EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Large data density peak clustering based on sparse auto-encoder and data space meshing via evidence probability distribution

Fang Lu[1,*]

[1]Harbin Finance University, Harbin 150000 China

Abstract

The development of big data analysis technology has brought new development opportunities to the production and management of various industries. Through the mining and analysis of various data in the operation process of enterprises by big data technology, the internal associated data of the enterprises and even the entire industry can be obtained. As a common method for large-scale data statistical analysis, clustering technology can effectively mine the relationship within massive heterogeneous multidimensional data, complete unlabeled data classification, and provide data support for various model analysis of big data. Common big data density clustering methods are time-consuming and easy to cause errors in data density allocation, which affects the accuracy of data clustering. Therefore we propose a novel large data density peak clustering based on sparse auto-encoder and data space meshing via evidence probability distribution. Firstly, the sparse auto-encoder in deep learning is used to achieve feature extraction and dimensionality reduction for input high-dimensional data matrix through training. Secondly, the data space is meshed to reduce the calculation of the distance between the sample data points. When calculating the local density, not only the density value of the grid itself, but also the density value of the *k* nearest neighbors are considered, which reduces the influence of the subjective selection truncation distance on the clustering results and improves the clustering accuracy. The grid density threshold is set to ensure the stability of the clustering results. Using the K-nearest neighbor information of the sample points, the transfer probability distribution strategy and evidence probability distribution strategy are proposed to optimize the distribution of the remaining sample points, so as to avoid the joint error of distribution. The experimental results show that the proposed algorithm has higher clustering accuracy and better clustering performance than other advanced clustering algorithms on artificial and real data sets.

Keywords: data density clustering, sparse auto-encoder, data space meshing, evidence probability distribution, transfer probability distribution strategy.

Received on 25 07 2024, accepted on 16 11 2024, published on 20 11 2024

Copyright © 2024 F. Lu, licensed to EAI. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetsis.6758

1. Introduction

In recent years, the world's stored information has grown at an annual rate of nearly 24% [1], and the explosive growth of data volume has accelerated the arrival of the era of big data, which has brought new opportunities and challenges to all walks of life. How to efficiently and automatically analyze and mine big data has become a major issue facing all industries.

As a data mining method that can explore the potential information inside the data, clustering has been widely used in image processing [2], economic analysis [3], biomedics [4], pattern recognition [5] and community detection [6]. According to different data processing

*Corresponding author. Email: lufang202407@163.com

approaches, clustering algorithms include many types, such as, partition-based k-means algorithm [7], grid-based STING (statistical information grid) algorithm [8], hierarchical BRICH (balanced iterative reducing and clustering hierarchies) algorithm [9] and density-based DBSCAN algorithm [10], etc. The k-means algorithm minimizes the sum of squares of the distance between samples and cluster centers by optimizing the objective function. This algorithm is easy to implement and fast, and has a good effect on spherical clusters, but it has a poor effect on non-spherical clusters. STING algorithm divides samples into different grids and uses grid relations for clustering, effectively reducing the time complexity. This algorithm is greatly affected by the underlying grid and has low accuracy for variable density data clustering. BRICH algorithm represents each cluster level based on clustering features and uses bottom-up strategy to combine samples to complete clustering. This algorithm has a fast clustering speed, sensitive to parameters and poor clustering effect on non-convex data sets.

The clustering results of the above-mentioned clustering algorithms are often unsatisfactory when facing the clustering problems of arbitrary shape and variable density data. As a classical density clustering algorithm, DBSCAN can identify arbitrarily shaped clusters based on the tightness of sample distribution. This algorithm needs to input more parameters, and the clustering algorithm is sensitive to parameters in the face of arbitrary shapes and variable density data clustering. Reference [11] proposed a new density clustering algorithm, density peak clustering (DPC). The DPC algorithm was based on two assumptions: first, the local density of the density peak was significantly greater than that of the surrounding sample, and second, the distance between any two density peaks was far. The algorithm combined the relation of sample density and distance to cluster. Its principle was simple, the clustering process did not need iteration, the input parameters were few, and arbitrary shape clusters could be identified. DPC algorithm also has some shortcomings, such as: using global truncation distance to define density. This method only considers the global distribution of samples, ignoring the local distribution, and cannot accurately describe the density of the samples with more concentrated distribution in the low-density cluster of the variable density data set. It is easy to generate multiple density peaks in the high-density cluster, which leads to poor effect of the algorithm on the clustering of variable density data. The distribution strategy is prone to the "domino" phenomenon, that is, the incorrect allocation of high-density samples will lead to the subsequent allocation errors of low-density samples, leading to the expansion of allocation errors.

Recently, many scholars have proposed different improvement strategies for the shortcomings of DPC algorithm. In terms of local density improvement, reference [12] proposed the density peaks clustering algorithm based on improved similarity and allocation strategy (DPCV). In this algorithm, sample variance was introduced into the density definition to reduce the density difference between clusters of variable density data sets. Reference [13] proposed a robust clustering algorithm based on core point identification and K-nearest neighbor kernel density estimation, which used K-nearest neighbors to calculate sample density and form delegations for clustering. In reference [14], an adaptive nearest neighbor DPC algorithm was proposed, which introduced sample adaptive neighbors to accurately define sample density and obtain density peaks of low-density clusters. Reference [15] proposed density peaks clustering based on weighted local density sequence and nearest neighbor assignment (DPCSA). The new local density was defined by considering the contribution of K-nearest samples to the density. Reference [16] introduced the fuzzy domain and the relative relation between samples and their neighbors into density calculation, and these algorithms had improved the density calculation method to a certain extent. In terms of sample allocation strategy improvement, reference [17] proposed a fast hierarchical clustering of local density peaks via an association degree based on the association degree transfer method (FHC-LDP). The algorithm divided the data set into different sub-clusters according to the density peak, and uses hierarchical clustering to merge the sub-clusters. Reference [18] proposed a density peak clustering with connectivity estimation (DPC-CE) algorithm, which used the connectivity of the maps to improve the calculation strategy of relative distances in sample allocation. Reference [19] proposed an adaptive two-stage density clustering algorithm with fuzzy connectivity. Combining the advantages of DPC and DBSCAN algorithms, the algorithm also considered distance and fuzzy connectivity between samples when determining density peak and membership degree of sample allocation, thus improving the clustering effect. Reference [20] proposed a shared neighbor DPC algorithm for manifold oriented data, which defined the similarity between samples by the neighbor relationship between samples and allocated samples based on this similarity. References [21,22] improved the distribution accuracy of DPC algorithm by introducing minimum spanning tree and minimum spanning forest strategies.

Because the local density value in the density peak clustering algorithm directly affects the clustering result, and the local density is related to the truncation distance of the subjective selection, some scholars have improved the defect. Maximo et al. [23] made use of the idea of adaptive multi-resolution meshing. After meshing the data set, density ratio estimation was used to calculate the local density of the data points, and then clustering was carried out. Reference [24] set a new formula for calculating local density in the density peak clustering algorithm. The average value of local density was used as the density threshold to screen outliers and eliminate them, and then the adaptive strategy was used to merge similar clusters. Campello et al. [25] extended the density peak clustering algorithm into a general hierarchical structure in order to avoid the impact of two user-specified parameters on the clustering effect, thereby improving the clustering accuracy to a certain extent.

Reference [26] proposed the sharing K-nearest neighbors and multiple assignment policies density peaks clustering algorithm (SKM-DPC). The algorithm defined the similarity between sample points based on shared neighbors, and introduced an amplification factor to redefine the local density. Based on the two-step allocation strategy of SNN-DPC algorithm, the condition of nearest neighbor reachability of inevitable dependent points was raised to improve the allocation result. The K-nearest neighbor majority voting principle was applied to the possible subordinate points that were not assigned. However, the distance information was not considered in the voting process, and it was easy to be affected by the distant nearest neighbor points. This could lead to incorrect decisions when dealing with the cluster boundary region, which in turn affected the subsequent allocation of points, and eventually leaded to a large number of allocation associated errors.

The above researchers basically believe that the density peak clustering algorithm provides an effective tool for finding clusters with different shapes and densities, but the algorithm has shortcomings such as high time complexity and easy to be affected by manual parameter adjustment. Therefore, this paper proposes a density peak clustering algorithm based on sparse auto-encoder and data space meshing via evidence probability distribution. Our main contributions are as follows.

Firstly, sparse auto-encoder is used to extract features and reduce dimensionality of high dimensional data matrix. The original density peak clustering algorithm needs to calculate the distance between each data point and all other data points, which causes the problem of high time complexity of the algorithm. The idea of grid is introduced to divide the data space into multiple grids.

Then only the distance between each data point and the data points in the adjacent grid is calculated, which can greatly reduce the computation amount and improve the efficiency of the clustering algorithm. In the K-nearest neighbor domain, the transfer probability is constructed based on shared proximity to prioritize the points that meet the conditions. In order to overcome the limitation of the majority voting principle, the evidence function is established according to the distance information and class cluster information of the points in the K-nearest neighbor domain under the framework of evidence reasoning. Then the evidence function is synthesized, and a truth-based reliability decision method is proposed to address the deficiency of Pignistic decision probability.

Finally, the reliability decision method is used to convert the synthesized evidence function into evidence probability for fine decision making and determine the class cluster belonging of sample points to be assigned. Different allocation methods are designed for different levels of sample points, which makes the algorithm in this paper more effective in alleviating the associated error of allocation. The experimental results show that the proposed algorithm has good clustering effect when dealing with complex manifold data and real data set.

2. Preliminaries

2.1. Density peak clustering (DPC)

Density peaks clustering algorithm has two basic assumptions: the density of the algorithm itself is high, while other data points with lower density are surrounded. The distance from data points greater than its own density is relatively large. In the density peak clustering algorithm, there are two important parameters, local density p_i and relative distance δ_i, the values of these two parameters are related to the clustering effect of the whole algorithm.

Definition 1. Local density. The two formulas for calculating local density in the density peak clustering algorithm are shown in equations (1) and (4).

When clustering large data sets, it uses truncation kernel to calculate local density:

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{1}$$

Where $\chi(x)$ is the indicative function.

$$\chi(x) = \begin{cases} 1, x < 0 \\ 0, x \geq 0 \end{cases} \tag{2}$$

$$d_{ij} = \sqrt{\sum_{i=1}^{n} |x_i - x_j|^2} \tag{3}$$

Where d_{ij} represents Euclidean distance. d_c stands for truncation distance. When the data set size is large, the clustering results are less affected by the truncation distance d_c. When the data set size is relatively small, the clustering results are greatly affected by the truncation distance d_c. In order to avoid the influence of truncation distance d_c on clustering results, the density peak clustering algorithm adopts formula (4) to calculate local density ρ for small-scale data sets.

When clustering small-scale data sets, Gaussian kernel is used to calculate the local density:

$$\rho_i = \sum_j exp(-\frac{d_{ij}^2}{d_c^2}) \tag{4}$$

It can be seen from equation (4) that local density ρ is also affected by truncation distance d_c. Therefore, proposed algorithm does not use truncation distance d_c to calculate local density ρ, but introduces the concept of attenuation factor and the idea of k-nearest neighbor to calculate local density ρ.

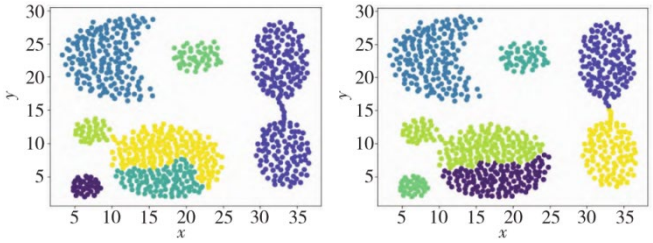
Definition 2. Relative distance. Relative distance δ_i represents the minimum Euclidean distance between a data point x_i and a data point x_j whose local density is higher than its own local density.

$$\delta_i = \begin{cases} \min\limits_{j:\rho_i<\rho_j} (d_{ij}) \\ \max\limits_{j} (d_{ij}) \end{cases} \tag{5}$$

For different data sets, the time and number of truncation distance d_c calculation are also different. The larger the number of data samples contained in the data set, the greater the distance between the sample points to be calculated, and the higher the time complexity. In the same sample data set, different truncation distance values will directly affect the local density calculation. In the density peak clustering algorithm, the quality of the clustering results is greatly affected by the clustering center. In this

algorithm, the selection of cluster center is determined by two parameters: local density ρ and relative distance δ. Moreover, the formula for calculating the local density includes the parameter truncation distance d_c, so the size of the truncation distance d_c will directly affect the clustering result of the data set.

The aggregation data set has seven classes, and the number of data points in different classes is not uniform, and the difference is large. Different class clusters will have a relatively large number of points connected. This paper takes the data set as an example to study the influence of different truncation distance values on clustering results. As can be seen from Figure 1, there are great differences in clustering results generated by different truncation distance values. The proposed method does not use truncation distance d_c to calculate local density ρ, but introduces the concept of attenuation factor and the idea of K nearest neighbor to reduce the negative effect of truncation distance in calculating local density, so as to avoid a series of effects brought by truncation distance d_c on clustering results.



(a) cluster result with d_c=0.5 (b) cluster result with d_c=1

Figure 1. Aggregation data clustering result

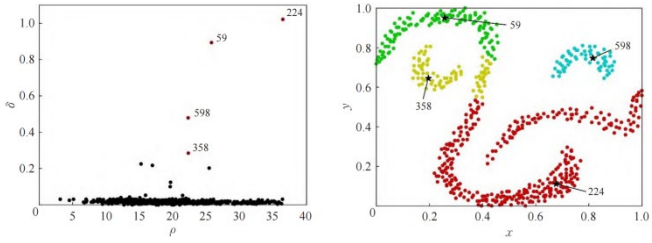
In the density peak clustering algorithm, firstly, the distance matrix D is obtained by using the sample set data, so that the local density ρ_i and the relative distance δ_i are obtained. Then, according to the product $\gamma_i = \rho_i \times \delta_i$ of local density ρ_i and relative distance δ_i, the decision graph is drawn to select the cluster center point. Clustering centers generally select data sample points with large local density and relatively far distance. The product of the two is used for selection to avoid the value of one item being too small, which makes the selection of clustering centers inaccurate. After selecting the correct cluster center, the sample data points of non-cluster center are allocated according to the distance principle. Finally, the result graph of clustering and various indexes are obtained.

It can be seen from the steps of the above algorithm that the time complexity of the density peak clustering algorithm mainly lies in the calculation and storage of the distance matrix. However, after the data is meshed, the Euclidean distance between the data points is no longer calculated. Instead, the time complexity is reduced by calculating the distance between the grids.

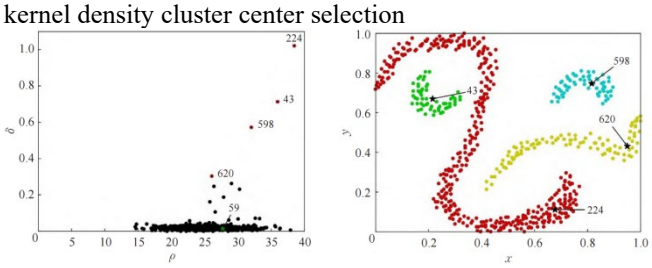
2.2. Defect analysis of DPC algorithm

Although DPC algorithm can cluster arbitrarily shaped clusters without iteration, especially on non-convex data sets with good performance, the DPC also has certain defects:

(1) The truncated kernel and Gaussian kernel of DPC algorithm can not fully reflect the local density information of sample points, the former only counts simply, and the latter requires the participation of all sample points, which easily leads to inaccurate density measurement. Manifold data is composed of some arc-shaped or ring class clusters. Due to the inaccuracy of density measurement, it is difficult for DPC algorithm to find the correct class cluster center. As shown in Figure 2, the Db data set is a complex manifold data, consisting of 4 arc-shaped clusters. The decision diagram of DPC algorithm on Db data set is shown in Figure 2(a), and the cluster center (black five-pointed star) selected according to the decision diagram is shown in Figure 2(b). It can be found that the DPC algorithm cannot correctly select the class cluster center, and sample points numbered 224 and 59 are selected as the class cluster center on the longest curved ribbon class cluster, among which sample point 59 is a multi-selected class cluster center. This is because the DPC algorithm considers the global distribution information of the sample points, ignoring the local distribution information around. In the face of manifold data with complex distribution structure, it is often difficult to accurately measure its density, which results in the excessive density of sample point No. 59, and its high-density nearest neighbor is located around sample point No. 224, which has the highest density, indirectly leading to its relative distance being too large. It ends up being the second most faulty candidate cluster center on the decision graph.



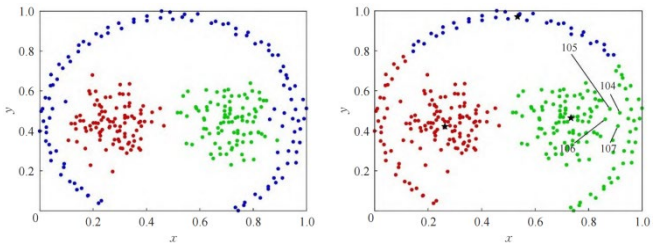
(a)Gaussian kernel density decision diagram (b) Gaussian kernel density cluster center selection



(c)Shared proximity density decision map (d) Shared proximity density cluster center selection

Figure 2. The cluster center selection by different density definition methods on the Db data set

(2) After selecting the center of the cluster, the DPC algorithm sorts each non-central point in descending order of density, and then assigns it to the cluster where its high-density nearest neighbor is located in turn. The clustering process is simple and efficient, but it is prone to assignment associated errors. When the high-density nearest neighbor of a point is assigned incorrectly, the point will also be assigned incorrectly. If that point is also a high-density nearest neighbor of another point, a chain reaction will result, with one misallocation of sample points causing more points to be misassigned. Experiments are performed on the Pathbased data set, as shown in Figure 3, which has two spherical clusters inside and one ring cluster outside. Although the DPC algorithm finds the correct cluster center, there is a wrong sample point allocation. In the boundary region of spherical cluster and annular cluster, the boundary between clusters is not obvious, and the density of the sample points of circular cluster is not only lower than that of the boundary points of spherical cluster, but also closer to the boundary points of spherical cluster. According to the distribution rule, sample points No. 105 and 106 of the ring cluster are allocated to the boundary points of the spherical cluster whose density is larger than that of the ring cluster and the distance between them is nearest. Due to these incorrect allocation, sample points No. 104 and 107 are also allocated incorrectly, which leads to a large number of ring cluster points being incorrectly allocated to the two inner spherical clusters.



(a)Pathbased data set (b) Gaussian kernel clustering result

Figure 3. Joint errors in allocation of DPC algorithm on Pathbased dataset

2.3. D-S evidence theory

D-S(Dempster-Shafer) evidence theory can deal with uncertainty, inaccuracy and incomplete information, and effectively integrate and reason these information, so it has a wide range of applications in data mining [27,28].

For a problem to be decided, it is assumed that the possible outcomes that can be recognized are represented by the set $\Theta = \{\theta_1, \theta_2, \cdots, \theta_n\}$, and Θ is called the recognition framework. For $\forall A \in 2^\Theta$, if the function $m: 2^\Theta \to [0,1]$ satisfies the following two conditions:

$$\begin{cases} m(\emptyset) = 0 \\ \sum_{A \subseteq \Theta} m(A) = 1 \end{cases} \tag{6}$$

Then m is called the basic probability assignment function (mass function). $m(A)$ represents the degree to which the evidence supports proposition A. If $m(A) > 0$, then A is called a focal element.

On the basis of the basic probability assignment function, trust function Bel and plausible function Pl can be used to express the lower limit and upper limit estimation of the degree of support for proposition A, and express the degree of uncertainty of proposition A to a certain extent:

$$\begin{cases} Bel(A) = \sum_{B \subseteq A} m(B), A \in 2^\Theta \\ Pl(A) = \sum_{B \cap A \neq \emptyset} m(B), A \in 2^\Theta \end{cases} \tag{7}$$

$Bel(A)$ indicates the degree to which the evidence trusts proposition A. $Pl(A)$ indicates the degree of trust that proposition A is not false, that is, the degree of no doubt about proposition A, which is of great significance for data analysis and decision making.

In order to combine information from multiple independent evidence sources, mass functions of multiple evidence can be combined by D-S synthesis rules, so as to achieve the fusion of multiple evidence and obtain a new global mass function, which is ready for the next step of decision analysis. Let m_1 and m_2 be two mass functions with focal elements A_i, and B_j, respectively. m is used to represent the mass function corresponding to the new evidence after the combination of m_1 and m_2, then the D-S synthesis rule is expressed as follows:

$$\mathrm{m(A)} = \begin{cases} \frac{\sum_{A_i \cap B_j} m_1(A_i) m_2(B_j)}{1-\mathrm{K}}, \mathrm{A} \neq \emptyset \\ 0, \mathrm{A} = \emptyset \end{cases} \tag{8}$$

Where $K = \sum_{A_i \cap B_j = \emptyset} m_1(A_i) m_2(B_j)$ is called the conflict coefficient. $\frac{1}{1-K}$ is called a regularization factor. m is also called the direct sum of m_1 and m_2, denoted $m = m_1 \oplus m_2$.

3. Proposed density peak clustering algorithm

A large number of scholars use cluster analysis to measure the similarity between different data sources, so as to find the relationship and rule between the data. The density peak clustering algorithm is a clustering algorithm using density. When the density peak clustering algorithm calculates the local density, it involves the truncation distance. The truncation distance is calculated according to the distance matrix of the sample data points, so it takes a long time for a large data set. In this paper, the sample data set is meshed, the density threshold is defined, the dense grid is screened, and the cluster center is selected directly from the dense grid, which greatly reduces the calculation time. In order to improve the clustering effect, the center sample points of non-class cluster are redistributed according to the idea of nearest neighbor.

3.1. Data dimensionality reduction based on sparse auto-encoder

Sparse auto-encoder is an unsupervised feature learning algorithm [29,30], and its structure is shown in Figure 4. It is a neural network model where the target output is the same as the input. Each sparse auto-encoder has two processes: the encoding process and the decoding process. During encoding, input is converted into hidden features; During decoding, the hidden features are reconstructed into the target output. It overcomes the disadvantage that auto-encoders cannot extract features effectively [31] and introduces sparse constraint term into the error function of auto-encoder to enforce dimensionality reduction expression on high-dimensional data. This method has a very broad application prospect.

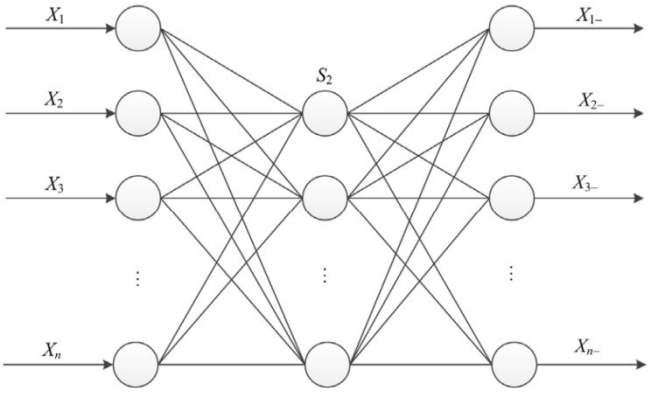


Figure 4. Structure of sparse auto-encoder

According to the structure of the sparse auto-encoder shown in Figure 4, the parameter of the sparse auto-encoder is set to (W, b). Where $W^{(l)}$ represents the connection weight between $l-th$ layer and $l+1-th$ layer. $b^{(l+1)}$ is the bias of $l+1-th$ layer. For a given input data, the output based on the forward propagation hidden layer is:

$$h_{W,b}(x) = \begin{bmatrix} a_1^{(2)} \\ a_2^{(2)} \\ \vdots \\ a_{S_2}^{(2)} \end{bmatrix} = \begin{bmatrix} f(\sum_{j=1}^{S_1} W_{1j}^{(1)} x_j + b_1^{(1)}) \\ f(\sum_{j=1}^{S_1} W_{2j}^{(1)} x_j + b_2^{(1)}) \\ \vdots \\ f(\sum_{j=1}^{S_1} W_{S_2 j}^{(1)} x_j + b_{S_2}^{(1)}) \end{bmatrix} \quad (9)$$

Where S_1 is the number of neurons in the input layer. S_2 is the number of hidden layer neurons. $f(\cdot)$ is the activation function defined for sigmoid, as shown in equation (10).

$$f(z) = \frac{1}{1+e^{-z}} \quad (10)$$

The sparse auto-encoder attempts to get the output vector $h_{W,b}(x)$ as close as possible to the input vector x. In order to obtain better sparse features, a sparse penalty term is added to the error expression, i.e.,

$$J_S(W, b; x) = \frac{1}{2}||h_{W,b}(x^{(i)}) - x^{(i)}||^2 + \beta \sum_{j=1}^{S_2} KL(\rho||\rho_j) \quad (11)$$

The first term is the sum of squares error term, which describes the difference between the entire training data. The second term is a sparse penalty term with a penalty coefficient β. Where ρ is a constant close to zero. ρ_j is the average activation of hidden unit j, its expression is:

$$\rho_j = \frac{1}{m}\sum_{i=1}^{m} a_j^{(2)}(x^{(i)}) \quad (12)$$

Where m is the number of samples.

Sparse constraint can be understood as making the average activity of neurons in the hidden layer extremely small. Sparse effect is best when $\rho_j \approx \rho$. KL divergence is introduced to limit the difference between the two. KL divergence is defined as:

$$\sum_{j=1}^{S_2} KL(\rho||\rho_j) = \sum_{j=1}^{S_2} [\rho ln \frac{\rho}{\rho_j} + (1-\rho) ln \frac{1-\rho}{1-\rho_j}] \quad (13)$$

When ρ is closer to ρ, the KL divergence value is smaller, and as the difference between the two becomes larger, the KL divergence value will increase accordingly. The optimization problem can be solved by neural activation forward transmission and error back propagation.

3.2. Data space meshing

For data space meshing, each dimension of data is evenly divided into the same number h of segments, the empty grid with 0 data points in the grid is deleted, and the number of remaining non-empty grids is M. Experimental results show that when the number of grid objects M is greater than $n/5$ of data samples, the clustering accuracy is higher [32].

For the number of mesh segments h, the size of h will affect the clustering result. The smaller h in mesh division denotes the larger grid length, which will lead to data sample points that do not belong to the same cluster being divided into the same cluster, resulting in a decrease in clustering accuracy. However, the larger h denotes the smaller mesh length, and even some grids contain only one data sample point, in this case, the accuracy of clustering is not much different from that before mesh division. So mesh division loses its meaning.

Homogeneous partition clustering of heterogeneous data sets can find the cluster center more directly and has low time complexity. Non-uniform partition may have a positive effect on data sets with different degrees of sparsity. However, how to quantify data sparsity needs to be analyzed according to different data. In addition, non-uniform partitioning will reduce the generality of the algorithm and increase the complexity of the calculation. Therefore, this paper chooses the method of evenly dividing the number of grid segments h for cluster analysis.

3.3. Local density calculation and allocation strategy

Definition 3. Local density. The local density of grid i is calculated as the product of the number of sample data points g_i in the grid plus the number of sample data points g_j in k nearest neighbor grid j and the function e. Where

$e^{-d_{ij}^2}$ is the attenuation factor. d_{ij} represents the distance between the grid i and j.

$$\rho_i = g_i + \sum_{j \in KNN_i} e^{-d_{ij}^2} g_j \quad (14)$$

Definition 4. Relative distance. The relative distance δ_i of the grid represents the Euclidean distance between grid i and the denser grid, calculated as follows:

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}) \quad (15)$$

Definition 5. Density threshold. The density threshold is calculated as follows:

$$\rho_t = sort(\rho_i) \times d\% \quad (16)$$

Where d represents the upper quartile. $sort(\rho_i)$ means to sort the local density values of each grid cell according to the principle from largest to smallest.

The steps of the allocation strategy are as follows:

(a) Take any unlabeled sparse grid g from the sparse grid set G; If all grids are marked, skip to step (d).

(b) For unlabeled sparse grids, identify the data points within the grids.

(c) Compare the Euclidean distance between the data points in the sparse grid and the center points of the surrounding cluster, and assign the remaining data sample points using the idea of nearest neighbor.

(d) Complete the assignment.

3.4. Non-central point allocation strategy

After determining the class cluster center, labels can be assigned to non-class cluster center points according to the allocation strategy. The single-step allocation strategy adopted by DPC algorithm is easy to cause a lot of allocation errors. In order to improve the fault-tolerance of DPC algorithm distribution, a two-step distribution strategy is proposed in this section. In the first step, transfer probability distribution is performed, and sample points are connected according to transfer probability to complete the distribution of core sample points of cluster and form the backbone of cluster. The second step is to carry out the evidence probability distribution. According to the evidence probability, the unassigned non-core sample points in the cluster boundary area are accurately classified into clusters, and the clustering of the whole data set is finally completed.

Most density clustering algorithms are considered from the connectivity between sample points. DBSCAN algorithm defines a cluster as a set of maximum density connected points, while DPC algorithm connects sample points with its high-density nearest neighbors. However, when the proximity of two connected points is low, allocation errors are likely to occur. The transfer probability distribution takes into account the shared proximity between two sample points. The idea is to start from the center of the cluster and find points with greater shared proximity in the K-nearest neighbor domain for label transfer. When the transfer point (labeled point) and the passed point (unlabeled point) are passed, it is considered that the two sample points are connected once. In this way, the label in the center of the class cluster is diffused to the whole class cluster. Those connected sample points constitute the backbone structure of the class cluster and can represent the core area of a class cluster.

In order to demonstrate the connection effect of transitive probability assignment, a simple data set containing two lunate class clusters is artificially generated. Figure 5 shows the effect diagram of connecting sample points according to transfer probability distribution. The blue five-pointed star represents the cluster center, and the yellow sample point is the sample point that does not participate in the connection and participates in the next stage of distribution without the arrow pointing from the other sample points. What needs to be considered in this process is to which sample points can an allocated sample point transmit its label, so as to reduce the probability of errors in the allocation process as much as possible. The following is the relevant definition to achieve this allocation process.

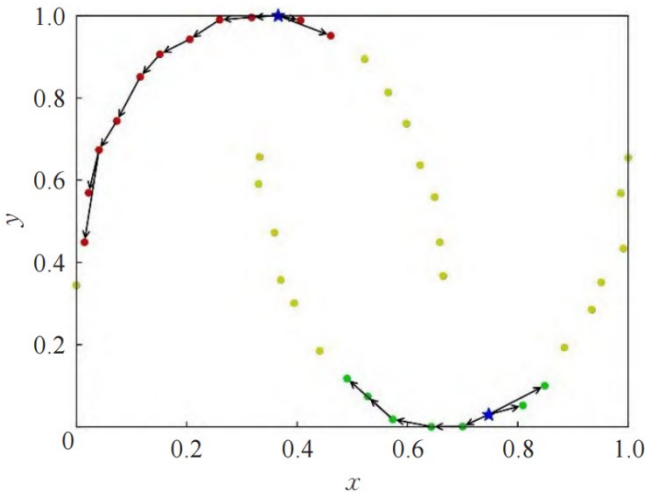


Figure 5. Transfer probability distribution of the sample point connection effect

Definition 6. Transfer probability. The transfer probability is defined by K-nearest neighbors and shared proximity:

$$P_{ij} = \begin{cases} \frac{Snd(i,j)}{\sum_{x_l \in KNN(x_i)} Snd(i,j)}, x_j \in KNN(x_i) \\ 0, x_j \notin KNN(x_i) \end{cases} \quad (17)$$

If sample point x_i has been assigned a cluster label, P_{ij} represents the probability that its cluster label will be passed to another sample point x_j, and $P_{ij} = 0$ is set that sample point x_i cannot pass the label to itself.

Definition 7. Transitive probability assignment. If sample point x_i has been assigned cluster labels, for $\forall x_j \in X$, its transfer probability assignment concept is as follows:

$$Lb(x_j) = \begin{cases} Lb(x_i), P_{ij} \geq \frac{1}{K} \; and \; Snd(i,j) \geq \mu \\ -1, \mu = \frac{\sum_{x_i \in X} \sum_{x_j \in X} Snd(i,j)}{n(n-1)} \end{cases} \quad (18)$$

Where $Lb(x_i)$ is the cluster label for x_i. $Lb(x_j) = -1$ indicates that point x_j does not pass labels. K is the number of nearest neighbors. μ is the mean of the shared proximity matrix. When x_i passes the label to x_j, this indicates that

the two points are very closely connected, and x_i and x_j can be considered to belong to the same class cluster.

Algorithm 1. Transitive probability distribution strategy

Input: cluster center $centers$, number of neighbors K, shared proximity matrix Snd.
Output: the core set of allocated points and the remaining sample set R to be allocated.
Step 1. Initializing an empty $Queue$, adding the class cluster $centers$ to the queue.
Step 2. Retrieves and deletes the first element x_p from $Queue$.
Step 3. Find the K-neighbor set $KNN(x_p)$ of x_p, and traverse every neighbor x_q of x_p. If x_q is not assigned, and x_q and x_p can be transitive probability assignment, then pass x_p's class cluster label to x_q, and add x_q to the end of $Queue$, otherwise, skip x_q to access the next neighbor.
Step 4. If the $Queue$ is empty, the algorithm will be ended. Otherwise, go to Step 2.

3.5. Evidence probability distribution

The core points of the cluster are clustered through the transfer probability distribution, and the backbone structure of the cluster can be formed in most cases. The non-core set R that does not satisfy the transitive probability assignment is often the boundary region of the class cluster. For this part of the point, the classification of its cluster can not be determined accurately, which will have a certain impact on the final clustering accuracy. Even if the cluster backbone cannot be formed, the information of the points allocated in the previous step should be fully used for label diffusion, and the wrong chain reaction in the allocation process should be cut off as much as possible.

Using the traditional KNN decision to cluster is to find the K-nearest neighbors of the unallocated sample points, and then to classify the unallocated sample points according to the voting principle based on their class cluster information. However, there are some problems, it does not take into account the difference between K neighbors, and it does not give the probability of belonging to each class cluster. As shown in Figure 6(a), it is obviously unreasonable for sample points not allocated according to the voting principle to belong to class cluster B. The EKNN algorithm proposed in reference [33] is a supervised classification algorithm that improves the traditional KNN algorithm by combining evidence theory. The category information of K neighbors is regarded as evidence, and each piece of evidence takes into account the distance from the sample point to be classified. The closer the distance from the sample point to be classified, the more important its category is to determine the category of the sample point to be classified. Evidence synthesis is to synthesize these information from the perspective of information fusion to obtain a new evidence function, which provides the basis for the next decision. Inspired by reference [33], considering the distance and cluster information of K-neighbors of unassigned points, an evidence function is established for each neighbor, and the evidence probability of the point belonging to each cluster is calculated after the K neighbor information is fused by the evidence synthesis rule. Using evidence probability to precisely guide the clustering of points can achieve higher clustering accuracy than the traditional KNN majority voting. As shown in Figure 6(d), the decision is made according to the principle of maximum probability, and the unallocated sample points will be classified to clustering A, which is obviously more reasonable.

When applying evidence theory to clustering, because clustering is an unsupervised learning algorithm, there may be other unassigned sample points in the k-nearest neighbor of an unassigned sample point. Therefore, it is also necessary to consider how to establish evidence functions for the nearest neighbor points without class cluster labels. The following gives the evidence function construction methods for two kinds of nearest neighbor points.

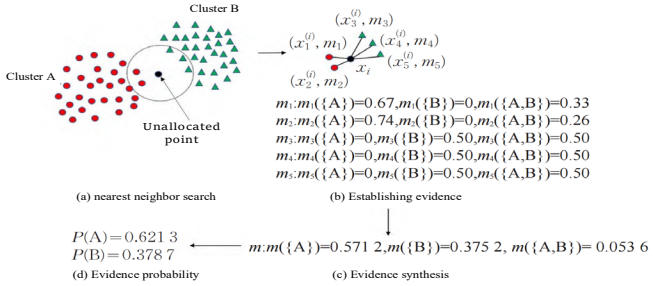


Figure 6. The process of generating evidence probability

Before evidence fusion, the cluster of data set X is used as the identification framework $\Omega = \{c_1, c_2, \cdots, c_M\}$. M is the number of class clusters. For $\forall x_i \in R$, the k-nearest neighbor set $KNN(x_i)$ of x_i is found, and the class cluster information of $x_j^{(i)} \in KNN(x_i)$ is regarded as evidence. If $x_j^{(i)}$ is an assigned point, it considers the distance relationship between it and x_i, and constructs a proof function for distance monotonicity. $m(\cdot\,|x_j^{(i)})$ indicates the confidence that x_i belongs to the class cluster $x_j^{(i)}$:

$$\begin{cases} m(c^{(j)}|x_j^{(i)}) = 0.95 \times exp(-d_{ij}^2) \\ m(\Omega|x_j^{(i)}) = 1 - 0.95 \times exp(-d_{ij}^2) \end{cases} \tag{19}$$

If $x_j^{(i)}$ is an unassigned point, it builds an evidence function $m(\cdot\,|x_j^{(i)})$ for it as well:

$$m(\cdot\,|x_j^{(i)}) = 1 \tag{20}$$

$c^{(j)} \in \Omega$ indicates the class cluster to which $x_j^{(i)}$ belongs. d_{ij} is the Euclidean distance between x_i and $x_j^{(i)}$. The smaller d_{ij} denotes the larger $m(c^{(j)}|x_j^{(i)})$, indicating that x_i and $x_j^{(i)}$ are more likely to belong to the same class

cluster. $m(\Omega|x_j^{(i)})$ represents the degree of ignorance and it shows the confidence in assigning x_i to all class clusters. The larger $m(\Omega|x_j^{(i)})$ denotes that the more difficult it is to judge the class cluster belonging of x_i.

Thus, the K neighbors of x_i provide K pieces of evidence for their cluster information $\{m_k|k = 1,2,\cdots,K\}$. In order to obtain a piece of evidence describing the class cluster assigned by x_i, the classical D-S evidence fusion rule is used to synthesize it $K-1$ times, and a fused evidence $m = m_1 \oplus m_2 \oplus \cdots \oplus m_K$ is obtained.

Definition 7. Evidence probability. The focal element in the fusion evidence m of x_i contains the class clusters and recognition frameworks to which K neighbors belong. In order to get the probability that x_i is assigned to each class cluster, the decision probability obtained by converting m with the truth probability is called the evidence probability of x_i.

The idea of the evidence probability distribution strategy is to continuously find the undistributed sample with the maximum evidence probability in the sample point set to be distributed and assign it first until the number of undistributed sample points reaches zero.

Algorithm 2. Evidence probability distribution strategy

Input: points R to be allocated, number of neighbors K, number of clusters M.

Output: Final clustering result S.

Step 1. Initialize an allocation matrix A with H rows and M columns, where $H = |R|$ is the number of unallocated points, each unallocated point corresponds to a row of the matrix, and each class cluster corresponds to a column of the matrix.

Step 2. For $\forall x_i \in R$, whose row number in the distribution matrix is denoted as h, calculate the x_i evidence probability $(P_1, P_2, \cdots, P_M)$, and then change the evidence probability $(P_1, P_2, \cdots, P_M)$ is added to all columns corresponding to the row in x_i. $A(h,m)$ is the element of row h, column m in matrix A, which represents the probability P_m assigned by x_i to class cluster c_m.

Step 3. Find the maximum value MV of the matrix A, if $MV \neq 1/M$, and have row h and column m corresponding to $A(h,m) = MV$. Assign the point x_i corresponding to row h to the class cluster c_m corresponding to column m, and then delete x_i from R, if $MV = 1/M$, increase the nearest neighbor search range, let $K = K + 1$.

Step 4. If R is not empty, perform Step 1 to continue allocating unallocated points in R, otherwise, end the algorithm.

4. Time complexity analysis

Assuming that the number of samples in the data set is n, the time complexity of the DPC algorithm mainly comes from calculating the distance matrix, calculating the local density of each sample point, and calculating the relative distance of each sample point. The time complexity of each part is $O(n^2)$, so the total time complexity of the DPC algorithm is $O(n^2)$.

The time complexity of this proposed algorithm is mainly composed of the following parts: (1) The time complexity of calculating the distance matrix is $O(n^2)$; (2) The time complexity of calculating the shared proximity matrix using K-nearest neighbors and shared nearest neighbors is $O(Kn^2)$; (3) The time complexity of calculating local density is $O(n)$; (4) The time complexity of calculating relative distance is $O(n^2)$; (5) In the worst case, the K-nearest neighbors of n core points need to be traversed, and the time complexity is $O(Kn)$; (6) The evidence probability distribution strategy also considers the allocation of n non-core points in the worst case, and the time complexity of forming a distribution matrix for each non-core point is $O(Kn)$. The time complexity of finding the maximum MV in the allocation matrix A is $O(Mn)$, regardless of the fact that n is decreasing every time allocation, so the time complexity of the second step allocation strategy is $O((K+M)n^2)$. To sum up, the time degree of the proposed algorithm is $O((K+M)n^2)$, and both K and M are much smaller than n, and the time complexity can be approximated to $O(n^2)$, which is the same as the time complexity of the DPC algorithm.

5. Experimental results and analysis

To evaluate the effectiveness of the proposed algorithm and detect its performance, this section presents a performance comparison compared to the comparison algorithm on both artificial and real datasets. WANG et al. [34] divided the artificial data set into 7 categories: uniform density data set, variable density data set, convex data set, non-convex data set, single-peak data set, multi-peak data set and cross-winding data set. Therefore, this paper adopts 7 manual datasets and 6 UCI datasets as shown in Table 1 and Table 2 to conduct simulation experiments. In addition to ED-Hexagon and Blood datasets, other manual datasets and UCI datasets are taken from reference [35], while the rest are taken from reference [34].

Table 1. Artificial data set

Data set	Sample number	Dimension	Number of categories
ED_Hexagon	361	2	2
Jain	373	2	2
D31	3100	2	31
Donutcurves	1000	2	4
Banana	4811	2	2
T4	8000	2	6
Chainlink	1000	3	2

Table 2. UCI data set

Data set	Sample number	Dimension	Number of categories
Liver	345	6	2
Wpbc	198	33	2
Glass	214	9	6
Ecoli	336	7	8
Blood	748	4	2
Wine	178	13	3

The experimental environment of this paper are Windows 64-bit operating system, CPU AMD Ryzen 7 6800H, AR 4.7GHz, 16.0GB RAM, PyCharm Community Edition 2023.2.1, Python 3.9.

5.1. Evaluation index

In this paper, Adjusted Rand Index (ARI), Normalized Mutual Information (NMI) and Fowlkes and Mallows Index (FMI) are selected as cluster evaluation indicators. NMI). ARI is an indicator to measure the similarity of two clustering results, and its value range is [-1,1]. The value is close to 1, the clustering result is better; the value is close to 0, the clustering result is worse. Its definition is shown in Equation (21):

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]} \quad (21)$$

$max(RI)$ represents the value if the clustering results are completely correct. $E[RI]$ is the expected value of RI.

NMI is an external indicator that measures the degree of information sharing between clustering results and real categories, with values ranging from [0,1]. When the value of NMI is 0, it means that the clustering results are completely independent of the real category, and the value of 1 means that the clustering results are perfectly corresponding to the real category. Its definition is shown in equation (22):

$$NMI = \frac{\sum_{i=1}^{k(C)} \sum_{j=1}^{k(T)} log_a(\frac{n \cdot n_{i,j}}{n_i \cdot n_j})}{\sqrt{(\sum_{i=1}^{k(C)} n_i \, log_a \frac{n_i}{n})(\sum_{i=1}^{k(C)} n_j \, log_a \frac{n_j}{n})}} \quad (22)$$

Where $k(C)$ is the number of clusters of clustering results. $k(T)$ is the number of clusters of real clustering results. n_i is the number of samples of cluster i. n_j is the number of samples of cluster j. $n_{i,j}$ is the number of samples belonging to cluster i in clustering result C and cluster j in real clustering result T. n is the total number of samples in the data set.

Based on the idea of pairwise comparison, FMI compares whether a pair of data points is assigned to the same cluster in two cluster results at the same time, and then measures the similarity between the given two clusters, whose value range is [0,1]. Where 1 indicates that the two clustering results are identical. 0 indicates that the clustering results are completely inconsistent. Its definition is shown in equation (23):

$$FMI = \frac{TP}{\sqrt{(TP+FP)(TP+FN)}} \quad (23)$$

TP refers to the number of data point pairs assigned to the same cluster in two clusters at the same time. FP is the number of pairs of data points that are assigned to the same cluster in one cluster, but not in another cluster. FN refers to the number of pairs of data points that are not the same cluster in one cluster but are in another cluster.

5.2. Experiments on manual data sets

In this paper, DPC algorithm, DBSCAN algorithm and ICKDC algorithm are used as comparison algorithms, and experiments are conducted on 7 different types of manual data sets. The experimental results and analysis are as follows:

(1) Uniform density data set. Different clusters have similar densities in the data set, as shown in Figure 7(a). In a data set with uniform density, most points have similar local densities, so it is difficult to distinguish points with significant density as cluster centers, which makes it difficult for DPC algorithm to accurately identify cluster centers. Secondly, because the density between data points does not change much, the cluster boundary becomes blurred, and the classification of cluster boundaries by DPC algorithm depends on the density difference, but in the case of uniform density, this difference almost does not exist. Therefore, in EX_Hexagon data set, DPC algorithm performs poorly in clustering.

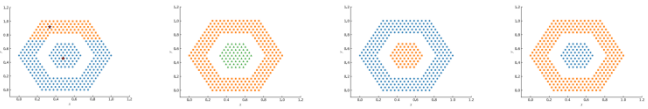


Figure 7. Experiments on uniform density data sets (from left to right: DPC on EX_Hexagon, DBSCAN on EX_Hexagon, ICKDC on EX_Hexagon, proposed on EX_Hexagon)

(2) Variable density data set. Different clusters have different densities in the data set, as shown in Figure 8(a). In Jain data set, the upper branch data is sparsely distributed, and the lower branch data is densely distributed. In such data sets, the local density dominates the decision value, while the correction effect of relative distance is weak, which leads to the wrong selection of the cluster center. As shown in Figure 8(a), Figure 8(b) and Figure 8(c), DPC algorithm, DBSCAN algorithm and ICKDC algorithm cannot effectively separate clusters.

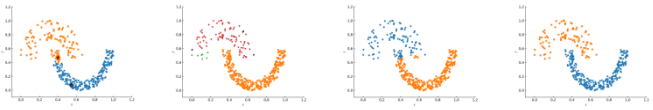


Figure 8. Experiments on variable density data sets (from left to right: DPC on Jain, DBSCAN on Jain, ICKDC on Jain, proposed on Jain)

(3) Convex data sets. In any cluster, all connections between data points are contained within the cluster region, as shown in Figure 9(a). In this kind of data set, there are usually obvious cluster boundaries and good spatial separation, which makes the cluster center easier to be identified. As shown in Figure 9(a), Figure 9(b) and Figure 9(c), DPC algorithm, DBSCAN algorithm, ICKDC algorithm and GDPC algorithm all perform well in clustering.

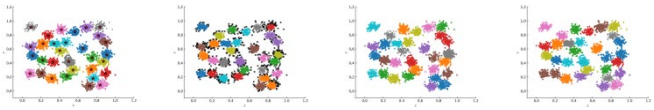


Figure 9. Experiments on convex density data sets (from left to right: DPC on D31, DBSCAN on D31, ICKDC on D31, proposed on D31)

(4) Non-convex data sets. In any cluster, all the connected parts between data points are contained within the cluster region, as shown in Figure 10(a). In the Donutcurves data set, three non-convex clusters and one convex cluster are included, since the remaining sample points are allocated based on the Euclidene distance from each point to the nearest high-density point by both the DPC and ICKDC algorithms. This means that a point may be assigned to the cluster where its closest high-density point is located, even if that point is spatially closer to another cluster center. Therefore, the DPC algorithm and ICKDC algorithm performed poorly in clustering on the Donutcurves data set.

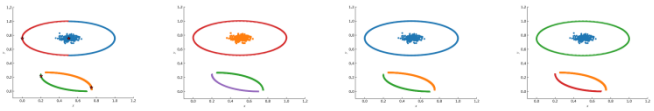


Figure 10. Experiments on non-convex density data sets (from left to right: DPC on Donutcurves, DBSCAN on Donutcurves, ICKDC on Donutcurves, proposed on Donutcurves)

(5) Unimodal data set. Any cluster has a density peak as the clustering center, as shown in Figure 11(a). On most unimodal data sets, DPC algorithm, DBSCAN algorithm, ICKDC algorithm and proposed algorithm perform better. However, in Banana data set, there are more data points with the characteristics of cluster center, and the parameter sensitivity of DPC algorithm and ICKDC algorithm is strong, which leads to the wrong selection of cluster center and the wrong clustering result.

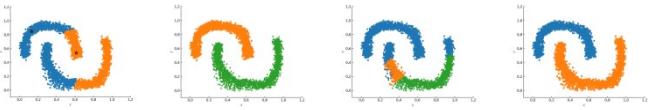


Figure 11. Experiments on unimodal density data sets (from left to right: DPC on Banana, DBSCAN on Banana, ICKDC on Banana, proposed on Banana)

(6) Multi-peak data set. Any cluster has multiple centers with high local density as clustering centers, as shown in Figure 12(a). T4 data set not only contains multiple points with high local density, but also contains a lot of noise. As shown in Figure 12(a) and Figure 12(b), the influence of noise causes the DPC algorithm and ICKDC algorithm to produce wrong clustering results. If the algorithm cannot effectively identify noise, it will not be able to effectively cluster on such data sets.

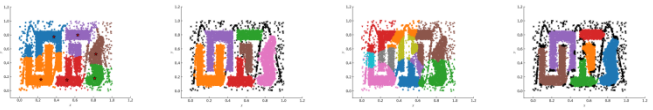


Figure 12. Experiments on multi-peak density data sets (from left to right: DPC on T4, DBSCAN on T4, ICKDC on T4, proposed on T4)

(7) Cross-wound data set. Two or more clusters of a data set make it difficult for the algorithm to accurately distinguish between different clusters, especially in the cluster boundary region. The DPC algorithm shown in Figure 13(a) and Figure 13(c) is affected by the chain allocation problem on the Chainlink data set, resulting in an intersection between errors, as shown in Figure 13(b). The clusters of the Chainlink data set are intertwined with each other and have complex and irregular shapes. The remaining points were incorrectly allocated. ICKDC algorithm produces large quantum clusters in order to avoid the problem of incorrect allocation, and the clustering performance is poor.

Figure 13. Experiments on cross-wound data sets (from left to right: DPC on Chainlink, DBSCAN on Chainlink, ICKDC on Chainlink, proposed on Chainlink

In summary, the proposed algorithm is effective and has improved accuracy and clustering performance compared with the comparison algorithms.

As shown in Tables 3,4,5, among the evaluation indexes of different clustering algorithms on artificial data sets, the clustering performance of the proposed algorithm is all good. The algorithm in this paper is a two-stage algorithm. In the first stage, more clustering centers are selected by using the idea of Gaussian distribution, so as to avoid the

wrong allocation problem caused by the influence of relative distance when allocating the remaining sample points. As shown in Figure 7(a), 10(a) and 13(a), after selecting the clustering center, DPC algorithm divides the data points belonging to the same cluster into another cluster due to the influence of relative distance, and the algorithm in this paper can cluster accurately. Secondly, in the second stage, the suture factor model proposed in this paper pays attention to the global characteristics of the data to a certain extent, and is more suitable for data sets with complex structures than the DPC algorithm. Finally, the proposed algorithm in this paper can not only screen cluster centers, but also identify noise, as shown in Figure 12(d). The reason is that the algorithm in this paper is based on data space meshing via evidence probability distribution. The remarkable characteristic of data space meshing is its symmetry, and its two tails are symmetric, which means that the probability of extreme values appearing on both sides of the mean value is equal, and the clustering center can be selected in the forward direction, and the noise can be identified in the reverse direction. Therefore, compared with DPC algorithm and ICKDC algorithm, the proposed algorithm performs better in clustering on T4 data set.

Table 3. ARI results on manual data set

Data	DPC	DBSCAN	ICKDC	Proposed
EX_hexagon	-0.0902	1.0000	1.0000	1.0000
Jain	0.7056	0.9732	0.7306	1.0000
D31	0.9359	0.5401	0.9498	0.9384
Donutcurves	0.7594	1.0000	0.7138	1.0000
Banana	0.0470	1.0000	0.3697	1.0000
T4	0.6056	0.9052	0.3914	0.9978
Chainlink	0.3313	1.0000	0.2893	1.0000

Table 4. NMI results on manual data set

Data	DPC	DBSCAN	ICKDC	Proposed
EX_hexagon	0.1117	1.0000	1.0000	1.0000
Jain	0.6448	0.9179	0.6093	1.0000
D31	0.9574	0.8379	0.9654	0.9584
Donutcurves	0.8479	1.0000	0.8572	1.0000
Banana	0.0332	1.0000	0.4157	1.0000
T4	0.7350	0.8983	0.5479	0.9957
Chainlink	0.4015	1.0000	0.5154	1.0000

Table 5. FMI results on manual data set

Data	DPC	DBSCAN	ICKDC	Proposed
EX_hexagon	0.6320	1.0000	1.0000	1.0000
Jain	0.8779	0.9896	0.8913	1.0000
D31	0.9379	0.5717	0.9514	0.9404
Donutcurves	0.8244	1.0000	0.8161	1.0000
Banana	0.5304	1.0000	0.6982	1.0000
T4	0.6805	0.9239	0.4954	0.9982
Chainlink	0.6954	1.0000	0.5377	1.0000

5.3. Experiments on UCI data set

The clustering performance of DPC algorithm, DBSCAN algorithm, ICKDC algorithm and proposed algorithm on UCI data sets is further evaluated. Table 2 shows the basic information of selected 6 UCI data sets in this experiment. These UCI data sets have significant differences in dimension, feature number, shape, etc., which can evaluate the performance of the proposed algorithm in this paper from different angles and they are representative to a certain extent. In the parameter setting of the comparison algorithm, the DPC algorithm determines parameter k according to the number of clusters. The values of parameters eps and $minPts$ in DBSCAN algorithm are [1,20] and [0.01,1], and the step sizes are 1 and 0.01, respectively. The optimal result is obtained by iterating 2000 times. Parameter γ in ICKDC algorithm value range is [0.01,2], step size is 0.01, after iterating 200 times, it obtains the optimal result. In this paper, the parameter $threshold$ and σ values range in the proposed algorithm are [0.01,1] and {1,2,3,4,5}, respectively, and the optimal result is obtained through 500 iterations. The comparison results of evaluation indicators of different clustering algorithms on UCI data set are shown in Tables 6,7,8.

For UCI data sets, the performance of the proposed algorithm is generally good, but it still performs poorly on some data sets with high dimensions and complex distribution, and the clustering effect is lower than other algorithms. As shown in the Blood data set and Wine data set in the experiment, the clustering performance of the algorithm in this paper is poor. This is due to the fact that in a high-dimensional space, the distance difference between different points becomes smaller, making it more difficult to distinguish between different data points. Secondly, with the increase of dimensions, the volume of the data space increases rapidly, resulting in the existing data becoming sparse, which means that even large data sets may appear inadequate in high-dimensional Spaces, resulting in problems of over-fitting and poor generalization ability. In general, the proposed algorithm performs well compared with other algorithms, but DPC algorithm, DBSCAN algorithm and ICKDC algorithm all fail to achieve satisfactory results on UCI data sets. Compared with the manual data set, the evaluation indexes on the high-dimensional data set are decreased.

Table 6. ARI results on UCI data set

Data	DPC	DBSCAN	ICKDC	Proposed
Liver	0.0002	0.0444	0.0046	0.0547
Wpbc	-0.0057	0.4516	0.4185	0.4732
Glass	-0.0241	0.0254	0.0034	0.0254
Ecoli	0.3397	0.6152	0.6766	0.7560
Blood	0.5682	0.6247	0.7114	0.5682
Wine	0.5055	0.4959	0.4528	0.5238

Table 7. NMI results on UCI data set

Data	DPC	DBSCAN	ICKDC	Proposed
Liver	0.0004	0.0453	0.0142	0.0532
Wpbc	0.0095	0.3561	0.4098	0.4141
Glass	0.0469	0.3394	0.0291	0.1314
Ecoli	0.5248	0.5541	0.6289	0.7183
Blood	0.7338	0.6634	0.7708	0.7338
Wine	0.5647	0.5660	0.4481	0.5634

Table 8. FMI results on UCI data set

Data	DPC	DBSCAN	ICKDC	Proposed
Liver	0.5492	0.6481	0.7125	0.7174
Wpbc	0.7223	0.7370	0.6723	0.7455
Glass	0.3215	0.1917	0.4981	0.3770
Ecoli	0.4985	0.7259	0.7643	0.8257
Blood	0.7716	0.7534	0.8085	0.7716
Wine	0.6803	0.6932	0.6725	0.6657

5.4. Parameter sensitivity experiment

In the running process of the proposed algorithm in this paper, in order to evaluate the impact of the parameter $threshold$ on the clustering results of the algorithm, under the optimal parameter σ, the value range is set as [0.001,0.05], the step size is 0.001, and 50 iterations are performed to observe the $threshold$ changes of the ARI, NMI and FMI on the Ecoli data set.

As shown in Figure 14, in multiple experiments, when the parameter $threshold$ changes slightly, the experimental results hardly changes, indicating that the parameter $threshold$ is less sensitive.

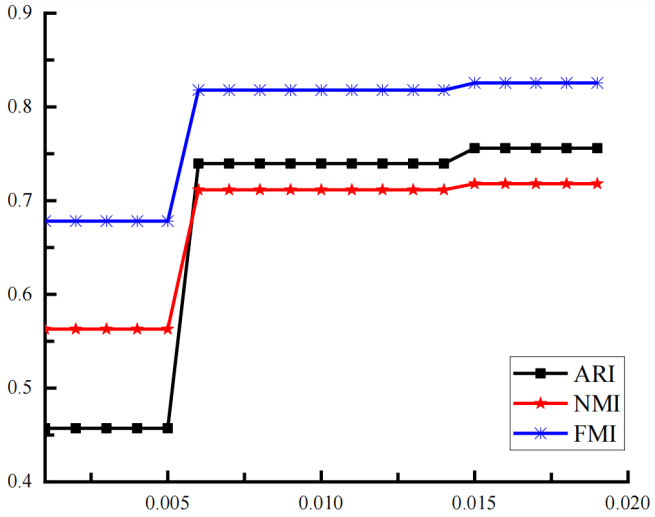


Figure 14. Parameter threshold sensitivity experiment

Similarly, in order to evaluate the influence of parameter σ on the clustering results of the proposed algorithm in this paper, under the optimal parameter $threshold$, the value range of σ is set to [0.1,3], the step size is 0.1, and the iteration is 30 times. The changes of parameter σ on the evaluation indicators ARI, NMI and FMI for the Ecoli data set are observed. As shown in Figure 15, when the parameter σ changes slightly, the experimental results change accordingly, but the change amplitude is small, which indicates that the sensitivity of parameter σ is small.

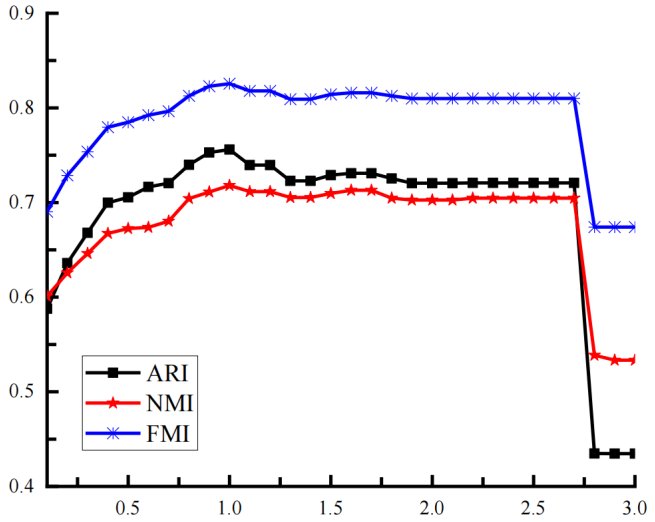


Figure 15. Parameter σ sensitivity experiment

Generally, the experimental results of the proposed algorithm in this paper will change dramatically only when the parameters change greatly. On the contrary, when the parameter change amplitude is small, the experimental results change amplitude unchanged or not much. Therefore, the experimental results show that the proposed

algorithm has low parameter sensitivity and high stability and generalization ability.

6. Conclusion

Data clustering will consume a lot of computing resources. In order to avoid consuming a lot of time in calculating Euclidean distance between data sample points by density peak clustering algorithm, this paper adopts sparse auto-encoder for dimensionality reduction of data set and maps it to the corresponding grid to reduce the running time. Aiming at the subjective selection of truncation distance d_c in density peak clustering algorithm, a local density calculation method is defined according to the K-nearest neighbor. This calculation method is independent of the size of the data set and is independent of the truncation distance d_c, which effectively avoids the influence of truncation distance d_c on the clustering effect. On the basis of setting density threshold, the workload of selecting cluster center is further reduced. The Euclidean distance is calculated not only from the center point of the nearest cluster, but also from the data points contained in the nearest cluster, which reduces the possibility of joint error caused by the allocation strategy. DPC algorithms may have limitations when dealing with high-dimensional data, because the sparsity and spatial complexity of high-dimensional data may cause the algorithm to not accurately reflect the similarity between data points. DPC algorithms may have limitations in identifying noise points in data sets, which may affect the accuracy of clustering results. However, how to determine the effective grid division mode under different data sets to further improve the clustering efficiency will be the direction of future research.

Acknowledgements.
None.

References

[1] Hilbert M, López P. The world's technological capacity to store, communicate, and compute information[J]. Science, 2011, 332(6025): 60-65.
[2] Anand R, Veni S, Aravinth J. An application of image processing techniques for detection of diseases on brinjal leaves using k-means clustering method[C]//2016 international conference on recent trends in information technology (ICRTIT). IEEE, 2016: 1-6.
[3] Hennig C, Liao T F. How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification[J]. Journal of the Royal Statistical Society Series C: Applied Statistics, 2013, 62(3): 309-369.
[4] Rahnenführer J, De Bin R, Benner A, et al. Statistical analysis of high-dimensional biomedical data: a gentle introduction to analytical goals, common approaches and challenges[J]. BMC medicine, 2023, 21(1): 182.
[5] Yu J, Lu Z, Yin S, et al. News recommendation model based on encoder graph neural network and bat optimization in online social multimedia art education[J]. Computer Science and Information Systems, vol. 21, no. 3, 989-1012, 2024.
[6] Rostami M, Oussalah M, Berahmand K, et al. Community detection algorithms in healthcare applications: a systematic review[J]. IEEE Access, 2023, 11: 30247-30272.
[7] Shi K, Yan J, Yang J. A semantic partition algorithm based on improved K-means clustering for large-scale indoor areas[J]. ISPRS international journal of geo-information, 2024, 13(2): 41.
[8] Hasan M K, Habib A K M A, Islam S, et al. DDoS: Distributed denial of service attack in communication standard vulnerabilities in smart grid applications and cyber security with recent developments[J]. Energy Reports, 2023, 9: 1318-1326.
[9] Ran X, Xi Y, Lu Y, et al. Comprehensive survey on hierarchical clustering algorithms and the recent developments[J]. Artificial Intelligence Review, 2023, 56(8): 8219-8264.
[10] Suchy D, Siminski K. GrDBSCAN: A granular density-based clustering algorithm[J]. International Journal of Applied Mathematics and Computer Science, 2023, 33(2).
[11] Wang Y, Wang D, Zhou Y, et al. VDPC: Variational density peak clustering algorithm[J]. Information Sciences, 2023, 621: 627-651.
[12] Ding S, Du W, Li C, et al. Density peaks clustering algorithm based on improved similarity and allocation strategy[J]. International journal of machine learning and cybernetics, 2023, 14(4): 1527-1542.
[13] Pourbahrami S. A neighborhood-based robust clustering algorithm using Apollonius function kernel[J]. Expert Systems with Applications, 2024, 248: 123407.
[14] Yan H, Wang M, Xie J. ANN-DPC: Density peak clustering by finding the adaptive nearest neighbors[J]. Knowledge-Based Systems, 2024, 294: 111748.
[15] Yu D, Liu G, Guo M, et al. Density peaks clustering based on weighted local density sequence and nearest neighbor assignment[J]. IEEE Access, 2019, 7: 34301-34317.
[16] Kumar A, Singh S K, Saxena S, et al. CoMHisP: A novel feature extractor for histopathological image classification based on fuzzy SVM with within-class relative density[J]. IEEE Transactions on Fuzzy Systems, 2020, 29(1): 103-117.
[17] Guan J, Li S, He X, et al. Fast hierarchical clustering of local density peaks via an association degree transfer method[J]. Neurocomputing, 2021, 455: 401-418.
[18] Guo W, Wang W, Zhao S, et al. Density peak clustering with connectivity estimation[J]. Knowledge-Based Systems, 2022, 243: 108501.
[19] Wang M, Zhang Y Y, Min F, et al. A two-stage density clustering algorithm[J]. Soft Computing, 2020, 24(23): 17797-17819.
[20] Cheng D, Huang J, Zhang S, et al. Improved density peaks clustering based on shared-neighbors of local cores for manifold data sets[J]. IEEE Access, 2019, 7: 151339-151349.
[21] Fan T, Li X, Hou J, et al. Density peaks clustering algorithm based on kernel density estimation and minimum spanning tree[J]. International Journal of Innovative Computing and Applications, 2022, 13(5-6): 336-350.
[22] Liang W, Schweitzer P, Xu Z. Approximation algorithms for capacitated minimum forest problems in wireless sensor networks with a mobile sink[J]. IEEE Transactions on Computers, 2012, 62(10): 1932-1944.
[23] Maximo A, Velho L, Siqueira M. Adaptive multi-chart and multiresolution mesh representation[J]. Computers & Graphics, 2014, 38: 332-340.

[24] Xu X, Ding S, Shi Z. An improved density peaks clustering algorithm with fast finding cluster centers[J]. Knowledge-Based Systems, 2018, 158: 65-74.
[25] Campello R J G B, Moulavi D, Zimek A, et al. Hierarchical density estimates for data clustering, visualization, and outlier detection[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2015, 10(1): 1-51.
[26] Hongxiang Z H U, Genxiu W U, Zhaohui W. Density Peaks Clustering Algorithm Based on Shared Neighbor Degree and Probability Assignment[J]. Journal of Computer Engineering & Applications, 2024, 60(12).
[27] Fang N, Cui J. An Improved Dempster–Shafer Evidence Theory with Symmetric Compression and Application in Ship Probability[J]. Symmetry, 2024, 16(7): 900.
[28] Tang Y, Zhang X, Zhou Y, et al. A new correlation belief function in Dempster-Shafer evidence theory and its application in classification[J]. Scientific Reports, 2023, 13(1): 7609.
[29] Bi J, Wang Z, Yuan H, et al. Self-adaptive teaching-learning-based optimizer with improved RBF and sparse autoencoder for high-dimensional problems[J]. Information Sciences, 2023, 630: 463-481.
[30] Saufi S R, Isham M F, Ahmad Z A, et al. Machinery fault diagnosis based on a modified hybrid deep sparse autoencoder using a raw vibration time-series signal[J]. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(4): 3827-3838.
[31] Yin S, Li H, Sun Y, et al. Data Visualization Analysis Based on Explainable Artificial Intelligence: A Survey[J]. IJLAI Transactions on Science and Engineering, 2024, 2(2): 13-20.
[32] Yin S, Li H, Laghari A A, et al. An anomaly detection model based on deep auto-encoder and capsule graph convolution via sparrow search algorithm in 6G internet-of-everything[J]. IEEE Internet of Things Journal, vol. 11, no. 18, pp. 29402-29411, 15 Sept.15, 2024.
[33] Rabie A H, Saleh A I. A new diagnostic autism spectrum disorder (DASD) strategy using ensemble diagnosis methodology based on blood tests[J]. Health Information Science and Systems, 2023, 11(1): 36.
[34] Wang Y, Qian J, Hassan M, et al. Density peak clustering algorithms: A review on the decade 2014–2023[J]. Expert Systems with Applications, 2023: 121860.
[35] Du H, Hao Y, Wang Z. An improved density peaks clustering algorithm by automatic determination of cluster centres[J]. Connection Science, 2022, 34(1): 857-873.