Learning Cluster Analysis Based on K-means Algorithm

Ning Shuaiyu
1024041123
Nanjing University of Posts and Telecommunications
School of Computer Science
Nanjing, China

***Abstract*—Cluster analysis is an important unsupervised learning method in machine learning, which is widely used in data mining, pattern recognition, and business intelligence. This paper takes the classical K-means algorithm as the core, analyses the related principles of clustering algorithm and K-means algorithm, defects, and briefly introduces the improved algorithm of K-means, K-means++, as well as some other excellent algorithms in the field of cluster analysis, and demonstrates the clustering effect of various cluster analysis algorithms through experiments on real data sets and briefly evaluates and analyses the algorithms through common evaluation metrics to briefly evaluate and analyse the algorithms.**

***Index Terms*—Cluster analysis, Unsupervised study, K-means, evaluation metrics**

I. Introduction

With the arrival of the big data era, how to discover hidden structures and patterns from massive data has become a core task. Cluster analysis, as a data mining technique, can group data based on similarity without supervised information, and is widely used in business, medicine, finance, image processing and other fields. Clustering belongs to unsupervised learning in machine learning classification. It is convenient to group data in the case of unlabelled datasets.

K-means clustering algorithm was proposed independently by Steinhaus 1955, Lloyd 1957, Ball & Hall 1965, McQueen 1967 in their different fields of scientific research.After K-means clustering algorithm was proposed, it has been widely researched and applied in different disciplines and a large number of different improved algorithms. Although the K-means clustering algorithm has been proposed for more than 50 years, it is still one of the most widely used algorithms for dividing clusters [1].

II. Clustering Algorithms

A. Definition of Clustering

Clustering is a process of dividing a data set into subsets and making data objects within the same set have a high degree of similarity, while data objects in different sets are dissimilar, and the measure of similarity or dissimilarity is determined based on the values of the descriptive attributes of the data objects, which is usually the use of the distances between the individual clusters to be described. The basic guiding principle of cluster analysis is to maximise the similarity of objects in a class and minimise the similarity of objects between classes.

Clustering is different from classification in that in a classification model, there exists sample data for which the class labels are known, and the purpose of classification is to extract the rules of classification from the training sample set for class identification of other objects whose class labels are unknown. In clustering, the information about the class of the target data is not known in advance, and it is necessary to classify all the data objects into clusters using some metric as a criterion. Hence, cluster analysis is also known as unsupervised learning.

B. Requirements for Clustering Algorithms

Cluster analysis is a challenging area of research, and each application imposes its own unique requirements. The following are some of the typical requirements for cluster analysis in data mining.

- Scalability. Many clustering algorithms work well with small datasets (less than 200 data objects): but a large database may contain millions of objects. Cluster analysis using sampling methods may give a biased result, and this is where scalable cluster analysis algorithms are needed.
- Ability to handle different types of attributes. Many algorithms are designed for interval-based numerical attributes. But some applications require the ability to work with other types of data, e.g., binary types, symbolic types, sequential types, or combinations of these data types.
- Discover clustering of arbitrary shapes. Many clustering algorithms perform clustering based on Euclidean distance and Manhattan distance. Clustering methods based on such distances can generally only discover circular or spherical clusters with similar size and density. In fact, a cluster can have an arbitrary shape, so it is very important to design clustering algorithms that can discover arbitrarily shaped class sets.
- Ability to handle noisy data. Most real-world databases contain anomalous data, unknown data, missing data, and noisy data, to which some clustering algorithms are very sensitive and lead to poor quality data.

C. Classification of Clustering Algorithms

Clustering algorithms can be categorised into divisional, hierarchical, density-based, grid-based and model-based approaches.

a) Methods of division: Given n objects, a division method constructs k divisions of the objects, each representing a cluster and $k \leq n$. Given the number k of divisions to be constructed, the division method first creates an initial division. It then employs an iterative relocation technique that attempts to improve the division by moving objects between divisions. A general criterion for a good division is that there is high similarity between objects in the same class and low similarity between objects in different classes. Two popular heuristics for division are K-means algorithm, K-medoids algorithm.

b) Hierarchical approach: The hierarchical approach performs a hierarchical decomposition of a given set of data objects. Based on the hierarchical decomposition, hierarchical methods can be categorised as cohesive and split. Cohesive methods, also known as bottom-up methods, start with each object as a separate group and then successively merge similar objects or groups until all combinations merge into one, or a termination condition is reached. Split methods, also known as top-down methods, begin by placing all objects in a cluster, and at each step of the selection generation, a cluster splits into smaller clusters until eventually each object is in a separate cluster, or a termination condition is reached.Algorithms such as BIRCH, CURE, and CHAMELEON are typical of hierarchical clustering.

c) Density-based methods: The vast majority of segmentation methods are based on the distance between objects for clustering, such methods can only find spherical clusters and encounter difficulties in finding clusters of arbitrary shapes, and consequently, density-based methods are proposed. The main idea is: as long as the density (number of objects or data points) of the neighbouring regions exceeds a certain queue value, the clustering will continue.DBSCAN, OPTICS and CLIQUE are three representative methods.

d) Grid-based methods: Grid-based methods quantise the object space into a finite number of cells, forming a grid structure. All clustering operations are performed on this grid structure. The main advantage of this method is that it is fast and its processing time is independent of the number of data objects and is only related to the number of cells in each dimension of the quantised space.STING is a typical algorithm for grid-based methods.

e) Model-based approach: A model-based approach assumes a model for each cluster and searches for the best fit of the data to the given model. A model-based algorithm may locate clusters by constructing densities that reflect the spatial distribution of the data points, or it may automatically decide on the number of clusters based on standard statistics that take into account 'noisy' data or isolated points to produce a robust clustering method.

III. K-MEANS ALGORITHM

A. Introduction to Algorithm Principles

K-means algorithm is a hard clustering algorithm, which is a representative of typical prototype-based objective function clustering methods, which is some kind of distance sum of data points to the prototype (category centre) as an objective function for optimization, and the adjustment rules for iterative operations are obtained by using the function to find the extreme value.K-means algorithm uses the Euclidean distance as a similarity measure, which is to solve for the vector corresponding to a certain initial clustering centre vector V= $(v, v_n \cdots, v_k)^T$ optimal classification that minimises the value of the evaluation metric J_c. The algorithm often uses the sum-of-squares-of-errors criterion function as the clustering criterion function, which is defined as $J_c = \sum_{i=1}^{k} \sum_{p \in C_i} \|p - M_i\|^2$. Where M_i is the mean value of the data objects in class C_i and p is the spatial points in class C_i. The algorithm also uses the error-squared criterion function as the clustering criterion function.

Analysis of the error squared and criterion functions reveals that the K-means algorithm is an optimization solution problem where there are many local minima in the objective function and only one is a global minimum. The search direction of the objective function is always along the direction where the error squared and criterion function decreases. Different initial values cause the clustering centre vector V to decrease the objective function along different paths. As shown in Fig. 1, the objective function is gradually reduced along the paths of V_A, V_B, and V_C. three different initial value vectors, respectively, to find their corresponding minima. Among them, only the minimum corresponding to point B is the global minimum, while the minimum corresponding to points A and C are local minima.The K-means algorithm is a kind of Hillclimbing algorithm, and the algorithm tends to find the local minima when it terminates.

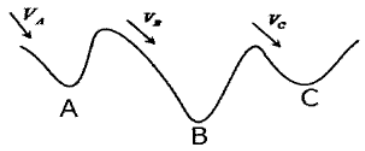


Fig. 1. Local minima and global optima.

The K-means algorithm uses the method of selective iterative updating: in each round of iteration, the surrounding points are formed into k clusters based on the k clustering centres, and the re-calculated centre of mass of each cluster will be used as the reference point for the next round of selective generation. Selection of generations makes the selected reference point closer and closer to the real cluster centre of mass, so the self-scaling function is getting smaller and smaller, and the clustering effect is getting better and better.

B. Introduction to Algorithm Principles

The core idea of K-means algorithm is to divide n data objects into k clusters so that the sum of squares of data points in each cluster to the centre of that cluster is minimized, algorithmic process:

- Input: number of clusters k, data set containing n data objects.

- Output: k clusters.
- Step 1: Arbitrarily select k objects from the n data objects as the initial cluster centres.
- Step 2: Calculate the distance from each object to each cluster centre separately and assign the object to the cluster with the closest distance.
- Step 3: After all objects are assigned, recalculate the centres of the k clusters.
- Step 4: Compare with the k clustering centres obtained from the previous calculation, if the clustering centre changes, go to Step 2, otherwise go to Step 5.
- Step 5: Output the clustering results.

C. K-means Algorithm Flaws

The disadvantages of K-means algorithm are given from the following [2]:

a) Initialisation of the number of clusters k in the K-means algorithm: The clustering results depend on the setting of the initial value, but the selection of the k-value often requires many experiments to find the optimal number of clusters. At present, the k value is determined mainly through the following methods:

- Select representative points empirically, according to the nature of the problem, data distribution, from the intuitively more reasonable representative points k.
- Randomly divide all the samples into k classes, calculate the centre of gravity of each class, and take these centres of gravity as the representative points of each class.
- Select representative points by density magnitude: Take each sample as the centre of the ball, and make a sphere with d as the radius; the number of samples falling inside the ball is called the density of the point, and it is sorted according to the density size. First select the largest density as the first representative point, i.e. the first clustering centre. Then consider the second largest density point, if the second largest density point from the first representative point of the distance is greater than d1 (artificially specified positive number) then the second largest density point as the second representative point, otherwise it can not be used as a representative point, so that according to the size of the density of the investigation goes on, the representative selected The distance between points is greater than d1. d1 is too small, too many representative points, d1 is too large, too small representative points, generally choose d1 = 2d. The density within the representative points is generally required to be greater than T. $\mathrm{T} > 0$ is a positive number specified.
- Use the first k sample points as representative points.

b) Selection of initial clustering centre: K-means algorithm uses random method to select the initial clustering centre, the clustering results may be different depending on the selected points, such dependence leads to the instability of the clustering results, and is easy to fall into the local optimal rather than globally optimal clustering results.Forgy proposed the method of randomly selecting points. Representative points can also be selected empirically as initial clustering centres.P.S.Bradley and Usama M.Fayyad proposed a sampling based approach to determine the initial clustering centres. Selection of initial points using density based clustering is also a method, but density based methods require the radius of the neighbourhood to be given, so density based selection is somewhat subjective and can have an effect on the clustering results.G.P.BabuandM.N.Murty proposed an algorithm based on Genetic Algorithms to find initial clustering centres.

c) Sensitive to noise points and isolated points: The K-medoid method considers not using the average value of the objects in the cluster as a reference point, and selects the most centrally located object in the cluster, i.e., the centroid. In this way the partitioning method is still performed based on the principle of minimising the sum of the dissimilarities between all the objects and their reference points.PAM (Partitioning around Medoid) is one of the first proposed K-medoid algorithms. It tries to give k divisions for n objects. After initially choosing k centroids randomly, the algorithm iteratively tries to find out better centroids. All possible pairs of objects are analysed and in each pair one object is seen as the centroid and the other is not. The quality of the clustering result is estimated for each of the possible combinations. An object is replaced by the object that can produce a reduction in the maximum squared error value. The set of best objects produced in one iteration becomes the centroid for the next selected generation. It is not executed as efficiently as the K-means algorithm.

d) Only spherical clusters can be found: The K-means algorithm often uses the Euclidean distance-based sum-of-squares-of-errors criterion function as the clustering criterion function, which is more effective if there is a clear distinction between the classes, i.e., the similarity between the classes is very low; but in the opposite case, further splitting of large clusters may occur. Literature [3] uses density-based multicentre clustering and combines it with a small class merging operation to solve the computationally spatial minutiae, the convergence progress is controlled, and each iteration of the algorithm is inclined to discover hyper-spherical clusters, which have good clustering ability especially for the extension-like irregular clusters.

D. K-means++ Algorithm

K-means++ is an improved version of the traditional K-means algorithm, focusing on optimising the selection process of initial cluster centroids in order to improve the clustering effect and reduce the possibility of falling into local optimums.K-means++ was proposed by David Arthur and Sergei Vassilvitskii [4] in 2007, the core of K-means++ is to select initial cluster centroids by a strategy based on the probability distribution, which makes the distribution of initial centroids more reasonable. The core of K-means++ is to select the initial cluster centroids through a strategy based on probability distribution, which makes the distribution of initial centroids more reasonable. K-means++ has the following significant advantages over traditional K-means:

- Better clustering quality: By choosing the initial points more reasonably, K-means++ can significantly improve the final clustering effect, especially when the data distribution is uneven or the cluster shape is complex.
- Faster convergence: The initial cluster centroids are well-distributed, allowing K-means to converge in fewer iterations.
- Reduced randomness: K-means++'s initial points are chosen with a certain degree of regularity, which greatly reduces the instability of the algorithm's results.
- Easy to implement: Although the initial point selection process of K-means++ adds some computational complexity, the added complexity is almost negligible compared to the overall time cost of the algorithm.

IV. Other Algorithms in the Field of Cluster Analysis

A. Hierarchical Clustering Algorithm

A hierarchical clustering method constructs and maintains a clustering tree of clusters and sub-clusters according to a given inter-cluster distance metric until some end condition is satisfied. Depending on whether the hierarchical decomposition is formed bottom-up or top-down, hierarchical clustering methods can be classified as agglomerative and divisive. The clustering quality of a pure hierarchical clustering method is limited by the fact that once a merge or division is performed, it cannot be corrected. [5]

a) agglomerative hierarchical clustering: This bottom-up strategy starts by treating each object as a cluster, and then merges these atomic clusters into larger and larger clusters until all objects are in a single cluster, or a certain end condition is satisfied. The vast majority of hierarchical clustering methods fall into this category, and they differ only in the definition of inter-cluster similarity.

b) divisive hierarchical clustering: this top-down strategy is the opposite of cohesive hierarchical clustering, in that it starts by placing all the objects in a single cluster, and then progressively subdivides them into smaller and smaller clusters until each object forms a cluster of its own, or a certain end condition is met, such as a desired number of clusters is reached, or the distance between the two closest clusters exceeds a certain threshold value.

B. DBSCAN

Density clustering method means that as long as the density of the points in a region is greater than a certain value, it will be added to the neighbouring classes.DBSCAN algorithm is a representative algorithm in the density clustering method, which defines the cluster as the largest collection of points connected by density, and as long as the density of the neighbouring regions (the number of objects or data points) exceeds a certain threshold, it will continue to be clustered, and the DBSCAN algorithm is highly resistant to 'noise'. 'noise' ability. However, DBSCAN algorithm also has certain shortcomings:

- In the clustering process, using the DBSCAN algorithm, once you find a core outward expansion, then in the process of expansion of the core object will be more and more objects that have not been clustered will remain in memory. If there is a large clustering in the original database, then solving the problem of storing core object information will be very tricky.
- Input parameter sensitivity . Determining the parameters Eps, MinPts is difficult and if not selected correctly, it will result in degradation of clustering quality.
- Since the variables Eps, MinPts are globally unique in the DBSCAN algorithm, the quality of clustering is poor when the data is unevenly distributed.

V. Experiment Content

This experiment focuses on clustering analysis of a dataset containing 3000 2D coordinates through various clustering algorithms mentioned above, and the clustering results are analysed through the evaluation metrics described below.

A. Evaluation indicators

a) Silhouette Coefficient: The profile coefficient measures the difference between the similarity of a data point to other points in its cluster and the similarity to the nearest cluster. The range is $[-1, 1]$, with larger values indicating better clustering.

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

- a: the average distance of a sample from other samples in the same cluster (intra-cluster closeness).
- The average distance of this sample from the samples in the nearest cluster (inter-cluster separation).
- Characteristics: suitable for assessing the closeness and separation of clusters, the larger the value the better.

b) Calinski-Harabasz Index: Calinski-Harabasz index (CH index), also known as Variance Ratio Criterion (VRC), is one of the metrics used to evaluate the quality of clustering. It measures the effectiveness of clustering by the ratio of inter-cluster dispersion to intra-cluster tightness.The formula for CH index is:

$$\mathrm{CH} = \frac{B_k(\mathrm{n} - \mathrm{k})}{W_k(\mathrm{k} - 1)}$$

- n is the total number of samples in the dataset, and k is the number of clusters.
- Inter-cluster Dispersion(B_k): Indicates the degree of separation of the clusters, the greater the distance between the cluster centre and the overall centroid, the greater the inter-cluster dispersion, indicating the better the separation between the clusters.
- Intra-cluster tightness(W_k): indicates the degree of tightness of the clusters, the smaller the distance from the cluster samples to the cluster centre, the smaller the intra-cluster tightness, indicating the better the tightness of the samples within the clusters.
- The larger the value of CH index, the better the separation between clusters, the stronger the intra-cluster tightness,

and the better the clustering effect. It is suitable for evaluating the clustering algorithm's ability to group data, but it is usually necessary to compare the CH index at multiple k values to select the optimal number of clusters.

c) Davies-Bouldin Index: The Davies-Bouldin index(DB index) is a metric that assesses the effectiveness of clustering by calculating the closeness of samples within clusters and the separation of samples between clusters.

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{\sigma_i + \sigma_j}{d_{ij}}\right)$$

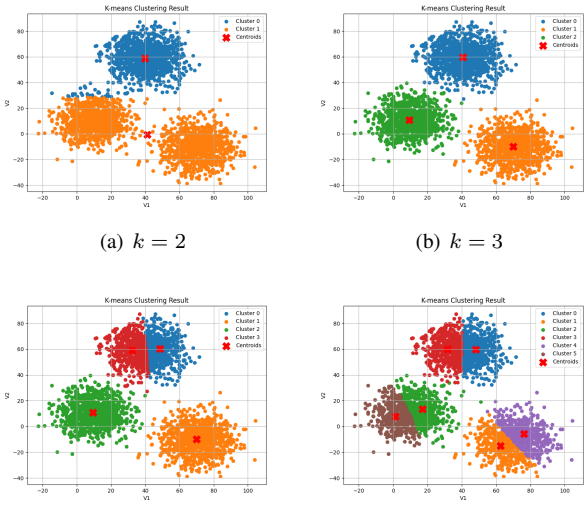
- k denotes the number of clusters and $sigma_i$ denotes the mean radius of cluster i (intra-cluster compactness).
- d_{ij} is the distance between cluster i and cluster j (inter-cluster separateness), which is usually calculated using the Euclidean distance:

$$d_{ij} = ||\mu_i - \mu_j||$$

In the above equation, μ_i is the centroid of the i cluster.
- The DB index measures the 'worst separation' of each cluster from other clusters, i.e. the closer the closeness and separation between clusters, the larger the DB index.
- Smaller values indicate better clustering because smaller DB values mean that the samples within clusters are closer together and the distances between clusters are more separated.

B. K-means And K-means++

The results of clustering analysis of the 2D coordinates in the xclara file using the K-means algorithm are shown in Fig. 2, which shows the results of setting the value of the number of clusters, k, to 2, 3, 4, and 6, respectively, and the results of setting the k values corresponding to the silhouette coefficient, the CH index, and the DB index are demonstrated in Table 1.



(a) $k = 2$ (b) $k = 3$

(c) $k = 4$ (d) $k = 6$

Fig. 2. K-means algorithm clustering results

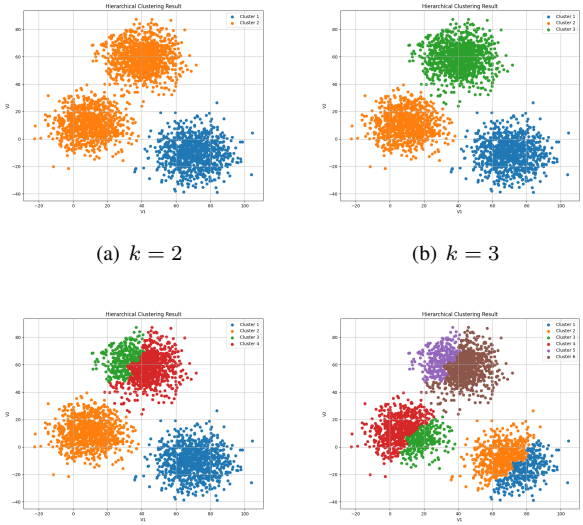
TABLE I
K-MEANS EVALUATION INDICATORS

k	Silhouette Coefficient	CH Index	DB Index
2	0.5093	3055.9875	0.7839
3	0.6946	10826.6006	0.4206
4	0.5393	8381.7464	0.8282
6	0.3110	6773.9895	1.2370

The data in the xclara file was then analysed by clustering using the K-means++ algorithm, and finally the results were found to be almost identical to the evaluation metrics of the clusters, so the results are not repeated.

C. Hierarchical Classification

The results of clustering analysis of the data in the xclara file using the Hierarchical Classification algorithm are shown in Fig. 3, which shows the results of setting the value of the number of clusters, k, to 2, 3, 4, and 6, respectively, and the results of setting the k values corresponding to the silhouette coefficient, the CH index, and the DB index are demonstrated in Table 2.



(a) $k = 2$ (b) $k = 3$

(c) $k = 4$ (d) $k = 6$

Fig. 3. K-means++ algorithm clustering results

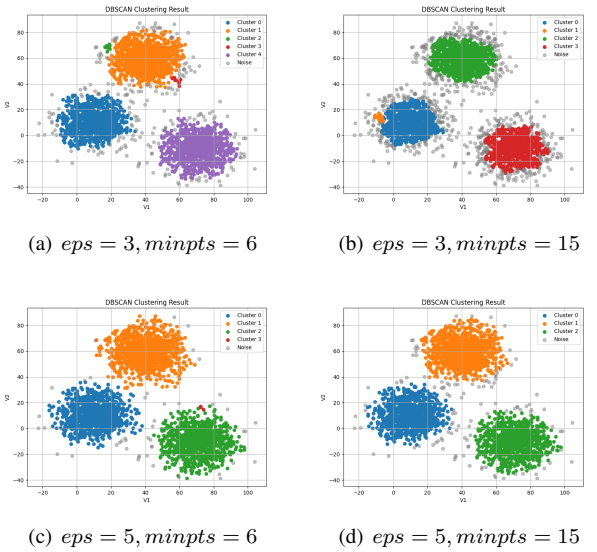
TABLE II
HIERARCHICAL CLASSIFICATION EVALUATION INDICATORS

k	Silhouette Coefficient	CH Index	DB Index
2	0.5428	3515.9397	0.6736
3	0.6942	10793.0639	0.4196
4	0.5333	8171.1109	0.8073
6	0.2917	6354.0829	1.2204

D. DBSCAN

The results of clustering analysis of data in xclara file using DBSCAN algorithm are shown in Fig. 4. Fig. 4 shows the

results of setting eps and minpts parameters in clustering algorithm to different values.Table 3 shows the silhouette coefficient, the CH index, and the DB index of clustering results after setting different parameters.



(a) $eps = 3, minpts = 6$ (b) $eps = 3, minpts = 15$

(c) $eps = 5, minpts = 6$ (d) $eps = 5, minpts = 15$

Fig. 4. K-means++ algorithm clustering results

TABLE III
K-MEANS++ EVALUATION INDICATORS

eps/minpts	Silhouette Coefficient	CH Index	DB Index
3/6	0.4872	6348.9954	0.4777
3/15	0.6056	9545.6728	0.5102
5/6	0.5586	7562.8897	0.4728
5/15	0.7151	12011.1569	0.3971

E. Algorithm Analysis and Comparison

Based on the results and metrics presented in the figure, we can analyse and compare the performance of different clustering algorithms. In K-means and K-means++ algorithms, the clustering results demonstrate the distribution of clusters at different k values (2, 3, 4, 6). From the table, it can be seen that the contour coefficient (0.6946) of K-means clustering reaches its maximum when k=3, indicating that the clusters are best separated at this point. In addition, the DB index also minimises at k=3 (0.4206), further indicating that the tightness within clusters and the separation between clusters are better at this time. Although the CH index is the highest at k=4 (8381.7464), the combined profile coefficient and DB index indicate that k=3 is a better choice. Meanwhile, the indexes of K-means and K-means++ are almost the same, indicating that the optimisation effect of K-means++ is not obvious under the current dataset, which may be due to the more regular data distribution.

In Hierarchical Clustering, the clustering results under different k values are similar to those of K-means algorithm. The evaluation metrics show that the contour coefficient (0.6942) reaches the maximum and the DB index is the smallest (0.4196) when k=3, which indicates that the intra-cluster closeness and inter-cluster separation are better. In addition, the CH index reaches the highest value (8171.1109) at k=4, indicating that the overall dispersion of clusters is good at k=4. However, on the whole, hierarchical clustering is consistent with K-means results at k=3, indicating that k=3 is a more reasonable choice. The advantage of hierarchical clustering is that it can visualise the hierarchical structure of clusters, but the computational complexity is higher.

For the DBSCAN algorithm, the results do not depend on a fixed number of clusters, but are controlled by the parameters eps (radius of neighbourhood range) and minPts (minimum number of core points). The clustering results in the figure show that different combinations of parameters have a large impact on the clustering effect. For example, at eps=5, minPts=15, the profile coefficient of DBSCAN is the highest (0.7151), CH index is the largest (12011.1569), and DB index is the smallest (0.3971), which indicates that the clustering result is optimal at this time. Compared with K-means and hierarchical clustering, DBSCAN is able to deal with noisy points (grey points) and performs well in non-spherical clusters, but has a high sensitivity to the parameters, which needs to be optimized through parameter tuning.

Comprehensive comparison, K-means and hierarchical clustering perform better when the data distribution is more regular, especially when k=3, both of them have the best effect; while DBSCAN performs better when there are noisy points or irregular shape of clusters, and it achieves the optimal clustering effect by the parameter combination of eps=5, minPts=15. In terms of evaluation indexes, the contour coefficient, CH index and DB index provide a consistent basis that can help effectively assess the performance and applicability of different clustering algorithms. In view of the actual needs and data characteristics, suitable clustering methods and parameters can be selected to achieve the best results.

CONCLUSION

In response to the above findings, K-means and hierarchical clustering perform well when the data distribution is regular and the cluster shape is close to spherical, especially at k = 3, where both methods achieve the best results with the highest profile coefficient and the lowest DB index. Although the improvement of K-means++ is theoretically able to optimise the selection of initial clustering centres, its effect is similar to that of K-means on the current dataset, suggesting that its optimisation is not obvious with regular data distributions.

In contrast, the DBSCAN algorithm does not rely on a fixed number of clusters, and is able to flexibly adjust the clustering results by the parameters epsilon (neighbourhood radius) and minPts (minimum number of core points), which is especially good in the presence of noisy points or irregular cluster shape. However, DBSCAN is more sensitive to the parameters and needs to optimise the parameter settings through experiments.

Taken together, K-means and hierarchical clustering are more suitable for regularly distributed data, while DBSCAN performs better when dealing with noisy points or non-spherical clusters. By combining the characteristics of clustering algorithms and the results of evaluation indexes, users can choose appropriate clustering methods and parameters according to the characteristics of the actual data, so as to achieve the best clustering effect.

REFERENCES

[1] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[3] 毛韶阳and 李肯立, "优化k-means 初始聚类中心研究," 计算机工程与应用, vol. 43, no. 22, pp. 179–181, 2007.

[4] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

[5] 杨俊闯and 赵超, "k-means 聚类算法研究综述," 计算机工程与应用, vol. 55, no. 23, pp. 7–14, 2019.