quadratic steepest descent and its application for clustering, Scientific Programming **2019**, 7151574 (2019).

[11] R. Sarmah, R.K. Prasad, M.T. Singh, S. Sarmah, A. Boruah, S. Majumder, M. Rahman, Mk-means: Improving partition-based clustering with optimized centroid initialization, IEEE Access **14**, 1758 (2025).