Cluster Analysis

ChenChen CAO
1024040803

***Abstract*—In the era of big data, the landscape of data analysis has undergone a profound transformation. The exponential growth of data volume, coupled with the increasing complexity of data types, has presented a multitude of challenges. Traditional analytical methods often fall short in effectively processing and extracting valuable insights from this vast and intricate data. Against this backdrop, cluster analysis emerges as a crucial technique. It plays a vital role in handling complex data by uncovering latent patterns and structures. These insights enable more informed decision-making across various industries, from business analytics to scientific research.This study delves deep into partitioning-based, hierarchical-based, and density-based cluster analysis methods, along with their classic algorithms: K-Means, Hierarchical Clustering, and DBSCAN. By leveraging a dataset sourced from GitHub, we calculate the distances between data points to generate a matrix for fine-tuning the DBSCAN algorithm's parameters, Employing the Silhouette Coefficient, Calinski-Harabasz Index, and Davies-Bouldin Index, we evaluate these algorithms. Our findings reveal that K-Means and Hierarchical Clustering exhibit comparable performance on this dataset, while DBSCAN fares worse. The dataset's spherical distribution likely favors the first two algorithms. This study offers valuable guidance for algorithm selection.**

***Index Terms*—Clustering, Similarity measures, Data mining.**

I. Introduction

Data mining, as a crucial data analysis technique, plays a pivotal role in the digital age today. Its core objective is to precisely extract information from very large databases (VLDB) that is not only of high value but also aligns with users' interests. Among the numerous technical means in data mining, clustering is undoubtedly one of the most critical techniques[1].

It achieves its functionality mainly by meticulously constructing specific mathematical models. These models are designed based on in-depth understanding and analysis of the data, enabling them to accurately capture the internal characteristics and structures of the data. In the actual operation process, clustering meticulously divides the massive data in the database into different clusters according to strict data similarity measurement criteria.

During this division process, clustering technology always endeavors to achieve an important goal: ensuring that the data within the same cluster possess a high degree of similarity as much as possible. This means that the data within the same cluster exhibit a high level of consistency in various dimensions, whether they are numerical features, text features, or other types of features, and are as close as possible. For example, in a customer information database, if clustering is performed according to consumption behavior, customers within the same cluster will have similar patterns in terms of consumption frequency, consumption amount, types of purchased goods, and so on. At the same time, clustering also focuses on making the data between different clusters show significant differences. That is to say, there are obvious distinctions in data characteristics among various clusters. This difference can be manifested in various ways. For instance, the data of different clusters may have significant differences in aspects such as mean value, variance, and distribution pattern. Take the field of image recognition as an example. Through clustering, different types of image data can be divided into different clusters, making the images in different clusters significantly different in terms of color, texture, shape, etc., thus facilitating subsequent classification and recognition tasks.

In this way, clustering technology can assist data miners in better understanding the distribution and structure of data, providing a strong foundation for further data analysis, knowledge discovery, and decision support.

II. Problem Statement

This experiment aims to conduct experiments on a public dataset using three classic algorithms: K-Means, Hierarchical Clustering, and DBSCAN. By modifying specific parameters and analyzing the principles of these three types of clustering algorithms, we will analyze their advantages and disadvantages on public dataset.

III. Solutions

A. Clustering Algorithm Introduction

- ***K-Means***[2]: The K-Means algorithm is the most widely used clustering algorithm. This algorithm represents each class by the weighted mean of the samples within the class (referred to as the centroid). It is only applicable to the clustering of numerical attribute data. The algorithm has a clear geometric and statistical significance but is relatively susceptible to interference.The algorithm steps are as follows:

  (1)Initialization: First, the number of clusters needs to be determined. Randomly select data points from the dataset as the initial centroids. The choice of these initial centroids can have a certain impact on the final clustering results, but usually, random selection can start the algorithm process.

  (2)Assign Samples: For each sample point in the dataset, calculate its Euclidean distance to the centroids. Then, assign this sample point to the cluster represented by the closest centroid. In this way, all sample points are divided into different clusters.

(3)Update Centroids: After all sample points have been assigned, recalculate the centroid of each cluster. Specifically, calculate the mean value of all sample points within the cluster in each dimension, and the obtained new mean is the new centroid of the cluster. This new centroid will serve as a reference point for the next iteration.

(4)Iteration: Repeat steps 2 and 3, continuously assigning sample points and updating centroids until a certain stopping condition is met. Common stopping conditions include that the centroid position no longer changes significantly (i.e., the moving distance of the centroid between two iterations is less than a certain threshold), or the preset maximum number of iterations is reached.

Typically, the sum of the Euclidean distances between each sample and its centroid is used as the objective function. The objective function can also be modified to the sum of the Euclidean distances between any two points within each class. This takes into account both the dispersion and the compactness of the classes. If the objective function is regarded as the logarithm of the likelihood ratio of a distribution-normalized mixture model, the K-Means algorithm can be considered as a generalization of the probabilistic model algorithm.

- ***DBSCAN***[3]: DBSCAN is a density-based spatial clustering algorithm widely used in data mining and machine learning. It aims to discover clusters of arbitrary shapes in a dataset while identifying noise points, which are data points that do not belong to any meaningful cluster. Its key concepts include density, the ε-neighborhood (for a data point p, it's the set of points within distance ε from p), core points (points with at least $MinPts$ points in their ε-neighborhood), border points (non-core points in a core point's ε-neighborhood), and noise points (neither core nor border points).

  The algorithm starts with initializing parameters ε and $MinPts$ and marking all data points as unvisited. Then it randomly selects an unvisited point , marks it as visited and determines its type. If p is a core point, a new cluster C is created and is added to it, with unvisited points in its ε-neighborhood added to a queue . If p has less than $MinPts$ points in its ε-neighborhood and is unassigned, it's marked as a noise point. While Q is not empty, points are removed from Q, marked as visited, and processed to expand the cluster. This process repeats until all data points are visited, identifying all clusters and noise points.

  DBSCAN has several advantages. It doesn't require prespecifying the number of clusters, can detect arbitrary-shaped clusters, and is robust to noise. However, it also has disadvantages. Its performance is highly sensitive to the selection of parameters ε and $MinPts$, and different values can lead to very different results, often requiring domain knowledge or much experimentation to find the optimal ones. Additionally, it has a high computational complexity with a time complexity of $O(n^2)$(where is the number of data points in the dataset).
- ***Hierarchical Clustering***[4]: Hierarchical Clustering algorithm is a widely used clustering algorithm in the fields of data analysis and machine learning. It achieves clustering by constructing hierarchical relationships among data points and can be divided into two types: Agglomerative and Divisive. For Agglomerative Hierarchical Clustering, initially, each data point is regarded as a separate class, and at this time, the number of classes is equal to the number of data points. Then, calculate the distance between every two classes (distance measurement methods such as single linkage, complete linkage, and average linkage can be used). Identify the two closest classes and merge them into a new class. Continuously repeat this process until all data points are merged into one class or a preset stopping condition is met.

  The specific steps of Divisive Hierarchical Clustering are as follows:

  (1) Initialization: Consider all data points as a single whole class, which is the starting state of the entire splitting process.

  (2) Select the Class to Split: Calculate the differences or distances among the data points within the class. Based on a specific criterion (such as the maximum distance, maximum variance, etc.), select the most suitable class to split from all the current classes. The choice of this criterion will affect the clustering results and quality.

  (3) Determine the Splitting Method: According to the selected distance measurement method (such as single linkage, complete linkage, average linkage, etc.) and the preset splitting strategy, determine the specific way to split the selected class into two subclasses. For example, if a distance-based splitting strategy is adopted, two subsets with the farthest distance may be divided into different subclasses.

  (4) Execute the Splitting: Split the selected class into two subclasses formally according to the determined splitting method.

  (5) Iteration: Repeat steps 2 to 4. Each split will increase the number of classes. Continue this process until each data point becomes a separate class or a preset stopping condition (such as the number of classes reaching a specified value) is met.

  The advantages of this algorithm are as follows. It doesn't require the pre-specification of the number of clusters. Instead, it automatically generates a hierarchical clustering structure, allowing users to select results according to their needs. It is highly flexible and can employ different distance metrics to adapt to different data and scenarios. Moreover, it has a good visualization effect. The hierarchical relationships and the clustering process can be intuitively displayed through a dendrogram.

B. Evaluation parameters

Evaluation metrics in clustering algorithms play a vital role. They quantify the quality of clustering results, enabling us to directly assess how well an algorithm performs. For example, the Silhouette Coefficient measures the cohesion within clusters and separation between clusters, with values closer to 1

indicating better results.These metrics also provide a common ground for comparing different algorithms. Whether it's K-Means, Hierarchical Clustering or DBSCAN, we can use the same metrics like the Davies-Bouldin Index to objectively evaluate their performance on a dataset.

Due to the fact that the dataset used in this article does not have real values for comparison to obtain external indicators, the following three internal indicators are selected for performance evaluation:

- ***Silhouette Coefficient***: The Silhouette Coefficient is a measure used to evaluate the quality of a clustering result. It provides a way to assess how well each data point fits within its assigned cluster and how distinct different clusters are from one another.

  Mathematically, for a data point i, the silhouette coefficient $s(i)$ is calculated as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

  where $a(i)$ is the average distance between data point i and all other points within the same cluster, and $b(i)$ is the minimum average distance between data point i and all points in the neighboring clusters.

  The value of the silhouette coefficient ranges from -1 to 1. A value close to 1 indicates that the data point is well-clustered within its own cluster and is far from other clusters. A value close to 0 means that the data point is close to the boundary of its cluster. A negative value implies that the data point might be misclassified and would be better placed in a different cluster.

  The overall Silhouette Coefficient for a clustering is the average of the silhouette coefficients of all data points in the dataset. This coefficient is useful for comparing different clustering algorithms or for determining the optimal number of clusters in a dataset.
- ***Calinski-Harabasz Index***: The Calinski-Harabasz Index, also known as the Variance Ratio Criterion, is another metric for evaluating the quality of a clustering. It measures the ratio of the between-cluster variance to the within-cluster variance.

  Let k be the number of clusters, n be the total number of data points, and n_j be the number of data points in the j-th cluster. Let μ be the centroid of the entire dataset, and μ_j be the centroid of the j-th cluster.The between-cluster sum of squares (SSB) is given by:

$$SSB = \sum_{j=1}^{k} n_j \|\mu_j - \mu\|^2$$

  The within-cluster sum of squares (SSW) is given by:

$$SSW = \sum_{j=1}^{k} \sum_{i \in C_j} \|\mathbf{x}_i - \mu_j\|^2$$

  where C_j is the set of data points in the j-th cluster.The Calinski-Harabasz Index (CHI) is then calculated as:

$$CHI = \frac{SSB/(k-1)}{SSW/(n-k)}$$

  A higher Calinski-Harabasz Index value indicates better-defined clusters. As the number of clusters increases, the within-cluster variance (SSW) should decrease, and the between-cluster variance (SSB) should increase. The optimal number of clusters is often considered to be the one that maximizes the Calinski-Harabasz Index. Davies-Bouldin Index
- ***Davies-Bouldin Index***: The Davies-Bouldin Index is a measure that quantifies the similarity between clusters. It is based on the ratio of the sum of within-cluster scatter and the distance between cluster centroids.

  For two clusters C_i and C_j with centroids μ_i and μ_j, let s_i and s_j be the average distances of points within clusters C_i and C_j from their respective centroids.

  The similarity between clusters C_i and C_j is defined as:

$$R_{ij} = \frac{s_i + s_j}{\|\mu_i - \mu_j\|}$$

  The Davies-Bouldin Index (DBI) for a clustering with k clusters is calculated as:

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} R_{ij}$$

  The Davies-Bouldin Index ranges from 0 to ∞. A lower value of the Davies-Bouldin Index indicates better clustering, as it implies that the clusters are well-separated and have low internal scatter. The optimal number of clusters is often the one that minimizes the Davies-Bouldin Index.

IV. Evaluation

A. Data Characteristics

Data sets for clustering can be downloaded from GitHub, address to https://github.com/mubaris/friendly-fortnight/blob/master/xclara.csv. The data scale is 3000*2, and the distribution of data points is shown in Figure 1.

B. Experimental results

As can be observed from the data distribution in Figure 1, all the data are approximately grouped into three clusters. Consequently, it is essential to specify the cluster parameters of both the K-means and hierarchical clustering algorithms as 3. The result screenshots of the K - means algorithm and hierarchical clustering are shown in Figures 2 and 3.

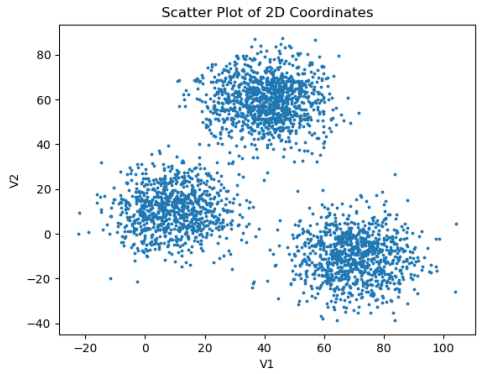

Fig. 1. Scatter Plot of 2D Coordinates

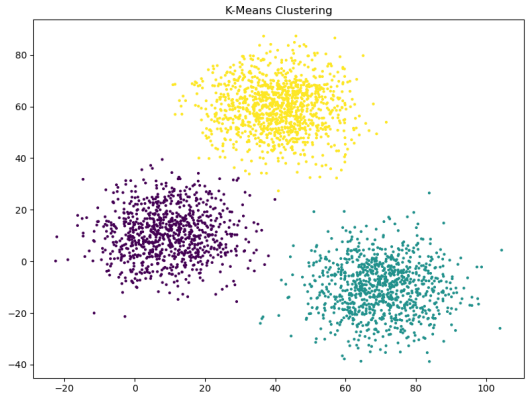

Fig. 2. K-means result

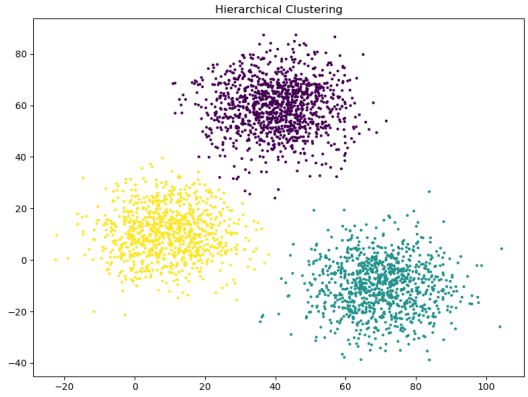

Fig. 3. Hierarchical clustering result

Unlike the methods mentioned above for directly observing the number of clustering clusters, in the parameter setting of the DBSCAN algorithm, visualization methods serve as effective auxiliary tools, primarily encompassing two approaches: distance matrix analysis and K-distance graph.

For distance matrix analysis, the distances between data points are first calculated to generate a matrix, which is then sorted row by row. Subsequently, the probability density distribution curves of the distance values are plotted column by column. By observing the position where the density significantly drops in the curve, the corresponding distance can be used to determine the value of eps.

Regarding the K-distance graph, for each data point, the distance to its k-th (usually between 5 and 10) nearest neighbor is calculated, and a graph is plotted. The distance corresponding to the "elbow" point, where the slope of the curve changes abruptly, is selected as a reference value for eps. Moreover, the value of k can be approximately used as a reference for $minPts$.

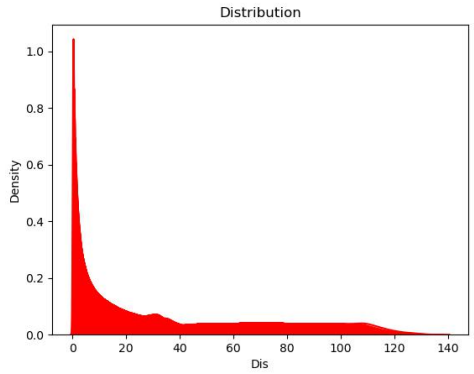

Fig. 4. Density distribution map

TABLE I
PARAMETER $minPts$ EVALUATION METRICS

	SC	CHI	DBI
$minPts$=25	0.663	6923	1.559
$minPts$=27	0.665	6870	1.582
$minPts$=30	0.661	6850	1.589
$minPts$=35	0.663	6710	1.528

The probability distribution diagram is shown in Figure 4. Analysis shows that when the distance value is about 8, the density begins to decrease significantly. Therefore, the initial selection of esp is 8.

Considering that there are 3000 data points in the dataset, $minPts$ can be set to about 1% of the total number of data points, which is 30, as the initial trial value. The three evaluation indicators introduced in the previous text are selected as the evaluation index, and the results are shown in Table I. Finally, esp of 8 and $minPts$ of 27 are selected as parameters for comparison, the corresponding clustering results are shown in Figure 5.

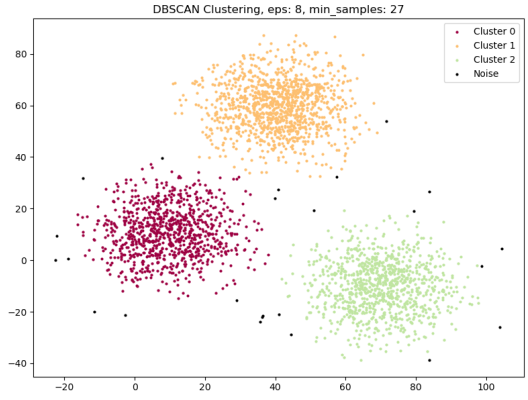

Fig. 5. DBSCAN result

TABLE II
CLUSTER RESULT EVALUATION

	SC	CHI	DBI
K-means	0.695	10826	0.421
Hierarchical clustering	0.694	10793	0.420
DBSCAN	0.665	6870	1.582

Analyze the three evaluation indicators in Table II. Under this dataset, the clustering effects of the K-means algorithm and the Hierarchical clustering algorithm are similar, and both are superior to the effect of the DBSCAN algorithm. Analyzing the dataset and the algorithm principles, we know that due to the spherical distribution of this dataset, the K-means and hierarchical clustering algorithms have a natural advantage in handling data with such a distribution.

The K-means algorithm divides clusters based on the centroid. For data with a spherical distribution, the centroids of each cluster can be determined relatively easily. The hierarchical clustering algorithm constructs a tree structure by calculating the similarity or distance between data points. The distance relationships between data points in a spherical distribution are relatively regular, which is conducive for the hierarchical clustering algorithm to find appropriate clustering levels and partitions.

However, the DBSCAN algorithm is more suitable for handling datasets with complex shapes. For a dataset with a spherical distribution, the density change is relatively gentle, and it is difficult to accurately divide different clusters through density differences. Therefore, it cannot give full play to its advantages on such a dataset.

Meanwhile, the parameters of the K-means and hierarchical clustering algorithms are relatively simple. In contrast, the settings of the two key parameters, eps and $minPts$, of the DBSCAN algorithm have a great impact on the clustering results[5].

V. CONCLUSION

The research delved deep into the realm of cluster analysis methods, exploring the fundamental principles of partitioning-based, hierarchical-based, and density-based clustering algorithms. The significance of clustering in big data analysis, as a crucial data preprocessing step for uncovering latent relationships within data, was clearly demonstrated.

Through experimental analysis on a specific dataset, we systematically preprocessed the data, calculated the distance matrix to fine - tune the DBSCAN algorithm's parameters, and comprehensively evaluated the performance of various algorithms using multiple internal metrics. The experimental results presented a clear picture: on the given dataset, the K - Means and Hierarchical Clustering algorithms exhibited similar performance levels, while the DBSCAN algorithm underperformed. This disparity was attributed to the spherical distribution of the dataset, highlighting the fact that even though the DBSCAN algorithm is designed for datasets with complex shapes, its effectiveness is contingent on the dataset's characteristics.

The findings of this research hold practical value. They offer researchers and practitioners a valuable reference for understanding different clustering algorithms, enabling them to make informed decisions when selecting and applying these algorithms in big data processing. By considering the unique characteristics of the data at hand, more appropriate clustering methods can be chosen, thereby enhancing the efficiency and accuracy of data analysis.

Looking ahead, future research should focus on further exploring the intricate relationship between various types of datasets and algorithm performance. This could involve investigating how different distributions, sizes, and dimensionalities of datasets impact the effectiveness of clustering algorithms. Additionally, optimizing algorithm parameters to adapt to diverse datasets will be a key area of study. By doing so, we can not only improve the clustering effect but also expand the application scope of these algorithms in the ever - evolving field of big data, ultimately facilitating more in - depth data exploration and knowledge discovery.

REFERENCES

[1] What Is Data Mining. *Introduction to data mining*. Springer, 2006.

[2] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. "The global k-means clustering algorithm". In: *Pattern recognition* 36.2 (2003), pp. 451–461.

[3] Erich Schubert et al. "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN". In: *ACM Transactions on Database Systems (TODS)* 42.3 (2017), pp. 1–21.

[4] Fionn Murtagh and Pedro Contreras. "Algorithms for hierarchical clustering: an overview". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2.1 (2012), pp. 86–97.

[5] Rui Xu and Donald Wunsch. "Survey of clustering algorithms". In: *IEEE Transactions on neural networks* 16.3 (2005), pp. 645–678.