Planar Sample Point Clustering: Comparison and Analysis of Multiple Algorithms

ZiJun Li
1024041116
Nanjing University of Posts and Telecommunications
School of Computer Science
Nanjing, China

***Abstract*—This paper evaluates the effectiveness of the K-means clustering algorithm in solving clustering problems of points on a plane. It identifies common issues such as random initial centroid selection and convergence to local optima, and proposes optimization strategies to improve the quality and efficiency of the clustering process. The study introduces the basic principles of the K-means algorithm, including centroid initialization, point assignment, and iterative centroid updating. Experiments were conducted using a dataset of 3,000 two-dimensional points from the xclara.csv file on GitHub, and a sensitivity analysis of the algorithm parameter K was performed to understand its impact on the clustering results. Such analysis is crucial for determining how the selection of K affects the stability and accuracy of the clustering results. Furthermore, the performance of the optimized K-means algorithm was compared with seven other clustering algorithms, including hierarchical clustering, DBSCAN, mean shift clustering, Gaussian mixture models, spectral clustering, fuzzy C-means clustering, and self-organizing maps. Comparative analysis indicates that the optimized k-means, k-means++, k-medoids, k-modes and k-prototypes algorithms significantly improve the stability and accuracy of clustering results for points on a plane. The findings provide practical guidance for solving clustering problems in the real world and lay a theoretical foundation for further exploration and improvement of existing clustering algorithms.**

***Index Terms*—K-means and extensions, Planar sampling point clustering, Multiple clustering algorithms, Sensitivity analysis.**

I. Introduction

WITH the advent of the information age and the explosion of data, big data analysis has become an essential tool for businesses, governments, and organizations. It enables the extraction of valuable insights from vast datasets, which is crucial for informed decision-making, streamlining business operations, mitigating risks, and maintaining a competitive edge in the global market [1][2][3][4][5]. Cluster analysis, a fundamental technique in the realms of data mining and machine learning, is gaining prominence. Operating in an unsupervised learning context, it uncovers the intrinsic structures within data, facilitating the understanding of complex datasets and the discovery of novel patterns. Clustering algorithms, including K-means, DBSCAN and hierarchical clustering, are being refined to accommodate evolving data characteristics and requirements. As big data technologies advance, the importance of cluster analysis in managing large-scale and high-dimensional datasets is magnified. It must not only address the volume and velocity of data but also ensure that the results are interpretable and visually accessible to non-technical stakeholders. Consequently, research in cluster analysis is not only propelling technological progress but also laying a theoretical foundation and offering practical guidance for addressing intricate real-world challenges.

The K-means algorithm was first proposed by Mac in 1967 as a simple and computationally efficient clustering method. It has become the foundation of exploratory big data analysis and has been widely applied in various fields, including market segmentation, image analysis, and bioinformatics [6][7][8][9]. Although the K-means algorithm is widely popular, it still has certain limitations due to its sensitivity to initial conditions and tendency to discover spherical clusters. To this end, researchers have proposed many variants and improvements, such as K-Means++ and K-Medoids to enhance robustness against outliers, and Elkan's K-Means to accelerate convergence speed [10][11]. Recently the algorithm has been extended to fuzzy logic and complex distance metrics to adapt to non Euclidean spaces.

In addition to the K-means algorithm and its variants, clustering algorithms are divided into five categories: partition-based, hierarchical, density-based, grid-based, and model-based. Common methods include DBSCAN, OPTICS, and Mean Shift, among others [12][13][14]. Each of these algorithms has its own characteristics and applications. For example, DBSCAN can identify clusters of any shape and handle noise data, while OPTICS is more effective when dealing with datasets that have regions of varying density. The Mean Shift algorithm determines the centers of clusters by iteratively finding the peaks of data point density. These methods provide more flexible and powerful clustering capabilities when dealing with large datasets, meeting the needs of data mining and analysis across different fields. In summary, the goal of clustering is to achieve high intra-cluster similarity and low inter-cluster similarity, maximizing the distance between clusters and minimizing the distance between samples within a cluster and its center.

II. K-means algorithm

A. The basic idea of K-means algorithm

Cluster analysis belongs to unsupervised learning, and its core function is to divide a set of data points into several groups or clusters, ensuring that data points within the same

cluster have a high degree of similarity, while data points between different clusters exhibit greater differences.

The K-means clustering algorithm, as a fundamental iterative clustering method, aims to divide an n-dimensional vector dataset D=xi — i=1, ..., n into k clusters [15][16][17]. In this algorithm, xi represents the ith data point. Commonly used distance measures in clustering algorithms include Euclidean distance, Manhattan distance, and Chebyshev distance, among others. Typically, the algorithm uses Euclidean distance as a measure of similarity and minimizes the sum of squared errors (SSE) as the objective function to evaluate the quality of clustering, achieving the division of data points into k clusters through multiple iterations.

a) Euclidean distance between data points:As is shown in Eq. 1, where μ_j is the cluster center of the j-th cluster; X is the data object in dataset D.

$$d(x_i, \mu_j) = \sqrt{\sum_{n=1}^{N}(x_{in} - \mu_{jn})^2} \tag{1}$$

b) Iterative update of centroid:

$$C_i = \{x_p : \|x_p - \mu_i\| \leq \|x_p - \mu_j\|, \forall j \in [1, K]\} \tag{2}$$

$$\mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j \tag{3}$$

For each cluster C_i, calculate the average of all data points within the cluster and use this average as the new centroid, as is shown in Eq. 2 and Eq. 3.

c) Optimization objective function:

$$SSE = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2 \tag{4}$$

As is shown in Eq. 4, the optimization goal of the K-Means algorithm is to minimize the sum of squared distances of all data points to their respective cluster centroids.

The K-means clustering algorithm is a dynamic clustering algorithm that achieves the optimization goal by minimizing the sum of squared errors between data points and the centers of the clusters they belong to. In each iteration, the algorithm recalculates the center of each cluster, which is the mean of all points within the cluster, and then reallocates the data points based on the new centers. This process continues until a stopping condition is met, indicating that it has converged to a stable distribution. For specific algorithm details, refer to Algorithm 1.

Algorithm 1 K-Means Clustering.

Input: Sample set: $D = \{x_1, x_2, x_3, x_4, \ldots, x_n\}$; The number of clusters K.

Algorithm process:

Randomly select K samples from D as the initial mean vectors $\{a_1, a_2, a_3, \ldots, a_k\}$.
repeat
  Leave F_i empty $(i = 1, 2, \ldots, K)$.
  for $j = 1, 2, \ldots, M$ **do**
    Calculate the distance between sample x_j and each mean vector a_i $(i = 1, 2, \ldots, K)$: $d_{ji} = \|x_j - a_i\|^2$.
    Determine the cluster label of x_j based on the nearest mean vector.
    Assign sample x_j to corresponding clusters.
  end for
  for $i = 1, 2, \ldots, K$ **do**
    Calculate a new mean vector a'_i for each cluster.
    if $a'_i \neq a_i$ **then**
      Update the current mean vector a_i to a'_i.
    else
      Keep the current mean vector unchanged.
    end if
  end for
until the current mean vectors are all updated.
Output: Cluster partition $C = \{C_1, C_2, C_3, \ldots, C_K\}$.

B. K-means optimization algorithm

The traditional K-means algorithm is random in the selection of initial centroids, which can lead to slow convergence and the final result being significantly affected by the choice of initial points. Additionally, research has shown that the algorithm has some drawbacks, such as sensitivity to the choice of initial centroids, a tendency to get stuck in local optima, sensitivity to noise and outliers, and the assumption that clusters are convex in shape. To overcome these limitations, researchers have proposed various improved k-means algorithms [18][19].

a) The K-Means++ algorithm: The k-means++ algorithm, proposed by Arthur and Vassilvitskii in 2007, is an improvement on the traditional k-means algorithm. It enhances clustering quality by intelligently selecting initial centroids, thereby reducing the algorithm's sensitivity to the choice of initial centroids and increasing convergence speed [20][21][22].

First, it randomly selects a data point as the first centroid. Then, for the remaining data points, it assigns a probability value to each point based on their distance from the selected centroids. Specifically, the further a data point is from the selected centroids, the higher the probability that it will be chosen as the next centroid. This method ensures that the initial centroids are relatively far apart, avoiding the situation where initial centroids are too concentrated and improving the convergence speed and clustering effect of the algorithm. After selecting the initial centroids, the k-means++ algorithm proceeds with the same iterative process as the traditional K-means algorithm. This process continues until a stopping condition is met, such as reaching a predetermined number of iterations or when the change in cluster centers is very small, indicating that the algorithm has converged to a stable distribution, refer to Algorithm 2.

b) The K-Medoids algorithm: The K-medoids algorithm is a clustering algorithm similar to k-means, designed to divide objects in a dataset into k clusters. However, unlike k-means, k-medoids selects cluster centers that are actual points in the dataset, known as medoids. This makes k-medoids more robust to outliers and noise data, as medoids are not as susceptible to the influence of outliers as centroids in k-means.

Algorithm 2 K-Means++ Clustering.

Input: Sample set: $D = \{x_1, x_2, x_3, x_4, \ldots, x_n\}$; The number of clusters K.

Algorithm process:

1: Select one data point x_{r1} from D randomly as the first centroid c_1;
2: **for** $i = 2, 3, \ldots, K$ **do**
3: For each data point x_j in D, calculate the distance to the closest centroid c_{closest} that x_j is associated with;
4: Choose one data point x_{ri} from D as the new centroid c_i with the maximum squared distance to the closest centroid;
5: **end for**
6: Repeat the K-Means clustering process as described in Algorithm 1 starting with the centroids initialized in steps 1 and steps 2;
7: **Output:** Cluster partition $C = \{C_1, C_2, C_3, \ldots, C_k\}$.

The Partitioning Around Medoids(PAM) algorithm is a classic method for implementing k-medoids, but it has the disadvantage of high computational complexity, especially when dealing with large datasets [23][24]. To address this issue, the improved algorithms such as Clustering LARge Applications(CLARA) and Clustering Large Applications based upon Randomized Search(CLARANS) have been proposed. Particularly, The CLARA selects medoids from multiple samples, whereas CLARANS searches for better medoids within randomly selected neighbors, thus enhancing the efficiency of the algorithm, refer to the Algorithm 3.

Algorithm 3 K-Medoids Clustering.

Input: Sample set: $D = \{x_1, x_2, x_3, x_4, \ldots, x_n\}$; The number of clusters K.

Algorithm process:

1: Randomly select K samples from D as the initial mean vectors $\{a_1, a_2, a_3, \ldots, a_k\}$.
2: **repeat**
3: Leave F_i empty for $i = 1, 2, \ldots, K$.
4: **for** $j = 1, 2, \ldots, M$ **do**
5: Calculate the distance between sample x_j and each mean vector a_i $(i = 1, 2, \ldots, K)$: $d_{ji} = \|x_j - a_i\|^2$.
6: Determine the cluster label of x_j based on the nearest medoid.
7: Assign sample x_j to corresponding clusters.
8: **end for**
9: **for** $i = 1, 2, \ldots, k$ **do**
10: Find the most centrally located point in each cluster to update the medoid.
11: **if** a new medoid is found **then**
12: Update the current medoid a_i.
13: **else**
14: Keep the current medoid unchanged.
15: **end if**
16: **end for**
17: **until** the current medoids are all updated.
18: **Output:** Cluster partition $C = \{C_1, C_2, C_3, \ldots, C_k\}$.

c) The K-Modes algorithm: The k-modes algorithm is a clustering algorithm designed for categorical data (discrete attributes), which updates cluster centroids using modes and is suitable for handling categorical data.

d) The K-Prototypes algorithm: The k-prototypes algorithm combines the features of k-means and k-modes algorithms, allowing it to handle both numerical and categorical data simultaneously, updating cluster centroids by defining a hybrid distance metric [25].

III. Related works

A. Selection of initial cluster centers

The core of K-Means is to first fix the centroids and adjust the category of each sample to reduce the loss value. Then, with each sample's category fixed, adjust the centroids to further decrease the loss value. These two processes alternate in a loop, with the loss value monotonically decreasing until it reaches a minimum, at which point both the centroids and the classification of samples converge.

a) Distance-based: Currently, scholars have proposed the concept of dissimilarity based on Euclidean distance, constructing a dissimilarity matrix to determine the differences between data sample points, and using this to judge whether data sample points can serve as initial clustering centers. Meng Zijian et al. [26] defined the dissimilarity between the k-th attribute of two data points and conducted standardization processing on the data.

$$r_{ij}^{(k)} = \frac{|x_{ik} - x_{jk}|}{\max x_{R_k} - \min x_{R_k}} \tag{5}$$

In Eq. 5 and Eq. 6, where x_{ik} and x_{jk} are the k-th attribute values of data points x_i and x_j, x_{R_k} is the set of all values for the k-th attribute of all data points in D.

$$\text{sum}(i) = \sum_{j=1}^{p} r_{ij} \tag{6}$$

Dong Qiuxian et al. [27] improved this flaw in the algorithm by identifying all sample points xi corresponding to the largest dissimilarity parameter value for the calculation.Through this algorithm, the first suitable initial clustering center can be found when the maximum value of dissimilarity parameters is not unique. Experiments show that this improved algorithm has higher accuracy than the original algorithm and also reduces the number of iterations.

b) Density-based: Many academics have suggested enhancements to the k-means clustering algorithm that leverage density-based approaches. Xue Yinxi et al. [28] proposed the KMS-GOSD algorithm, a globally optimized K-means clustering algorithm based on sample density. This algorithm obtains a pre-estimated density of all clustering centers during the initial iteration using a Gaussian model and incorporates a bias operation during the iterative process of updating the clustering centers. Cai Yuhao et al. [29] proposed the WLV-K-means algorithm, which optimizes the initial clustering centers by weighting the local variance.

B. Selection of K value

During the selection of the K value, when performing cluster analysis on any dataset, it is often difficult to precisely choose the most appropriate K value because the determination of the number of clusters K usually relies on the subjective experience of the researcher. Many researchers proposed that in most cases, the number of clusters should be between 2 and the square root of the number of samples in the dataset n, and provided corresponding methods for estimating the K value, in practical applications, whether for text data or image data, the determination of the K value remains unclear. Researchers find it difficult to arrive at the so-called correct answer through subjective analysis [30][31][32]. The determination of this value depends on the distribution characteristics of the dataset and the clustering resolution desired by the user.

a) Elbow method: The elbow method is a technique that involves plotting the within-cluster sum of squares (SSE) against the number of clusters (K) and identifying the elbow point on the curve as the optimal number of clusters.The sum of squared errors (SSE) is then calculated, and the resulting E-K curve is generated. However, the limitation of this method is that for some sets of sample objects, the E-K curve may not have a clear elbow point, or the point may not correspond to the optimal number of clusters, which can occur frequently [33].

b) Optimization algorithms: Optimization algorithms can also determine the K value [34]. Dong Shipenget al. [35] proposed a method for determining the K value based on Euclidean distance and spatial distance between neighboring points, for secondary screening of neighboring points.

c) Silhouette coefficient: The Gap statistic in Eq. 7 and Eq. 8 is based on principles of information theory and statistics, aiming to balance the complexity of the model with the goodness of fit to the data. By comparing the clustering performance of actual data with reference data under different numbers of clusters, the Gap statistic can identify the intrinsic structure of the data, avoiding issues of overfitting or underfitting.

$$\text{Gap}(K) = E(\log D_{\tilde{k}}) - \log D_{\tilde{k}} \tag{7}$$

$$k^* = \arg\max_k \text{Gap}(k) \tag{8}$$

C. Outlier preprocessing

During the clustering process, the presence of outliers to some extent affects the selection of initial clustering centers. Currently, research on outliers has also become a popular direction, with many researchers attempting to preprocess the data using various methods to reduce the impact of outliers on clustering, thereby improving the efficiency of the algorithm.

a) Optimization algorithms: Distance-based outlier detection methods are computationally simple and can be applied to any dataset where data points can be computed, and are independent of the distribution of the dataset. Song Sinan [36] proposed the WDNForest outlier detection algorithm, which defines local and global outlier functions based on the differences in density and distance between normal data points and outlier data points, and calculates outlier values to detect outlier data.

b) Density-based: Yang Hong et al. [37] used the LOF algorithm for outlier detection, conducted data preprocessing on the dataset, and obtained the dense point dataset iris-1 and the outlier dataset iris-2. Then, they performed K-means clustering on the iris-1 dataset and obtained the clusters based on the criterion function.

$$SSE_2 = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{SSE}{(c_i - c_j)} \tag{9}$$

The algorithm shown in Eq. 9 mainly improved upon the traditional criterion functions, which only considered intra-class similarity, by taking into account inter-class differences, further optimizing the clustering results.

D. Multiple clustering algorithms

In addition to partition-based clustering algorithms, hierarchical clustering, density-based clustering, grid-based clustering, and model-based clustering methods have also been extensively studied.

a) Hierarchical clustering: The hierarchical clustering method decomposes a given dataset into a hierarchical structure until a certain condition is met. Among these methods, the BIRCH algorithm is a very efficient clustering technique for Euclidean vector space data. It clusters this type of data effectively with just one pass, efficiently handling outliers.

b) DBSCAN clustering: DBSCAN clustering is a density-based clustering algorithm [38]. It categorizes data points into core points, border points, and noise points, forming clusters by continuously expanding density-connected sample points. DBSCAN is capable of discovering clusters of arbitrary shapes and effectively identifying noise points in space, as is shown in Eq. 10.

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{S_i + S_j}{d(C_i, C_j)} \right) \tag{10}$$

where Si is the average distance from points to the cluster center, and d(Ci,Cj) is the distance between cluster centers.

c) Mean Shift clustering: Mean shift clustering is a non-parametric density-based clustering algorithm. It iteratively updates the positions of data points, moving them towards areas of higher density, and eventually aggregates to form clusters. It does not require the number of clusters to be specified in advance and has advantages in dealing with clusters of complex shapes.

d) Gaussian Mixture Model Clustering: Gaussian Mixture Models refer to the linear combination of multiple Gaussian distribution functions, typically used to address situations where data within the same set contains multiple different distributions. This method can fit any type of distribution and is suitable for complex datasets.

e) Spectral Clustering: Spectral clustering is a graph-based clustering algorithm. It considers data points as nodes in a graph, their similarities as edges, and then performs clustering using the graph's Laplacian matrix. Spectral clustering can handle non-spherical data and is robust to high-dimensional data and noise.

f) Fuzzy C-Means clustering: Fuzzy C-Means clustering is a soft clustering algorithm that allows a data point to belong to multiple clusters with varying degrees of membership. Fuzzy C-Means clustering optimizes an objective function to find the best cluster centers and membership matrix, minimizing the weighted sum of distances from data points to cluster centers.

g) Self-Organizing Map Clustering: Self-Organizing Maps (SOM) are a type of neural network model used for clustering and data visualization. They map data points onto a low-dimensional, usually two-dimensional grid, while preserving the topological relationships between data points. This approach is particularly useful for handling high-dimensional data and uncovering latent structures within the data [39].

E. Performance evaluation indicators

Validation methods for unsupervised internal indicators evaluate clustering partitions based on the set structure information of the dataset, considering aspects such as compactness, separability, connectivity, and overlap. Common internal indicators [40][41][42] include the Silhouette Coefficient, Calinski-Harabasz index, and Davies-Bouldin Index.

a) Silhouette Coefficient Index: For a set of samples, its Silhouette Coefficient is the average of the Silhouette Coefficients of all the samples. The Silhouette Coefficient ranges from [-1,1], the closer the samples within the same class are to each other, and the further the samples of different classes are from each other, the higher the score.

b) Calinski-Harabasz Index: The essence of the Calinski-Harabasz index is the ratio of the distance between clusters to the distance within clusters, also known as the variance ratio criterion. When clustering a dataset X of size N into K categories, compactness is measured by the sum of squared distances from each point within a cluster to the cluster center, while separation is measured by the sum of squared distances from each cluster center point to the center point of the dataset.

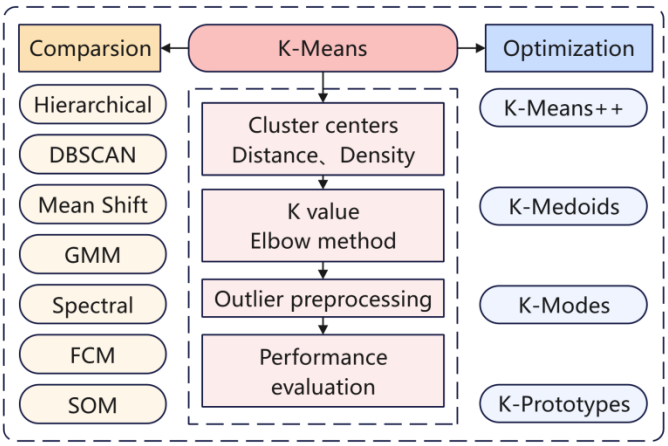

Fig. 1: Overview of the research work

c) Davies-Bouldin Index: The Davies-Bouldin index was proposed by David L. Davies and Donald Bouldin and is a measure used to evaluate clustering algorithms. DB calculates the sum of the average within-cluster distances for any two clusters divided by the distance between their centers, and then finds the maximum value. A smaller DB indicates smaller within-cluster distances and larger between-cluster distances.

IV. Sample Clustering

A. Overview Framework

The work of this paper includes a systematic study of the k-means algorithm, and based on its advantages and disadvantages, an improved algorithm for k-means clustering is implemented, with a comparative analysis of k-means++, k-modes, and k-medoids. In addition to analyzing partition-based clustering, this paper compares the other four major categories of clustering algorithms as is shown in Figure 1, validating and analyzing the performance of different clustering algorithms using a Python experimental environment and datasets.

B. Datasets

In the data preprocessing stage, we completes outlier detection and elimination based on density, ensuring the efficiency and robustness of subsequent clustering analysis. For clustering analysis, we import the dataset xclara.csv into Python and convert it into a two-dimensional array. After data preprocessing, outliers are filtered out, and normalization is performed on two-dimensional features. The distribution visualization is shown Figure 2, for specific details refer to the appendix.

Fig. 2: Data Plane Distribution plot

C. Parameter determination

a) Initial cluster centers: The original K-means algorithm initially randomly selects k points from the dataset as the clustering centers. The selection of the positions of the k initialized centroids has a significant impact on the final clustering results and the running time, hence it is necessary to choose appropriate k centroids. If the selection is purely random, it could lead to slow convergence of the algorithm. Compared to k-means, k-means++ selects the initial centroids using a roulette wheel method that favors data points further away from the current centroids.

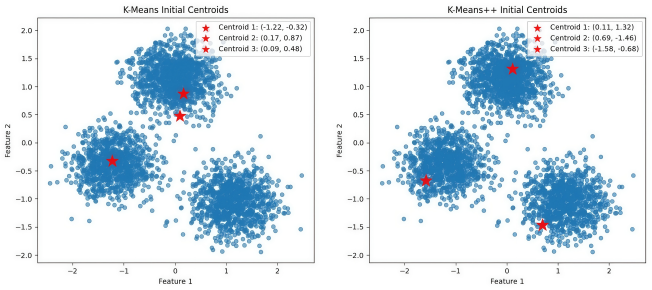

Fig. 3: Initial centroids of k-means and k-means++

Figure 3 shows that for the xclara.csv dataset, the k-means algorithm randomly selects initial centroids, whereas k-means++ selects them based on probability, thereby achieving faster convergence speed. In detection applications, the k-means++ algorithm introduces a weighted probability mechanism to increase the spacing between centroids by improving the strategy for selecting initial clustering centers. This strategy helps the centroids to more comprehensively cover the diversity of the dataset, and experiments verify that it enhances the model's performance in object detection tasks.

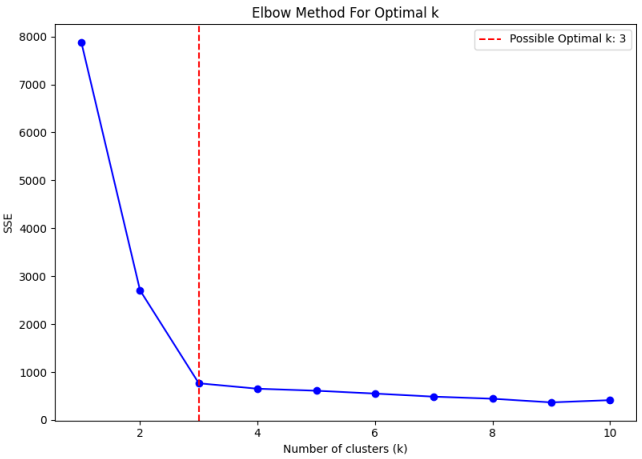

Fig. 4: Elbow method

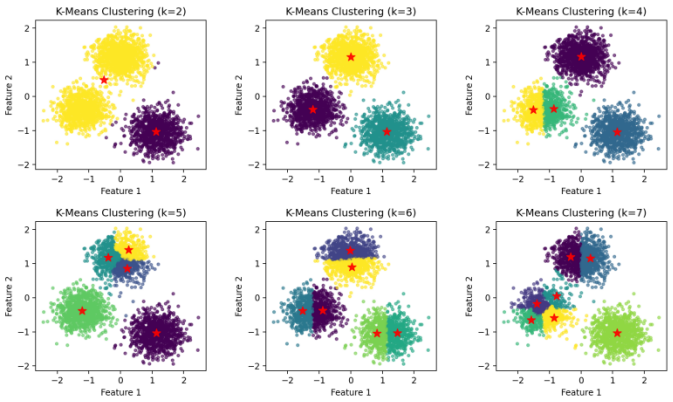

Fig. 5: K-means clustering with different K values

b) K-value sensitivity analysis: Figure 4 shows that, according to the elbow method, the optimal K value is determined to be 3, dividing the area into 3 clustering regions. Additionally, by visualizing the clustering effects, Figure 5 indicates that choosing other K values does not align with the actual situation and results in lower clustering performance.

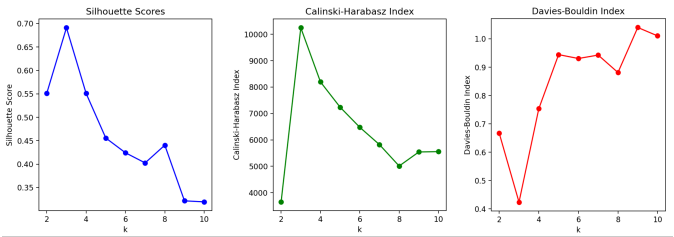

Fig. 6: K-value sensitivity analysis

Figure 6 shows the sensitivity of the k value to three major evaluation metrics, which again verifies that k=3 is the optimal value for the dataset in this paper.

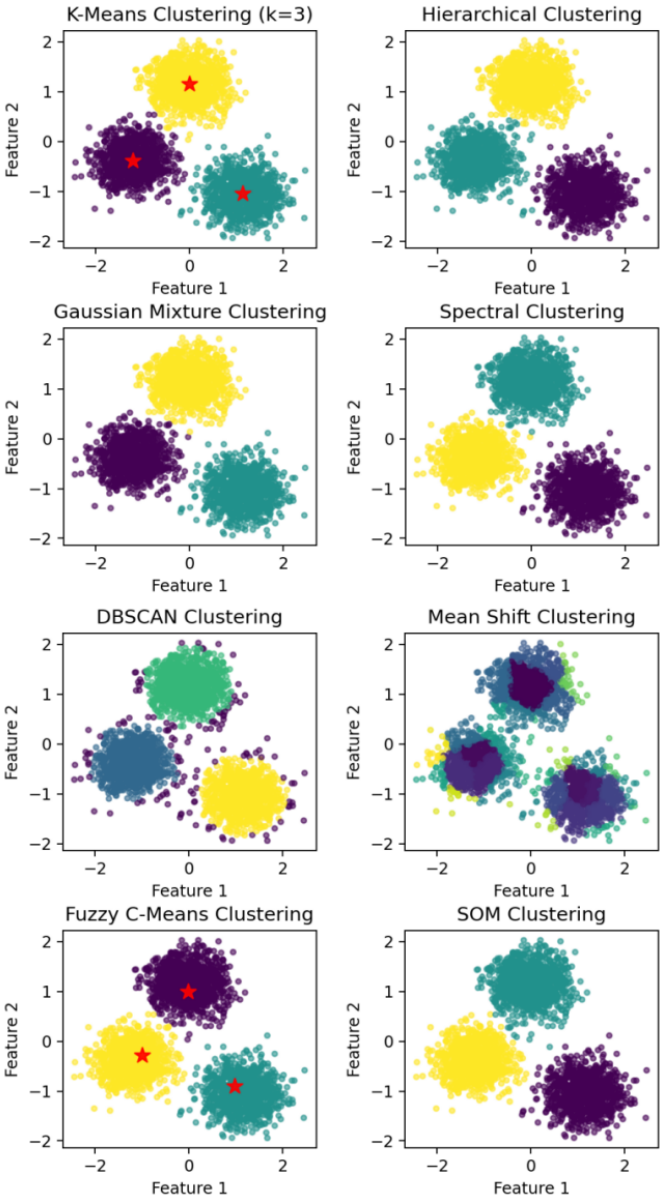

Fig. 7: Multiple clustering algorithms

D. Multiple cluster analysis

This paper, in addition to studying the k-means algorithm and its improved algorithms, conducts a comparative analysis with other clustering algorithms such as hierarchical clustering, DBSCAN, mean shift clustering, Gaussian mixture models, spectral clustering, fuzzy C-means clustering, and self-organizing maps, which are widely used in existing research. Visualizing the results of different clustering algorithms in Figure 7 indicates that, apart from the DBSCAN and Mean Shift algorithms, the performance of the other algorithms is within the same range as the k-means algorithm.

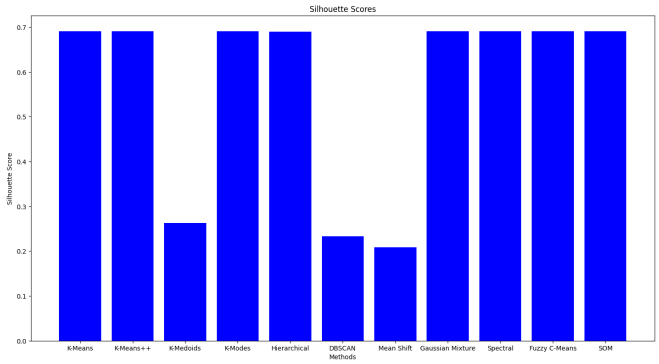

Fig. 8: Silhouette Coefficient Index

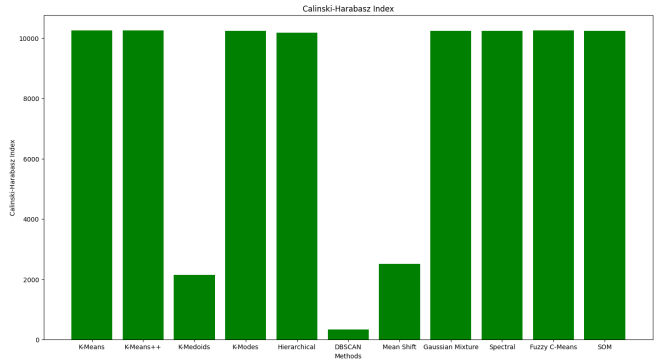

Fig. 9: Calinski-Harabasz Index

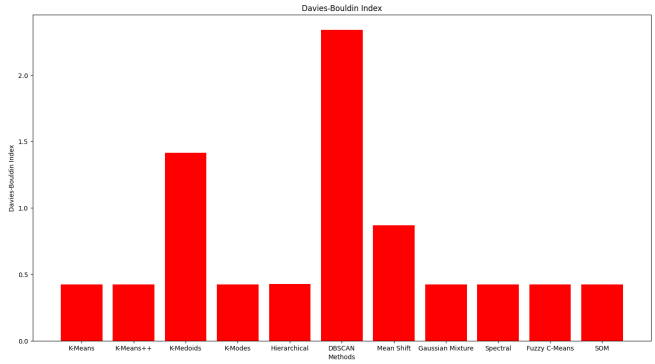

Fig. 10: Davies-Bouldin Index

Experiments show that the original DBSCAN and Mean Shift algorithms perform poorly when clustering the two-dimensional data in the paper, with specific results shown in Figure 8, 9 and 10, and the data can be found in the appendix. The clustering effect of RWR K-Means++ on the dataset in our work is shown in Figure 11.

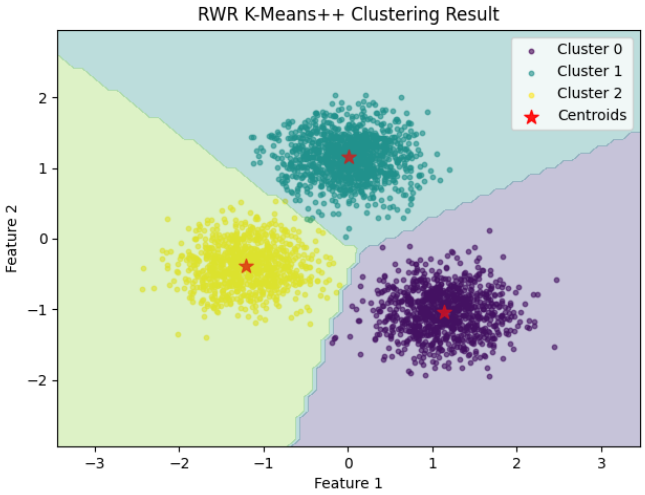

Fig. 11: RWR K-Means++ clustering results

Further analysis indicates that the DBSCAN algorithm is sensitive to the density of data distribution [43] [44] [45]. If the data is uniformly distributed in two-dimensional space and there are clear gaps between clusters, DBSCAN may consider them as noise. Additionally, DBSCAN is particularly picky about the choice of initial points, and local optima can lead to unstable clustering results, especially in the presence of noise or non-spherical clusters. Mean Shift, as a density-based clustering algorithm, is capable of capturing the peaks of high-density regions, but in two-dimensional space, it is prone to being influenced by local dense points, leading to local optima. In contrast, K-Means is relatively efficient in two-dimensional environments due to its preset clustering centers and simple assignment process, and it is suitable for spherical data. However, for problems with varying shapes and an uncertain number of clusters, Mean Shift may have more advantages.

E. K-Means++ algorithm for roulette reconstruction

Research indicates that k-means++ still has room for improvement. The more recent method is the k-means++ algorithm based on roulette reconstruction, which aims to improve the overall structure of the roulette and increase the probability of sequences further from the center being selected. Specifically, if initial centroid have been selected when selecting the next center point, the probability of its selection will be determined according to a more refined calculation, as shown in Eq. 11:

$$g_x = \frac{\min\limits_{x}(x - c_j)^2}{\sum\limits_{x}\min\limits_{j\in[1,n]}(x - c_j)^2}, \quad j \in [1, n] \tag{11}$$

To avoid selecting sequences that are too close to the already chosen centers, we set a random probability lower bound

based on the probability distribution of the squared distance to the roulette-selected center, preventing unnecessary random exploration. Specifically, the random seeds are set according to the number of clusters according Eq. 12., which is finely adjusted according to the number of clusters, which further optimizes the rationality of sequence selection.

$$s = \left[\frac{1}{k}, 1\right) \tag{12}$$

The improved roulette method ensures a certain degree of exploration while avoiding the issue of sequences that are too close being selected as weaker class centers. It also increases the probability of selecting sequences that are further away, alleviating the problem of excessively random behavior when dealing with high-dimensional data. The algorithm flowchart is shown in Figure 12.

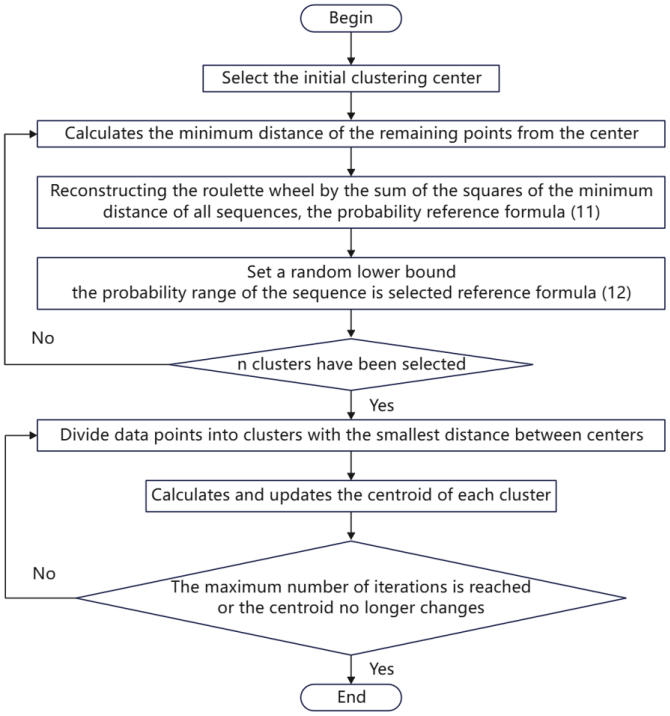

Fig. 12: RWR K-Means++ algorithm flowchart

Regarding time complexity and time consumption, the running time of the RWR K-Means++ algorithm is less than that of K-Means++. Through analysis, it is clear that the RWR K-Means++ method only reconstructs the probability roulette of the initial values, and therefore its time complexity is consistent with K-Means++. However, the K-Means++ method changes the way of selecting initial points based on K-Means during the iteration process, which is why the time complexity of K-Means++ is higher than that of the K-Means method.

V. Conclusion

The core principle of cluster analysis is similarity and distance [46] [47]. During the clustering process, the similarity between data points is usually measured by distance or similarity metrics, and clusters are formed by data points that are closer in distance or have higher similarity.

Since its proposal, the K-means algorithm has been widely applied in various fields due to its simple and understandable concept, fast clustering speed, and good clustering results. This paper primarily discusses the basic principles of selecting initial clustering centers, the value of K, and the identification of outliers. It also describes related work such as cluster analysis, including k-means++, k-modes, k-medoids, etc. After a thorough analysis of the dataset, the elbow method is used to determine the parameter K for the K-means algorithm. Initial centroid selection is based on optimized k-means like k-means++, and other clustering algorithms are simultaneously verified, including hierarchical clustering, DBSCAN, Mean Shift clustering, Gaussian Mixture Model Clustering, Spectral Clustering, Fuzzy C-Means clustering, and Self-Organizing Map Clustering. Three major internal indicators are used to validate the clustering performance of different algorithms.

This paper systematically compares and studies the clustering algorithms currently widely used in various fields, providing a good reference for improving the performance of clustering analysis in subsequent research. In future work, research will be conducted on high-dimensional hybrid feature data to enhance the adaptability and learning ability of clustering algorithms, with a focus on improving the efficiency of clustering algorithms, reducing time overhead, and further deepening the extensive application of clustering analysis in the field of big data analysis.

VI. Discussion

By improving the selection of initial clustering centers, K-means++ significantly improves the performance of the traditional K-means algorithm, reduces the risk of falling into the local optimal solution, and accelerates the convergence speed. The optimized initialization strategy makes the clustering results more accurate and effective. In addition, the algorithm is sensitive to outliers, and is more suitable for identifying spherical or convex clusters, and has limited processing ability for complex clusters. Although the calculation cost is increased in the initialization phase, the number of iterations is reduced in general, which may reduce the total calculation time. In the future, we believe that the improved k-means can further improve the accuracy and efficiency of data clustering, and the k-means algorithm is expected to show better robustness when dealing with data sets containing noise and outliers. In addition, with the development of big data technology, future research will focus on optimizing algorithm performance, reducing computational complexity, and exploring its practical application potential in different fields.

Acknowledgment

During this semester, I had the privilege of taking a course on Big Data Analysis, where I systematically learned and mastered methods for analyzing and processing information in data mining. In this experiment, I chose to conduct K-means and other clustering on planar sample points as my research topic and performed a comparative analysis on multiple dimensions based on this. Ultimately, I sincerely thank Professor Zou Zhiqiang for his diligent teaching this semester, which has laid a solid foundation for my future research career.

REFERENCES

1 Xurring, H. and Wei, Y., "Research on international experience and frontier issues of big data statistical applications," *Statistical Research*, vol. 2024, no. 41(09), pp. 3–12, 2024.

2 Ren, Y., Pu, J., Yang, Z., Xu, J., Li, G., Pu, X., Yu, P. S., and He, L., "Deep clustering: A comprehensive survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 4, pp. 5858–5878, 2025.

3 Wayment-Steele, H. K., Ojoawo, A., Otten, R., Apitz, J. M., Pitsawong, and et al., "Predicting multiple conformations via sequence clustering and alphafold2," *Nature*, vol. 625, no. 7996, pp. 832–839, 2024.

4 Barrio-Hernandez, I., Yeo, J., and et al., "Clustering predicted structures at the scale of the known protein universe," *Nature*, vol. 622, no. 7983, pp. 637–645, 2023.

5 Fang, U., Li, M., Li, J., Gao, L., Jia, T., and Zhang, Y., "A comprehensive survey on multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12 350–12 368, 2023.

6 SK, S., PB, N., NC, J., and et al., "K-means and fcm clustering of biological oxygen demand of water," in *Proceedings of the 2nd International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*. Trichy: IEEE, 2023, pp. 1743–1748.

7 Tang, C., Li, Z., Wang, J., Liu, X., Zhang, W., and Zhu, E., "Unified one-step multi-view spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 6449–6460, 2022.

8 Li, Y., Hu, P., Liu, Z., Peng, D., Zhou, J. T., and Peng, X., "Contrastive clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 10, 2021, pp. 8547–8555.

9 Wang, H., Song, Y., Chen, W., Luo, Z., Li, C., and Li, T., "A survey of co-clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 9, pp. 1–28, 2024.

10 Congying, D., Bo, Y., Jiangou, M., and et al., "Fuzzy evaluation of dynamic characteristics of cnc machine tool pose variation based on k-means clustering and probabilistic neural network," *Journal of Instrumentation*, vol. 2020, no. 41(12), pp. 227–235, 2020.

11 Yang, M., Li, Y., Hu, P., Bai, J., Lv, J., and Peng, X., "Robust multi-view clustering with incomplete information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1055–1069, 2022.

12 Ping, Y., "Improved grasshopper optimization algorithm integrating modified dbscan clustering and multiple evolutionary strategies," *Instrument Technique and Sensor*, vol. 2024, no. 05, pp. 98–105+12, 2024.

13 Shutaywi, M. and Kachouie, N. N., "Silhouette analysis for performance evaluation in machine learning with applications to clustering," *Entropy*, vol. 23, no. 6, p. 759, 2021.

14 Liang, J., Cui, Y., Wang, Q., Geng, T., Wang, W., and Liu, D., "Clusterfomer: clustering as a universal visual learner," *Advances in neural information processing systems*, vol. 36, pp. 64 029–64 042, 2023.

15

16 Liang, J., Cui, Y., Wang, Q., Geng, T., Wang, W., and Liu, D., "Clusterfomer: clustering as a universal visual learner," *Advances in neural information processing systems*, vol. 36, pp. 64 029–64 042, 2023.

17 Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., and Heming, J., "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, vol. 622, pp. 178–210, 2023.

18 Miao, Y., Li, S., Wang, L., Li, H., Qiu, R., and Zhang, M., "A single plant segmentation method of maize point cloud based on euclidean clustering and k-means clustering," *Computers and Electronics in Agriculture*, vol. 210, p. 107951, 2023.

19 Xu, G., Wen, J., Liu, C., Hu, B., Liu, Y., Fei, L., and Wang, W., "Deep variational incomplete multi-view clustering: Exploring shared clustering structures," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 14, 2024, pp. 16 147–16 155.

20 Hong, L., Jie, D., Feian, C., and et al., "Research on regional classification of ders based on comprehensive weighting and improved k-means++," *J Smart Power*, vol. 2020, no. 48(12), pp. 51–57+78, 2020.

21 Yang, X., Jiaqi, J., Wang, S., Liang, K., Liu, Y., Wen, Y., Liu, S., Zhou, S., Liu, X., and Zhu, E., "Dealmvc: Dual contrastive calibration for multi-view clustering," in *Proceedings of the 31st ACM international conference on multimedia*, 2023, pp. 337–346.

22 Zang, L., Xue, Q., Wang, X., Tian, B., and Zhu, S., "An optimized product model based on the k-means++ algorithm," in *Proceedings of the 2024 4th International Conference on Artificial Intelligence, Big Data and Algorithms*, 2024, pp. 677–684.

23 Jikong, Z., Yan, L., and et al., "Research and application of clustering algorithms in data mining," *Network Security Technology and Application*, vol. 2023, no. 12, pp. 39–41, 2023.

24 Schreiber, U., "Pulse-amplitude-modulation (pam) fluorometry and saturation pulse method: an overview," *Chlorophyll a fluorescence: a signature of photosynthesis*, pp. 279–319, 2004.

25 Jiajia, C., Wang, Z., Donghai, L., and et al., "A hybrid data k-prototypes algorithm incorporating metric," *Statistics and Decision Making*, vol. 2023, no. 39(10), pp. 16–22, 2023.

26 Zujian, M. and Jiangrong, M., "A k-means algorithm with optional initial clustering centers," *Statistics and Decision*, vol. 2014, no. 12, pp. 12–14, 2014.

27 Quxian, D. and Zanshen, Z., "Anevkmeans algorithm for selecting initial clustering center," *Statistics and Decision*, vol. 2020, no. 36(16), pp. 32–35, 2020.

28 Yinxu, X., Hongwen, X., and Ling, L., "Global optimized k-means clustering algorithm based on sample density," *Computer Engineering and Applications*, vol. 2018, no. 54(14), pp. 143–147, 2018.

29 Yuhao, C., Yongquan, L., Jiancong, F., and et al., "Optimizing initial cluster centroids by weighted local variance in k-means algorithm," *Journal of Frontiers of Computer Science and Technology*, vol. 2016, no. 10(5), pp. 732–741, 2016.

30 Qingjie, L., "Research on the improvement and application of heuristic k-means clustering algorithm," Master's thesis, Dalian Jiaotong University, 2023.

31 Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., and Heming, J., "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, vol. 622, pp. 178–210, 2023.

32 Adnan, R. M., Khosravinia, P., Karimi, B., and Kisi, O., "Prediction of hydraulics performance in drain envelopes using kmeans based multivariate adaptive regression spline," *Applied Soft Computing*, vol. 100, p. 107008, 2021.

33 Yu, H., Qiang, W., and Lina, L., "Optimization study of k-means clustering algorithm," *Software*, vol. 2023, no. 44(10), pp. 58–61, 2023.

34 Fenggu, S., Weifei, M., Xuannrang, Z., and et al., "An improved method for partitioning clustering k-value and center initialization," *Computer Engineering*, vol. 2023, no. 49(11), pp. 85–93, 2023.

35 Shipeng, D. and Shaobo, W., "Indoor positioning algorithm based on fuzzy clustering and dynamic k value," *Internet of Things Technology*, vol. 2023, no. 13(01), pp. 38–41, 2023.

36 Sinan, S., "Research and application of outlier detection method based on nearest neighbor," Master's thesis, Taiyuan University of Science and Technology, 2022.

37 Hong, Y., Danning, L., and Yaie, W., "K-means algorithm based on lof," *Communications Technology*, vol. 2019, no. 52(8), pp. 1848–1848, 2022.

38 Schubert, E., Sander, and et al., "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

39 Heyovyan, M., Yukun, H., Shuang, F., and et al., "Research on bearing fault diagnosis algorithm based on improved k-means clustering and deep neural network," *Journal of Engineering of Heilongjiang University (Chinese English Russian)*, vol. 2023, no. 14(04), pp. 55–63, 2013.

40 Lixiang, X., Lijie, G., and Zhanbo, L., "Analysis of multi-dimensional data de-duplication clustering algorithm based on sparse autoencoder," *Computer Simulation*, vol. 2024, no. 41(03), pp. 542–547, 2021.

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

42 Yu, H., Jiang, L., Fan, J., Xie, S., and Lan, R., "A feature-weighted suppressed possibilistic fuzzy c-means clustering algorithm and its application on color image segmentation," *Expert Systems with Applications*, vol. 241, p. 122270, 2024.

43 Fenggu, S., Weifei, M., Xuannrang, Z., and et al., "An improved method for partitioning clustering k-value and center initialization," *Computer Engineering*, vol. 2023, no. 49(11), pp. 85–93, 2023.

44 Golalipour, K., Akbari, E., Hamidi, S. S., Lee, M., and Enayatifar, R., "From clustering to clustering ensemble selection: A review," *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104388, 2021.

45 Shi, W. B. W. S. e. a., "A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm." *J Wireless Com Network*, vol. 31, 2021.

46 Li, Z. D. W. Z. D. L. H. W. S., "Forecasting crude oil prices based on variational mode decomposition and random sparse bayesian learning." *Appl. Soft Comput.*, vol. 113, 2021.

47 Shi, W. T. J. D. M. L. H. L. B., "Detecting spatiotemporal extents of traffic congestion: A density-based moving object clustering approach." *Int. J. Geogr. Inf. Sci.*, vol. 35, 2021.

APPENDIX

A. Experimental Environment Configuration

The hardware environment required for the experiment is a laptop with a configuration of an Intel® Core i7-8550U CPU @1.80GHz4.00GHz quad-core processor, a 256 GB solid-state drive, Intel® UHD Graphics 620, and NVIDIA GeForce MX150 dual graphics cards.The software environment for this experiment is Windows 10 Home Chinese Edition 64-bit operating system, Python 3.8 interpreter, PyCharm, and third-party Python libraries.

B. Original dataset

For clustering analysis, we import the dataset xclara.csv into Python and convert it into a two-dimensional array.

TABLE I: Original Dataset

Feature1	Feature2
1.083576	7.319176
11.12067	14.40678
23.71155	2.557729
24.16993	32.02478
21.66578	4.892855
4.693684	12.34217
19.21191	-1.121366
4.230391	-4.441536
7.314846	3.309312
11.94122	8.122487

Visual processing can treat 3000 pairs of data as points on the plane, as is shown in Table I, in which the first column of features is used as the X coordinate and the second column of features is used as the Y coordinate, which is convenient for subsequent clustering analysis.

C. DBSCAN Parameter Optimization

Through comparative experiments, this paper finds that the DBCSCAN algorithm performs relatively poorly in clustering the provided dataset, so researchers have adjusted and optimized the parameters of DBSCAN. The main work is to add the find-optimal-parameters function, which iterates over given eps and min-samples values, using the silhouette coefficient as a performance metric to find the optimal parameters. The experimental results indicate that for the paper dataset, the best results are achieved with eps set to 0.2 and min-samples set to 9, as shown in Figure 13.

D. Cluster performance

As previously mentioned, this paper conducts a comparative analysis of the performance of the k-means algorithm versus hierarchical clustering, DBSCAN, mean shift clustering, Gaussian mixture models, spectral clustering, fuzzy C-means clustering, and self-organizing maps. Refer to Table II for specific experimental results. We have comprehensively considered the three clustering analysis indicators to comprehensively and reliably evaluate the clustering results.

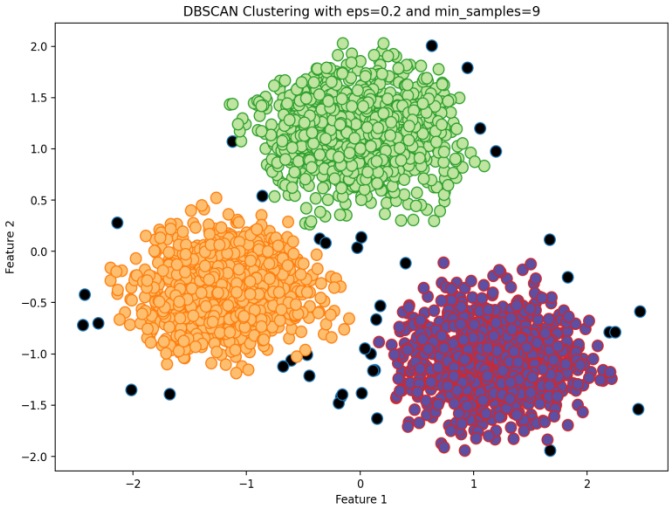

Fig. 13: Optimization of DBSCAN parameters

During the experiment, it was found that the initial performance of the DBSCAN algorithm was low, with an initial choice of eps set to 0.2 and min-samples set to 25. After improvements, the algorithm showed significant enhancement in terms of silhouette coefficient and Calinski-Harabasz index, but there was no significant improvement in the Davies-Bouldin index. Through research, it was discovered that this is due to DBSCAN's high sensitivity to noise points. If there are a large number of noise points in the dataset, they may affect the clustering results, leading to a decrease in the separation between clusters, which in turn causes a higher Davies-Bouldin index.

TABLE II: Evaluation Indicators

Clustering algorithm	**SC**	**CH**	**DB**
K-Means (k=3)	0.69110	10247.433	0.42364
Hierarchical	0.68973	10177.274	0.42741
DBSCAN	0.63214	5298.768	1.38595
DBSCAN(Optimized)	0.65478	6334.681	1.40413
Mean Shift	0.20875	2507.820	0.87074
Gaussian Mixture	0.69098	10240.788	0.42425
Spectral	0.69105	10241.753	0.42366
Fuzzy C-Means	0.69104	10246.602	0.42423
SOM	0.69092	10236.780	0.42333

Overall, algorithms such as k-means and its optimized variants like k-means++, as well as other types of clustering methods, have mostly achieved a silhouette coefficient above 0.65, indicating relatively good overall performance. This research process has opened up a broader perspective for future research, and subsequent work will focus on improving algorithm efficiency.