J Inf Process Syst, Vol.20, No.2, pp.185~199, April 2024
https://doi.org/10.3745/JIPS.04.0307

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

An Improved Automated Spectral Clustering Algorithm

Xiaodan Lv*

Abstract

In this paper, an improved automated spectral clustering (IASC) algorithm is proposed to address the limitations of the traditional spectral clustering (TSC) algorithm, particularly its inability to automatically determine the number of clusters. Firstly, a cluster number evaluation factor based on the optimal clustering principle is proposed. By iterating through different k values, the value corresponding to the largest evaluation factor was selected as the first-rank number of clusters. Secondly, the IASC algorithm adopts a density-sensitive distance to measure the similarity between the sample points. This rendered a high similarity to the data distributed in the same high-density area. Thirdly, to improve clustering accuracy, the IASC algorithm uses the cosine angle classification method instead of K-means to classify the eigenvectors. Six algorithms—K-means, fuzzy C-means, TSC, EIGENGAP, DBSCAN, and density peak—were compared with the proposed algorithm on six datasets. The results show that the IASC algorithm not only automatically determines the number of clusters but also obtains better clustering accuracy on both synthetic and UCI datasets.

Keywords

Cosine Angle Classification Method, Cluster Number Evaluation Factor, Density-Sensitive Distance, Spectral Clustering, UCI

1. Introduction

Clustering analysis is a highly effective method for mining deep information and correlating data. The goal of clustering is to divide a dataset into several clusters. The optimal criterion for clustering is characterized by high intraclass similarity and low interclass similarity [1]. Recently, cluster analysis has been widely used for image processing [2,3], biological information [4–6], and pattern recognition [7–9], among others.

The K-means [10] algorithm randomly initializes the cluster center, calculates the distance from each sample point to each center, and divides the category of sample points according to the distance from the center. Subsequently, the K-means algorithm calculates new center points for each cluster and circulates this process until the positions of all the center points no longer change. The concept of K-means algorithm is straightforward; however, it has notable disadvantages, including sensitivity to initial centers and outliers, as well as the lack of automatic determination of the number of clusters. To enhance the reliability of the K-means algorithm, researchers introduced the flexible partition idea of fuzzy mathematics into cluster analysis, leading to the development of fuzzy C-means (FCM) algorithm. FCM algorithm [10] uses a membership function to classify different sample points. It minimizes the cost by repeatedly calculating the membership function and clustering center to obtain the clustering result.

Manuscript received November 22, 2022; first revision February 14, 2023; accepted March 11, 2023.
* **Corresponding Author:** Xiaodan Lv (20210051@huat.edu.cn)
Institute of Automotive Engineers, Hubei University of Automotive Technology, Shiyan, China (20210051@huat.edu.cn)

Compared with the K-means algorithm, the FCM algorithm has better reliability; however, the clustering amount cannot be automatically determined. Ester et al. [11] presented a DBSCAN algorithm to discover clusters with random shapes. As a classical density-based clustering algorithm, DBSCAN calculates the tightness of the samples to recognize the categories of sample points. In the sample space, particles with a high aggregation density are divided into one cluster. This algorithm avoids the requirement of inputting a specific number of clusters and efficiently handles arbitrarily shaped datasets. However, DBSCAN is sensitive to its parameters and requires manual adjustment of the eps and MinPTs values to achieve better cluster results. To address this problem, Rodriguez and Laio [12] proposed a density peaks (DP) algorithm. The DP algorithm assumes that cluster centers exhibit a higher local density than neighbors and are relatively far from any other point with a higher local density. DP algorithm calculates each data point's local density ρ_i and its distance δ_i from other higher density points. Then, points with a high δ_i value and relatively high ρ_i value are manually selected as clustering centers based on the decision graph. Finally, the DP algorithm assigns the remaining points to the same categories as its nearest neighbor, which has a higher density. The result of the DP algorithm is robust with a parameter cutoff distance dc. Despite the significant improvement over DBSCAN, the clustering performance exhibited by the DP algorithm on non-convex datasets is not optimal. To enhance the performance of the algorithm on non-convex datasets, researchers have proposed a traditional spectral clustering (TSC) algorithm [13]. The TSC builds an undirected graph through the connections between sample points. In this graph, the nodes represent each data point, and the sides represent the similarity between the data points. The TSC then cuts this graph to minimize the weight of the cutting edge. At this point, the clustering problem is transformed into a graph-segmentation problem. The TSC algorithm improves the processing ability of high-dimensional data and performs well on non-convex datasets. Although the TSC is a competitive clustering algorithm, the clustering number cannot be determined automatically.

In order to make TSC algorithm can automatically determine the number of clusters, this paper presents an improved automated spectral clustering (IASC) algorithm. In this study, the clustering number evaluation factor was defined based on the clustering principle of "higher intra-class similarity and lower inter-class similarity." By iterating with different k values, the IASC algorithm selects the k value corresponding to the largest evaluation factor as the final number of clusters. In addition, to ensure that data distributed in the same-density region share higher similarity, IASC uses density-sensitive distance to calculate similarity. Simultaneously, to improve clustering accuracy, the IASC algorithm classifies feature vectors using the cosine angle method. After conducting the comparative experiment, the results reveal that the IASC algorithm obtains the cluster number automatically based on better clustering accuracy, which is an effective improvement on TSC.

2. Principle of TSC

The TSC algorithm is derived from spectrum division theory [14,15]. From a graph theory perspective, clustering is equivalent to the optimal partitioning of an undirected graph. All samples in the dataset are considered as node set V, and the connections (or similarities) between samples are considered as the edge set E of the undirected graph. Together, they constitute an undirected graph $G = (V, E)$. The task of spectral clustering is to determine an optimal partitioning method for the graph such that the resulting subgraphs exhibit the following characteristics: the total edge weight between different subgraphs is minimized, while the total edge weight between nodes within the same subgraph is maximized. Thus, the

objective function can be defined as:

$$\text{RatioCut}(A_1, A_2 \dots \dots . A_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{|A_i|}. \tag{1}$$

In Eq. (1), $A_1, A_2 \dots \dots . A_k$ represents k clusters after partitioning the dataset, $W(A_i, \overline{A_i})$ represents the similarity between cluster A_i and the other clusters, while $|A_i|$ represents the number of nodes in cluster A_i. Therefore, the problem of spectrum clustering is transformed into the problem of determining the minimum value of Eq. (1).

In general, it is difficult to find the minimum value of Eq. (1) directly. A better solution is to consider the continuous relaxation form of the problem and convert the problem of minimizing the objective function into a spectral decomposition problem of a Laplace matrix [10]. Firstly, we constructed an adjacency matrix W in graph G. W_{ij} in the matrix represents the similarity between the sample points x_i and x_j, which is defined as follows:

$$W_{ij} = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}. \tag{2}$$

The TSC algorithm then builds a Laplacian matrix, as defined in Eq. (3). In Eq. (3), D is the degree matrix of W.

$$L = D - W. \tag{3}$$

Definition $f = (f_1, f_2, f_3 \dots \dots . f_n)' \in R_n$ and

$$f_i = \begin{cases} \sqrt{\frac{|\bar{A}|}{|A|}} v_i \in A \\ -\sqrt{\frac{|A|}{|\bar{A}|}} v_i \in \bar{A} \end{cases}. \tag{4}$$

According to the definition of vector f, it can be inferred that:

$$\begin{aligned} f'Lf &= f'Df - f'Wf \\ &= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij} \\ &= \frac{1}{2}\left[\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2\right] \\ &= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \\ &= \frac{1}{2}\left[\sum_{i\in A, j\in \bar{A}} w_{ij}\left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}}\right)^2 + \sum_{i\in \bar{A}, j\in A} w_{ij}\left(-\sqrt{\frac{|\bar{A}|}{|A|}} - \sqrt{\frac{|A|}{|\bar{A}|}}\right)^2\right] \\ &= |V|\,\text{RatioCut}(A, \bar{A}), \end{aligned} \tag{5}$$

$$\| f \|^2 = \sum_{i=1}^{n} f_i^2 = |A| \frac{|\bar{A}|}{|A|} + |\bar{A}| \frac{|A|}{|\bar{A}|} = n, \tag{6}$$

$$f \cdot E = \sum_{i=1}^{n} f_i = \sum_{i \in A} \sqrt{\frac{|\bar{A}|}{|A|}} - \sum_{i \in \bar{A}} \sqrt{\frac{|A|}{|\bar{A}|}} = |A| \sqrt{\frac{|\bar{A}|}{|A|}} - |\bar{A}| \sqrt{\frac{|A|}{|\bar{A}|}} = 0. \tag{7}$$

In Eq. (5), $|V|$ represents the number of nodes in graph G. In Eq. (7), E represents the unit matrix. Eq. (5) shows that minimizing the objective function RatioCut is equivalent to the minimization of $f'Lf$, while simultaneously satisfying the constraint conditions in Eqs. (6) and (7). Assuming that the eigenvalue of the Laplace matrix L is denoted by λ, and that the eigenvector is f, a new equation is obtained:

$$f'Lf = f'\lambda f = \lambda f'f = \lambda n. \tag{8}$$

In Eq. (8), n is a constant representing the number of samples in the dataset. Minimization of the objective function RatioCut was used to acquire the minimum eigenvalue of the Laplace matrix. In addition, the Laplace matrix L is a symmetric semidefinite matrix, and all eigenvalues are not less than zero.

$$L^*E = (D - W) * E = 0^*E. \tag{9}$$

According to Eq. (9), the lowest eigenvalue of the Laplace matrix is 0. However, the associated eigenvector is E, which does not satisfy the constraint conditions as shown in Eq. (7). According to the Rayleigh–Ritz theory, it is possible to obtain the second-smallest eigenvalue and its corresponding eigenvector. Furthermore, we obtained the first k smallest eigenvalues and their corresponding eigenvectors of L. These k eigenvectors were then arranged to form an $N \times K$ matrix. K-means clustering was performed on the row vectors of this matrix to obtain k clusters, and the final clustering results were mapped back to the original space.

The workflow of TSC is as follows [13,15–17]:

Step 1: Construct a matrix to represent the relationship between the sample points in the dataset, the similarity matrix W. W_{ij} represents the similarity between x_i and x_j, calculated using Eq. (2).

Step 2: Obtain the degree matrix D of the similarity matrix W and calculate the Laplace matrix using $L = D - W$.

Step 3: Arrange the eigenvalues of L from smallest to largest and obtain the first k eigenvalues and their corresponding eigenvectors.

Step 4: Combine these k eigenvectors into an $N \times k$ matrix, where each row represents a k-dimensional vector. After performing K-means clustering, the classes assigned to each row in the clustering result represent the categories of each sample in the original datasets.

3. IASC Algorithm

The TSC algorithm is primarily focused on the size of the dataset rather than the dimensions of the data. This feature enhances its ability to process high-dimensional data effectively. Additionally, as a

non-central clustering algorithm, it is capable of producing optimal clustering results for non-convex datasets. However, the TSC algorithm exhibits three notable disadvantages. Firstly, it requires the manual input of the number of clusters. Predicting the appropriate number of clusters can be challenging in practical scenarios, presenting an obstacle to the widespread application and promotion of TSC [18]. Therefore, it is necessary to design an automated spectral clustering algorithm. Secondly, as a highly competitive algorithm, its success is inseparable from that of the similarity measure method. Although the algorithm uses the Euclidean distance to measure the similarity between samples, this approach is sensitive to scale parameters and fails to account for the global consistency characteristics of the data distribution. After spectral decomposition and K-means clustering, the error caused by the lack of features is amplified, affecting the clustering results. Thirdly, the TSC algorithm uses K-means clustering to classify eigenvectors, which may not perform optimally with high-dimensional vectors.

Therefore, it is imperative to achieve automatic clustering and enhance clustering accuracy by improving the similarity measurement method and optimizing K-means clustering techniques.

3.1 Cluster Number Evaluation Factor

Numerous classical clustering algorithms, such as K-means, FCM, and TSC, require manual specification of the value of k. However, it is impossible to determine the number of clusters, k, through prior knowledge, which poses an obstacle to the advancement and application of clustering. Therefore, the automatic determination of clustering numbers has become a popular research topic in recent years.

Kong et al. [18] ranked all the eigenvalues of the Laplace matrix from smallest to largest and then utilized the eigengap to describe the difference between adjacent eigenvalues. They automatically obtained clustering numbers by determining the location of the largest eigengap value. However, this method is difficult to understand and requires abundant calculations, resulting in poor efficiency when the data dimensions increase [19]. Porter and Canagarajah [20] found that the objective function decreases monotonically with the number of clusters. Initially, the objective function decreases rapidly as the value of the independent variable (number of clusters) increases. After reaching a certain number of clusters, the rate of decline slowed down. Therefore, the k value corresponding to the critical point is calculated as the best clustering number. However, this method only considers compactness within a class and does not consider dispersion between classes [21]. Chen et al. [22] constructed a cumulative adjacency matrix by integrating multiple FCM clustering results and segmenting the cumulative adjacency matrix using an iterative method to output the final results.

By combining the advantages of the above algorithms, we designed a cluster number evaluation factor and introduced it into a spectral clustering algorithm to automatically determine the number of clusters. The evaluation factor comprehensively considers the compactness within a class and dispersion between classes. A larger evaluation factor indicates a more compact intraclass and a distinct interclass. Simultaneously, this study integrates the results of multiple spectral clusters. By iterating through different cluster numbers to obtain different evaluation factor values, the IASC algorithm considers the cluster number corresponding to the maximum evaluation factor value as the optimal cluster number. The evaluation factor V is calculated as follows:

$$V = \frac{\sum_{x_i, x_j \in A_1} W_{ij}}{\sum_{i,j=1}^{n} W_{ij} - 2\sum_{x_i, x_j \in A_1} W_{ij}}. \tag{10}$$

In Eq. (10), W_{ij} represents the element value in the *i*th row and *j*th column of the similarity matrix W, which denotes the similarity between x_i and x_j. Here, $x_i, x_j \in A_1$, indicating that data x_i and data x_j belong to the same class. The numerator *V* represents the sum of the similarities between all data points belonging to one class, while the denominator *V* represents the sum of the similarities between data points belonging to different classes. The evaluation factor *V* effectively evaluates the tightness intraclass and dispersion interclass. The larger the value of *V*, the more compact the intraclass data and the more distinct the interclass data.

3.2 Density Sensitive Distance Similarity

For complex problems, adapting to the characteristics of the spatial distribution of data is almost impossible. Typically, sample points belonging to the same category are often distributed within high-density areas, while those of different categories may reside in low-density regions. Unfortunately, the similarity calculated using Euclidean distance does not meet this requirement.

In such cases, it is necessary to design a new similarity measurement method that elongates paths passing through low-density regions while shortening others. To address this challenge, Wang et al. [23] designed a density-adjustable line segment.

$$L_{ij} = \rho^{\mathrm{dist}\,(x_i, x_j)} - 1. \tag{11}$$

In Eq. (11), $dist(x_i, x_j)$ represents the Euclidean distance between data points x_i and x_j, and the stretching factor ρ is greater than 1.

In Fig. 1 [11], based on the definition of a density-adjustable line segment, we obtain

When $\rho = 2$, $L(a, f)+L(f, e)+L(e, d)+L(d, c)+L(c, b) = 1+1.8+3+1+1.8 = 8.6 < L(a, b) = 31$;

When $\rho = 3$, $L(a, f)+L(f, e)+L(e, d)+L(d, c)+L(c, b) = 2+4.2+8+2+4.2 = 20.4 < L(a, b) = 241$.

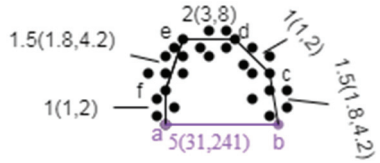


Fig. 1. Illustration of density-adjustable line segments. The numbers in parentheses are the distances between two adjacent points when ρ is equal to 2 and 3.

As shown in Fig. 1, the distance calculated using the density-sensitive line segment makes the line segment passing through the high-density region shorter than that passing through the low-density region. For example, $L(a, b) > L(a, f)+L(f, e)+L(e, d)+L(d, c)+L(c, b)$. This new distance measurement method reflects the global consistency of the spatial distribution of data and lays the foundation for similarity measurements between data.

Based on the density-adjustable line segment, we designed a method to calculate the similarity between sample points. The density-sensitive distance is calculated as follows:

$$D_{ij} = \min_{p \in P_{ij}} \sum_{k=1}^{l-1} L(p_k, p_{k+1}). \tag{12}$$

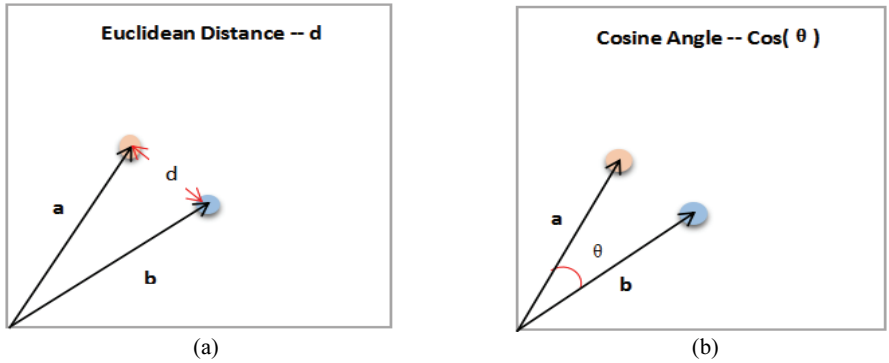
In Eq. (12), P_{ij} stands for the set of connection paths between points x_i and x_j, L represents the density adjustable line segment's size, and D_{ij} represents the shortest path between x_i and x_j. Furthermore, the similarity matrix W is defined based on the density-sensitive distance, and the calculation formula for W_{ij} is as follows:

$$W_{\mathrm{ij}} = \frac{1}{D_{ij}}. \tag{13}$$

In Eq. (13), D_{ij} is the density-sensitive distance. This similarity calculation method reflects the reverse correlation between density-sensitive distance and similarity. That is, the data in the same-density area had higher similarity, and the data in different-density areas had lower similarity. It considers the global consistency of the spatial distribution of data. Compared to the Gaussian kernel function proposed by Chapelle and Zien [24], this similarity calculation method reduces the number of parameters. Compared with the method proposed by Yang et al. [25], this similarity calculation method simplifies the calculation process and improves the calculation efficiency, which is evident when the amount of data is large.

3.3 Cosine Angle Classification Method

K-means was used to classify the eigenvectors according to the algorithm flow of the TSC. The core principle of the K-means algorithm is to classify vectors according to their Euclidean distance, which means that two vectors with shorter distances are more similar and vice versa. The Euclidean distance was used to measure the shortest distance between the two vectors. It is widely used owing to its simplicity. However, the Euclidean distance is difficult to normalize and does not perform well for high-dimensional vectors. The cosine angle value is a normalized quantity that is more suitable for measuring the similarity between higher-dimensional vectors (Fig. 2).



(a) (b)

Fig. 2. (a) Euclidean distance and (b) cosine angle.

Because most eigenvectors are higher-dimensional vectors, based on this consideration, we propose a cosine-angle classification method instead of K-means. The cosine of the vector angle is calculated as follows:

$$\cos(\theta) = \frac{x \cdot y}{|x \parallel y|}. \tag{14}$$

In Eq. (14), θ represents the angle between the vector x and the vector y. $|x|$ represents the magnitude of the vector x, and $|y|$ represents the magnitude of the vector y. According to the definition of the cosine function, the smaller θ is, the bigger $\cos(\theta)$ is, indicating that x is more similar to y. Thus, IASC uses the cosine angle instead of the K-means algorithm.

3.4 Main Processes of the IASC Algorithm

This paper proposes an IASC algorithm based on the TSC algorithm. In IASC, the cluster number evaluation factor is utilized to automatically determine the cluster number, the density-sensitive distance is used to measure the similarity between data, and the cosine angle classification method is applied for classifying eigenvectors instead of K-means. The IASC algorithm consists of two stages: Stage 1 involves the selection of cluster numbers, while Stage 2 involves obtaining the final clusters.

Stage 1. Selection of cluster numbers

Input:	Dataset X
Output:	Cluster numbers k
1:	**for** k = 2 **to** K **do**
2:	Calculate L_{ij} based on Equation (11) and acquire matrix L.
3:	Calculate D_{ij} based on Equation (12) and acquire matrix D.
4:	Calculate W_{ij} based on Equation (13) and acquire the similarity matrix W.
5:	Calculate degree matrix D1 of W based on D1= diag(sum(W, 2)), and then obtain the Laplace matrix L based on L= D1-W.
6:	Using [V,~] = eigs(L, k,' SM') to obtain eigenvectors related to the smallest k eigenvalues of L.
7:	Each row of matrix V is clustered using the K-means algorithm. In the clustering results, the category of each row represents the category of each sample in the original dataset.
8:	Calculate V_k based on Equation (10).
9:	**end**
10:	Return k value corresponding to the maximum evaluation factor V_k.

The input of Stage 1 includes dataset X and the maximum number of clusters, K. Here, line 1 specifies the range of k, that is, [2, K], and iterates the integer values in this range. Subsequently, operations from lines 2–8 are performed repetitively. Line 2 uses Eq. (11) to calculate the length of the density-sensitive line segments between any data point to acquire matrix L. Line 3 uses Eq. (12) to calculate the shortest path between any two points and form matrix D based on the Floyd algorithm [26]. Line 4 uses Eq. (13) to calculate the similarity W_{ij} between any two points and obtain the similarity matrix W. Line 5 calculates the degree matrix and Laplace matrix based on $D1 = diag(sum(W, 2))$ and $L = D1 - W$. Line 6 arranges the eigenvalues of L from small to large to obtain the first k eigenvalues and the corresponding eigenvectors. Line 7 arranges these k eigenvectors to constitute an $N \times k$ matrix. After applying K-means to each row of the matrix, the category of each row was the category of each sample in the original dataset. Line 8 calculates the evaluation factor according to Eq. (10) and adds it to array V. Line 10 returns the k value corresponding to the maximum evaluation factor value V_k as cluster numbers.

Stage 2. Obtain the final clusters

Input:	Dataset X, Cluster numbers k
Output:	Final Clusters C, Where C_k represents k-th cluster
1:	Calculate L_{ij} between X_i and X_j based on Equation (11) and acquire matrix L.
2:	Calculate D_{ij} based on Equation (12) and acquire matrix D.
3:	Calculate W_{ij} based on Equation (13) and acquire the similarity matrix W.
4:	Calculate degree matrix D1 of W based on D1= diag(sum(W, 2)), and then obtain the Laplace matrix L based on L = D1-W.
5:	Using [V,~] = eigs(L, k,'SM') to obtain eigenvectors related to the smallest k eigenvalues of L.
6:	Each row of matrix V is clustered using the cosine angle classification method. In the clustering results, the category of each row is the category of each sample in the original dataset X.
7:	Return the final clusters C.

The input of Stage 2 includes the dataset X and cluster number k. Line 1 uses Eq. (11) to calculate the length of the density-sensitive line segments between any data point to acquire matrix L. Line 2 uses Eq. (12) to calculate the shortest path between any two points and form matrix D based on the Floyd algorithm [26]. Line 3 uses Eq. (13) to calculate the similarity W_{ij} between any two points and obtain the similarity matrix W. Line 4 calculates degree matrix D1 and Laplace matrix L based on $D1 = diag(sum(W, 2))$ and $L = D1 - W$. Line 5 arranges eigenvalues of L from small to large and obtains the first k eigenvalues and corresponding eigenvectors. Line 6 arranges these k eigenvectors to constitute an $N \times k$ matrix: After applying the cosine-angle classification method to each row of the matrix, the category of each row was the category of each sample in the original dataset. Line 7 returns to the final cluster C.

4. Experimental Results

To evaluate the performance of the IASC algorithm, we verified the verification process on six datasets, including three synthetic datasets and three UCI datasets, which are real-world datasets from the University of California, Irvine [27]. The test datasets are presented in Table 1. From Table 1, we extracted the names, attributes, clusters, instances, and data sources. The experiment comprised two parts. Part 1 studies the determination of cluster numbers, and Part 2 studies the clustering accuracy of the different algorithms.

For comparison, six additional clustering algorithms were implemented to validate the IASC algorithm. These algorithms include K-means [1], FCM [1], TSC [16], EIGENGAP [19], DBSCAN [11], and DP

Table 1. Introduction to the datasets

Dataset	Attributes	Clusters	Instances	Data source
Lines	2	4	400	Synthetic
Luoxuan	2	2	252	Synthetic
TwoMoon	2	2	600	Synthetic
BreastCancer	10	2	683	UCI
Transfusion	4	2	748	UCI
Ecoli	7	2	272	UCI

[12]. To ensure the principle of one variable, the TSC and IASC parameters were identical. The test environment [28] of this experiment is as follows: central processing unit (CPU) is Intel Core I5-6200U CPU @2.30 GHz 2.4 kHz; memory space is 4 GB; programming environment is MATLAB; programming language is m.

Table 2. Experimental results of the selection of the cluster amount

Dataset	K-means	FCM	TSC	EIGENGAP	DBSCAN	DP	IASC
Lines	-	-	-	5	**4**	**4**	2
Luoxuan	-	-	-	3	1	**2**	**2**
TwoMoon	-	-	-	**2**	**2**	3	**2**
BreastCancer	-	-	-	-	3	1	**2**
Transfusion	-	-	-	4	17	1	**2**
Ecoli	-	-	-	5	1	**2**	**2**

The correct numbers of clusters are indicated in bold.

First, we conducted a cluster selection experiment. In Table 2, the dashed symbol indicates that the algorithm could not automatically obtain a cluster number. The K-means, FCM, and TSC algorithms were incapable of automatically obtaining cluster amounts on all datasets. This is because these algorithms must manually input clusters in advance. The EIGENGAP algorithm adopts the concept of EIGENGAP. A higher EIGENGAP value indicates a more stable subspace constructed using the selected k eigenvectors. The EIGENGAP algorithm considers the position of the first maximum of the intrinsic gap sequence as the number of categories. Because the eigenvalues of the matrix may be real or complex, the effect of this approach is not ideal. The DBSCAN algorithm calculates the tightness of the samples for classification. However, it performs well only on the Lines and TwoMoon datasets because the clustering effect of DBSCAN is sensitive to the parameters. The DP algorithm selects points with a high δ_i value and relatively high ρ_i value as clustering centers manually based on the decision graph. However, cluster results are influenced by many factors, such as human experience and data distribution shapes. Thus, the DP algorithm only obtained the correct number of clusters for three datasets. However, the IASC obtains a reasonable number of clusters on most datasets. The IASC algorithm calculates the corresponding evaluation factor value by iterating through varied k values and outputs the k values corresponding to the maximum evaluation factor as the final number of clusters to achieve automatic clustering. This demonstrated the competitive advantage of the IASC algorithm.

Second, a clustering accuracy experiment was conducted on six datasets. Because EIGENGAP and DBSCAN are sensitive to the parameters, we compared the IASC algorithm with the K-means, FCM, TSC, and DP algorithms. Table 1 contains three two-dimensional manual datasets; therefore, we used graphics to show the clustering results for ease of reading. The results for the manual datasets are shown in Figs. 3–5. In addition, there are three UCI datasets with high-dimensional attributes in Table 1, hence the results of the UCI datasets are displayed in Table 3.

Table 3. Clustering accuracy on UCI datasets

Dataset	K-means	FCM	TSC	DP	IASC
BreastCancer	0.3499	0.6032	0.3441	0.3438	**0.6428**
Transfusion	0.2607	0.2928	0.5267	0.2396	**0.6845**
Ecoli	0.0221	0.2022	0.1875	**0.7045**	0.5110

Bold indicates the algorithm with the highest accuracy in testing each dataset.

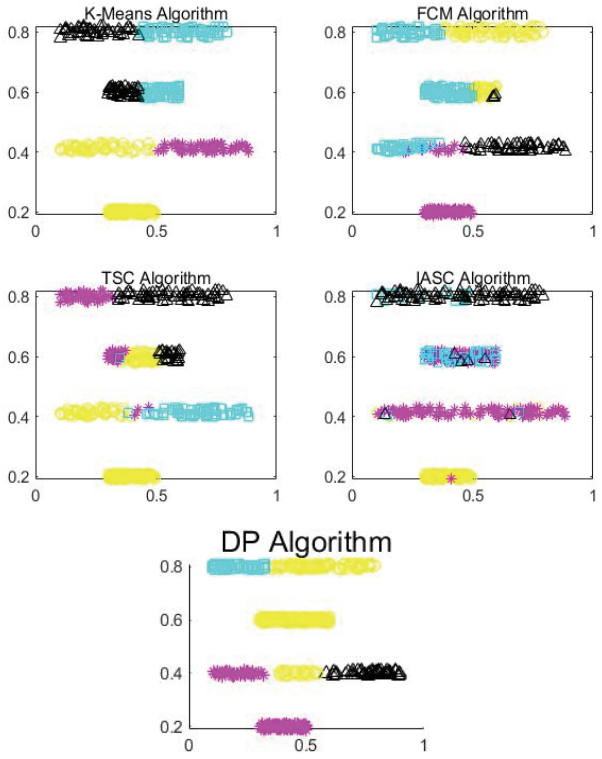


Fig. 3. Clustering result on the lines dataset (the same color represents the same category).

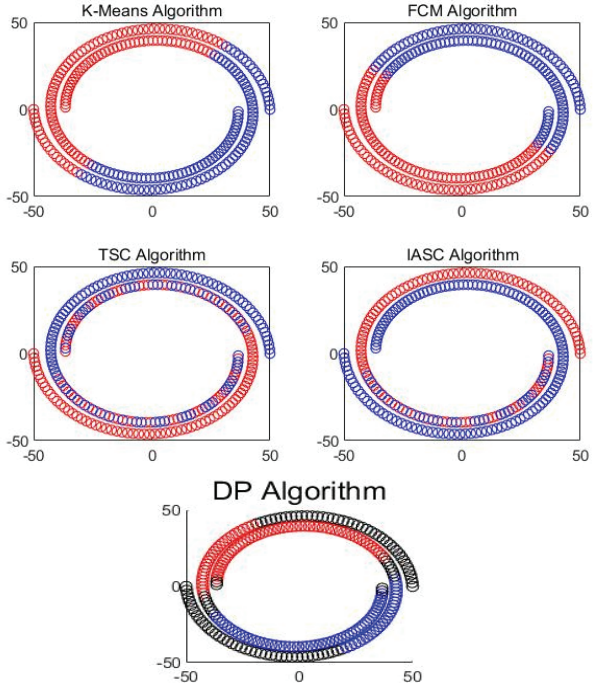


Fig. 4. Clustering result on the Luoxuan dataset (the same color represents the same category. In addition, black represents noise points in the DP algorithm).

The results in Figs. 3–5 indicate that the IASC algorithm achieves a better clustering effect on the three non-convex manual datasets, comparing with K-means, FCM, TSC, and DP. The clustering accuracy of five algorithms on the UCI datasets is presented in Table 3. The clustering accuracy of IASC is higher than those of K-means, FCM, TSC, and DP on most datasets. For example, on the Transfusion dataset, the K-means clustering accuracy was 0.2607, FCM's clustering accuracy was 0.2928, TSC's clustering accuracy was 0.5267, DP's clustering accuracy was 0.2396, and IASC's clustering accuracy was 0.6845.

In K-means, the Euclidean distance is used to estimate the similarities between points and centers. The algorithm allows each point to select the category of the center with the smallest distance as its own category. The FCM uses the Euclidean distance to build a cost function. When the cost function reaches its minimum, the algorithm converges and outputs the results. In the TSC, the Euclidean distance is adapted to calculate the similarity of sample points; the closer the Euclidean distance, the higher the similarity. The DP algorithm uses the Euclidean distance build decision diagram to select the cluster center and assigns sample points to different categories. However, the Euclidean distance only considers the local consistency of the spatial distribution of data and does not reflect global consistency; therefore, it is difficult for the above algorithm to achieve good clustering accuracy on non-convex datasets. The IASC algorithm uses density-sensitive distances to estimate the similarity between sample points, which reflects the characteristics of the spatial distribution of the data. This causes the points to be distributed in a high-density area with high similarity. In addition, the last step of the IASC algorithm uses the cosine angle method instead of K-means to classify the feature vectors because the cosine angle is normalized and is more suitable for measuring the similarity between higher-dimensional vectors.

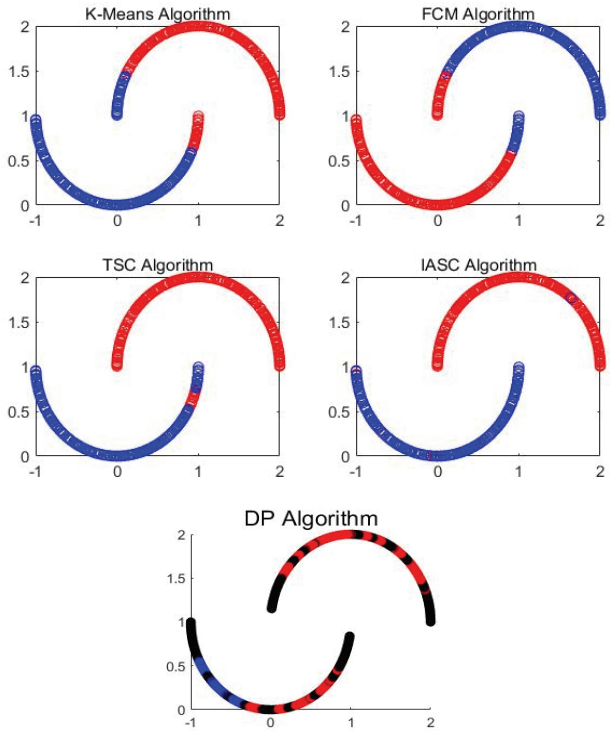


Fig. 5. Clustering result on the Twomoon dataset (the same color represents the same category. In addition, black represents noise points in the DP algorithm).

In summary, compared to other algorithms, the IASC algorithm greatly improves clustering accuracy through the density-sensitive similarity measure and cosine angle classification method. This is proof of the superiority of the IASC algorithm.

5. Conclusion

In this study, we propose the IASC algorithm for data analysis. To achieve automated clustering, the IASC algorithm introduces an evaluation factor into the spectral clustering. The corresponding evaluation factor value was calculated by iteratively varying the k values, and the k value corresponding to the maximum evaluation factor was selected as the final number of clusters.

The IASC algorithm then uses a density-sensitive distance to measure the similarity between samples, which makes the data distributed in a high-density area have a higher similarity. Furthermore, to improve cluster accuracy, the IASC algorithm adopts the cosine-angle method to classify the feature vectors.

It is concluded that the IASC algorithm is capable of automatically obtaining the correct cluster amount and demonstrating better cluster accuracy on most datasets than the other algorithms. Therefore, the IASC is more effective than the TSC algorithm.

Acknowledgement

This study was supported by the 2022 Annual Scientific Research Plan Project of the Hubei Provincial Department of Education (No. B2022352).

References

[1] L. Bai, X. Zhao, Y. Kong, Z. Zhang, J. Shao, and Y. Qian, "Survey of spectral clustering algorithms," *Computer Engineering and Applications*, vol. 57, no. 14, pp. 15-26, 2021. https://doi.org/10.3778/j.issn.1002-8331.2103-0547

[2] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594-2608, 2016. https://doi.org/10.1109/TIFS.2016.2590944

[3] K. Xia, X. Gu, and Y. Zhang, "Oriented grouping-constrained spectral clustering for medical imaging segmentation," *Multimedia Systems*, vol. 26, pp. 27-36, 2020. https://doi.org/10.1007/s00530-019-00626-8

[4] Z. Yu, H. Chen, J. You, J. Liu, H. S. Wong, G. Han, and L. Li, "Adaptive fuzzy consensus clustering framework for clustering analysis of cancer data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 4, pp. 887-901, 2015. https://doi.org/10.1109/TCBB.2014.2359433

[5] X. Jiang, M. Chen, W. Song, and G. N. Lin, "Label propagation-based semi-supervised feature selection on decoding clinical phenotypes with RNA-seq data," *BMC Medical Genomics*, vol. 14(Suppl 1), article no. 141, 2021. https://doi.org/10.1186/s12920-021-00985-0

[6] U. Agrawal, D. Soria, C. Wagner, J. Garibaldi, I. O. Ellis, J. M. S. Bartlett, D. Cameron, E. A. Rakha, and A. R. Green, "Combining clustering and classification ensembles: A novel pipeline to identify breast cancer profiles," *Artificial Intelligence in Medicine*, vol. 97, pp. 27-37, 2019.

[7] D. Xu, C. Li, T. Chen, and F. Lang, “A Novel low rank spectral clustering method for face identification,” *Recent Patents on Engineering*, vol. 13, no. 4, pp. 387-394, 2019. https://doi.org/10.2174/1872212112666180828124211

[8] S. Wazarkar and B. N. Keshavamurthy, “A survey on image data analysis through clustering techniques for real world applications,” *Journal of Visual Communication and Image Representation*, vol. 55, pp. 596-626, 2018. https://doi.org/10.1016/j.jvcir.2018.07.009

[9] Z. Ding, J. Li, H. Hao, and Z. R. Lu, “Structural damage identification with uncertain modelling error and measurement noise by clustering based tree seeds algorithm,” *Engineering Structures*, vol. 185, pp. 301-314, 2019. https://doi.org/10.1016/j.engstruct.2019.01.118

[10] Q. Wu, “Research and implementation of Chinese text clustering algorithm,” Ph.D. dissertation, Xidian University, Xi’An, China, 2010.

[11] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, Portland, OR, USA, 1996, pp. 226-231.

[12] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” *Science*, vol. 344, no. 6191, pp. 1492-1496, 2014. https://doi.org/10.1126/science.1242072

[13] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: analysis and an algorithm,” *Advances in Neural Information Processing Systems*, vol. 14, pp. 849-856, 2001.

[14] M. Fiedler, “Algebraic connectivity of graphs,” *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298-305, 1973. http://dx.doi.org/10.21136/CMJ.1973.101168

[15] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, 2000. https://doi.org/10.1109/34.868688

[16] H. Liu, J. Chen, J. Li, L. Shao, L. Ren, and L. Zhu, “Transformer fault warning based on spectral clustering and decision tree,” *Electronics*, vol. 12, no. 2, article no. 265, 2023. https://doi.org/10.3390/electronics12020265

[17] L. Hagen and A. B. Kahng, “New spectral methods for ratio cut partitioning and clustering,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 9, pp. 1074-1085, 1992. https://doi.org/10.1109/43.159993

[18] W. Z. Kong, Z. H. Sun, C. Yang, G. J. Dai, and C. Sun, “Automatic spectral clustering based on eigengap and orthogonal eigenvector,” *Acta Electronica Sinica*, vol. 38, no. 8, pp. 1880-1885+1891, 2010.

[19] Z. Hu and J. Weng, “Adaptive spectral clustering algorithm based on artificial bee colony algorithm,” *Journal of Chongqing University of Technology (Natural Science Edition)*, vol. 34, no. 3, pp. 137-144, 2020. https://doi.org/10.3969/j.issn.1674-8425(z).2020.03.020

[20] R. Porter and N. Canagarajah, “A robust automatic clustering scheme for image segmentation using wavelets,” *IEEE Transactions on Image Processing*, vol. 5, no. 4, pp. 662-665, 1996. https://doi.org/10.1109/83.491343

[21] C. Gao and X. Wu, "An automatic technique to determine cluster number for complex biologic datasets," *China Journal of Bioinformatics*, vol. 8, no. 4, pp. 295-298, 2010. https://doi.org/10.3969/j.issn.1672-5565.2010.04.003

[22] H. Chen, X. Shen, J. Long, and Y. Lu, “Fuzzy clustering algorithm for automatic identification of clusters,” *Acta Electronica Sinica*, vol. 45, no. 3, pp. 687-694, 2017.

[23] L. Wang, L. Bo, and L. Jiao, "Density-sensitive spectral clustering," *Acta Electronica Sinica*, vol. 35, no. 8, pp. 1577-1581, 2007.

[24] O. Chapelle and A. Zien, “Semi-supervised classification by low density separation,” in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, Bridgetown, Barbados, 2005, pp. 57-64.

[25] P. Yang, Q. Zhu, and B. Huang, “Spectral clustering with density sensitive similarity function,” *Knowledge-Based Systems*, vol. 24, no. 5, pp. 621-628, 2011. https://doi.org/10.1016/j.knosys.2011.01.009

[26] R. W. Floyd, “Algorithm 97: shortest path,” *Communications of the ACM*, vol. 5, no. 6, pp. 345-345, 1962. https://doi.org/10.1145/367766.368168
[27] UCI Machine Learning Repository, “Machine learning datasets,” c2023 [Online]. Available: https://archive.ics.uci.edu/.
[28] X. Xu, S. Ding, L. Wang, and Y. Wang, “A robust density peaks clustering algorithm with density-sensitive similarity,” *Knowledge-Based Systems*, vol. 200, article no. 106028, 2020. https://doi.org/10.1016/j.knosys.2020.106028

Xiaodan Lv **https://orcid.org/0000-0003-3102-1528**

She is a lecturer in the Institute of Automotive Engineers, Hubei University of Automotive Technology. She received her M.S. degree in Electronic and Communication Engineering from Guizhou University in 2021. Her research interests include machine learning, smart medical care, and data analysis..