AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 8(12): 28926–28950.
DOI:10.3934/math.20231482
Received: 06 August 2023
Revised: 07 October 2023
Accepted: 23 October 2023
Published: 25 October 2023

Research article

An automatic density peaks clustering based on a density-distance clustering index

Xiao Xu[1,2,*], **Hong Liao**[1] **and Xu Yang**[1]

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China
[2] Xuzhou First People's Hospital, Xuzhou 221116, China

* **Correspondence:** Email: xu_xiao@cumt.edu.cn.

Abstract: The density peaks clustering (DPC) algorithm plays an important role in data mining by quickly identifying cluster centers using decision graphs to identify arbitrary clusters. However, the decision graph introduces uncertainty in determining the cluster centers, which can result in an incorrect number of clusters. In addition, the cut-off distance parameter relies on prior knowledge, which poses a limitation. To address these issues, we propose an improved automatic density peaks clustering (ADPC) algorithm. First, a novel clustering validity index called density-distance clustering (DDC) is introduced. The DDC index draws inspiration from the density and distance characteristics of cluster centers, which is applicable to DPC and aligns with the general definition of clustering. Based on the DDC index, the ADPC algorithm automatically selects the suitable cut-off distance and acquires the optimal number of clusters without additional parameters. Numerical experimental results validate that the introduced ADPC algorithm successfully automatically determines the optimal number of clusters and cut-off distance, significantly outperforming DPC, AP and DBSCAN algorithms.

Keywords: DPC algorithm; automatic clustering; decision graph; optimal number of clusters; parameter selection
Mathematics Subject Classification: 91C20

1. Introduction

Clustering represents a versatile conceptual and algorithmic framework employed in diverse domains like pattern recognition, image segmentation, data mining and genetic disease detection, among others [1–3]. The fundamental objective of clustering is to categorize data points into meaningful clusters based on their similarity characteristics [4]. The overarching objective is to optimize similarity within clusters while minimizing it between distinct clusters [5, 6]. Over time, numerous clustering methods have emerged, encompassing the likes of k-means, AP, SC algorithms and others [7–9]. However, the performance of most conventional algorithms is constrained when dealing with datasets that exhibit arbitrary shapes and densities [10–12].

In 2014, the density peaks clustering (DPC) algorithm was introduced on the Science Journal, presenting two key features for identifying cluster centers [13]. First, cluster centers exhibit higher local density compared to their neighboring points. Second, cluster centers are positioned at relatively large distances from each other. By capitalizing on these unique traits, the DPC algorithm effectively identifies cluster centers through the construction of a decision graph. In addition, DPC does not require iterative processes or excessive input parameters [14]. As a simplistic yet highly efficient density-based clustering technique, DPC has played an eminent role in diverse domains, including data mining, community exploration, genetic disease investigation, biology and other related areas [15–19].

However, the DPC algorithm has a limitation as it may inaccurately estimate the number of clusters when selecting cluster centers based on the decision graph [20]. In addition, determining the appropriate input parameter d_c for satisfactory clustering performance requires prior knowledge [21]. In recent years, several approaches have been proposed to address these limitations. Xu et al. [22] utilized a linear fitting method based on the distribution of parameters to select all potential centers. Chen et al. [23] employed a linear regression model and residuals analysis to automatically determine the cluster centers. Liu et al. [24] introduced the ADPC-KNN algorithm, which selects initial cluster centers and then aggregates density-reachable sub-clusters. Masud et al. [25] presented the I-nice algorithm, inspired by human observation of mountains during field exploration to automatically detect the number of clusters and select their centers. d'Errico et al. [26] proposed that density peaks can be automatically identified using a point adaptive k-nearest neighbor density estimator. Despite the theoretical and practical advantages of the algorithms mentioned above, they introduce new parameters to facilitate obtaining an exact number of clusters. Alternatively, the algorithms may be complex and not scalable. Consequently, the challenge of automatically obtaining the optimal number of clusters and a suitable parameter persists.

To address the challenges faced by the DPC algorithm, we present an innovative automatic density peaks clustering (ADPC) algorithm. First, based on the Silhouette Coefficient [27], a clustering index named density-distance cluster (DDC) is defined. Then, ADPC introduces the DDC index to identify accurate number of clusters and select a suitable parameter automatically. The new features of the ADPC algorithm are (i) a novel DDC index is proposed based on the characteristics of the DPC algorithm and while simultaneously fulfilling the clustering definition. The DDC index is specifically designed to be appropriate for DPC, and (ii) suitable parameter d_c is selected according to the optimal DDC value. Thus, ADPC detects the number of a clusters automatically without any additional parameters.

To summarize, the major contributions of our work are

- The proposed novel clustering validity index leverages both density and distance characteristics of cluster centers. Notably, this index is not only suitable for the DPC algorithm but also aligns harmoniously with the broader definition of clustering.
- An improved automatic density peaks clustering algorithm is proposed based on the DDC index, which automatically selects a suitable cut-off distance and determines the optimal number of clusters without the need for additional parameters.
- The experimental results validate the effectiveness of the ADPC algorithm in automatically

determining the optimal number of clusters and cut-off distance.

The subsequent chapters of this paper are structured as follows: Section 2 presents the fundamental principles of the DPC algorithm. Section 3 introduces the innovative DDC index and details the enhanced ADPC algorithm. In Section 4, experiments are designed to demonstrate the efficiency of the ADPC algorithm. A comparison of the DPC, AP and DBSCAN algorithms is conducted on diverse datasets. The paper concludes with a summary of the key findings and generalization and outlines potential challenges for future research.

2. Related works

This section introduces the major ideas of the Silhouette Coefficient and the DPC algorithm.

2.1. Silhouette coefficient

Clustering algorithms are evaluated based on two significant factors: Within-cluster similarity and between-cluster dissimilarity. As far as we know, the Silhouette Coefficient is a clustering validity index that reflects the compactness and separation of clusters [27,28]. Its value rang is between [-1, 1], and the larger the indicator value, the better the clustering effect of the clustering results.

Assuming that dataset $X = \{x_1, x_2, \cdots, x_n\}$ has been divided into $C = \{C_1, C_2, \cdots, C_k\}$ clusters through clustering algorithms. To calculate the total Silhouette Coefficient of the clustering result, we first calculate the Silhouette Coefficient of each sample point in the dataset separately. First, the average distance between point and other points in the same cluster is calculated as

$$a(i) = \frac{1}{|C_i| - 1} \sum_{x_j \in C_i, i \neq j} d(x_i, x_j), \tag{2.1}$$

where $d(x_i, x_j)$ represents the Euclidean distance between x_i and x_j. $a(i)$ determines the degree to which x_i is assigned to this cluster. Second, calculate the minimum average distance from x_i to other clusters

$$b(i) = \min_{m \neq i} \frac{1}{|C_m|} \sum_{x_m \in C_m} d(x_i, x_m), \tag{2.2}$$

$b(i)$ represents the dissimilarity between clusters. Third, the Silhouette Coefficient of x_i is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{2.3}$$

It can be seen that the closer $s(i)$ approaches 1, the better the compactness and separation of the clustering results. Finally, average the Silhouette Coefficient of all points to obtain the total Silhouette Coefficient of the clustering result

$$SC = \frac{1}{n} \sum_{i=1}^{n} s(i). \tag{2.4}$$

To calculate the Silhouette Coefficient for each sample point, $O(|C_i| - 1)$ and $O(n - |C_i|)$ time complexity are required to obtain $a(i)$ and $b(i)$, respectively. An iteration requires calculating the entire data point to obtain SC, and the distance between two data points will be calculated. Therefore, the total time complexity of the Silhouette Coefficient is $O(n^2)$.

The Silhouette Coefficient validity reflects the compactness of datasets within clusters and the separation between clusters. However, the computational complexity of the Silhouette Coefficient is high. Therefore, we design a new clustering index DDC based on the concept of the Silhouette Coefficient for DPC, which not only meets the density and distance characteristics of DPC, but also meets the definition of clustering.

2.2. DPC algorithm

The DPC algorithm is introduced as an efficient method for identifying cluster centers and creating arbitrary clusters [29]. It is a straightforward approach with substantial potential, leading to significant interest from the research community [30–32]. The algorithm is built upon two fundamental assumptions:

Assumption 1. Cluster centers exhibit a higher density compared to their neighboring data points. Local density ρ_i is calculated for each data as

$$\begin{aligned} \rho_i &= \sum_j \chi\left(d_{ij} - d_{\mathrm{c}}\right), \\ \chi(x) &= \begin{cases} 1, x < 0, \\ 0, x \geq 0, \end{cases} \end{aligned} \tag{2.5}$$

where d_{ij} involves the dissimilarity between objects x_i and x_j, computed using the Euclidean distance. The cut-off distance, d_{c}, is a parameter in the DPC algorithm. It is defined as the 2% percentile value of the similarity matrix, which is obtained by sorting the similarities in ascending order. The cutoff distance serves as a threshold to determine the neighborhood of each data point. In addition, ρ_i can be obtained using a Gaussian kernel function when dealing with a small dataset:

$$\rho_i = \sum_j \exp\left(-\frac{{d_{ij}}^2}{{d_{\mathrm{c}}}^2}\right). \tag{2.6}$$

Assumption 2. The distance between any two cluster centers is relatively large. For each data point, it calculates the algorithm calculates the distance to these cluster centers

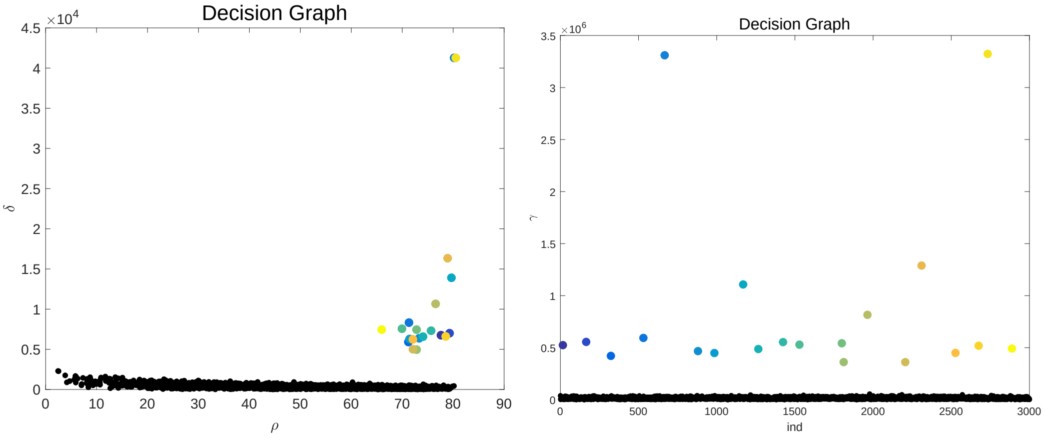
$$\delta_i = \min_{j:\rho_j > \rho_i}\left(d_{ij}\right). \tag{2.7}$$

For the data with the highest density, the following can be observed:

$$\delta_i = \max_j\left(d_{ij}\right). \tag{2.8}$$

Once the local density ρ and the minimum distance to higher-density neighbors δ are calculated for all data points, the DPC algorithm constructs a decision graph. Figure 1(a) visually represents this decision graph, where ρ is plotted on the abscissa (x-axis), and δ is plotted on the ordinate (y-axis). The decision graph provides valuable insights into the distribution of data points and helps in identifying cluster centers. Subsequently, the DPC algorithm selects the cluster centers based on Figure 1(b). The decision graph is generated using parameter γ, which is computed as follows:

$$\gamma_i = \rho_i \cdot \delta_i. \tag{2.9}$$



(a) (b)

Figure 1. Different methods of drawing the decision graph; (a) Decision graph based on ρ and δ; (b) Decision graph based on γ.

Subsequently, in Figure 1(a) or Figure 1(b), data points with higher values of ρ and δ and larger values of γ can be identified as cluster centers. These points are distinguishable as they are more separated from the remaining data points. In the figures, the colored objects represent the identified cluster centers. Finally, the DPC algorithm assigns the non-center points to their nearest neighbors with higher densities [33]. This step ensures that each data point is allocated to the cluster represented by its nearest higher-density neighbor, thus completing the clustering process.

It is worth noting that the DPC algorithm's key advantage lies in its decision graph, which plays a crucial role in identifying cluster centers based on the parameters ρ and δ, or relying solely on γ [34]. However, one limitation of the decision graph is that the cluster centers it identifies may not always be distinctly separate from the remaining data points, making it challenging to precisely define the boundaries of larger clusters [35]. Consequently, correctly determining cluster centers solely based on a decision graph can be a non-trivial task [36]. Additionally, the performance of the DPC algorithm can be influenced by the selection of the parameter d_c. Imperfect choices of d_c may fail to highlight the characteristics of cluster centers, leading to suboptimal clustering results [37]. The appropriate selection of d_c is crucial for achieving accurate and meaningful cluster assignments.

For example, Figure 2(a)–(f) show different decision graphs generated by the DPC algorithm on the Aggregation dataset, which consists of 7 clusters [13]. In Figure 2(a)–(c), the decision graphs are constructed using ρ and δ with $d_c = 1$, $d_c = 2$, and $d_c = 4$, respectively. Figure 2(d)–(f) demonstrate decision graphs drawn using γ with $d_c = 1$, $d_c = 2$ and $d_c = 4$, respectively. Figure 2(a) and (d) illustrate the challenge of accurately identifying the actual number of cluster centers, regardless of whether the decision graph is based on ρ and δ or on γ. The larger values of ρ, δ or γ can lead to ambiguity in determining the cluster centers. As observed in Figure 2(b) and (e), when $d_c = 2$, according to the DPC algorithm, 10 data points exhibit both larger ρ and δ values or have larger γ values. Consequently, misclassification can occur, resulting in the division of the Aggregation dataset into 10 clusters instead of the correct 7 clusters. Figure 2(c) and (f) demonstrate that by setting $d_c = 4$, seven cluster centers are identified, accurately capturing the underlying seven clusters in the Aggregation dataset. This emphasizes the need for an appropriate selection of the parameter d_c in order to obtain satisfactory clustering results with the DPC algorithm. In summary, Figure 2(a)–(f) provide further evidence of the limitations of the DPC algorithm, as it struggles to automatically select the optimal number of cluster centers. Additionally, the selection of d_c plays a critical role in achieving desirable clustering outcomes.

To address the limitations mentioned above, this paper introduces an improved ADPC algorithm based on the DDC index. To obtain DDC values iteratively, the DPC algorithm is executed multiple times using different values of the parameter dc and varying the number of cluster centers. Finally, satisfactory clustering results can be achieved by identifying the optimal DDC index, along with selecting a suitable value for the parameter d_c and determining the optimal number of clusters.

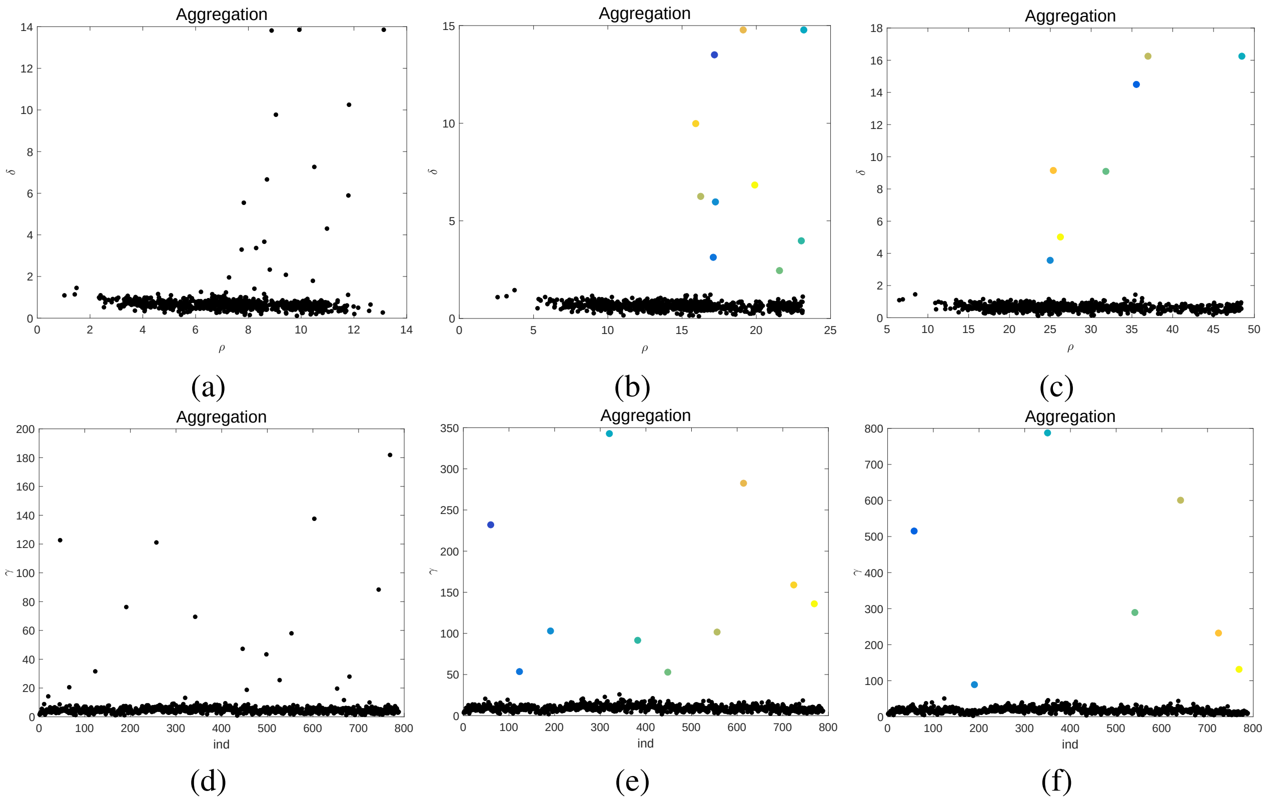


Figure 2. Decision graphs of DPC on Aggregation; (a) Decision graph based on ρ and δ with $d_c = 1$; (b) Decision graph based on ρ and δ with $d_c = 2$; (c) Decision graph based on ρ and δ with $d_c = 4$; (d) Decision graph based on γ with $d_c = 1$; (e) Decision graph based on γ with $d_c = 2$; (f) Decision graph based on γ with $d_c = 4$.

3. ADPC algorithm

3.1. Design of DDC index

Absolutely, assessing within-cluster similarity and between-cluster dissimilarity are essential aspects when evaluating clustering algorithms [38]. In the case of the DPC algorithm, the optimal partitioning of data should maximize within-cluster similarity while minimize between-cluster similarity [39]. Furthermore, DPC is grounded on density and distance assumptions, wherein cluster centers exhibit higher local density and are relatively distant from each other. Consequently, to effectively capture within-cluster compactness and between-cluster separation in datasets while adhering to the characteristics of higher local density and larger distance, a novel clustering validity index, known as the density-distance clustering (DDC) index, is introduced.

Let $X=\{x_1, x_2, \cdots, x_n\}$ represents the set of data samples. Assuming that there are n samples clustered into k clusters, we denote the cluster center in the *ith* cluster as u_i. In the subsequent definitions, $d(x_i, x_j)$ denotes the dissimilarity between x_i and x_j, which is computed using the Euclidean distance.

Definition 1. *We take the average similarity between each data samples of ith cluster and its corresponding cluster center u_i as the within-cluster similarity $a(i)$:*

$$a(i) = \frac{1}{|C_i|}\sum_{j=1}^{|C_i|} d(x_j, u_i), \tag{3.1}$$

where $|C_i|$ represents the scale of the ith cluster, the smaller $a(i)$ is, the higher the local density is.

Definition 2. *The between-cluster similarity, denoted as $b(i)$, is defined as the minimum similarity between the cluster center u_i of the ith cluster and each cluster center u_m of other clusters. It can be calculated as:*

$$b(i) = \min_{1\le m\le k, m\ne i} d(u_i, u_m). \tag{3.2}$$

The larger $b(i)$ is, the larger the relative distance between cluster centers is.

Definition 3. *To optimize the within-cluster similarity while minimizing the between-cluster similarity, we define the $DDC(i)$ of the ith cluster as follows:*

$$DDC(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))}. \tag{3.3}$$

The denominator is set to ensure that the $DDC(i)$ ranges from -1 to 1.

Definition 4. *We define the average $DDC(i)$ of the k clusters obtained as the DDC of this data partition:*

$$\mathrm{DDC}_k = \frac{1}{k}\sum_{i=1}^{k} DDC(i). \tag{3.4}$$

Definition 5. *The number of clusters corresponding to the maximum average DDC value is determined as the optimal number of clusters, denoted as k:*

$$k_{best} = \arg\max_{3\le k<n}\{\mathrm{DDC}_k\}. \tag{3.5}$$

The DDC index is illustrated in Figure 3, which depicts the clustering of all data samples into three clusters: A, B and C. The cluster centers corresponding to these clusters are denoted as a, b and c, respectively. In addition, let's assume that there are eight data points in cluster A. Consequently, we can assess the within-cluster similarity of cluster A by computing the average distance between each data sample and the cluster center a.

$$a(A) = \frac{d(a, a_1) + d(a, a_2) + d(a, a_3) + d(a, a_4) + d(a, a_5) + d(a, a_6) + d(a, a_7) + d(a, a_8)}{8}. \tag{3.6}$$

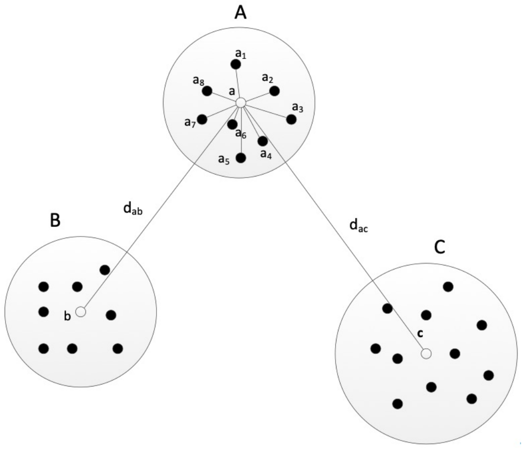


Figure 3. Illustration of DDC index.

The between-cluster similarity of cluster A can be measured by determining the minimum similarity between the cluster center a and the other cluster centers b and c:

$$b(A) = \min(d(a, b), d(a, c)). \tag{3.7}$$

Then, we calculate $DDC(A)$ for cluster A according to (3.3). As for calculating $DDC(A)$, we can obtain $DDC(B)$ and $DDC(C)$. The DDC of this partition is the average of $DDC(A)$, $DDC(B)$ and $DDC(C)$ according to (3.4):

$$\mathrm{DDC}_3 = \frac{DDC(A) + DDC(B) + DDC(C)}{3}. \tag{3.8}$$

The DDC index is designed to effectively optimize the within-cluster similarity while minimizing the between-cluster similarity by incorporating both local density and distance characteristics in DPC. In Figure 3, $a(i)$ represents intra-class point aggregation, indicating how closely the points within a cluster are grouped together. A smaller value of $a(i)$ signifies higher similarity among the points within the cluster, indicating a higher local density around that cluster center. This observation aligns with Assumption 1 of the DPC algorithm. On the other hand, $b(i)$ represents inter-class distinctions, indicating the dissimilarity between objects in different clusters. A higher value of $b(i)$ implies greater distinctiveness between the clusters, indicating that the cluster center is farther away from other centers. This aligns with the distance assumption of the DPC algorithm. In summary, the DDC index not only fulfills the objectives of the clustering algorithm by optimizing similarity and dissimilarity measures but also aligns with the two fundamental assumptions of the DPC algorithm. It can be considered a novel clustering index suitable for the DPC algorithm.

3.2. Main processes of ADPC algorithm

Using the DDC index as a foundation, we propose the Automatic Density Peaks Clustering (ADPC) algorithm to enhance the DPC algorithm. Algorithm 1 summarizes the steps involved in the ADPC algorithm.

Algorithm 1 ADPC algorithm

```
Input: Dataset $X = \{x_1, x_2, \cdots, x_n\}$;
Output: The optimal number of clusters and optimal $d_c$ and the clustering result.
Step 1: For $k = 3$ to $k = \sqrt{n}$;
  a: Select one unvisited value from $d_c = [1, 2, 4]$;
  b: Use (2) to calculate $\rho_i$ for each data sample;
  c: Use (3) and (4) to calculate $\delta_i$ for each data sample;
  d: Use (5) to calculate $\gamma_i$ for each data sample;
  e: Sort $\gamma$ as a decreasing order;
  f: Take $k$ points corresponding to the first $\gamma$ values as the cluster centers;
  g: Class the dataset into $k$ clusters;
  h: Use (8) to calculate the DDC value of a single cluster;
  i: Use (9) to calculate the average DDC value of $k$ clusters as DDC for this clustering partition;
  j: If $d_c$ has been not traversed, then go to a;
Step 2: Use (10) to obtain the optimal number of clusters;
Step 3: Output suitable $d_c$ and clustering results.
```

In the ADPC algorithm, there are three major processes. First, it takes the number of clusters k, as an input parameter. The algorithm allows k to range from 3 to $\sqrt{n}$, where n represents the number of data samples. Through experience and literature [13, 30], DPC performances satisfactory when $d_c = 1$, or $d_c = 2$, or $d_c = 4$ generally. To avoid an increase in algorithm complexity, ADPC set d_c to be the same as in DPC. The algorithm performs DPC iteratively with different combinations of k and d_c, obtaining multiple clustering results. Second, after obtaining each clustering result from the DPC process, the DDC index is calculated for each clustering result. This index quantifies the within-cluster similarity and between-cluster dissimilarity, considering the density and distance characteristics. It allows for an objective evaluation of the quality of each clustering result. Finally, based on the calculated DDC index values, the clustering result with the best DDC value is selected. This indicates the clustering result that achieves satisfactory performance, with an optimal number of clusters and a suitable d_c value. By considering the DDC index, ADPC aims to find the clustering configuration that maximizes within-cluster similarity while minimizing between-cluster similarity.

Through these three processes, the ADPC algorithm iteratively explores different combinations of k and d_c, calculates the DDC index for each clustering result and selects the clustering configuration with the best DDC value. This approach helps in achieving satisfactory clustering performance with an optimal number of clusters and an appropriate d_c value.

3.3. Analysis of ADPC algorithm

The primary principle of the ADPC algorithm is to iteratively search for the best DDC value. The DDC index is specifically designed to be applicable to the DPC algorithm, considering the local density and distance characteristics. The best DDC value represents the optimal performance of the clustering algorithm. This optimal performance corresponds to the ideal number of clusters and parameter values.

Assume that $|C_i|$ is the size of the *ith* cluster, and k stands for the number of clusters. In the ADPC algorithm, the time complexity is determined by both the DDC and DPC processes. The major time-consuming tasks in DPC include constructing the similarity matrix and calculating density and distance. Each operation has a time complexity of $O(n^2)$, where n is the number of data samples. Therefore, the total time complexity of DPC is $O(n^2)$. As for the DDC process, the time complexity is determined by both $a(i)$ and $b(i)$. It takes $O(|C_i|)$ to compute $a(i)$ and $O(k-1)$ to compute $b(i)$. Since k and $|C_i|$ are typically much smaller than $n(|C_i| \ll n, k \ll n)$, the time complexity of DDC is approximately $O(k(|C_i| + k))$, ensuring that the algorithm maintains efficiency.

The ADPC algorithm achieves the optimal number of clusters using the novel DDC index and suitable d_c automatically, surpassing the performance of DPC. Additionally, the ADPC algorithm provides satisfactory clustering results without the need for any parameters, while maintaining the same level of efficiency as DPC.

4. Experiments and results

4.1. Experimental design

The effectiveness of the proposed ADPC algorithm was demonstrated through extensive experimentation on a diverse set of datasets, including eight synthetic datasets, eight real-world datasets and the well-known Olivetti face dataset [40]. Table 1 provides a description of these datasets, including the number of clusters and the scale, which vary from small to large. To compare the performance of ADPC with DPC, we also applied a widely used AP algorithm [41] and the DBSCAN algorithm [42] on the UCI datasets and the Olivetti face dataset. These two algorithms, AP and DBSCAN, do not require prior determination of cluster centers, thus serving as benchmarks to validate the superiority of the proposed ADPC algorithm. Furthermore, we provide a discussion on the DDC index to demonstrate its applicability to the DPC algorithm.

The experiments were conducted on a desktop computer equipped with a 3.10 GHz Intel Core i5 processor, running the MacOS 10.14.6 operating system and equipped with 4 GB of RAM. The experiments were executed using MATLAB 2015 as the programming environment.

4.2. Results and discussion

4.2.1. Experiments on synthetic datasets

Table 1 presents the different characteristics of the eight synthetic datasets used in the experiments. These datasets consist of clusters with various shapes and densities, allowing for a comprehensive evaluation of clustering algorithm performance. To showcase the effectiveness of our ADPC algorithm in achieving satisfactory clustering results with the optimal number of clusters and a suitable parameter, Table 2 presents the number of clusters achieved by different algorithms. Additionally, the value of d_c obtained by ADPC is also included in Table 2.

Given that the datasets are two-dimensional, visualizing the clustering results of the ADPC algorithm, along with the compared DPC, AP and DBSCAN algorithms, using colored plots would offer a more straightforward interpretation. This visualization method allows for a clearer understanding of the performance of each algorithm. Additionally, we have considered the references [13,41,42] to determine the parameters of the DPC, AP and DBSCAN algorithms. Through careful selection, we have chosen the optimal parameters for these algorithms to ensure fair and accurate comparisons in our experiments.

Table 1. Characteristics of different datasets.

Datasets	Samples	Attributes	Clusters
Aggregation	788	2	7
D31	3100	2	31
S	1765	2	5
Twenty	1000	2	20
Square	1000	2	4
S1	5000	2	15
A3	7500	2	50
S3	5000	2	15
Iris	150	4	3
Seeds	210	7	3
Waveform	5000	21	3
Vertebral	310	6	3
Soybean	47	35	4
X8D5K	1000	8	5
Leuk	72	39	3
Wine	178	13	3
Olivetti face	100	92×112	10

Table 2. Number of clusters on synthetic datasets obtained by different algorithms.

Datasets	ADPC/d_c	DPC	AP
Aggregation	7/4	10	17
D31	31/1	31	31
S	5/4	6	27
Twenty	20/4	20	20
Square	4/4	4	20
S1	15/1	15	24
A3	50/1	32	50
S3	15/1	15	53

As shown in Table 2, ADPC effectively determines the optimal number of clusters for the eight datasets. Additionally, ADPC can determine a suitable value for d_c, rather than relying on the default parameter d_c=2 used in DPC. If DPC is applied with d_c=2 as described in reference [13], it fails to produce a reasonable number of clusters for the Aggregation, S and A3 datasets, as indicated by the decision graph. This means that DPC, relying on visual identification of cluster centers based on a decision graph, suffers from a significant limitation. Furthermore, the AP and DBSCAN algorithms only determine the optimal number of clusters for the Twenty dataset. Moreover, these algorithms require parameter adjustments and are sensitive to the chosen parameters [41,42]. As a result, ADPC outperforms DPC, AP and DBSCAN in terms of clustering results, as demonstrated in Figures 4–11.

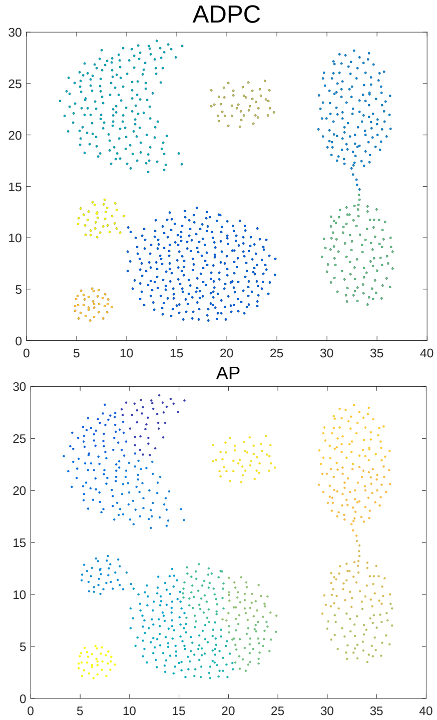


Figure 4. Clustering results on Aggregation.

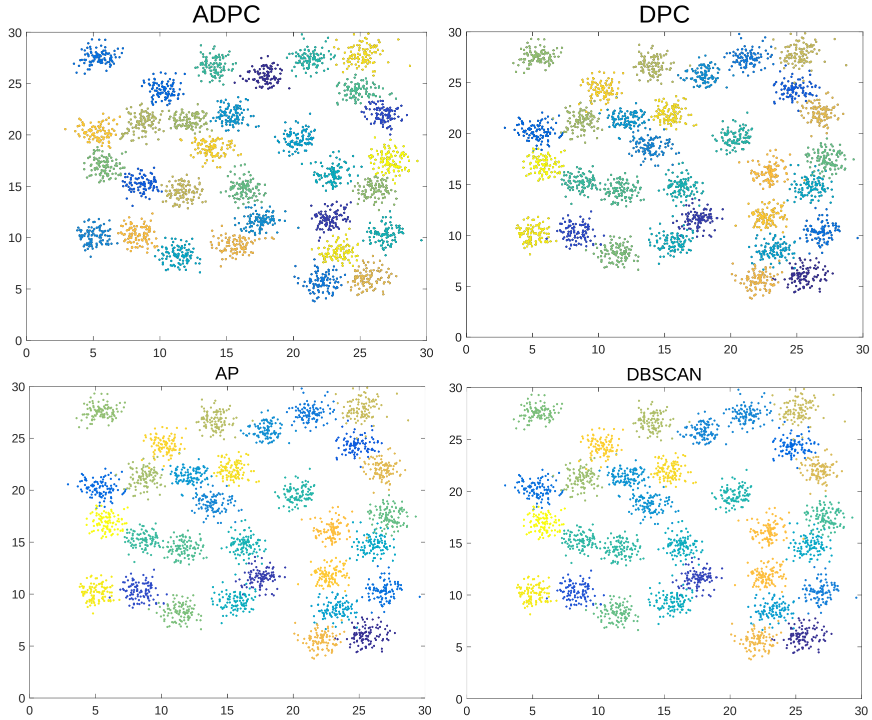


Figure 5. Clustering results on D31.

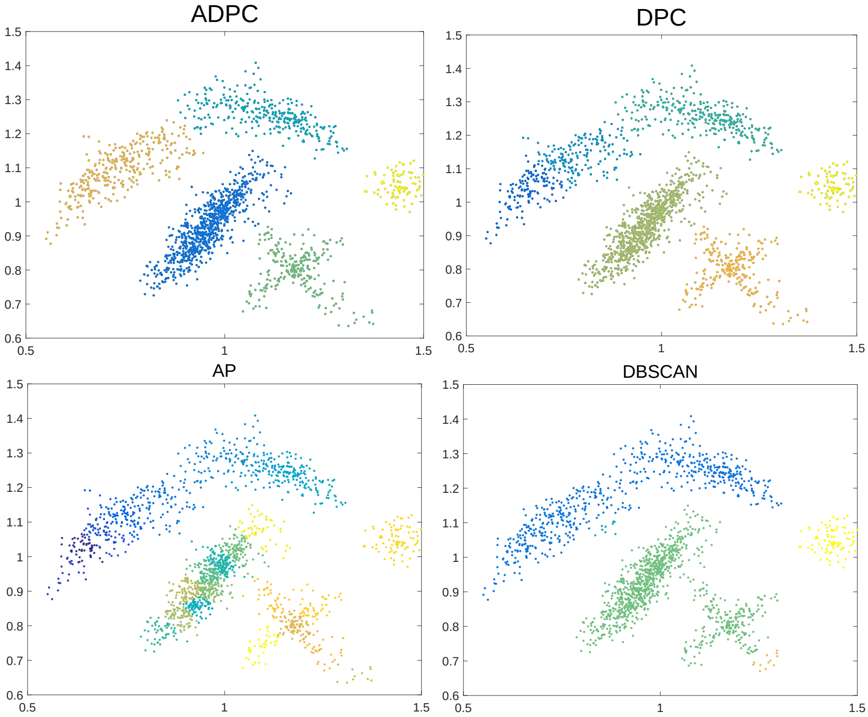


Figure 6. Clustering results on S.

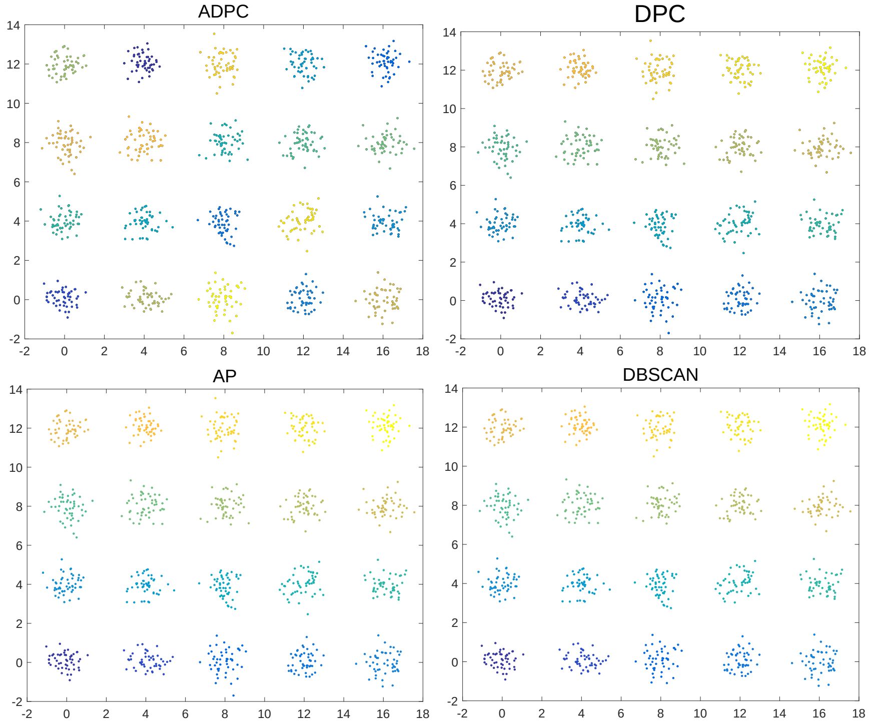


Figure 7. Clustering results on Twenty.

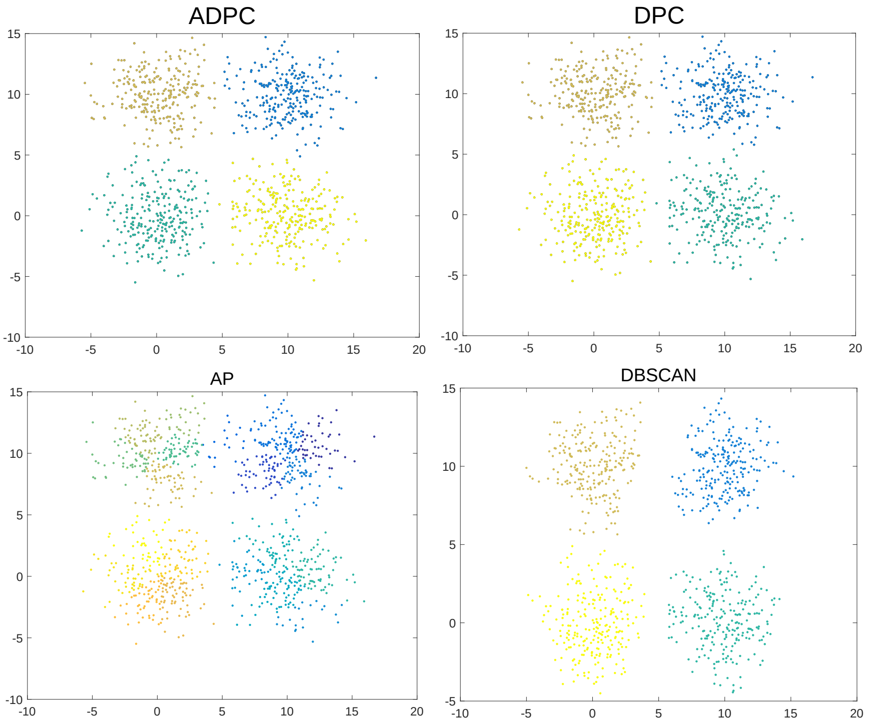


Figure 8. Clustering results on Square.

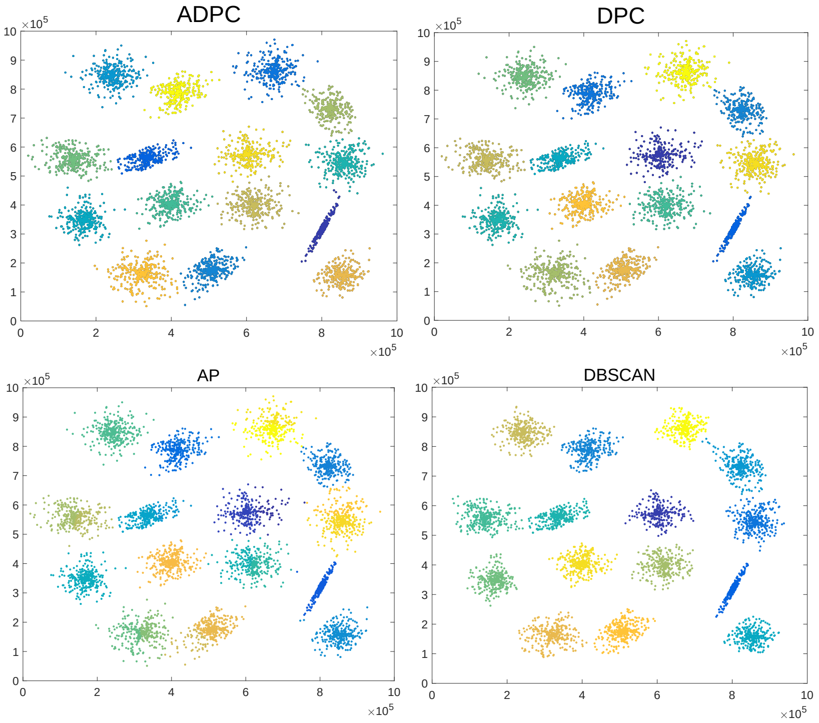


Figure 9. Clustering results on S1.

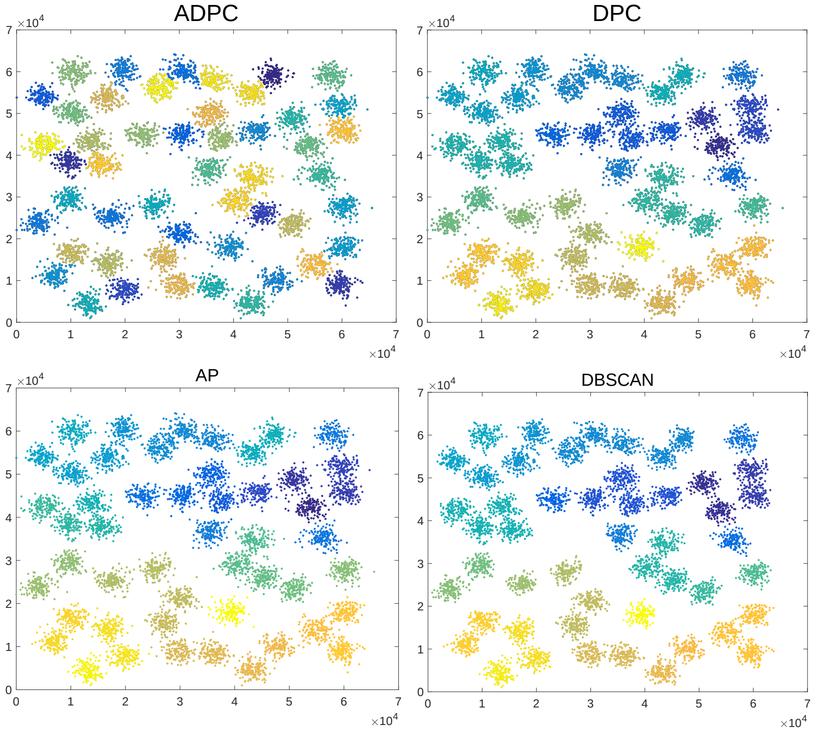


Figure 10. Clustering results on A3.

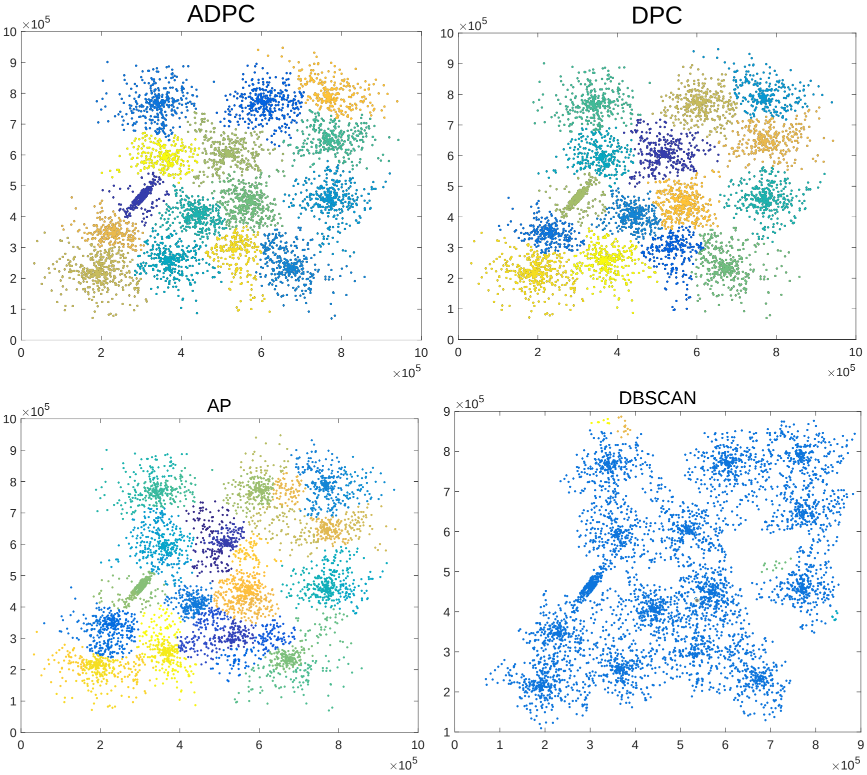


Figure 11. Clustering results on S3.

Figures 4–11 display the clustering performance of the ADPC, DPC, AP and DBSCAN algorithms on each of the synthetic datasets. In these figures, each color represents a distinct cluster. It is evident that the ADPC algorithm consistently achieves satisfactory clustering results across all eight synthetic datasets. On the other hand, the DPC algorithm fails to produce reasonable clustering results on the Aggregation, S and A3 datasets. Similarly, the AP and DBSCAN algorithms only exhibit good clustering performance on the Twenty dataset. These findings further validate the effectiveness of the ADPC algorithm, which utilizes the DDC index, as it consistently outperforms DPC, AP and DBSCAN in terms of clustering accuracy and robustness across diverse synthetic datasets.

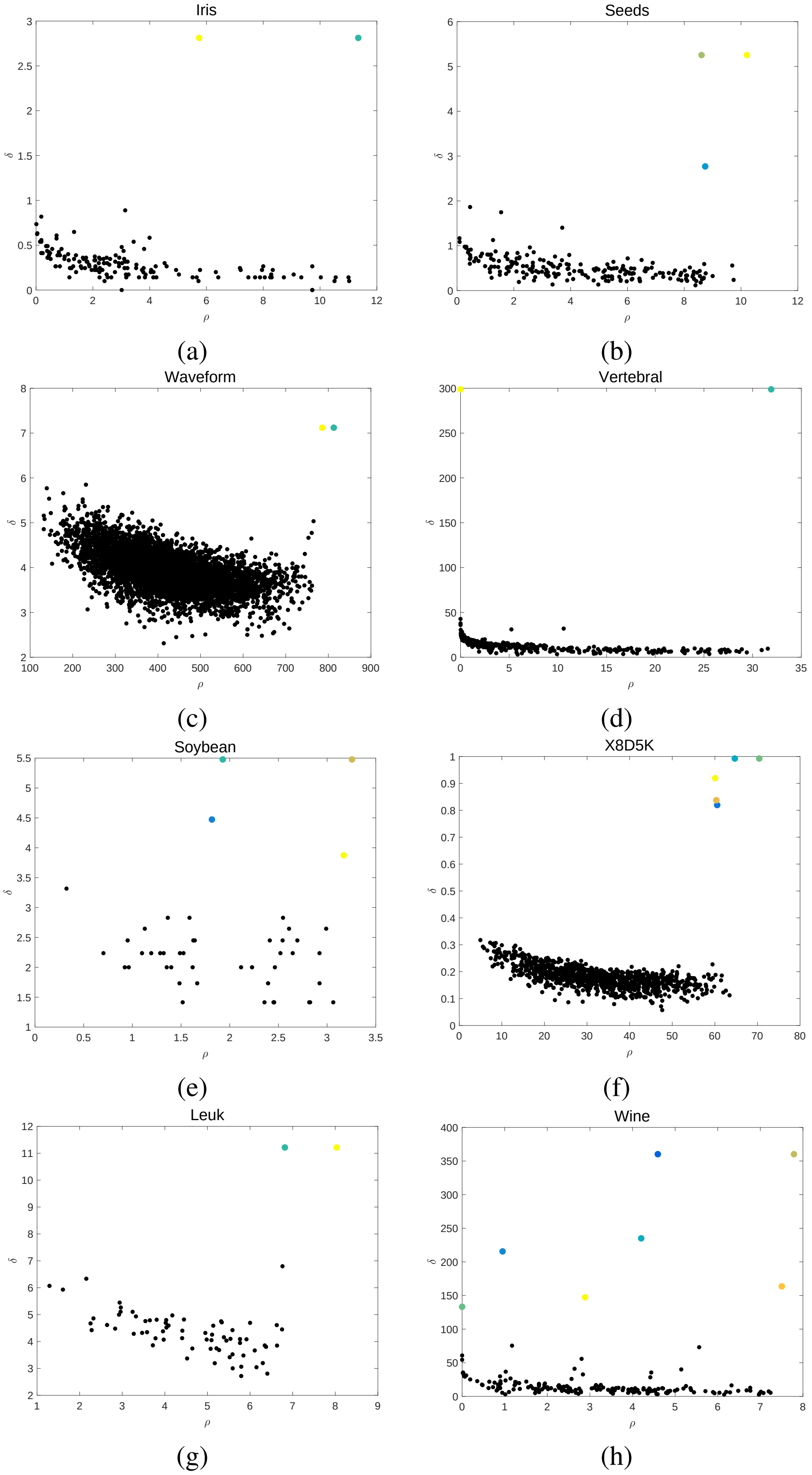
4.2.2. Experiments on UCI datasets

In this subsection, we present the results of applying the introduced ADPC algorithm on eight real-world datasets to demonstrate its superiority. Table 3 provides a comparison of the number of clusters obtained by the ADPC, DPC, AP and DBSCAN algorithms on these datasets ('-' means the algorithm either identifies only one cluster or fails to find any clusters).

Table 3. The number of clusters on the UCI datasets obtained through different algorithms.

Datasets	ADPC	DPC	AP	DBSCAN
Iris	3	2	6	2
Seeds	3	3	11	-
Waveform	3	2	139	-
Vertebral	3	2	21	-
Soybean	4	4	5	-
X8D5K	5	5	14	2
Leuk	3	2	6	3
Wine	3	7	8	-

Table 3 shows that the AP and DBSCAN algorithms are unable to determine the optimal number of clusters for the eight UCI datasets. Additionally, these algorithms require parameter adjustments and exhibit sensitivity to the chosen parameters. On the contrary, DPC fails to obtain a reasonable number of clusters among the UCI datasets, with the exception of Soybean, X8D5K and Seeds datasets. To gain deeper insights into the challenges faced by DPC in accurately identifying the number of clusters on these datasets, refer to Figure 12, which showcases the decision graphs produced by DPC for the same datasets. The decision graphs vividly demonstrate the difficulties and limitations encountered by DPC in cluster determination for these specific datasets.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 12. Decision graphs of DPC on UCI datasets.

We can see from Figure 12(b), (e), (g) and (h) that the cluster centers in these decision graphs are not clearly separated from the surrounding points. This lack of clear separation makes it difficult to accurately identify the correct number of cluster centers. Similarly, in Figure 12(a), (c) and (d), the correct number of cluster centers may be challenging to determine due to the use of d_c=2 as described in reference [13]. The reliance of DPC on human-based selection represents a significant limitation. In contrast, the ADPC algorithm can automatically determine the optimal number of clusters on these eight UCI datasets and establish correct clusters without the need for any parameters. Consequently, the ADPC algorithm outperforms DPC, AP and DBSCAN in terms of clustering results, offering a more effective and reliable approach.

To provide further evidence of the superiority of the introduced ADPC algorithm, we present a detailed comparison with the DPC algorithm in Table 4. The table presents the clustering results of ADPC and DPC on all eight UCI datasets, measured in terms of Accuracy (Acc) [43], Adjusted Mutual Information (AMI) [44] and Adjusted Rand Index (ARI) [45].

Table 4. Clustering results of ADPC and DPC on UCI datasets.

	ADPC					DPC				
	ACC	AMI	ARI	d_c	NC	ACC	AMI	ARI	d_c	NC
Iris	0.9067	0.7960	0.7592	2	3	0.6667	1	0.5681	2	2
Seeds	0.8952	0.6741	0.7170	1	3	0.8857	0.6926	0.7027	2	3
Waveform	0.5794	0.3482	0.2962	1	3	0.582	0.4978	0.2422	2	2
Vertebral	0.6387	0.2618	0.2850	4	3	0.4806	0.0054	-0.0022	2	2
Soybean	0.8936	0.8061	0.7251	1	4	0.8936	0.8061	0.7251	2	4
X8D5K	1	1	1	1	5	1	1	1	2	5
Leuk	0.9583	0.8573	0.8809	1	3	0.7083	0.0781	0.5413	2	2
Wine	0.7921	0.5534	0.5054	1	3	0.5393	0.3168	0.2802	2	7

Table 4 indicates that ADPC generally outperforms DPC on most datasets, demonstrating superior results. ADPC has the ability to automatically adjust the value of d_c to achieve optimal clustering outcomes. The only exception is observed in the Waveform datasets, where DPC performs slightly better than ADPC. This can be attributed to the fact that Waveform comprises three clusters, with each cluster occupying approximately 33% of the data. Additionally, ADPC yields consistent results with DPC on the Soybean and X8D5K datasets. This consistency arises from the fact that DPC produces the same outcomes regardless of whether d_c=1 or d_c=2. These findings further validate that our ADPC algorithm, based on the DDC index, can effectively determine the optimal number of clusters and adjust without the need for additional parameters.

4.2.3. Experiments on Olivetti Face datasets

To further evaluate the performance of ADPC, the ADPC algorithm is tested on the famous Olivetti Face Database. The database consists of 40 subjects, each having 10 different images. For this evaluation, we utilized the first 100 images, dividing them into 10 clusters. To measure the similarity between each pair of images, we utilized the Structural Similarity Index (SSIM) [46], defined by formula (4.1)

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{({\mu_x}^2 + {\mu_y}^2 + c_1)({\sigma_x}^2 + {\sigma_y}^2 + c_2)}, \tag{4.1}$$

where c_1 and c_2 are the constants taken to maintain the stability. μ_x and μ_y represent the average of x and y, σ_x and σ_y respectively represent the variances of x and y, σ_{xy} represents the covariance of x and y. Images were allocated to a cluster only if their distance is less than d_c. Figure 13 depicts the cluster allocation results obtained through the ADPC algorithm on the Olivetti Face Dataset. On the other hand, Figure 14 shows the clustering outcomes achieved through the DPC algorithm on the same dataset. In both figures, images belonging to the same cluster are represented with the same color. However, images displayed in gray indicate incorrect classifications.

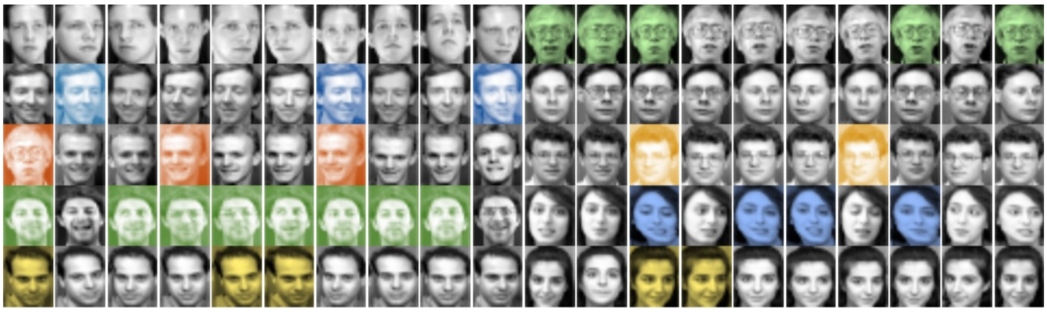


Figure 13. ADPC clustering on Olivetti Face Database (the first 100 images).

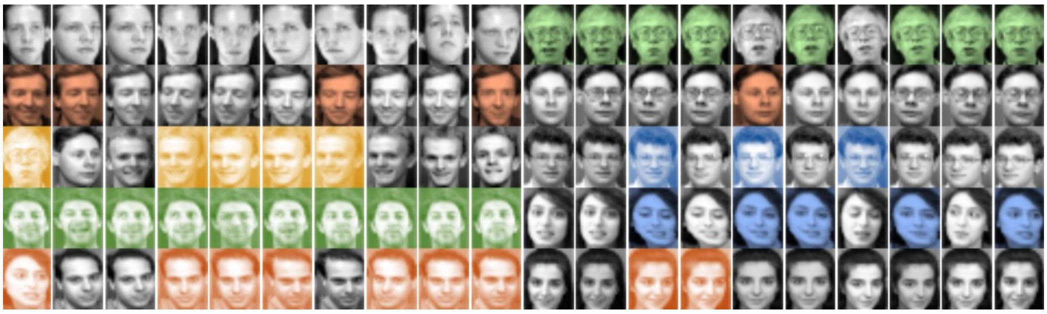
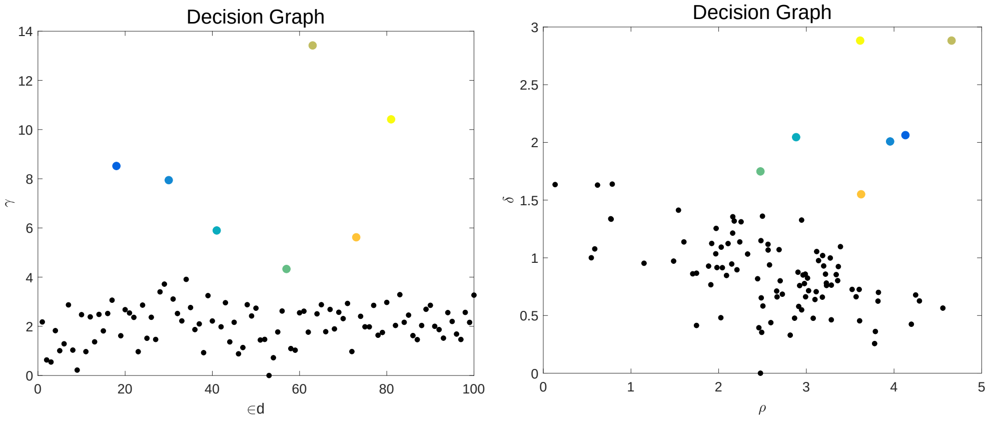


Figure 14. DPC clustering on Olivetti Face Database (the first 100 images).

In accordance with reference [13], the optimal number of clusters for DPC on the Olivetti Face dataset is determined to be 7. Remarkably, ADPC has also successfully identified 7 subjects out of 10, aligning with the results achieved by the DPC algorithm. However, upon examining Figure 15, we observe that when relying on ρ and δ or $\gamma = \rho \cdot \delta$ to draw decision graphs, the DPC algorithm fails to accurately select cluster centers. In contrast, although ADPC's accuracy may not be as high as that of DPC, it compensates for the limitations of DPC by automatically identifying clusters. Consequently, the ADPC algorithm improves upon the challenge faced by DPC in establishing clusters automatically. Through this comparison, it becomes evident that ADPC offers a valuable enhancement to DPC, enabling the automatic identification of clusters and addressing the shortcomings of DPC in this particular context.



(a) (b)

Figure 15. DPC clustering on Olivetti Face Database (the first 100 images); (a) Decision graph based on ρ and δ; (b) Decision graph based on γ.

4.2.4. Discussion of the DDC index

In this section, we discuss the applicability of the DDC index to DPC. First, in order to verify the advantages of using the density and distance characteristics of the cluster centers, we plotted the clustering accuracy of DPC with other similar assumptions on the two-dimensional datasets in Table 1 based on DDC and Silhouette Coefficient, as shown in Figures 16 and 17. Using visualizations will provide a more direct performance comparison.

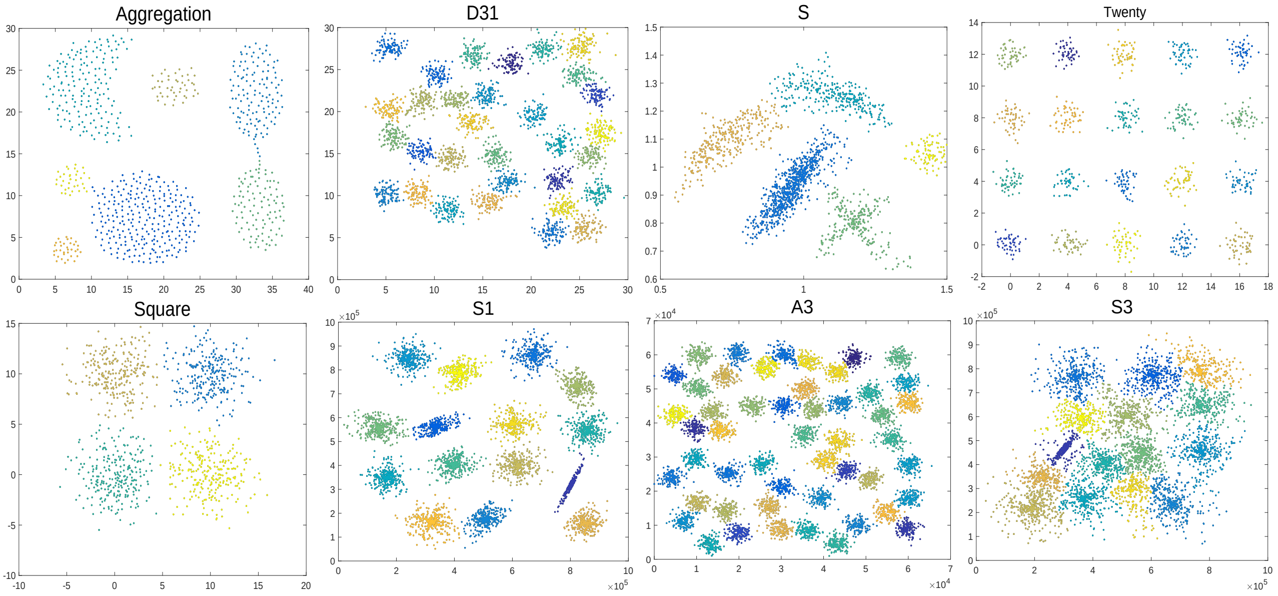


Figure 16. DPC clustering results based on the DDC index.

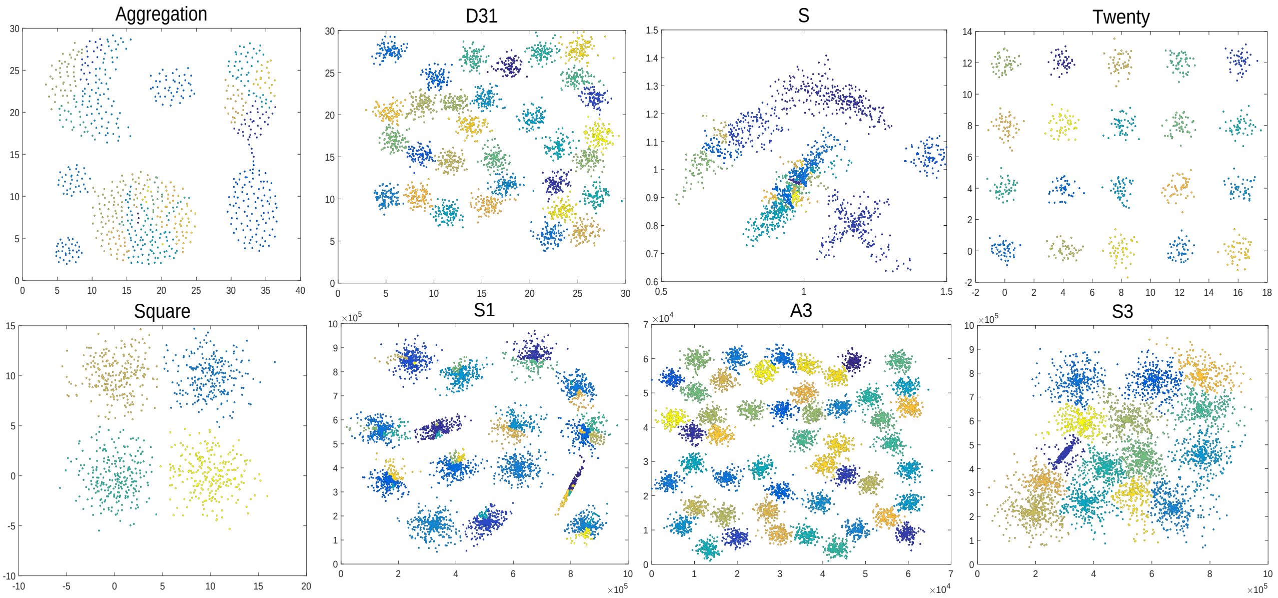


Figure 17. DPC clustering results based on the Silhouette Coefficient.

From Figures 16 and Figure 17, it is evident that the DDC index is more suitable for the DPC algorithm than the Silhouette Coefficient and achieves better clustering accuracy. This is because the DDC index not only considers within-cluster similarity and between-cluster dissimilarity, but also effectively utilizes the density and distance characteristics of cluster centers, which is more in line with the cluster center hypothesis in DPC. The Silhouette Coefficient cannot guide the DPC algorithm in finding the optimal number of clusters on Aggregation, S and S1. To further demonstrate the effectiveness of the DDC index, we present a comparison of the clustering time between them in Figure 18. Both algorithms are run 10 times to obtain the average results.

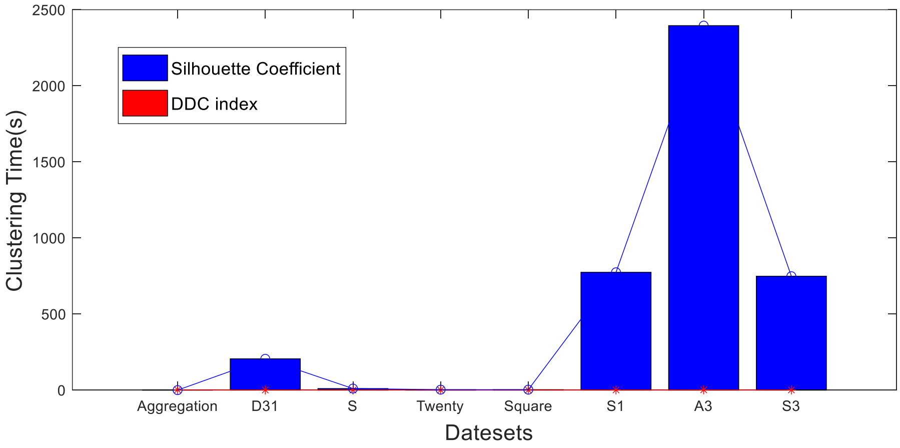


Figure 18. Clustering time comparison.

We can see from Figure 18 that the execution efficiency of DDC is higher because it does not need to calculate the similarity between all data points, and its complexity $O(k(|C_i| + k))$ is much smaller than the complexity $O(n^2)$ of the Silhouette Coefficient. Therefore, the DDC index is more suitable for DPC regarding clustering accuracy and efficiency, and its design is reasonable.

5. Conclusions

We introduce a novel clustering validity index called the DDC index, specifically designed for the DPC algorithm. Building upon the concept of the DDC index, we propose a new algorithm called ADPC. The ADPC algorithm aims to achieve desirable clustering results by determining the optimal number of clusters and identifying a suitable parameter. ADPC begins with a similar approach as DPC, utilizing DPC to calculate the DDC values iteratively and considering different numbers of clusters and parameters. The DDC value serves as an indicator of the quality of the clustering results, with larger DDC values indicating better clustering outcomes.

ADPC successfully addresses two significant limitations of DPC. First, it resolves the issue of inaccurate visual identification of cluster centers on the decision graph. Second, it tackles the problem of selecting an unsuitable parameter d_{c}, which can negatively impact the clustering results. The DDC index, which constitutes the foundation of ADPC, consists of two essential evaluation factors for clustering algorithms: the within-cluster parameter and the between-cluster parameter. By incorporating these factors, the DDC index not only aligns with the assumptions of DPC but also remains consistent with the objectives of clustering algorithms. Experimental evaluations conducted on both synthetic and real-world datasets demonstrate that ADPC outperforms conventional DPC. ADPC not only automatically selects the optimal number of clusters but also determines a suitable value for the parameter d_{c}. Notably, ADPC achieves this without the need for additional parameters.

Indeed, while the ADPC and DPC algorithms have shown promising results in clustering tasks, they may still encounter challenges when applied to manifold datasets, such as S4-D and S5-B in the paper describing the original DP algorithm. These challenges arise due to the inherent complexities and intricacies present in such datasets. In order to improve the clustering performance on complex datasets, further exploration and research are necessary.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the National Natural Science Foundations of China (No. 62206296) and the Key Research and Development Projects in Xuzhou (No. KC22142).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. H. Kim, Geospatial data-driven assessment of earthquake-induced liquefaction impact mapping using classifier and cluster ensemble, *Appl. Soft Comput.*, **140** (2023), 110266. https://doi.org/10.1016/j.asoc.2023.110266

2. E. Ivannikova, H. Park, T. Hämäläinen, K. Lee, Revealing community structures by ensemble clustering using group diffusion, *Inform. Fusion*, **42** (2018), 24–36. https://doi.org/10.1016/j.inffus.2017.09.013
3. X. Zeng, A. Chen, M. Zhou, Color perception algorithm of medical images using density peak based hierarchical clustering, *Biomed. Signal Proces.*, **48** (2019), 69–79. https://doi.org/10.1016/j.bspc.2018.09.013
4. Y. Slimen, S. Allio, J. Jacques, Model-based co-clustering for functional data, *Neurocomputing*, **291** (2018), 97–108. https://doi.org/10.1016/j.neucom.2018.02.055
5. Q. Zhang, C. Zhu, L. Yang, Z. Chen, L. Zhao, P. Li, An incremental cfs algorithm for clustering large data in industrial internet of things, *IEEE T. Ind. Inform.*, **13** (2017), 1193–1201. https://doi.org/10.1109/TII.2017.2684807
6. A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Zomaya, et al., A survey of clustering algorithms for big data: taxonomy and empirical analysis, *IEEE T. Emerg. Top. Com.*, **2** (2014), 267–279. https://doi.org/10.1109/TETC.2014.2330519
7. D. Wang, T. Li, P. Deng, F. Zhang, W. Huang, P. Zhang, et al., A generalized deep learning clustering algorithm based on non-negative matrix factorization, *ACM T. Knowl. Discov. D.*, **17** (2023), 99. https://doi.org/10.1145/3584862
8. M. Shahzad, S. Riazul Islam, M. Hossain, M. Abdullah-Al-Wadud, A. Alamri, M. Hussain, Gafor: genetic algorithm based fuzzy optimized re-clustering in wireless sensor networks, *Mathematics*, **9** (2021), 43. https://doi.org/10.3390/math9010043
9. W. Zhao, C. Deng, C. Ngo, k-means: a revisit, *Neurocomputing*, **291** (2018), 195–206. https://doi.org/10.1016/j.neucom.2018.02.072
10. Y. Zhu, K. Ting, M. Carman, Density-ratio based clustering for discovering clusters with varying densities, *Pattern Recogn.*, **60** (2016), 983–997. https://doi.org/10.1016/j.patcog.2016.07.007
11. Chaomurilige, How klfcm works—convergence and parameter analysis for klfcm clustering algorithm, *Mathematics*, **11** (2023), 2285. https://doi.org/10.3390/math11102285
12. H. Ling, J. Wu, Y. Zhou, W. Zheng, How many clusters? A robust pso-based local density mode, *Neurocomputing*, **207** (2016), 264–275. https://doi.org/10.1016/j.neucom.2016.03.071
13. A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, *Science*, **344** (2014), 1492–1496. https://doi.org/10.1126/science.1242072
14. R. Liu, H. Wang, X. Yu, Shared-nearest-neighbor-based clustering by fast search and find of density peaks, *Inform. Sciences*, **450** (2018), 200–226. https://doi.org/10.1016/j.ins.2018.03.031
15. X. Xu, S. Ding, Y. Wang, L. Wang, W. Jia, A fast density peaks clustering algorithm with sparse search, *Inform. Sciences*, **554** (2021), 61–83. https://doi.org/10.1016/j.ins.2020.11.050
16. J. Xu, G. Wang, T. Li, W. Deng, G. Gou, Fat node leading tree for data stream clustering with density peaks, *Knowl.-Based Syst.*, **120** (2017), 99–117. https://doi.org/10.1016/j.knosys.2016.12.025
17. S. Ding, M. Du, T. Sun, X. Xu, Y. Xue, An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood, *Knowl.-Based Syst.*, **133** (2017), 294–313. https://doi.org/10.1016/j.knosys.2017.07.027

18. M. Karaayvaz, S. Cristea, S. Gillespie, A. Patel, R. Mylvaganam, C. Luo, et al., Unravelling subclonal heterogeneity and aggressive disease states in tnbc through single-cell rna-seq, *Nat. Commun.*, **9** (2018), 3588. https://doi.org/10.1038/s41467-018-06052-0

19. X. Li, K. Wong, Evolutionary multiobjective clustering and its applications to patient stratification, *IEEE T. Cybernetics*, **49** (2019), 1680–1693. https://doi.org/10.1109/TCYB.2018.2817480

20. T. Xu, J. Jiang, A graph adaptive density peaks clustering algorithm for automatic centroid selection and effective aggregation, *Expert Syst. Appl.*, **195** (2022), 116539. https://doi.org/10.1016/j.eswa.2022.116539

21. L. Bai, X. Cheng, J. Liang, H. Shen, Y. Guo, Fast density clustering strategies based on the k-means algorithm, *Pattern Recogn.*, **71** (2017), 375–386. https://doi.org/10.1016/j.patcog.2017.06.023

22. J. Xu, G. Wang, W. Deng, Denpehc: density peak based efficient hierarchical clustering, *Inform. Sciences*, **373** (2016), 200–218. https://doi.org/10.1016/j.ins.2016.08.086

23. J. Chen, H. He, A fast density-based data stream clustering algorithm with cluster centers self-determined for mixed data, *Inform. Sciences*, **345** (2016), 271–293. https://doi.org/10.1016/j.ins.2016.01.071

24. Y. Liu, Z. Ma, F. Yu, Adaptive density peak clustering based on k-nearest neighbors with aggregating strategy, *Knowl.-Based Syst.*, **133** (2017), 208–220. https://doi.org/10.1016/j.knosys.2017.07.010

25. M. Masud, J. Huang, C. Wei, J. Wang, I. Khan, M. Zhong, I-nice: a new approach for identifying the number of clusters and initial cluster centres, *Inform. Sciences*, **466** (2018), 129–151. https://doi.org/10.1016/j.ins.2018.07.034

26. M. D'Errico, E. Facco, A. Laio, A Rodriguez, Automatic topography of high-dimensional data sets by non-parametric density peak clustering, *Inform. Sciences*, **560** (2021), 476–492. https://doi.org/10.1016/j.ins.2021.01.010

27. P. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *J. Comput. Appl. Math.*, **20** (1987), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

28. L. Lovmar, A. Ahlford, M. Jonsson, A. Syvanen, Silhouette scores for assessment of SNP genotype clusters, *BMC Genomics*, **6** (2005), 35. https://doi.org/10.1186/1471-2164-6-35

29. X. Xu, S. Ding, Z. Shi, An improved density peaks clustering algorithm with fast finding cluster centers, *Knowl.-Based Syst.*, **158** (2018), 65–74. https://doi.org/10.1016/j.knosys.2018.05.034

30. J. Xie, H. Gao, W. Xie, X. Liu, P. Grant, Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors, *Inform. Sciences*, **354** (2016), 19–40. https://doi.org/10.1016/j.ins.2016.03.011

31. M. Du, S. Ding, H. Jia, Study on density peaks clustering based on k-nearest neighbors and principal component analysis, *Knowl.-Based Syst.*, **99** (2016), 135–145. https://doi.org/10.1016/j.knosys.2016.02.001

32. S. Ding, C. Li, X. Xu, L. Ding, J. Zhang, L. Guo, et al., A sampling-based density peaks clustering algorithm for large-scale data, *Pattern Recogn.*, **136** (2023), 109238. https://doi.org/10.1016/j.patcog.2022.109238

33. Z. Liang, P. Chen, Delta-density based clustering with a divide-and-conquer strategy: 3dc clustering, *Pattern Recogn. Lett.*, **73** (2016), 52–59. https://doi.org/10.1016/j.patrec.2016.01.009

34. M. Chen, L. Li, B. Wang, J. Cheng, L. Pan, X. Chen, Effectively clustering by finding density backbone based-on knn, *Pattern Recogn.*, **60** (2016), 486–498. https://doi.org/10.1016/j.patcog.2016.04.018

35. M. Wang, F. Min, Z. Zhang, Y. Wu, Active learning through density clustering, *Expert Syst. Appl.*, **85** (2017), 305–317. https://doi.org/10.1016/j.eswa.2017.05.046

36. B. Wu, B. Wilamowski, A fast density and grid based clustering method for data with arbitrary shapes and noise, *IEEE T. Ind. Inform.*, **13** (2017), 1620–1628. https://doi.org/10.1109/TII.2016.2628747

37. Z. Li, Y. Tang, Comparative density peaks clustering, *Expert Syst. Appl.*, **95** (2018), 236–247. https://doi.org/10.1016/j.eswa.2017.11.020

38. K. Ting, Y. Zhu, M. Carman, Y. Zhu, Z. Zhou, Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure, *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, 1205–1214. https://doi.org/10.1145/2939672.2939779

39. S. Ding, W. Du, X. Xu, T. Shi, Y. Wang, C. Li, An improved density peaks clustering algorithm based on natural neighbor with a merging strategy, *Inform. Sciences*, **624** (2023), 252–276. https://doi.org/10.1016/j.ins.2022.12.078

40. F. Samaria, A. Harter, Parameterisation of a stochastic model for human face identification, *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, 1994, 138–142. https://doi.org/10.1109/ACV.1994.341300

41. B. Frey, D. Dueck, Clustering by passing messages between data points, *Science*, **315** (2007), 972–976. https://doi.org/10.1126/science.1136800

42. D. Ienco, G. Bordogna, Fuzzy extensions of the DBScan clustering algorithm, *Soft Comput.*, **22** (2018), 1719–1730. https://doi.org/10.1007/s00500-016-2435-0

43. J. Jiang, X. Yan, Z. Yu, J. Guo, W. Tian, A Chinese expert disambiguation method based on semi-supervised graph clustering, *Int. J. Mach. Learn. Cyber.*, **6** (2015), 197–204. https://doi.org/10.1007/s13042-014-0255-z

44. H. Jia, S. Ding, M. Du, Y. Xue, Approximate normalized cuts without eigen-decomposition, *Inform. Sciences*, **374** (2016), 135–150. https://doi.org/10.1016/j.ins.2016.09.032

45. N. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: is a correction for chance necessary? *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, 1073–1080. https://doi.org//10.1145/1553374.1553511

46. M. Sampat, Z. Wang, S. Gupta, A. Bovik, M. Markey, Complex wavelet structural similarity: a new image similarity index, *IEEE T. Image Process.*, **18** (2009), 2385–2401. https://doi.org/10.1109/TIP.2009.2025923