


ITC 3/49 Information Technology and Control Vol. 49 / No. 3 / 2020 pp. 395-411 DOI 10.5755/j01.itc.49.3.23405	Adaptive Density Peak Clustering Based on Dimension-Free and Reverse K-Nearest Neighbours	
	Received 2019/05/20	Accepted after revision 2020/07/20
	 http://dx.doi.org/10.5755/j01.itc.49.3.23405	

HOW TO CITE: Wu, Q., Zhang, Q., Sun, R., Li, L., Mu, H., Shang, F. (2020). Adaptive Density Peak Clustering Based on Dimension-Free and Reverse K -Nearest Neighbours. *Information Technology and Control*, 49(3), 395-411. <https://doi.org/10.5755/j01.itc.49.3.23405>

Adaptive Density Peak Clustering Based on Dimension-Free and Reverse K -Nearest Neighbours

Qiannan Wu, Qianqian Zhang, Ruizhi Sun*, Li Li, Huiyu Mu, Feiyu Shang

College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
 Scientific Research Base for Integrated Technologies of Precision Agriculture (Animal Husbandry),
 the Ministry of Agriculture, Beijing 100083, China

e-mails: wuqiannan@cau.edu.cn, qqzhang@cau.edu.cn, sunruizhi@cau.edu.cn, lili_2018@cau.edu.cn,
 B20183080630@cau.edu.cn, shangfy_zgnydx@cau.edu.cn

*Corresponding author: sunruizhi@cau.edu.cn

Cluster analysis is a crucial component in consumer behaviour segmentation. The density peak clustering algorithm (DPC) is a novel density-based clustering method, but it performs poorly in high-dimension datasets and local density for boundary points. In addition, the DPC fault tolerance is affected by the one-step allocation strategy. To overcome these disadvantages, an adaptive density peak clustering algorithm based on dimension-free and reverse k -nearest neighbours (ERK-DPC) is proposed in this paper. First, we compute the Euler cosine distance to obtain the similarity of sample points in high-dimension datasets. Second, the adaptive local density formula is used to measure the local density of each point. Finally, the reverse k -nearest neighbour approach is added onto the two-step allocation strategy, which assigns the remaining points accurately and effectively. The proposed clustering algorithm was applied in experiments on several benchmark datasets and real-world datasets. After comparing the benchmarks, the results demonstrate that the ERK-DPC algorithm is superior to selected state-of-the-art methods.

KEYWORDS: Density peaks, Clustering, Local density, Euler cosine distance, Reverse k -nearest neighbour.

1. Introduction

With the development of information technology, an increasing number of consumption and production data have emerged. The question of how to find certain rules and consumption patterns for these large amounts of data is a problem of concern in various fields. Clustering is a research hotspot in the field of data mining, and it is also a typical unsupervised learning method [14]. Clustering methods can find dense and sparse areas of data without any prior knowledge, and thus can understand the global distribution of data and the relationship between data attributes. Clustering has been widely applied in many fields, (e.g., pattern recognition [17], market analysis [25], image processing [8], time series analysis [19], information retrieval [35] and social networking [5], among others). According to different clustering methods, several broad categories are defined, namely, which are hierarchical-based, partitioning-based, density-based, model-based and grid-based approaches [18].

The partitioning-based representative algorithms are the K-means algorithm [24] and fuzzy c-means clustering algorithm (FCM) [16]. The hierarchical clustering representative algorithms includes the BIRCH [36] and CURE [15] methods. Frey and Dueck proposed an affinity propagation clustering algorithm, which is an exemplar-based method [13]. A typical density-based clustering algorithm is DBSCAN [11], which can discover arbitrary shapes of clusters. However, those methods are sensitive to parameters. Due to the shortcomings of DBSCAN, a series of improved algorithms such as OPTICS [1], GDBSCAN [28], STDBSCAN [3] and GRIDEN [7] have been proposed. The density peak clustering algorithm (DPC) [27] is a novel density-based clustering method proposed by Rodriguez and Laio. DPC does not need to preset the cluster numbers, and has achieved promising efficiency and accuracy for non-spherical data and unbalanced data. DPC uses only one cutoff distance as an adjustable parameter, does not require iteration in the clustering process, and has lower time consumption than other clustering algorithms.

However, certain limitations exist in the DPC algorithm: (1) It does not perform well on high-dimensional data and suffers from the impact of “dimensional disasters”. In the traditional DPC algorithm,

the Euclidean distance is used as a measure of similarity between data, but in many cases in high-dimensional space, the concept of similarity no longer exists, which creates a severe test of high-dimensional data clustering. In addition, the Euclidean distance treats the importance of each dimension attribute equally, which at times fails to meet the actual needs. The calculation of the local density in the DPC algorithm and the distance to the higher local density point are both related to the distance between the sample points. Therefore, the distance metric in the DPC algorithm becomes a serious task when working with high-dimensional data; (2) The boundary points of the traditional local density are susceptible to the data distribution shape of different clusters, which has a great impact on the final clustering effect; (3) The traditional one-step allocation strategy has poor fault tolerance. Although many improved allocation methods can improve the clustering accuracy to a certain extent, the trade-off is higher time consumption.

To overcome the above problems, this paper proposes an adaptive density peak clustering algorithm based on dimension-free and reverse k -nearest neighbours (ERK-DPC). The organization of this paper is listed as follows. In Section 2, the related work is introduced. In Section 3, the concepts of the traditional DPC algorithm are reviewed as preliminaries for later sections. In Section 4, the ERK-DPC algorithm proposed in this paper is described in detail, and the time complexity is analysed. In Section 5, the proposed algorithm is tested on multiple synthetic datasets and UCI datasets, and the clustering accuracy and parameter sensitivity are analysed. Finally, selected conclusions and future work are presented in the final section.

2. Related Work

In this section, we focus on recent research advancements in the DPC algorithm.

Because the value of the cut-off distance is sensitive to the clustering results, many effective local density metrics have been proposed. Du et al. [9] used the k -nearest neighbour to replace the cutoff distance, thus considering the global density and local density of the data. In the case of poor clustering of high-di-

mensional data, the idea of principal component analysis was introduced. The application of k -nearest neighbours reduced the parameter sensitivity and improved the accuracy of the clustering results. Wang et al. [33] proposed rapid clustering using adaptive density peak detection, which estimates the local density through a nonparametric multivariate kernel. The nonparametric concept is a hot topic in the DPC algorithm. The proposed SNN-DPC [22] algorithm measured the similarity based on shared neighbours, redefined the local density and distance from the nearest larger density point, and introduced the shared neighbours and local density information between points. The advantage of this approach that it can effectively address variable-density clusters. The concept of geodesic distance was introduced into the DPC algorithm [10] because the traditional DPC algorithm cannot effectively solve multi-manifold structures and data distributions with arbitrary shape clusters.

The DPC algorithm suffers from the limitation of manual selection of the clustering centres according to the decision graph. Bie et al. [4] proposed a fuzzy CFSFDP adaptive selection centre cluster, which searched for all density peaks and treated each peak as a local cluster before merging the local clusters to find the global cluster. However, this method only considers the distance property when merging clusters, and thus the performance in complex data is not satisfactory. Wang and Song [32] proposed a new clustering algorithm to automatically detect the cluster centres via statistical tests. Similar to the statistical tests, selected adaptive thresholds are applied to the DPC algorithm. When the local density and the cluster centre distance of the data points are greater than the selected thresholds, it is considered to be the cluster centre.

The final allocation strategy for the DPC algorithm is prone to continuity errors. Xie et al. [34] proposed a fuzzy weighted k -nearest density peak clustering (FKNN-DPC) method. Two new allocation strategies were introduced. First, the core points and outliers in the data set are screened out. Second, the core points are distributed from the k -nearest neighbour points of each cluster centre. Finally, the outliers are assigned by calculating the membership of each non-cluster centre point to each cluster. Seyed Amjad Seyedi et al. [29] proposed dynamic graph-based label

propagation for density peak clustering. The clustering centres are identified using local densities, and these centres are subsequently used to form the cluster cores. Finally, novel graph-based label propagation is applied to spread the labels of the cluster cores to the remaining instances. This algorithm combines the DPC algorithm with a semi-supervised label propagation method to improve efficiency. However, iterations are performed during label propagation, which undoubtedly increases the complexity. Liu and Huang et al. [21] proposed a constraint-based density peak clustering algorithm that combines semi-supervised constraints, density clustering, and hierarchical clustering for the first time and is a semi-supervised robust clustering algorithm.

The extended application of combining the density peak clustering algorithm with other algorithms has also become a topic of high interest. SVDD [6] used the cut-off distance-based local density and support vector data description to improve the performance of outlier detection for noise or uncertain data. Tang et al. [30] proposed an enhanced density-based clustering method (E-FDPC) for selection of hyperspectral bands. In this method, the weighted distance between the normalized local density and the intra-cluster density is controlled by introducing parameters. Zhang [37] applied the density peak clustering algorithm to multiple document summaries. Bai et al. [2] proposed an overlapping population detection algorithm based on density peaks to explore the community structure in the network. This method used a similarity method to set the distances among nodes, and it tends to perform better on those simple structure networks than on infrequently complicated networks.

3. Density Peak Clustering

In this section, we introduce the selected concepts related to density peak clustering.

The establishment of the DPC [27] algorithm is based on two assumptions. One assumption is that the density of the cluster centre is higher than that of the surrounding neighbours, and the other is that the distance between one cluster centre and another cluster centre is relatively larger.

The DPC algorithm is a density-based clustering method, the main idea of which is to find high-density

peaks surrounded by low-density points. In the DPC algorithm, the local density and the distance to the cluster centre point are directly related to the distance between the data points. The distance between sample points is generally calculated using the Euclidean distance, and d is the dimension of the sample point.

$$d(x_i, x_j) = \sqrt{\sum_{c=1}^d (x_{ic} - y_{ic})^2}. \quad (1)$$

The local density of the point i denoted by ρ_i in the DPC algorithm, is represented by Equation (2).

$$\rho_i = \sum_{j \neq i} \chi(d(x_i, x_j) - d_c),$$

$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x > 0 \end{cases}. \quad (2)$$

where $d(x_i, x_j)$ is the Euclidean distance between data points i and j . d_c is the cutoff distance and must be specified in advance, and d_c is equivalent to a radius of all data points. The range of value is generally the value of the first 1-2% of the distance matrix D . Additionally, D is a set of distances between every two data points, which is sorted from small to large. Another formula that uses the Gaussian kernel formula to represent the local density of data points is Equation (3), which applies to small data.

$$\rho_i = \sum_j \left(-\frac{d(x_i, x_j)^2}{d_c^2} \right). \quad (3)$$

The closest distance from each point to a higher local density point is represented by δ_i , defined in Equation (4) as

$$\delta_i = \begin{cases} \min_{j: \rho_j > \rho_i} (d(x_i, x_j)), & \text{if } \exists j \text{ s.t. } \rho_j > \rho_i \\ \max_j (d(x_i, x_j)), & \text{otherwise} \end{cases}. \quad (4)$$

It can be observed from Equation (4) that if sample point i has the largest local density, its corresponding δ_i is also the largest.

After calculating ρ_i and δ_i of all data points, the decision graph is drawn based on the two variables. Figure 1(a) shows a clustering example with five clusters, and

Figure 1

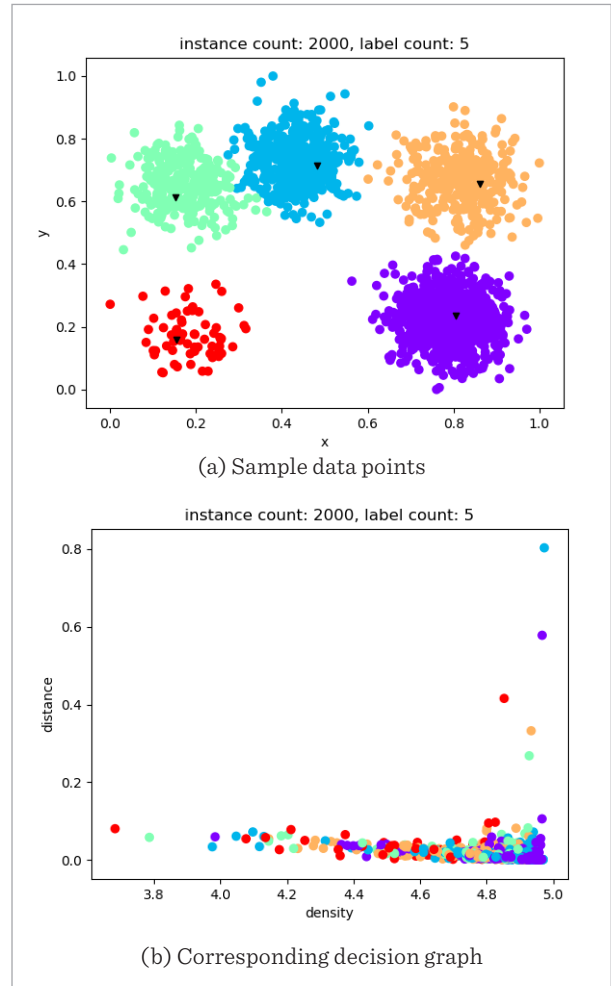


Figure 1(b) shows the corresponding decision graph. A point with a higher local density and a larger distance to a higher local density on the upper right side of the decision graph is also viewed as a cluster centre point. To avoid the influence of data points with small local density but a large clustering centre distance, another strategy for determining the cluster centre point is proposed. Thus, γ_i can be calculated by Equation (5) as an indicator to measure whether the data point is the cluster center.

$$\gamma_i = \rho_i * \delta_i. \quad (5)$$

The γ_i values are sorted. If the value of the point is larger, it might be selected as the clustering centre. When the

cluster number and the cluster centre are determined, the remaining non-clustered centre points are sorted according to the local density, from largest to smallest. From the highest local density point, each non-clustered centre point is assigned to the nearest cluster.

4. The Proposed ERK-DPC Algorithm

The main contributions of the algorithm proposed in this paper can be summarized in three aspects. (1) A new Euler cosine distance formula is proposed that can solve the problem in which the Euclidean distance cannot correctly represent the distance between sample points in high-dimensional data; (2) An adaptive local density formula is used to solve the problem of boundary clustering. As such, (1) and (2) can improve the robustness of the algorithm; (3) The idea of reverse k -nearest neighbours is used to improve the clustering accuracy of the non-cluster central points and the algorithm without reducing the density and increasing the density.

4.1. Main Idea of ERK-DPC

This section describes the main idea of ERK-DPC, including the Euler cosine distance, the adaptive local density, and the reverse k -nearest neighbours sample allocation strategy.

1 Euler cosine distance. Liwicki et al. [23] presented a cosine-based metric formula in the Euler principal component analysis algorithm. In Equation (6), x_j and x_q belong to the sample point of the data set \mathfrak{R}^p , and $x_j(c)$ represents the c -th dimension of x_j .

$$d(x_j, x_q) = \sum_{c=1}^p \left\{ 1 - \cos(\alpha\pi(x_j(c) - x_q(c))) \right\}. \tag{6}$$

More specifically, the procedure for obtaining Equation (6) is given as follows. First, x_j is mapped from a p -dimensional real space to a complex reproducing kernel Hilbert space (RKHS).

$$z_j = \frac{1}{\sqrt{2}} \begin{bmatrix} e^{i\alpha\pi x_j(1)} \\ \vdots \\ e^{i\alpha\pi x_j(p)} \end{bmatrix}. \tag{7}$$

The relationship between z_j and $d(x_j, x_q)$ is given as follows.

$$\begin{aligned} d(x_j, x_q) &= \|z_j - z_q\|^2 = \\ &= \frac{1}{2} \left\| \left(\cos(\alpha\pi x_j) + i \sin(\alpha\pi x_j) \right) - \left(\cos(\alpha\pi x_q) + i \sin(\alpha\pi x_q) \right) \right\|^2. \tag{8} \\ &= \sum_{c=1}^p \left\{ 1 - \cos(\alpha\pi(x_j(c) - x_q(c))) \right\} \end{aligned}$$

In Equation (8), the distance between data points is a cosine-based metric calculated in complex space, but it actually represents a real distance. Because the range of the cosine function varies on the interval $[-1, 1]$, the influence of noise characteristics on the distance is reduced to a certain extent compared with the Euclidean distance, without increasing the data complexity and data dimension. The Euler cosine metric defined in this paper is defined as follows in this paper.

$$d(x_i, x_j) = \sum_{c=1}^p \left\{ 1 - \cos(\alpha\pi w_c(x_i(c) - x_j(c))) \right\}. \tag{9}$$

In Equation (9), d represents the number of attributes of the sample point, and α is an adjustment coefficient. In this paper, α takes on a value of 1, and w_c is the weight of each dimension attribute in each sample point. If the dispersion of the m attribute is greater than n in the dataset, it is necessary to assign a larger weight of m attribute to adjust the attribute space. This step is essential to accurately reflect the similarity measurement among data samples.

The coefficient of variation can describe the degree of dispersion of the data. Affected by the idea of coefficient of variation, to reduce the influence of the attribute dispersion degree on the distance between sample points, we apply Equation (10) to adjust the weight of each feature of the sample points. In this idea, the weight of p attributes of any two sample points is considered to be the same, and thus only p values in the data need to be calculated. Let $w = \{w_1, w_2, \dots, w_c\}$, and w is the set of weights in the attribute set.

$$w_c = \frac{s_j}{x_c} = \frac{1}{x_c} \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ic} - \bar{x}_c)^2}. \tag{10}$$

In Equation (10), w_c represents the weight for c -th at-

tributes in the sample point, n is the total number of sample points. \bar{x}_c represents the mean of all sample points in the c -th dimension attributes, and x_{ic} represents the value corresponding to the c -th dimension attributes in the i -th sample points. The new Euler cosine distance not only considers the weight of different attributes in the sample points, but also considers the noise influence of high-dimensional data.

2 Adaptive local density. In the concept of using k -nearest neighbours to optimize local density, the local density considers the equivalent effect of the k -nearest neighbours around the data point i . Because the k -th nearest neighbour of data point i is susceptible to noise points, it easily causes errors in the local density of the boundary points. As shown in Figure 2, point a and point b belong to different clusters, whereas point a and point c belong to the same cluster. When calculating the local density in combination with k -nearest neighbours, points a and b tend to consider each other as k -nearest neighbours, and their local densities are closer. Therefore, the probability that point a and point b are misclassified into the same class is greater than that of point a and point c .

Based on the proposed idea of combining local density with k -nearest neighbours, a new adaptive k -nearest neighbour local density is proposed. The algorithm uses the adaptive neighbourhood local density, which emphasizes the flexibility of local density and corresponds to one scale parameter for each sample. This idea overcomes the limitations of the single global

scale parameter in the traditional local density formula, thus achieving automatic selection of scale parameters.

Intuitively, $d(x_a, x_b) < d(x_a, x_c)$, and the sum of the distances of the k neighbours of point a and point

b is equal, $w(i, j) = \frac{d(x_i, x_j)}{\sum_{x_j \in knn(x_i)} d(x_i, x_j)}$, where $w(i, j)$ is

less than 1. For a, b and c , $w(a, b) < w(a, c)$. Then de-

fine the adaptive scale as $\sigma = 1 - \ln \frac{d(x_i, x_j)}{\sum_{x_j \in knn(x_i)} d(x_i, x_j)}$,

and thus, $1 - \ln(w(a, b)) > 1 - \ln(w(a, c))$. The value of the adaptive scale σ is greater than 1. Obviously, this distance adjustment method can increase the distance between clusters of different densities, which is beneficial to the algorithm in distinguishing different clusters. The new adaptive scale local density formula is written as follows.

$$\rho_i = \sum_{x_j \in knn(x_i)} \exp \left(- \left(1 - \ln \frac{d(x_i, x_j)}{\sum_{x_j \in knn(x_i)} d(x_i, x_j)} \right) d(x_i, x_j) \right). \quad (11)$$

In Equation (11), $d(\cdot)$ represents the distance between two sample points, k is the number of neighbours of one point, and $knn(x_i)$ represents the set of k -nearest neighbours of the x_i point. Equation (11) considers both the distribution information on the k -nearest neighbours and the state of the distance between two points in all k -nearest neighbours. This approach can better represent the local density information on a point without searching the global data points.

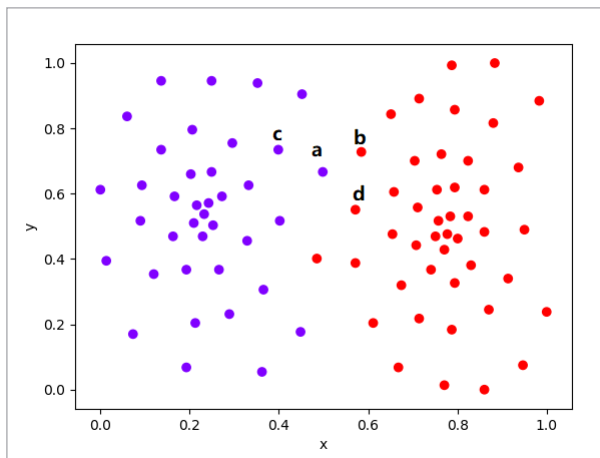
3 New allocation strategy. The new allocation strategy uses a two-step allocation strategy and introduces the reverse k -nearest neighbour. The reverse k -nearest neighbour of i is defined as follows:

$$N_{rk}(i) = \{o \mid o \in D, i \in N_k(o)\}. \quad (12)$$

Similar to the friend relationship model, the more people that are friends with others, the more popular the person. Therefore, the more neighbours of a point as another point, the more concentrated this point is in the sample point. Therefore, this point is not an isolated point. High-density points generally exist in data points of more reverse neighbours. First, based

Figure 2

Example of boundary points



on the previously obtained distance matrix (Eulerian cosine distance), we can obtain the k -nearest neighbour matrix of each data point and further calculate the number of reverse k -nearest neighbours of each data point. We define the number of reverse k -nearest neighbours as $rKnn[i]$, $i = 0, \dots, n$. where (n is the number of sample points), if $i \in N_k(o)$, $o \in D$, $N_k(o)$ is the k -nearest neighbour set of o , thus $rKnn[i] = rKnn[i] + 1$, where K is the number of loops. We define the global average reverse k -nearest neighbour number as $avgrknn$.

$$avgrknn = \frac{1}{n} \sum_{i=0}^n rknm[i]. \quad (13)$$

The two-step allocation strategy in this paper states that if the number of reverse k -neighbours of point i is greater than or equal to $avgrknn$, then allocation strategy 1 is adopted, and if the number of reverse k -nearest neighbours of point i is less than $avgrknn$, allocation strategy 2 is adopted. Allocation strategy 1 is based on the reverse k -nearest neighbour number and uses the breadth-first search method to start from the point of the largest reverse k -nearest neighbour number. If there is a labelled sample point in its reverse k -nearest neighbour, the label is propagated to the point. Allocation strategy 2 uses the density deviation degree of each sample point as the probability of label propagation. The propagation probability calculation formula is shown in Equation (14). When the value of P_i is greater than or equal to 1, the degree of deviation indicating the point i is relatively low. The closest point of the label is propagated to point i in the inverse k -nearest of point i .

$$P_i = \frac{\rho_i}{\frac{1}{k} \sum_{j \in knm(i)} \rho_j}. \quad (14)$$

4.2. Main Step Description of ERK-DPC Algorithm

This section describes the process of the ERK-DPC algorithm and the process of the two-step allocation strategy. The ERK-DPC algorithm follows the basic concept of the traditional DPC algorithm but optimizes and improves its key steps. The entire process is still divided into four steps: Calculate the distance

matrix, measure ρ_i and δ_i , select the cluster centres and assign the non-cluster centre points. The specific algorithm flow is described as follows.

Algorithm 1: ERK-DPC algorithm

Input: Data set of $X \in \mathfrak{R}_{n \times m}$, n sample points and m attributes; number of neighbours k

Output: Clustering results with labels $Y \in \mathfrak{R}_{n \times 1}$

Begin algorithm

Step1: All sample points are normalized

Step2: Calculate distance matrix according to Equation (9)

Step3: Calculate ρ_i for point i according to Equation (11)

Step4: Calculate δ_i for point i according to Equation (12)

Step5: Plot decision graph and select cluster centers

Step6: Calculate the reverse k -nearest neighbors for each point, Calculate the $avgrknn$ according to Equation (13)

Step7: Apply Algorithm 2 to remaining point where reverse k -nearest neighbors is greater than $avgrknn$

Step8: Apply Algorithm 3 to point where reverse k -nearest neighbors is less than $avgrknn$

Step9: The remaining unallocated points are regarded as noise points and allocated to the cluster of the nearest allocated points.

Step10: Return y

End algorithm

The following is a description of Algorithm 2.

Algorithm 2: Assign reverse k -nearest neighbour to a point greater than $avgrknn$

Input: set of centers $C_r = \{C_1, C_2, \dots, C_m\}$, number of neighbours k , distance matrix.

Output: preliminary result $M = \{C_1, C_2, \dots, C_m\}$

Begin algorithm

1. Select a cluster center point C_m in C_r and mark it as visited

2. Put the $knn(C_m)$ into an empty queue L , and sort the $rknm(C_m)$ in the queue from large to small.

3. While L is not empty

4. Select the first point X_p in L

5. If $cluster(X_p) \neq -1$ and $rknm(X_p) > avgrknn$

6. Put the marked points in into list L

```

7.   If  $L$  is not null:
8.     Assign the label closest to  $X_p$  in  $L$  to  $X_p$ 
9.     Else no operation
10.    End if
11.  End if
12.  Inserting unclassified point in  $knn(X_p)$  into queue  $L$ 
13.  Remove the  $X_p$  in the queue and repeat execution
    from line 5
14.  End while
15.  If all points  $r_{knn}(i) < avgrknn$ 
16.  Break
17.  Return  $M$ 
End algorithm

```

The assignment process of data points with low reverse k -nearest neighbours is included in Algorithm 3.

Algorithm 3: Assign reverse k -nearest neighbour to a point less than $avgrknn$

Input: preliminary result $M=\{C_1, C_2, \dots, C_m\}$, number of neighbours k .

Output: Final result $R=\{r_1, r_2, \dots, r_m\}$

Begin algorithm

```

1.  Sort the reverse  $k$ -nearest neighbours of the
    remaining unallocated points from large to small and
    put them in a queue  $L$ 
2.  While  $L$  is not empty
3.    Select the first point  $X_p$  in  $L$ 
4.    Calculate  $P_{xp}$  according to Equation (14)
5.    If  $cluster(X_p) \neq -1$  and  $P_{xp} > 1$ 
6.      Put the marked points in  $knn(X_p)$  into list  $L$ 
7.      If  $L$  is not null:
8.        Assign the label closest to  $X_p$  in  $L$  to  $X_p$ 
9.        Else no operation
10.       End if
11.     End if
12.     Remove  $X_p$  from the queue
13.  End while
14.  Return  $R$ 
End algorithm

```

4.3. Time Complexity of ERK-DPC Algorithm

This section primarily analyses the time complexity of the ERK-DPC algorithm, and the time complexity depends on main three components: (1) the time needed to calculate the distance between sample points, (2) the time needed to measure the local density of ρ_i and the cluster centre distance δ_i , and (3) the time needed to assign the non-clustered centre

points. The size of sample points is n , the number of cluster centres is c , and k is the number of neighbours. The algorithm proposed in this paper is analysed according to Algorithm 1 as follows.

Step1: The time complexity of standardizing the initial data is the required $O(n)$.

Step2: The Euler cosine distance is computed to form a distance matrix between the sample points, the required time complexity of which is $O(n^2)$. In other words, it costs $O(n)$ to calculate the weight of each dimension attribute. Thus, the overall time complexity of calculating the sample point distance is $O(n^2)$.

Step3: The local density ρ_i of each point is calculated. The time complexity requires $O(n)$ to filter the k -nearest neighbours of each point. Second, the time complexity of n points is $O(n^2)$.

Step4: The time complexity required to calculate δ_i is completed in Step3.

Step5: To obtain the cluster centres, ρ_i and δ_i are sorted, the time complexity of which is $O(n \log(n))$.

Step6: The calculation of the reverse k -nearest neighbour number of each sample point is $O(nk)$.

Step7: The time complexity is $O(n_i k)$, applying allocation strategy 1, and n_i is the number of points, where the inverse k -nearest neighbour number is greater than $avgrknn$.

Step8: The allocation strategy 2 supplies $O(n_2 k)$, where $n_2 = n - n_1$.

Therefore, the total complexity is $O(n) + O(n^2) + O(n^2) + O(n \log(n)) + O(nk) + O(n_i k) + O(n_2 k)$, which can be approximated as $O(n^2)$. The ERK-DPC algorithm has the same complexity as the traditional DPC algorithm.

5. Experiments and Analysis

To prove the effectiveness of the proposed algorithm, this paper uses artificial synthetic datasets [12] and UCI real-word datasets [20] for experimental testing and evaluation. The comparison algorithms include FKNN-DPC [34], traditional DPC [27], DBSCAN [11], K -means [25] and AP [13]. The ERK-DPC algorithm proposed in this paper and the four comparison algorithms are implemented in Python language. The results shown are the optimal results after parameter adjustment. Because we did not obtain the source code of the FKNN-DPC algorithm, we implemented the process by referring to the original paper. The pa-

parameter of the FKNN-DPC algorithm to be adjusted is the number of k -nearest neighbours. The traditional DPC algorithm and DBSCAN were implemented using the source code supplied by the original author. In this paper, we did not consider the *halo* portion of the DPC algorithm, and the parameter to be adjusted is the cut-off distance. The two parameters that DBSCAN must adjust are ε and *minpts*, where ε is a floating point number, and *minpts* is an integer. K -means and AP are implemented in the sklearn library [26] of Python. K -means only needs to determine the correct clusters, and AP algorithms also have only one adjustable parameter in the sklearn library, known as “preference”. The synthetic data sets and UCI data sets used in the experiments are described in Tables 1 and 2. In this paper, the ERK-DPC algorithm and the comparison algorithm are all run on Python 3.6.3. The installation configuration environment is an *Inter(R) Xeon(R)*, with *2.10 GHz CPU* and *64.0 GB* running *RAM*.

5.1. Data Preprocessing and Evaluation Metrics

Before the experiment, the data were preprocessed using the normalization method. Each attribute value of the data is mapped to the interval $[0, 1]$ to eliminate the impact of the different dimensions, as shown in Equation (15).

$$x^* = \frac{x - \min}{\max - \min}. \quad (15)$$

To validate the effectiveness of the proposed algorithm, four well-known external evaluation metrics were used in the synthetic datasets and the UCI datasets. These measures are accuracy(ACC), adjusted mutual information(AMI), adjusted rand-index(ARI) and normalized mutual information(NMI) [31]. All evaluation metrics need to know the results of real clustering in advance. The value range of NMI is $[0, 1]$. The range of values for ACC, ARI and AMI are $[-1, 1]$. Evaluation metrics are used to measure the degree of agreement between the two data distributions. The upper bounds of the evaluation metrics are 1, and the larger the value, the better the clustering result.

5.2. Results and Analysis on Synthetic Datasets

This paper selects several representative synthetic datasets for visualization points of view, namely, the Aggregation, Flame, R15, Spiral, D31 and S1 data sets.

Table 1

Synthetic datasets used in the experiments

Datasets	Objects	Attribute	Cluster
Aggregation	788	2	7
Flame	240	2	2
Spiral	312	2	3
R15	600	2	15
D31	3100	2	31
DIM512	1024	512	16

Table 2

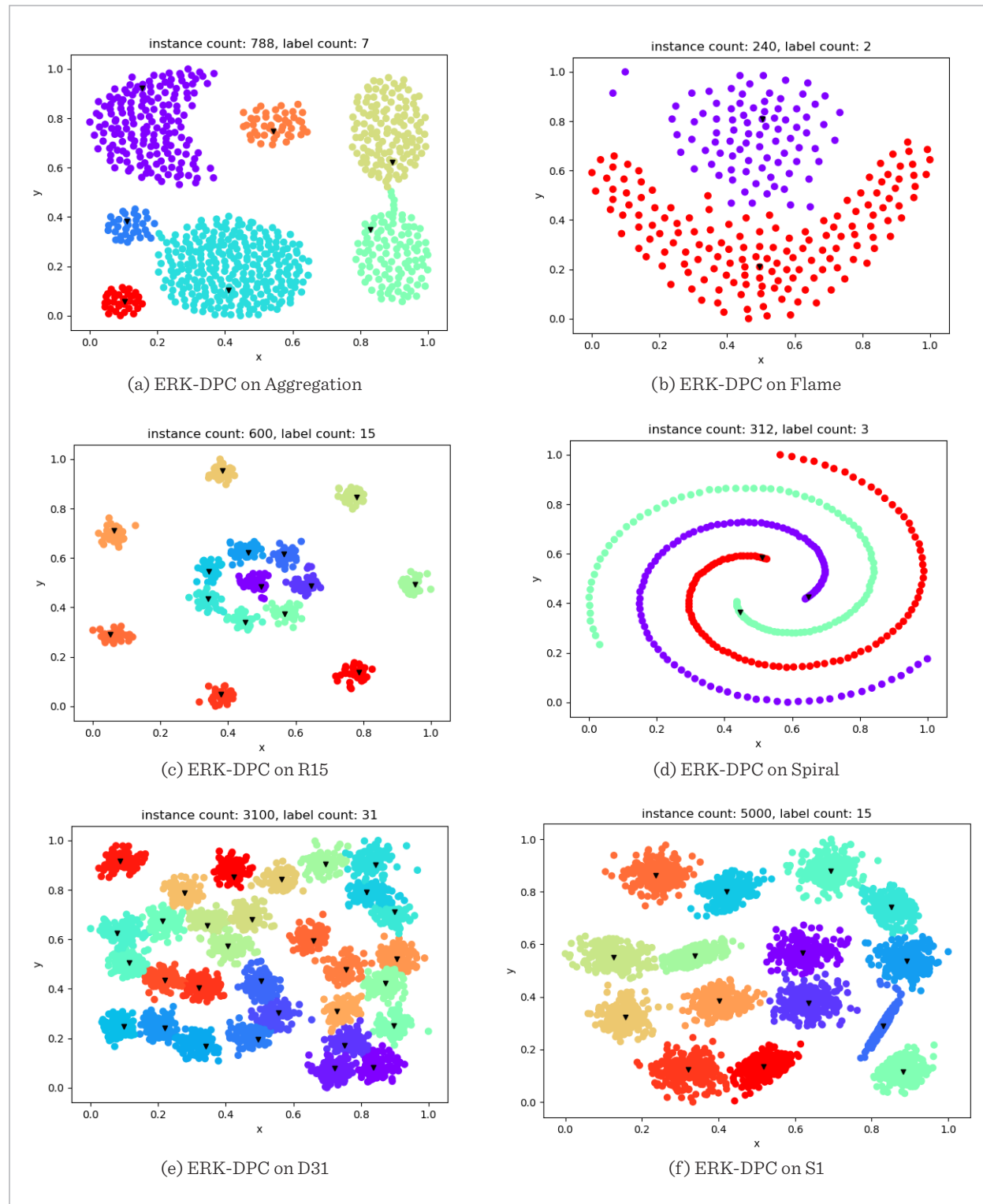
UCI datasets used in the experiments

Datasets	Objects	Attribute	Cluster
Iris	150	4	3
Balancescale	625	4	3
Ecoli	336	8	8
Glass	214	9	6
Breast	271	9	2
Vote	435	16	2
Zoo	101	19	7
Parkinson	195	23	2
Wpbc	198	33	2
Sonar	208	60	2
Leave	1600	64	100
Libras movement	360	90	15

These data sets differ in data distribution, number of attributes, number of samples, number of categories, etc. Many experiments have been run on each synthetic dataset. Figure 3 shows the clustering results of several synthetic datasets on a two-dimensional graph. The black triangles in the ERK-DPC algorithm results graph represent the cluster centre points of each cluster. The ERK-DPC algorithm does not consider the influence of noise points. The different colours in the resulting graph represent a cluster. The ERK-DPC, FKNN-DPC and DPC algorithms determine the number of clusters according to the distribution of points in the decision graph. DBSCAN and AP do not need to determine the number of clusters in advance. This paper directly determines the correct number of clusters of the K -means algorithm.

Figure 3

The clustering results by ERK-DPC algorithms



The Aggregation dataset consists of seven unbalanced clusters of different sizes, two of which have connection points. The identification of the connection points and boundary points presents difficulty in clustering. Figure 3(a) shows that the ERK-DPC algorithm not only correctly identifies the seven clustering centres but also better processes the connection points of four clusters. The Flame dataset contains two clusters of different sizes and shapes, one of which is a manifold shape. Figure 3(b) clearly illustrates that the proposed ERK-DPC method can correctly separate the two clusters and that the data points at the junction of the two clusters are correctly identified. In Figure 3(c), the R15 data set contains 15 clusters, and the seven clusters of the periphery are located far away from each other, and thus it is easy to distinguish them correctly. However, overlap exists between the eight clusters of the middle. The ERK-DPC algorithm can correctly identify 15 clustering centres, but several overlapping data points have not been correctly separated. The Spiral dataset con-

tains three circular nested clusters. Because the three rings are nested with each other, it is easy to divide the different classes together in the traditional DPC algorithm. In Figure 3(d), the ERK-DPC algorithm can correctly identify the centre point and correctly allocate it. D31 contains 31 clusters, and there are overlapping components in the clusters. Figure 3(e) shows that the ERK-DPC can completely find the cluster centres of 31 clusters, and the cluster centre of each cluster is located. The middle position of the cluster is also ideal for the distribution of the overlapping portions of the cluster and the surrounding noise points. For the S1 dataset shown in Figure 3, the ERK-DPC algorithms correctly identify all clusters. The S1 dataset includes many boundary points and overlapping points.

Table 3 shows a comparison of the evaluation metrics for several algorithms in the synthetic datasets. The best results of each evaluation metric are shown in bold, and Par is the corresponding parameter when the result is optimal.

Table3

Comparison of evaluation metrics on synthetic datasets

Algorithm	ACC	AMI	ARI	NMI	Par	ACC	AMI	ARI	NMI	Par
	Aggregation					Flame				
ERK-DPC	0.995	0.991	0.994	0.993	19	1.000	1.000	1.000	1.000	25
FKNN-DPC	0.996	0.992	0.992	0.988	7	0.996	0.962	0.983	0.963	6
DPC	1.000	1.000	1.000	1.000	3.4	1.000	1.000	1.000	1.000	2.8
DBSCAN	0.985	0.955	0.979	0.968	1.5/8	0.958	0.783	0.923	0.851	1/6
K-means	0.786	0.833	0.762	0.879	3	0.838	0.386	0.453	0.399	3
AP	0.841	0.683	0.665	0.763	-2.56	0.775	0.367	0.441	0.454	-5.25
	Spiral					D31				
ERK-DPC	1.000	1.000	1.000	1.000	2	0.956	0.955	0.934	0.957	20
FKNN-DPC	1.000	1.000	1.000	1.000	4	0.953	0.949	0.947	0.957	10
DPC	1.000	1.000	1.000	1.000	2.0	0.956	0.955	0.936	0.957	0.6
DBSCAN	1.000	1.000	1.000	1.000	2.5/2	0.864	0.849	0.709	0.879	0.6/9
K-means	0.343	0.006	0.005	0.006	3	0.711	0.529	0.304	0.681	31
AP	0.387	-0.005	-0.005	0.004	-2.08	0.631	0.417	0.192	0.579	0.65
	R15					DIM512				
ERK-DPC	0.997	0.994	0.993	0.994	15	1.000	1.000	1.000	1.000	4
FKNN-DPC	0.996	0.994	0.993	0.994	3	1.000	1.000	1.000	1.000	4
DPC	0.997	0.994	0.993	0.994	0.6	1.000	1.000	1.000	1.000	2.0
DBSCAN	0.953	0.929	0.933	0.944	0.4/8	0.985	0.976	0.982	0.983	2.0/4
K-means	0.997	0.994	0.993	0.994	15	1.000	1.000	1.000	1.000	16
AP	0.925	0.931	0.988	0.983	-0.16	0.974	0.945	0.921	0.962	-1.07

Table 4

Comparison of evaluation metrics on UCI datasets

Algorithm	ACC	AMI	ARI	NMI	Par	ACC	AMI	ARI	NMI	Par
	Iris					Ecoli				
ERK-DPC	0.967	0.883	0.904	0.885	4	0.821	0.633	0.738	0.711	3
FKNN-DPC	0.934	0.809	0.818	0.813	5	0.747	0.584	0.707	0.659	4
DPC	0.907	0.759	0.793	0.806	2.0	0.494	0.531	0.405	0.582	0.3
DBSCAN	0.667	0.651	0.562	0.713	0.7/3	0.677	0.412	0.507	0.506	0.2/9
K-means	0.893	0.826	0.730	0.758	3	0.584	0.515	0.454	0.598	8
AP	0.82	0.571	0.548	0.68	-0.25	0.623	0.411	0.307	0.512	-0.15
	Balancescale					Glass				
ERK-DPC	0.426	0.226	0.208	0.230	8	0.411	0.328	0.175	0.357	13
FKNN-DPC	0.342	0.091	0.118	0.102	7	0.383	0.176	0.172	0.263	6
DPC	0.007	0.000	-0.001	0.007	0.3	0.277	0.177	0.292	0.323	6
DBSCAN	0.006	-0.001	0	0	0.04/2	0.294	0.273	0.231	0.356	0.2/3
K-means	0.141	0.129	0.160	0.143	3	0.313	0.288	0.166	0.323	6
AP	0.159	0.097	0.141	0.159	-0.15	0.356	0.162	0.165	0.356	1.57
	Vote					Sonar				
ERK-DPC	0.901	0.545	0.643	0.554	12	0.659	0.075	0.096	0.086	4
FKNN-DPC	0.882	0.483	0.585	0.492	11	0.563	0.004	0.011	0.009	2
DPC	0.848	0.409	0.484	0.418	2.0	0.620	0.041	0.053	0.045	4
DBSCAN	0.483	0.315	0.410	0.391	0.8/3	0.409	0.073	0.004	0.121	0.5/2
K-means	0.866	0.459	0.536	0.469	2	0.552	0.005	0.006	0.005	2
AP	0.503	0.388	0.418	0.343	-0.61	0.417	0.017	0.022	0.025	-0.42
	Zoo					Libras movement				
ERK-DPC	0.878	0.798	0.842	0.829	4	0.403	0.601	0.351	0.662	4
FKNN-DPC	0.757	0.675	0.563	0.757	10	0.367	0.549	0.324	0.367	3
DPC	0.604	0.658	0.497	0.722	2.0	0.346	0.485	0.261	0.581	2.0
DBSCAN	0.249	0.220	0.145	0.615	0.5/1	0.279	0.359	0.235	0.658	0.9/1
K-means	0.753	0.765	0.679	0.816	7	0.365	0.534	0.315	0.599	15
AP	0.561	0.494	0.578	0.561	2.65	0.257	0.281	0.147	0.438	4.06
	Wpbc					Breast				
ERK-DPC	0.757	0.037	0.138	0.048	2	0.736	0.058	0.154	0.072	3
FKNN-DPC	0.737	0.013	0.099	0.027	4	0.718	0.048	0.135	0.055	2
DPC	0.676	0.019	-0.006	0.030	2.1	0.711	0.042	0.122	0.014	0.1
DBSCAN	0.459	0.025	0.044	0.042	0.6/4	0.438	0.035	0.083	0.065	0.8/1
K-means	0.601	0.020	0.035	0.027	2	0.509	-0.001	-0.003	0.001	2
AP	0.269	0.015	0.012	0.026	-0.96	0.671	0.036	0.057	0.067	1.15
	Parkinson					Leave				
ERK-DPC	0.851	0.273	0.391	0.351	3	0.308	0.510	0.189	0.724	8
FKNN-DPC	0.825	0.168	0.324	0.206	6	0.222	0.423	0.144	0.628	5
DPC	0.635	0.047	0.144	0.053	2.5	0.212	0.466	0.194	0.681	0.1
DBSCAN	0.754	0.097	0.164	0.159	0.15/3	0.127	0.243	0.029	0.500	0.15/2
K-means	0.669	0.213	0.052	0.213	2	0.299	0.507	0.282	0.710	100
AP	0.703	0.053	0.043	0.115	0.23	0.152	0.274	0.036	0.571	3.25

From the above results, it can be concluded that the proposed method performs well and can correctly cluster different types of data sets. The ERK-DPC algorithm not only correctly classifies the complex data and ring data but also better addresses the connection points and boundary points. The comparison results show that the ERK-DPC algorithm in this paper is only slightly lower than the traditional DPC and FKNN-DPC in Aggregation and D31, and the evaluation metrics in other data sets are not lower than the other four algorithms. In this paper, the ERK-DPC algorithm obtains 100% clustering results for the Flames, spiral and DIM512 datasets. These results indicated that the Euler cosine distance proposed in this paper has a good clustering effect on the two-dimensional data set and also has obvious advantages in high-dimensional data.

5.3. Results and Analysis on UCI Datasets

In this section, to further test the performance of the ERK-DPC algorithm, this paper selects 12 real data sets (as shown in Table 2) from the UCI database for experiments, in an attempt to obtain instructive effects. These datasets differ in the number of sample points, feature number and cluster number.

We compare the results of the cluster evaluation metrics of the five algorithms. ACC, AMI, ARI and NMI were used to evaluate the clustering results of real data sets. Table 4 shows the ERK-DPC algorithm and the clustering results of the comparison algorithms FKNN-DPC, DPC, DBSACN, *K*-means and AP on 12 UCI datasets. Par represents the parameters corresponding to the optimal results. The best results of each evaluation metric are shown in bold. The results show that the proposed

method achieved good results in most cases.

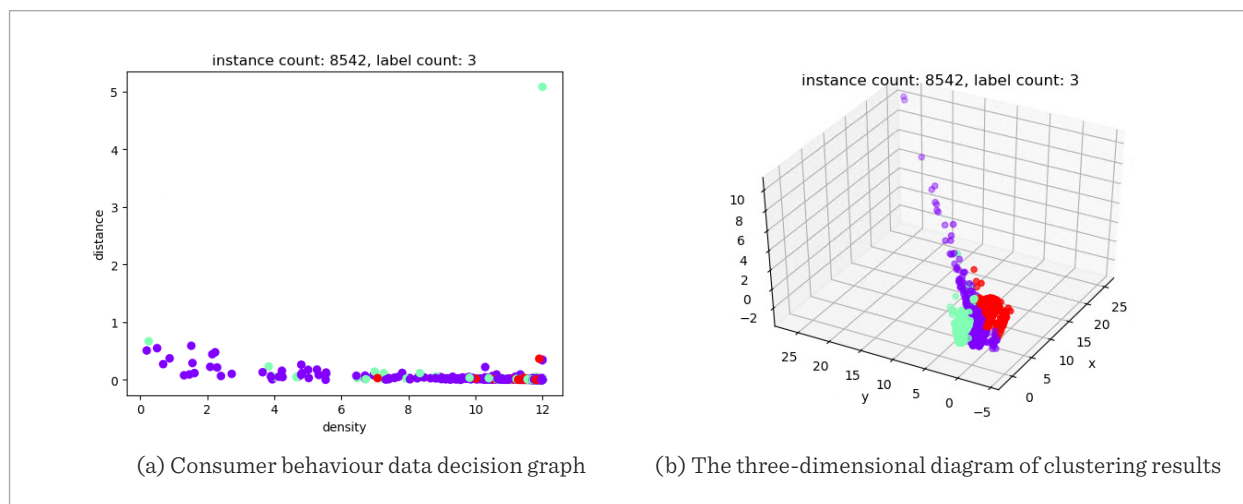
ERK-DPC outperforms other algorithms in low dimensional datasets, such as Iris, Ecoli, Balancescale and Glass. Iris is the data set most commonly used in clustering analysis and contains three class clusters, two of which are non-linearly separable. This algorithm can correctly distinguish the second and third clusters. However, this algorithm has better performance on relatively high-dimensional data in the Sonar, Leave, and Libras Movement datasets. Specifically, the higher the feature value of the datasets, the greater the advantage of ERK-DPC. Leave contains 100 class clusters, each with 64 characteristics. Each cluster has a small number of samples, which creates certain difficulties in clustering. However, in the Leave data, the algorithm proposed in this paper is superior to the other algorithms. Overall, the ERK-DPC algorithm achieved more satisfactory results on most data sets than other algorithms.

5.4. Analysis of Experimental Results of Consumer Behaviour Data

The ERK-DPC clustering algorithm is applied to the consumer behaviour segment. Consumption data are obtained by scanning a two-dimensional code during promotional activities. The data from January 1, 2018 to March 15, 2018 are selected. In this period of time, consumers display repeated consumption behaviour. The number of data sample points is 8542, and five features are selected: consumption amount, consumption frequency, recent consumption time, consumer scan code longitude and latitude.

It can be determined from the decision graph of the consumption data in Figure 4(a) that the data can be

Figure 4



grouped into three categories. Figure 4(b) presents a three-dimensional representation of the clustering results.

Because there is no known clustering label for consumption data, this paper uses an internal evaluation indicator. The Silhouette coefficient applies to situations in which the actual category information is unknown. The formula can be expressed as

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

For a single data sample point, a represents the distance of a sample point from other samples of its class, and b is the average distance of samples in the different categories from which it is closest. For a dataset, its Silhouette coefficient is the average of the Silhouette coefficient of all sample points, and the range is $[-1, 1]$. The closer the distance between the same category sample and the farther the distance between the different types of samples, the larger the Silhouette coefficient value.

Figure 5 shows the density relationship of consumer data attributes among three clusters, where M is consumption amount, F is consumption frequency, and T is recent consumption time. In Figure 5(a), the consumer's recent consumption time is relatively close to the current date, and the consumption amount and consumption frequency in the high-density area are relatively large. This result shows that consumers in this category are valuable users. Enterprises need to focus on the objects. In Figure 5(b), the consumption time period is relatively large, but the consumption amount and consumption frequency are lower. This result shows that the user repurchase rate of this category is relatively high, but it is a general value user. In Figure 5(c), the recent consumption time is far from the current value, and the consumption amount and consumption frequency are not too high, indicating that the users in this category are losing users or are about to lose users.

Table 5 compares the evaluation metric for several algorithms on the consumption data. It can be observed that the EKP-DPC algorithm has better Silhouette coefficient scores in the consumption behaviour data than other algorithms. In summary, the EKP-DPC algorithm effect is better.

Figure 5

Three attribute density analysis graphs in consumer behaviour data

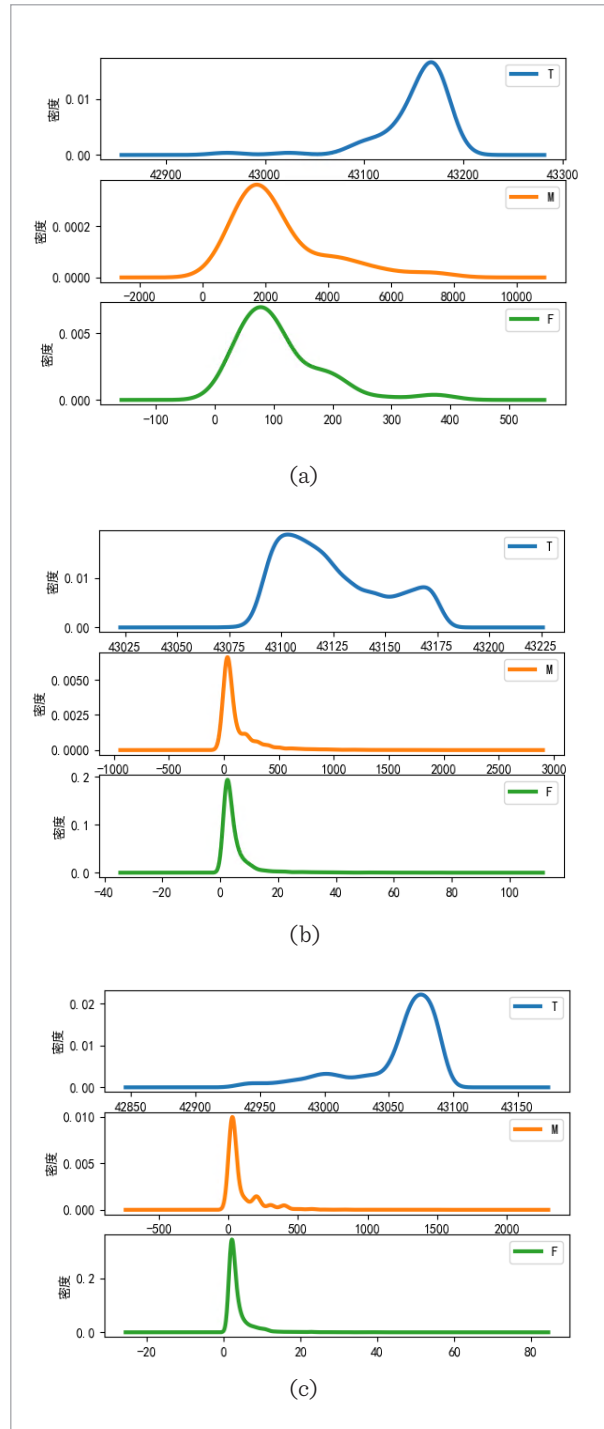


Table 5

Performance comparison of algorithms on consumption data

Algorithm	Silhouette coefficient	Par	Times(s)
ERK-DPC	0.6057	7	646.3344
FKNN-DPC	0.5028	7	690.0486
DPC	0.1865	0.1	771.0434
DBSCAN	0.4149	0.15/3	519.4163
K-means	0.6005	3	4.2433
AP	0.4025	1.83	602.7957

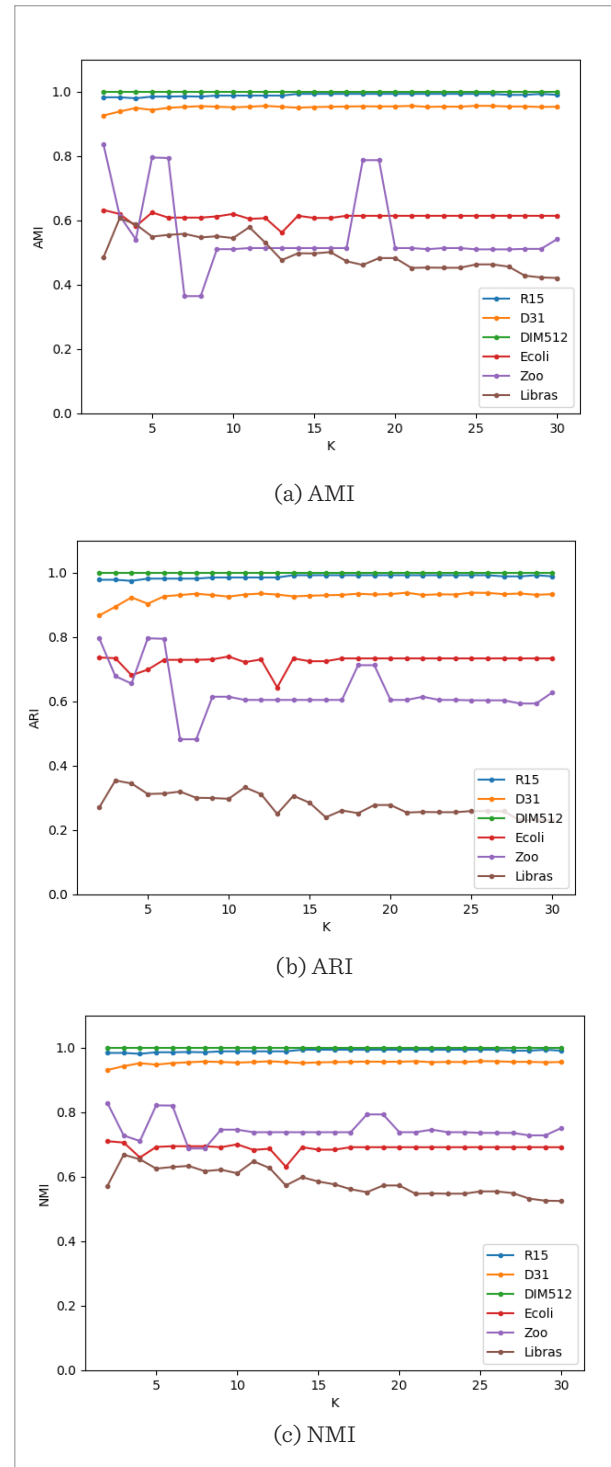
5.5. Sensitivity Analysis

In this section, we discuss the influences of the k parameter.

The k parameter can be adjusted to the resulting optimization in the ERK-DPC algorithm. To analyse the influence of the selection of the parameter k on the clustering result, different k values are selected for the result analysis. The range of k is $[2, 30]$, and k is incremented by one each time. As shown in Figure 6, the results are compared with different dataset experiments using the evaluation criteria, including AMI, ARI, and NMI. The datasets of Aggregation, R15 and D31 in the synthetic datasets and certain real-world datasets are selected. The dimensions of the datasets Ecoli, Zoo and Libras are 8, 19 and 90, respectively. As can be observed from Figure 6, the three synthetic data sets show little fluctuation with the change of k value. In the UCI data set, certain fluctuations occur in the values of the indicators. In Figure 6(a) and (b), the Zoo data set fluctuates slightly when the k value is small, but when k is greater than 20, the three index values to tend to be stable. The datasets of Ecoli and Libras show little fluctuation as k increases, which indicates that the ERK-DPC algorithm has robustness in both low-dimensional and high-dimensional data. This result verifies the robustness of the algorithm.

6. Conclusions and Future Work

In this paper, an adaptive density peak clustering based on dimension-free and reverse k -nearest neighbours is proposed. In the process of clustering, the idea of reverse

Figure 6Result on different datasets with different k argument

k -nearest neighbours is introduced to further optimize the one-step allocation strategy in the traditional DPC algorithm (known as ERK-DPC). The improved component of the method is divided into three main aspects. First, because the high-dimensional data will be greatly affected, and the traditional Euclidean distance cannot measure the distance between sample points correctly, and thus this paper introduces a novel Euler cosine distance formula. The Euler cosine distance formula can measure the distance more accurately without reducing the data dimension (dimension-free). This distance formula can avoid the effects of noise and sparsity in high dimensional data, and can effectively represent the true distance between sample points. Second, an adaptive local density formula is that can better solve the problem of local density calculation

in the boundary points of clusters. Third, aimed at the problem of continuous error in the one-step allocation strategy of the traditional DPC algorithm, a novel two-step allocation strategy based on the number of reverse k -nearest neighbours is proposed. To evaluate the effectiveness of the proposed method, a large number of experiments were conducted on synthetic datasets and UCI real-world datasets. The results demonstrate the effectiveness and robustness of the proposed method. The ERK-DPC method is applied to consumer behaviour data, which proves that the method can be used to effectively segment consumers.

To obtain the distance matrix, calculation of the distance between two pairs of sample points is required, which limits the algorithm to be run on large datasets. In future work, we will attempt to reduce the time complexity and conduct incremental clustering.

References

1. Ankerst, M., Breunig, M. M., Kriegel, H. P., Sander, J. OPTICS: Ordering Points to Identify the Clustering Structure. ACM SIGMOD International Conference on Management of Data, Philadelphia, Pa, Jun 01-03, 1999, 49-60. <https://doi.org/10.1145/304182.304187>
2. Bai, X., Yang, P., Shi, X. An Overlapping Community Detection Algorithm Based on Density Peaks. *Neurocomputing*, 2017, 226(22), 7-15. <https://doi.org/10.1016/j.neucom.2016.11.019>
3. Birant, D., Kut, A. ST-DBSCAN: An Algorithm for Clustering Spatial-Temporal Data. *Data & Knowledge Engineering*, 2007, 60(1), 208-221. <https://doi.org/10.1016/j.datak.2006.01.013>
4. Bie, R., Mehmood, R., Ruan, S., Sun, Y., Dawood, H. Adaptive Fuzzy Clustering by Fast Search and Find of Density Peaks. *Personal and Ubiquitous Computing*, 2016, 20(5), 785-793. <https://doi.org/10.1007/s00779-016-0954-4>
5. Chang, M., Chen, L., Hung, L., Rossmann, P., Wu, G. Exact Algorithms for Problems Related to the Densest K -set Problem. *Information Processing Letters*, 2014, 114(9), 510-513. <https://doi.org/10.1016/j.ipl.2014.04.009>
6. Chen, G., Zhang, X., Wang, Z., Li, F. Robust Support Vector Data Description for Outlier Detection with Noise or Uncertain Data. *Knowledge-Based Systems*, 2015, 90, 129-137. <https://doi.org/10.1016/j.knsys.2015.09.025>
7. Deng, C., Song, J., Sun, R., Cai, S., Shi, Y. GRIDEN: An Effective Grid-Based and Density-Based Spatial Clustering Algorithm to Support Parallel Computing. *Pattern Recognition Letters*, 2017, 109(15), 81-88. <https://doi.org/10.1016/j.patrec.2017.11.011>
8. Ducournau, A., Bretto, A., Rital, S., Laget, B. A Reductive Approach to Hypergraph Clustering: An Application to Image Segmentation. *Pattern Recognition*, 2012, 45(7), 2788-2803. <https://doi.org/10.1016/j.patcog.2012.01.005>
9. Du, M., Ding, S., Jia, H. Study on Density Peaks Clustering Based on K -Nearest Neighbors and Principal Component Analysis. *Knowledge-Based Systems*, 2016, 99, 135-145. <https://doi.org/10.1016/j.knsys.2016.02.001>
10. Du, M., Ding, S., Xu, X., Xue, Y. Density Peaks Clustering Using Geodesic Distances. *International Journal of Machine Learning and Cybernetics*, 2017, 9(8), 1335-1349. <https://doi.org/10.1007/s13042-017-0648-x>
11. Ester, M., Kriegel, H. P., Sander, R., Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, AAAI Press, Portland, Oregon, 1996, 226-231.
12. Fränti, P., Sieranoja, S. K -means Properties on Six Clustering Benchmark Datasets. *Applied Intelligence*, 2018, 48 (12), 4743-4759. <https://doi.org/10.1007/s10489-018-1238-7>
13. Frey, B. J., Dueck, D. Clustering by Passing Messages Between Data Points. *Science*, 2007, 315(5814), 972-976. <https://doi.org/10.1126/science.1136800>
14. Gionis, A., Mannila, H., Tsaparas, P. Clustering Aggregation. *ACM Transactions on Knowledge Discovery from Data*, 2007, 1(1), Article 4. <https://doi.org/10.1145/1217299.1217303>
15. Guha, S., Rastogi, R., Shim, K. Cure: An Efficient Clustering Algorithm for Large Databases. *Proceedings of*

- the ACM Sigmod Record, ACM, NY, USA, 1998, 73-84. <https://doi.org/10.1145/276304.276312>
16. Havens, T. C., Bezdek, J. C., Leckie, C., Hall, L. O. Fuzzy C-means Algorithms for Very Large Data. *IEEE Transactions on Fuzzy Systems*, 2012, 20(6), 1130-1146. <https://doi.org/10.1109/TFUZZ.2012.2201485>
 17. Haghtalab, S., Xanthopoulos, P., Madani, K. A Robust Unsupervised Consensus Control Chart Pattern Recognition Framework. *Expert Systems with Applications*, 2015, 42(19), 6767-6776. <https://doi.org/10.1016/j.eswa.2015.04.069>
 18. Jain, A. K. Data Clustering: 50 Years Beyond K-Means. *Pattern Recognition Letters*, 2010, 31(8), 651-666. <https://doi.org/10.1016/j.patrec.2009.09.011>
 19. Lai, C., Chung, P., Tseng, V. A Novel Two-Level Clustering Method for Time Series Data Analysis. *Expert Systems Applications*, 2010, 37(9), 6319-6326. <https://doi.org/10.1016/j.eswa.2010.02.089>
 20. Lichman, M. UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml>. Irvine: University of California, 2013.
 21. Liu, R., Huang, W., Fei, Z., Wang, K., Liang, J. Constraint-Based Clustering by Fast Search and Find of Density Peaks. *Neurocomputing*, 2019, 330(22), 223-237. <https://doi.org/10.1016/j.neucom.2018.06.058>
 22. Liu, R., Wang, H., Yu, X. Shared-Nearest-Neighbor-Based Clustering by Fast Search and Find of Density Peaks. *Information Sciences*, 2018, 450, 200-226. <https://doi.org/10.1016/j.ins.2018.03.031>
 23. Liwicki, S., Tzimiropoulos, G., Zafeiriou, S., Pantic, M. Euler Principal Component Analysis. *International Journal of Computer Vision*, 2013, 101(3), 498-518. <https://doi.org/10.1007/s11263-012-0558-z>
 24. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Oakland, CA, USA, 1967, 281-297.
 25. Morris, K., McNicholas, P. D. Clustering, Classification, Discriminant Analysis, and Dimension Reduction Via Generalized Hyperbolic Mixtures. *Computational Statistics & Data Analysis*, 2016, 97, 133-150. <https://doi.org/10.1016/j.csda.2015.10.008>
 26. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. Scikit-learn: Machine Learning in Journal of Machine Learning Research, 2011, 12 (Oct), 2825-2830.
 27. Rodriguez, A., Laio, A. Clustering by Fast Search and Find of Density Peaks. *Science*, 2014, 344(6191), 1492-1496. <https://doi.org/10.1126/science.1242072>
 28. Sander, J., Ester, M., Kriegel, H. P., Xu, X. Density-Based Clustering in Spatial Databases: The Algorithm Gdbscan and Its Applications. *Data Mining and Knowledge Discovery*, 1998, 2(2), 169-194. <https://doi.org/10.1023/A:1009745219419>
 29. Seyedi, S. A., Lotfi, A., Moradi, P., Qader, N. N. Dynamic Graph-Based Label Propagation for Density Peaks Clustering. *Expert Systems with Applications*, 2019, 115, 314-328. <https://doi.org/10.1016/j.eswa.2018.07.075>
 30. Tang, G., Jia, S., Li, J. An Enhanced Density Peak-Based Clustering Approach for Hyperspectral Band Selection. *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, Milan, Italy, 2015. <https://doi.org/10.1109/IGARSS.2015.7325966>
 31. Vinh, N., Epps, J., Bailey, J. Information Theoretic Measures for Clusterings Comparison: Is A Correction for Chance Necessary? *Proceedings of ICML'09*, Montreal, 1073-1080, 2009. <https://doi.org/10.1145/1553374.1553511>
 32. Wang, G., Song, Q. Automatic Clustering Via Outward Statistical Testing on Density Metrics. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(8), 1971-1985. <https://doi.org/10.1109/TKDE.2016.2535209>
 33. Wang, X., Xu, Y. Fast Clustering Using Adaptive Density Peak Detection. *Statistical Methods in Medical Research*, 2015, 26(6), 2800-2811. <https://doi.org/10.1177/0962280215609948>
 34. Xie, J., Gao, H., Xie, W., Liu, X., Grant, P. Robust Clustering by Detecting Density Peaks and Assigning Points Based on Fuzzy Weighted K-Nearest Neighbors. *Information Sciences*, 2016, 354(1), 19-40. <https://doi.org/10.1016/j.ins.2016.03.011>
 35. Xiong, H., Wu, W., Shekhar, S. Clustering and Information Retrieval. *Network Theory & Applications*, 2005. <https://doi.org/10.1007/978-1-4613-0227-8>
 36. Zhang, T., Ramakrishnan, R., Livny, M. Birch: An Efficient Data Clustering Method for Very Large Databases. *Proceedings of the ACM Sigmod Record*, ACM, NY, USA, 1996, 103-114. <https://doi.org/10.1145/233269.233324>
 37. Zhang, Y., Xia, Y., Liu, Y., Wang, W. Clustering Sentences with Density Peaks for Multi-Document Summarization. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Association for Computational Linguistics, Denver, Colorado, 2015, 1262-1267. <https://doi.org/10.3115/v1/N15-1>

