International Journal of Machine Learning and Cybernetics (2020) 11:1179–1195
https://doi.org/10.1007/s13042-019-01031-3

ORIGINAL ARTICLE

$M_{k\text{-}NN}$G-DPC: density peaks clustering based on improved mutual K-nearest-neighbor graph

Jian-cong Fan[1,2] · Pei-ling Jia[1] · Linqiang Ge[3]

Received: 22 October 2018 / Accepted: 31 October 2019 / Published online: 13 November 2019

Abstract
Clustering by fast search and detection of density peaks (DPC, Density Peaks Clustering) is a relatively novel clustering algorithm published in the Science journal. As a density-based clustering algorithm, DPC produces better clustering results while using less parameters than other relevant algorithms. However, we found that the DPC algorithm does not perform well if clusters with different densities are very close. To address this problem, we propose a new DPC algorithm by incorporating an improved mutual *k*-nearest-neighbor graph ($M_{k\text{-}NN}$G) into DPC. Our $M_{k\text{-}NN}$G-DPC algorithm leverages the distance matrix of data samples to improve the $M_{k\text{-}NN}$G, and then utilizes DPC to constrain and select cluster centers. The proposed $M_{k\text{-}NN}$G-DPC algorithm ensures an instance to be allocated to the fittest cluster. Experimental results on synthetic and real world datasets show that our $M_{k\text{-}NN}$G-DPC algorithm can effectively and efficiently improve clustering performance, even for clusters with arbitrary shapes.

Keywords Clustering · Mutual k-nearest-neighbor graph · Density peak

1 Introduction

Clustering analysis is an unsupervised learning technique [1] that recognizes different groups (clusters) underlying data. Clustering analysis has been applied to many fields such as pattern recognition [2], information retrieval [3], business intelligence [4], and so on. In the literatures, there are many clustering algorithms published, see for example [5–15]. Among those algorithms, hierarchy-based clustering [8–10] builds a binary tree to group similar data points. In density-based clustering [11, 12], a cluster is defined as a contiguous region of high density of data in the space. Model-based clustering assumes that the data is generated by a mixture of probability models. In general, clustering analysis involves two phases including *class* and *function* (also called a model). For a given data sample *D*, the *class* conducts rule-based or concept-based partitioning of *D*, while the *function* returns an array of labels corresponding to different clusters of *D*. Despite the tremendous efforts in the past a few decades, clustering analysis needs continued efforts with the emergence of new types of data in the era of big data.

In this paper, we focus on the state-of-the-art density peaks clustering (DPC) [16, 33–35]. As a density-based clustering algorithm, DPC produces better results while using less parameters than other clustering algorithms. However, it is found that the performance of DPC deteriorates significantly if clusters with different densities are very near. To address this problem, we propose to incorporate the mutual *k*-nearest-neighbor graph into DPC, called $M_{k\text{-}NN}$G-DPC. This proposed method can avoid assigning neighboring instances belonging to different clusters into the same cluster. Through constructing a mutual nearest neighbor graph, the theoretical analysis and the experimental results show that our $M_{k\text{-}NN}$G-DPC algorithm outperforms the DPC algorithm. In addition, our $M_{k\text{-}NN}$G-DPC algorithm can deal with clusters with arbitrary shapes.

The rest of this paper is organized as follows. In Sect. 2, the classical and well-performed clustering algorithms are briefly reviewed. In Sect. 3, the mutual *k*-nearest-neighbor

✉ Jian-cong Fan
fanjiancong@sdust.edu.cn

1 College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China

2 Provincial Key Lab. for Information Technology of Wisdom Mining of Shandong Province, Shandong University of Science and Technology, Qingdao, China

3 Department of Computer Science, Georgia Southwestern State University, Americus, GA, USA

graph and the DPC algorithm are introduced. In Sect. 4, we present the improved DPC algorithm, $M_{k\text{-}NN}$G-DPC, which combines the improved mutual *k*-nearest-neighbor graph with DPC algorithm. In Sect. 5, experimental studies are conducted to verify the effectiveness of our proposed algorithm. Section 6 concludes the paper.

2 Related work

Clustering analysis commonly differentiates objects from various groups (clusters) by the similarities or distances between pairs of objects. Typically, clustering algorithms are categorized based on a different set of rules for defining the similarity among data points [1, 21, 40–44, 55, 56]. As an example, in density-based clustering, a cluster is defined as an area with higher density than the other areas. The algorithm DBSCAN [11] is a classic density-based clustering method and designed to find arbitrary-shaped clusters. Through quantifying the neighborhood of a data point, DBSCAN can find a cluster with high dense in data space. The main drawback is that it is hard to systematically detect the density border since the cluster density decreases continuously, so that most of the parameters on density need to be specified manually.

Hierarchical clustering is another well-known clustering method. Hierarchical clustering (HC) methods represent data objects in a hierarchy or "tree" structure of clusters [15]. Hierarchical processes have two strategies: bottom-up or agglomerative, and top-down or divisive [1, 45–47]. Generally, HC is categorized into distance-based method or density-based method. For example, Gabor et al. [21] proposed a distance-based hierarchical clustering method that extends Ward's minimum variance by defining a cluster distance and objective function in terms of Euclidean distance. On the other hand, density-based HC applied the density connectivity to investigate the reachable distance of all data points and hierarchical structures. For example, Campello et al. [36] proposed HDBSCAN as an improvement over the classic OPTICS algorithm by measuring the clustering stability. They formalized the problem to maximize the overall stability of selected clusters. In addition, there are some theoretical studies on the above research areas [48–54].

In recent years, there are many research works focusing on improving the performance of the existing algorithms. For example, Li et al. [37] proposed a divisive projected clustering (DPCLUS) algorithm to partition the dataset into clusters in a top-down manner for detecting correlation clusters in highly noisy data. Zhang et al. [38] proposed a density-based multiscale analysis to reliably separate "noisy" objects from "clustered" objects and is applicable to clusters of arbitrary shapes. Clustering by fast search and detection of density peaks (abbrev. DPC) [16] is a novel algorithm published in *Science* by Rodriguez and Laio in 2014. DPC can find arbitrary shaped clusters without requiring multiple parameters. Moreover, it is insensitive to noise underlying the data. DPC is our interest and focus of this paper.

3 Preliminary

In this section, we introduce the standard DPC algorithm and the notion of mutual *k*-nearest-neighbor graph. Here, we will also investigate several improved DPC algorithms.

3.1 DPC Algorithm

DPC includes two distinct procedures. In the first procedure, DPC uses the input parameters to compute the local density of a group of data points and find the density peaks, which are considered as cluster centers. In the second procedure, each data point is assigned to the nearest cluster with the largest density peak. The best cluster center should satisfy two constraints: the largest density and the maximum margin. For data sample *i*, its local-density ρ_i and mini-distance δ_i are defined in Eqs. (1) and (2), respectively:

$$\rho_i = \sum_{j \neq i} \chi(d_{ij} - d_c) \tag{1}$$

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}) \tag{2}$$

where d_{ij} is the distance between sample *i* and *j*, d_c is the truncation distance. The piecewise function $\chi(x)$ is defined as:

$$\chi(\mathrm{x}) = \begin{cases} 1, & x < 0 \\ 0, & others \end{cases}$$

Besides Eq. (1), there is another approach to compute ρ_i by using Gaussian function as follows [6]:

$$\rho_i = \sum_{j \neq i} e^{-(\frac{d_{ij}}{d_c})^2} \tag{3}$$

In Eq. (1) ρ_i is actually the number of data samples whose distances to sample *i* is less than d_c. In Eq. (3), however, ρ_i is the sum of weighted distance of all samples to sample *i*. Since the probability that different samples have the same local densities is small, Eq. (3) is a suitable measure of local density, especially for small-scale datasets. The value of d_c directly influences clustering results. In DPC algorithm, d_c is set to such a value that the number of neighbors of each data sample is about one to two percent of the total data. DPC uses ρ and δ to construct decision graph, and then selects data points with the large ρ and δ values as the cluster centers. Each cluster center represents a cluster, and each data point is assigned to its nearest cluster.

The prominent advantage of DPC algorithm is that it can detect noisy data more precisely than other algorithms. DPC defines the boundary of a cluster as the member data points of the cluster. DPC considers the maximum density in a boundary of one cluster as the upper bound ρ_b of local density for this cluster. Then the data whose density is less than ρ_b is defined as noisy data.

DPC can find clusters with arbitrary shapes and dimensionalities. It works well in finding convex clusters. For clusters with very different densities, DPC cannot work well with the unique parameter ρ and the unique parameter δ.

Figure 1 illustrates two clustering results obtained by Eqs. (1) and (3), respectively. In the dataset, there are three clusters with different densities including a ring structure and two circular structures. The density peaks of three cluster structures are greatly different, but their boundaries are not obvious. The result in Fig. 1a is generated by the algorithm using Eq. (1), where point A is a data point with maximum value of local density. Point B is another point with larger local density value than point A and is close to point A. Based on the assignment criterion of DPC, point A should be assigned to the circular cluster centered by Center1 that point B belongs to. Obviously, this is incorrect because point A belongs to the ring cluster centered by Center3. The clustering result illustrated in Fig. 1b reveals the similar problem.

3.2 Mutual *k*-nearest-neighbor graph

A *k*-nearest-neighbor (abbrev. *k-NN*) graph [26] is a directed graph $G_{k\text{-}NN} = (V, E)$, where V is the set of vertexes, each of which is a group of data points, and E is the set of edges. There is a connection from vertex X_i to X_j if X_j is the *k-NN* of X_i. Like the *k-NN* algorithm, in *k-NN* graph, the choice of k is crucial for a good performance.

In a *k-NN* graph $G_{k\text{-}NN} = (V, E)$, for two arbitrary vertexes v_q and v_p, they are called *k*-mutual-neighbor if v_q belongs to *k-NN* (v_p) and v_p belongs to *k-NN* (v_q). For a dataset with the size n, a *k*-nearest-neighbor graph $G_{k\text{-}NN} = (V, E)$ is called mutual *k*-nearest-neighbor graph [26, 27] (abbrev. $M_{k\text{-}NN}G$) if and only if v_q and v_p are *k*-mutual-neighbor, which is written as:

$$M_{k-NN}G(v_p, v_q) = \begin{cases} 1, \text{ if } v_q \in k-NN(v_p) \ \ and \ \ v_p \in k-NN(v_q) \\ 0, \qquad\qquad \text{otherwise} \end{cases} \tag{4}$$

From Eq. (4), we know that an edge generated in an $M_{k\text{-}NN}G$ requires two vertexes of that edge to be *k*-mutual-neighbor for each other. Information are extracted from the relationships of vertexes in graph. In contrast with *k*-nearest-neighbor approach, information transmission of $M_{k\text{-}NN}G$ is bidirectional and connected each other. For example, The data point B is one of the $2{\cdot}k$ nearest neighbors of point A, while A does not belong to the set of $2{\cdot}k$ nearest neighbors of point B. Then there is no connection between A and B. In other words, the condition that A is connected to B is that A and B belong to the set of the other $2{\cdot}k$ nearest neighbors at the same time. So $M_{k\text{-}NN}G$ is an undirected graph, while *k-NN* graph is a directed graph.

4 Our approach

The above examples show that the standard DPC algorithm incorrectly groups the data as illustrated in Fig. 1. One reason is that the standard DPC has limitation in data assignment. When clusters with different densities are very close,

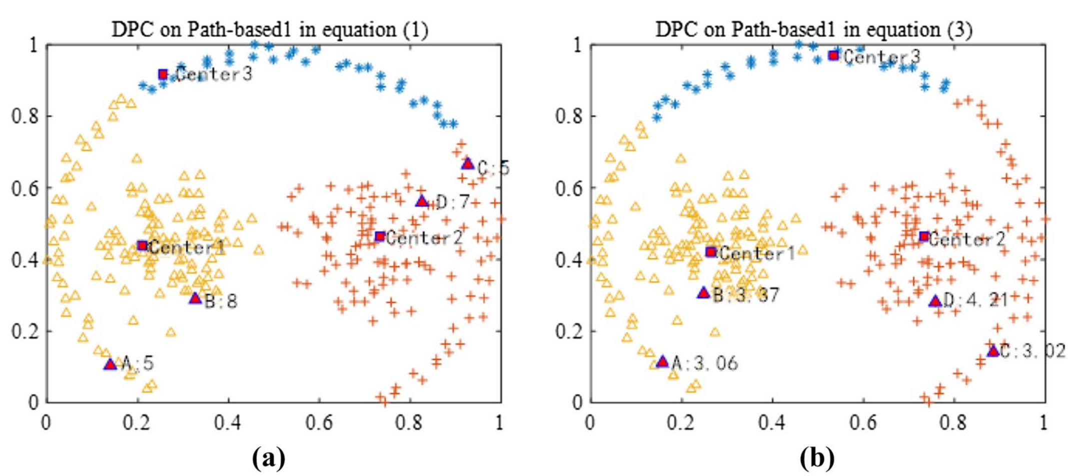


Fig. 1 Clustering results by DPC

a data instance in the boundary region can be easily assigned incorrectly to a cluster with higher density. For example, in Fig. 1a, the point A locates at the area with higher density. In fact, A belongs to the ring-shape cluster with lower density. According to the chained allocation criteria of DPC algorithm the data point A will be assigned to the cluster that B locates in because that cluster has the highest density near to A. This will cause the consequent misallocation. Figure 1b illustrates the clustering result by using another estimation approach of local density, the estimation of Gaussian kernel function. Also, it is can be seen that the point C is erroneously assigned to the nearest cluster with higher density that D locates in.

Xie et al. proposed an improved DPC algorithm KNN-DPC to address the problem discussed above [28]. Although KNN-DPC considers the strength of *k*-nearest-neighbor approach, other problems emerge. First, when the densities of clusters are irregular, inappropriate assignment strategy may result in unreliable output. That is, the data instances in the lower density cluster are probably assigned to higher density cluster. Second, the existing DPC-based algorithms use a single instance as the cluster center (the representative of a cluster), which may be insufficient to represent the actual shape of the cluster.

To address these problems, we can consider the mutual *k*-nearest-neighbor graph ($M_{k\text{-}NN}G$). However, we found that if $M_{k\text{-}NN}G$ is directly applied to DPC, the result is not much ideal, which will be illustrated in the following section. In this paper, we improve the basic $M_{k\text{-}NN}G$ and fuse it into DPC to develop a novel DPC-based algorithm.

4.1 $M_{k\text{-}NN}G$ and novel DPC

If the basic $M_{k\text{-}NN}G$ is directly applied to clustering analysis, we find the benefit is limited. We illustrate this problem by using an experiment with a two-dimensional dataset shown in Fig. 2. There are four clusters with different densities and shapes in this dataset, but the marginal similarity between arbitrary two clusters is very high. In other words, the objects in the boundary regions are highly similar or "close to" each other.

There are two possible scenarios when using the basic $M_{k\text{-}NN}G$ to deal with the dataset shown in Fig. 2. Figure 3 shows the clustering results when the parameter *k* is set to 3 and 4, respectively. For low-density clusters, if *k* is set as smaller values than the actual value, some points in low-density sections may become isolated points or clusters, for example, region C in Fig. 3a. If *k* is set as larger values, the points in the boundary regions may be connected to the same graph (the same cluster) even if they belong to different clusters. Region D in Fig. 3b is just such an example.

To address the limitations of the basic $M_{k\text{-}NN}G$ in handling clustering problems, we develop an improved version

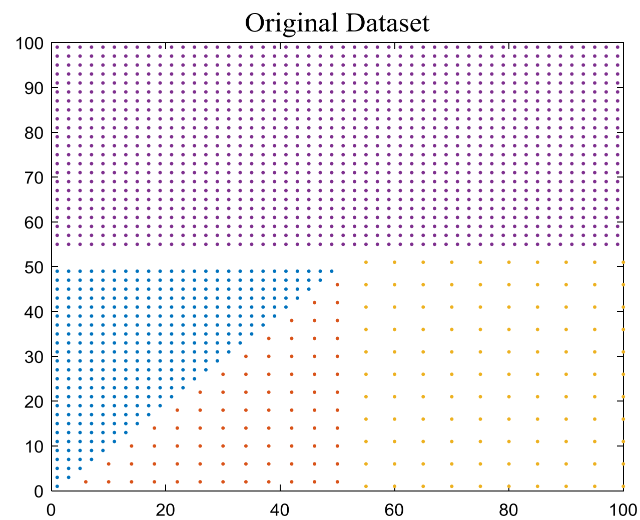


Fig. 2 Plot of a two-dimensional dataset

of $M_{k\text{-}NN}G$. Figure 4 shows the clustering result using the improved $M_{k\text{-}NN}G$. If $k=2$ and $k=3$, the improved $M_{k\text{-}NN}G$ leads to better clustering results than the basic $M_{k\text{-}NN}G$.

We now explain how to improve the basic $M_{k\text{-}NN}G$. We firstly give the following definitions.

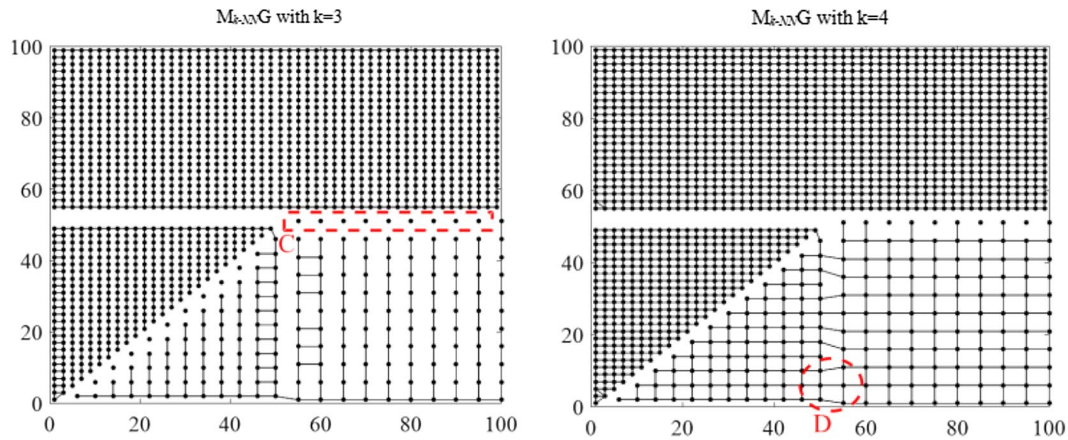
Definition 1 Let *i* and *j* represent different data samples, and $DS_{knn}(i)$ is a set which is formalized as Eq. (5). $DS_{knn}(i)$ is a ***k*-nearest-neighbor sample set**, where $knn(i)$ is the set containing *k* nearest neighbors of *i*, and $2{\cdot}knn(j)$ represents a set having $2{\cdot}k$ nearest neighbor samples of sample *j*.

$$DS_{knn}(i) = \{j | i \in 2 \cdot knn(j)\} \tag{5}$$

$DS_{knn}(i)$ is an index set of other different data samples connected to sample *i*. According to Eq. (5), the size of $DS_{knn}(i)$ is uncertain because of the different densities of data distribution. For those samples in a high-density space, there exists a great deal of nearest-neighbor samples of a sample *i* and the size of $DS_{knn}(i)$ may be greater than $2{\cdot}k$. otherwise, for those samples in a low density space, $DS_{knn}(i)$ may only include a few elements.

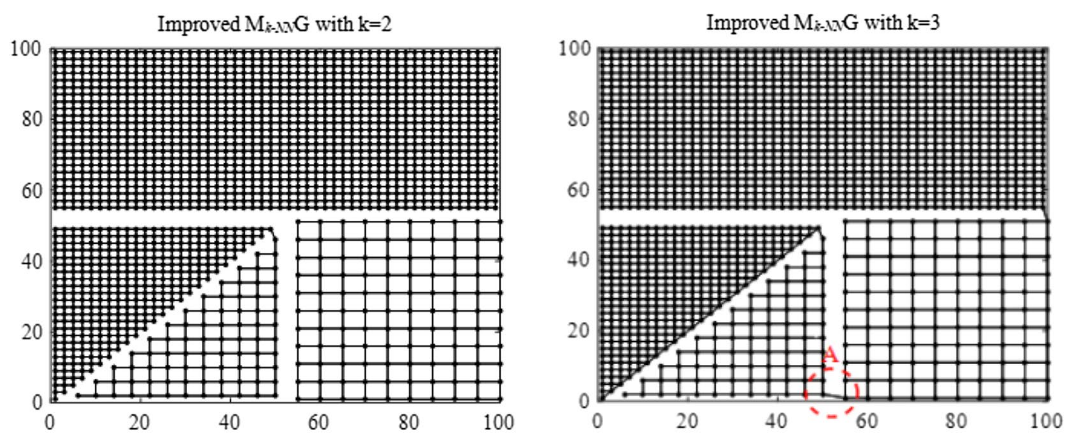
Definition 2 If the size of $DS_{knn}(i)$ is larger than or equal to *k*, a **distance-upper-bound point** $p_{DUB}(i)$ is the k^{th} larger number, otherwise, $p_{DUB}(i)$ is the data point of the most far nearest-neighbor. The description of $p_{DUB}(i)$ is formalized as Eq. (6), where $|DS_{knn}(i)|$ is the size of $DS_{knn}(i)$, and $distance(i, j)$ is a function which is used to calculate the distance of sample *i* and *j*.

$$p_{DUB}(i) = \begin{cases} kth \text{ nearest neighbor from } DS_{knn}(i), & |DS_{knn}(i)| \geq k \\ \underset{j \in DS_{knn}(i)}{\operatorname{argmax}}(distance(i,j)), & otherwise \end{cases} \tag{6}$$



(a) Clustering result when parameter k=3 **(b)** Clustering result when parameter k=4

Fig. 3 Clustering results when using the basic $M_{k\text{-}NN}$G



(a) Clustering result when parameter k=2 **(b)** Clustering result when parameter k=3

Fig. 4 Clustering results when using the improved $M_{k\text{-}NN}$G

Definition 3 The distance between sample i and $p_{DUB}(i)$ is called **distance level**, $dis_{level}(i)$, which is formalized as Eq. (7).

$$dis_{level}(i) = distance(i, p_{DUB}(i)) \tag{7}$$

We use the nearest-neighbor set DS_{knn} to estimate the distance level of each data sample. According to Eq. (7), distance level $dis_{level}(i)$ is actually a measure of density that can be used to distinguish those regions with different densities.

Definition 4 If the distance between sample i and j is less than the distance level $dis_{level}(i)$ and $dis_{level}(j)$, all such samples j constitute a set called **mutual k-nearest-neighbor set** ($S_{M_{k-NN}}$), which is formalized as Eq. (8). The graph generated from $S_{M_{k-NN}}$ is the **improved $M_{k\text{-}NN}$G**.

$$S_{M_{k-NN}}(i) = \{j | distance(i,j) < dis_{level}(j) \wedge distance(i,j) < dis_{level}(i)\} \tag{8}$$

According to Eq. (8), there is no upper bound of the connectivity for some nodes in the improved $M_{k\text{-}NN}$G because the number of mutual nearest neighbors for some samples in high density region may be greater than k.

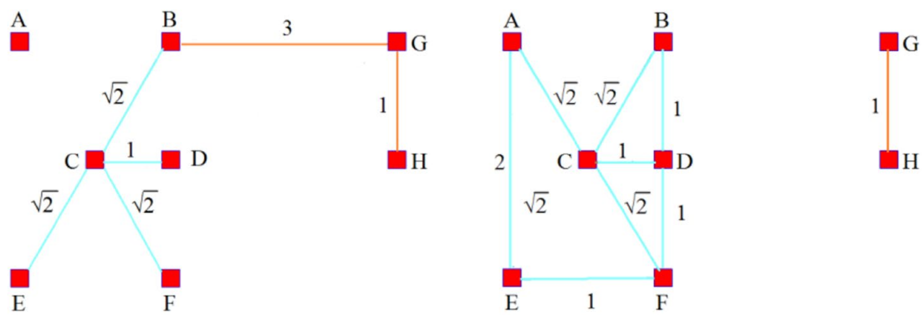
The improved $M_{k\text{-}NN}$G has high significance to lift the clustering capacities, especially for density-based technique. When sample i is in the boundary of a high-density region and sample j *is* in the region of low-density, if i and j are "near", the value of $dis_{level}(i)$ is small and the value of $dis_{level}(j)$ is large. Conseqently, i and j are not the mutual neighbors.

We next illustrate how to construct an improved $M_{k\text{-}NN}$G. When $k=2$, Fig. 5a is the nearest neighbor graph of samples C and G, and (b) is the improved version of (a). For samples C and G, $p_{DUB}(C) = \{B, D, F, E\}$, and $p_{DUB}(G) = \{B, H\}$. According to Eq. (6) and (7), $dis_{level}(C) = \sqrt{2}$, and $dis_{level}(G) = 3$. Figure 5 (b) can then be obtained according to Eq. (8). Because sample C is located in a high-density region, its mutual nearest neighbors are more than 2 k (k is 2, but the total number of neighbors is 5). For sample G, however, it is in a low-density region, and only has one neighbor.

Figure 6 illustrates the results of using the basic $M_{k\text{-}NN}G$ and its improved version to cluster the two-dimensional dataset illustrated in Fig. 2. Figure 6a, b are the results of the improved DPC algorithm with the improved $M_{k\text{-}NN}G$ and the basic $M_{k\text{-}NN}G$, respectively. Obviously, the result in (a) is more accurate than the result in (b).

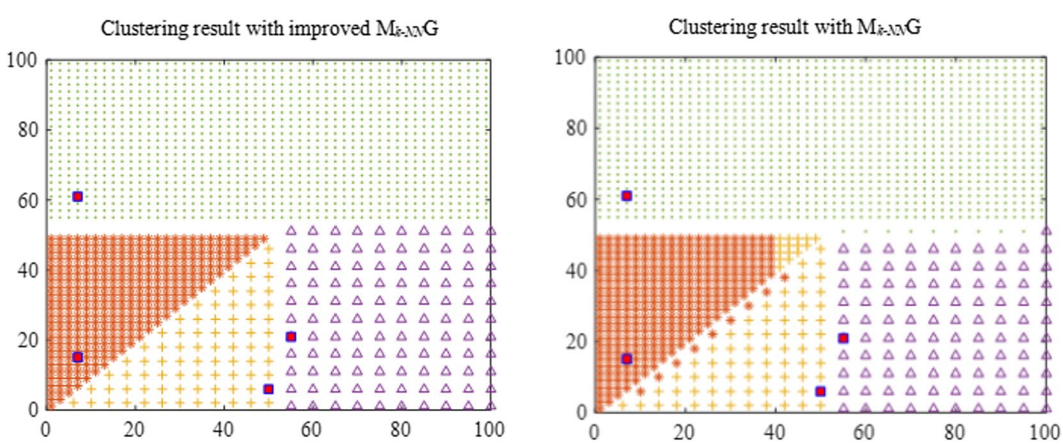
We also test the DPC algorithm with the improved $M_{k\text{-}NN}G$ using real-world datasets. Table 1 shows the NMI (Normalized Mutual Information) [30] results. By using the improved $M_{k\text{-}NN}G$, the DPC achieves much better results than it basic version for six real datasets.

The above results shows that DPC algorithm combined with the improved $M_{k\text{-}NN}G$ outperforms the basic $M_{k\text{-}NN}G$. In the improved $M_{k\text{-}NN}G$, the distance level of each sample i must be calculated to decide whether two arbitrary samples are mutual nearest neighbors. We give Theorem 1 to state the relationship of mutual nearest neighbors and distance level.



(a) An $M_{k\text{-}NN}G$ **(b)** the improved $M_{k\text{-}NN}G$

Fig. 5 Illustration of $M_{k\text{-}NN}G$ and its improved version



(a) Results with the improved $M_{k\text{-}NN}G$ **(b)** Results with the basic $M_{k\text{-}NN}G$

Fig. 6 Results of the improved DPC algorithm with the basic $M_{k\text{-}NN}G$ and its improved version ($k = 3$)

Table 1 NMI Results of the improved DPC algorithm with different $M_{k\text{-}NN}G$ on real datasets

DPC algorithm	Datasets					
	Seeds	WDBC	Glass	Movement	Ecoli	Iris
Improved $M_{k\text{-}NN}G$	72.73	72.02	53.13	63.09	67.13	90.09
Basic $M_{k\text{-}NN}G$	69.08	67.94	33.45	43.25	54.19	88.46

Theorem 1 *Samples i and j are mutual nearest neighbors, if and only if the distance between i and j is less than their distance levels* $dis_{level}(i)$ *and* $dis_{level}(j)$.

Proof Given an arbitrary sample i, let $knn(i)$ be the set containing k nearest neighbors of i. According to the definition of $DS_{knn}(i)$, if the size of $DS_{knn}(i)$ is larger than k, written as $|DS_{knn}(i)| > k$, $knn(i) \subset DS_{knn}(i)$; if $|DS_{knn}(i)| = k$, $knn(i) = DS_{knn}(i)$; if $|DS_{knn}(i)| < k$, $DS_{knn}(i) \subset knn(i)$. Therefore, if we want to prove that $distance(i, j) \leq dis_{level}(i)$ and $distance(i, j) \leq dis_{level}(j)$ hold, we need to prove that i belongs to $knn(j)$ and j belongs to $knn(i)$.

According to Definition 3, when $|DS_{knn}(i)| \geq k$, the distance level $dis_{level}(i)$ indicates the distance between sample i and the kth ordered neighbor in $DS_{knn}(i)$; when $|DS_{knn}(i)| < k$, the $dis_{level}(i)$ is the largest distance between sample i and the farthest neighbor. For sample i, let $MaxDis(i)$ Eqals to $\max_{j \in DS_{knn}(i)} (distance(i,j))$, where j is an arbitrary sample in $DS_{knn}(i)$. If $distance(i, j) \leq dis_{level}(i)$, j belongs to $knn(i)$ according to Definition 3, thereby obtaining $distance(i, j) \leq MaxDis(j)$. Similarly, j belongs to $knn(i)$ and thereby obtaining $distance(i, j) \leq MaxDis(i)$.

According to the above, we can prove that Theorem 1 holds.

Comparing the improved $M_{k\text{-}NN}G$ with the basic $M_{k\text{-}NN}G$ for DPC, the improved one prefers to those samples with similar density. Towards this nature, a traditional constraint should be added to the improved DPC algorithm.

Constraint 1 If samples i and j can be clustered together, one of the necessary constraints is that they are mutual k-nearest-neighbor.

Based on Constraint 1, the arbitrarily shaped and complex clusters can be obtained.

4.2 Assignment strategy

Through the above analysis, we know that the $M_{k\text{-}NN}G$ for DPC needs Constraint 1 to improve its clustering performance. Thus, besides measuring the local density, distance and finding the cluster centers by decision graph, the DPC algorithm based on the improved $M_{k\text{-}NN}G$ can take the advantage of the theories stated in Sect. 4.1 to assign data instances to those nearest clusters with the highest local densities. Especially, we need reasonable assignment strategy for those data instances in the boundary of clusters. We give Definition 5.

Definition 5 Let $C_1, C_2, \ldots, C_m$ be m different clusters, a set is called a cluster boundary if satisfying Eq. (9),

$$S_{boundary} = \{j | j \in S_{M_{k-NN}}(i_1), j \in S_{M_{k-NN}}(i_2), \ldots, j \in S_{M_{k-NN}}(i_m), \quad i_1 \in C_1, i_2 \in C_2, \ldots, i_m \in C_m\} \tag{9}$$

where $i_1, i_2, \ldots, i_m$ are m instances taken from $C_1, C_2, \ldots, C_m$, respectively.

According to Definition 5, an instance j belonging to $S_{boundary}$ means that j is the mutual nearest neighbor of other instances from multiple clusters. That is, there are multiple directed edges connected to node j. Then, the assignment strategy of j is that j is assigned to a cluster C if there are the maximum edges connected to it from C.

4.3 Proposed algorithm

In this section, we apply the improved mutual nearest neighbor graph and the assignment strategy for data instances to develop an improved DPC algorithm, called $M_{k\text{-}NN}G$-DPC. To avoid assigning neighboring instances belonging to different clusters into the same cluster, it is necessary to construct a

Algorithm 1.

Input: Dataset D, the truncation distance d_c, the parameter k of k-NN;
Output: Data instances with labels

Procedure:
Step 1. Derive the distance matrix M of D;
Step 2. Use M to Calculate ρ and δ according to Eq. (1), (2) and (3), respectively;
Step 3. Apply ρ and δ to construct the mutual nearest neighbor graph, and choose cluster centers from the graph;
Step 4. Derive $DS_{knn}(i)$ and $dis_{level}(i)$ of each instance i by Eq. (5) - (7) in turn;
Step 5. Calculate $S_{M_{k-NN}}$ by Eq.(8);
Step 6. Assign the data instances in the boundary by *Constraint* 1 and Definition 5;
Step 7. Repeat Step 5 and 6 until each instance assigned to a cluster;

Fig. 7 Description of $M_{k\text{-}NN}G$-DPC algorithm mutual nearest neighbor graph to distinguish such neighbors. The idea is to use distance (near neighbors) and distribution density simultaneously, as stated in Sects. 4.1 and 4.2. The $M_{k\text{-}NN}$G-DPC algorithm is described in Fig. 7.

4.4 Analysis of computational complexity

For a data set D containing n instances, the computational complexities of calculating the distance matrix M and the parameter δ are all $O(n^2)$. The complexity of calculating local-density ρ using Eq. (1) is $O(n^2)$ as well. It is necessary to scan D to derive the distances between different instances less than a threshold d_c. If we use Eq. (3) to calculate ρ, the complexity is also $O(n^2)$ because the sum of weighted distances of all instances must be derived. Therefore, the overall complexity from Step 1–3 is $O(n^2)$. The calculation of $DS_{knn}(i)$ in Step 4 involves finding 2 k neighbors of each instance i, and thus the complexity for n instances is $O(n^2)$. Because the complexity of deriving $p_{DUB}(i)$ is $O(2\,kn)$, considering that the elements in $p_{DUB}(i)$ needs to be ordered, the worst-case computational complexity of calculating $dis_{level}(i)$ is $O(n(2\,k)^2)$, where k is usually far less than n. According to Eq. (8), the computational complexity of Step 5 is $O(n^2)$. Thus, the overall complexity of the $M_{k\text{-}NN}$G-DPC algorithm is $O(n^2)$. In fact, the time complexities of DPC algorithm and its variations are all $O(n^2)$ [16, 28, 33, 34, 36], so the computational complexity of the proposed algorithm is approximate to or equal to the other DPC-based algorithms.

5 Experimental analysis

In this section, we test the proposed $M_{k\text{-}NN}$G-DPC algorithm on 18 datasets and compare it with nine classical clustering algorithms.

5.1 Experimental methodology

5.1.1 Data sets

We test our proposed algorithm $M_{k\text{-}NN}$G-DPC with eighteen benchmark datasets from UCI machine learning repository [29]. The important statistics of the benchmark datasets are summarized in Table 2. The datasets listed in Table 2 are all real and multi-dimensional. And all the class labels are removed from these datasets to generate unlabeled data.

5.1.2 Compared algorithms

We compare the $M_{k\text{-}NN}$G-DPC algorithm with nine well-known clustering algorithms including the standard DPC [16], DBSCAN [11], Spectral clustering [19], BIRCH [8], Mean shift [18], Gaussian Mixture Model (GMM) [23], K-means [6], FCM [24] and Affinity Propagation (AP) [17].

DBSCAN is a representative method which models clusters as dense regions in the data space and can discover arbitrary clusters. Spectral clustering is a representative method in high dimensional data applications, which combines feature extraction approaches with clustering strategies. Spectral clustering uses matrix theory including affinity matrix, computation of eigenvectors, and transformation of vector spaces. BIRCH is another representative algorithm in hierarchical clustering. BIRCH integrates bottom-up strategy with a kind of data structure, namely, clustering feature tree, resulting in multiphase hierarchical clustering. Mean shift clustering is a typical non-parametric feature-space analysis technique with the characteristics of application-independence and non-assumption of any predefined shape on data clusters. The GMM can be viewed as a mixture of a number of Gaussian components. The aim of GMM is to maximum the log-likelihood function. The K-means clustering algorithm is based on distance measurement. FCM is based on Euclidean distance function and associated with fuzzy mathematics. The AP method is one of the state-of-the-art methods proposed recently, which takes as input measures of similarity between pairs of data points. These well-known models apply probability-, evolution-, or ML-based ideas, which is also applied in our proposed approaches.

5.1.3 Evaluation metrics

We use five metrics to evaluate the performances of the clustering algorithms:

Table 2 The benchmark datasets used in this paper

DataSet	Size/attribute	Cluster/source	DataSet	Size/attribute	Cluster/source
Compound	399/2	6/[17]	Seeds	210/7	3/[23]
Atom	800/3	2/[18]	Art	300/4	3/[23]
Path-based1	300/2	3/[19]	Ionosphere	351/34	2/[23]
Zelnik6	238/2	3/[20]	Libras Movement	360/90	15/[23]
Dim256	1024/256	16/[21]	WDBC	569/30	2/[23]
Dim1024	1024/1024	16/[21]	Glass	214/10	6/[23]
S3	5000/2	15/[22]	Waveform	5000/21	3/[23]
S4	5000/2	15/[22]	Image Segmentation	2310/19	7/[23]
Iris	150/4	3/[23]	Ecoli	336/7	8/[23]

(1) Clustering accuracy: the accuracy of a clustering algorithm is the ratio of correct assignments to the whole dataset. The computational equation of accuracy (Acc) is

$$\mathrm{Acc} = \frac{\sum_{i=1}^{k} a_i}{|D|} \tag{10}$$

In Eq. (10), the parameter a_i denotes the number of data objects that are correctly assigned to cluster C_i, k is the number of clusters, and $|D|$ means the size of dataset D.

(2) Clustering F-score: a weighted average of the Precision and Recall, where it reaches its best value at 1 and worst at 0, or it is viewed as the harmonic mean of Precision and Recall. Precision means the precision of the clustering results, which is the degree to which repeated measurements show the same results if conditions unchanged. Precision (PR) is calculated by:

$$\mathrm{PR} = \frac{\sum_{i=1}^{k} \frac{a_i}{a_i+b_i}}{|D|} \tag{11}$$

In Eq. (11), if a data set D contains k clusters, a_i denotes the number of data objects that are correctly assigned to cluster C_i while the parameter b_i denotes the data objects that are incorrectly assigned to C_i.

Recall (RE) is the ratio of the number of data objects but are falsely clustered. Assuming that c_i denotes the number of data objects belonging to the ith cluster but are falsely assigned to other clusters, RE is defined as:

$$\mathrm{RE} = \frac{\sum_{i=1}^{k} \frac{a_i}{a_i+c_i}}{|D|} \tag{12}$$

Using Eq. (11) and (12), F-score is defined as:

$$\text{F-score} = \frac{2 * \mathrm{PR}*\mathrm{RE}}{\mathrm{PR} + \mathrm{RE}} \tag{13}$$

(3) NMI (Normalized Mutual Information) [30]: a commonly used index in evaluating the effectiveness of a clustering algorithm. NMI is calculated by:

$$\mathrm{NMI}(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}} \tag{14}$$

where X and Y are random variables, $I(X, Y)$ means the mutual information between X and Y, and $H(X)$ is the entropy of X. $I(X, Y)$, $H(X)$ and $H(Y)$ are obtained by Eqs. (15), (16) and (17), respectively,

$$I(X,\ Y) = \sum_{h=1}^{k^{(a)}} \sum_{l=1}^{k^{(b)}} n_{h,l} \log \left(\frac{n \cdot n_{h,l}}{n_h^{(a)} n_l^{(b)}} \right) \tag{15}$$

$$H(X) = \sum_{h=1}^{k^{(a)}} n_n^{(a)} \log \frac{n_h^{(a)}}{n} \tag{16}$$

$$H(Y) = \sum_{l=1}^{k^{(b)}} n_l^{(b)} \log \frac{n_l^{(b)}}{n} \tag{17}$$

where $n_h^{(a)}$ is the number of data objects in cluster h when their class-label is a, and $n_l^{(b)}$ is the number of data instances in cluster l when their class-label is b. And $n_{h,\,l}$ is the number of objects that are in cluster h associated with a as well as in group l associated with b.

(4) Clustering Purity: a simple clustering index in evaluating the proportion of correctly clustered samples. Purity is calculated by:

$$\mathrm{Purity} = \sum_{i=1}^{K} \frac{m_i}{m} P_i \tag{18}$$

where m_i is the number of all members in cluster i, and m is the number of members involved in the whole cluster partition. P_i is proportion of the largest number of members in cluster i to all members of this cluster. And K is the number of clusters.

(5) ARI (adjusted rand index): a variation of clustering evaluation index RI, ARI is to calculate the similarity of random uniform distribution between real class label and predicted class label. The larger ARI is, the better the clustering effect is. RI and ARI are calculated by:

$$RI = \frac{a + b}{C_2^{n_{samples}}} \tag{19}$$

$$ARI = \frac{RI - E(RI)}{\max(RI) - E(RI)} \tag{20}$$

where a *is* the sample log in same label between real label and predicted label, b is the sample log in different label between real label and predicted label. $C_2^{n_{samples}}$ is all possible combinations of pairs of samples. $E(RI)$ is the expected value of RI.

5.2 Results and analysis on synthesis data

We first use three two-dimensional datasets to illustrate the performance of the standard DPC, the basic $\mathrm{M}_{k\text{-}NN}$G-based and the improved $\mathrm{M}_{k\text{-}NN}$G-based DPC. Each of these synthesis datasets contains several clusters with different shapes and distribution densities. In this section, we employ the Euclidean distance as the distance measure. To facilitate computation in the algorithm, the truncation distance d_c is represented by percentages. We sort all the distances in ascending order based on the distance matrix, and then take the value at position n %×size (D) of these ordered distances, where n is a tuning parameter and D is a dataset, as the value of d_c.

Figures 8, 9, and 10 illustrate the clustering results on three synthesis datasets, Zelnik6, Path-based1, and Compound, respectively, produced by the standard DPC, and the proposed $M_{k\text{-}NN}$G-DPC algorithms.

Zelnik6 dataset has 238 data instances and contains 3 clusters. The circle-shaped cluster in Zelnik6 has low-density, as shown in Fig. 8a, the clusters obtained by the

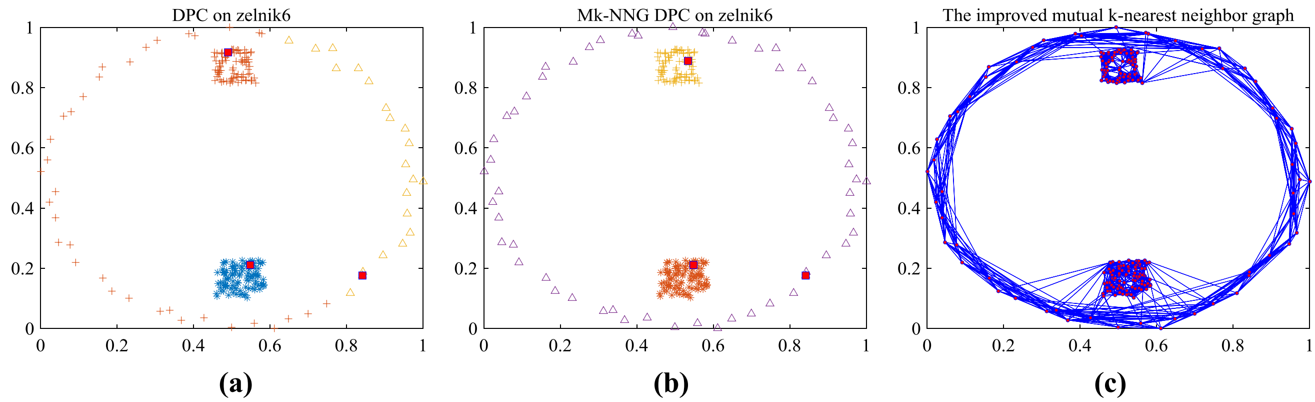


Fig. 8 Clustering results on the synthesis dataset Zelnik6

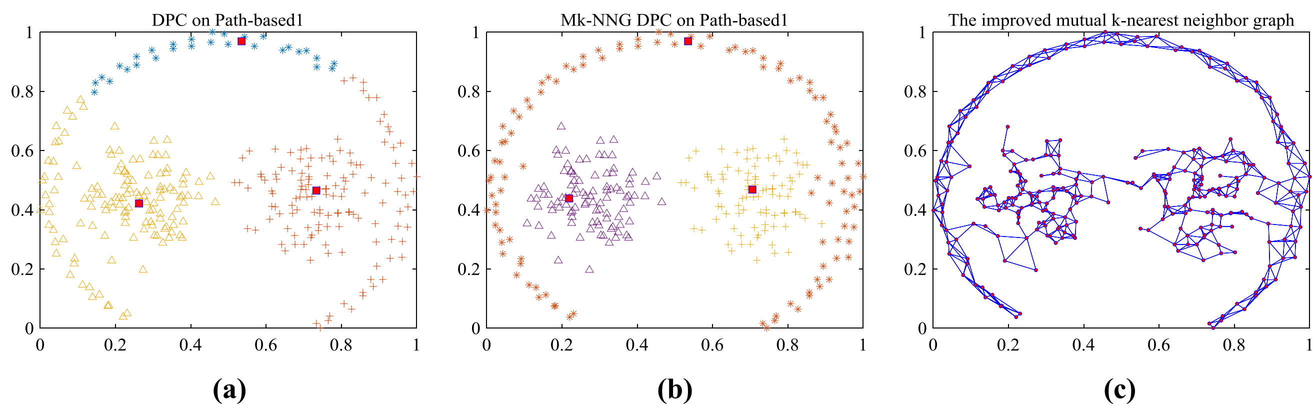


Fig. 9 Clustering results on the synthesis dataset Path-based1

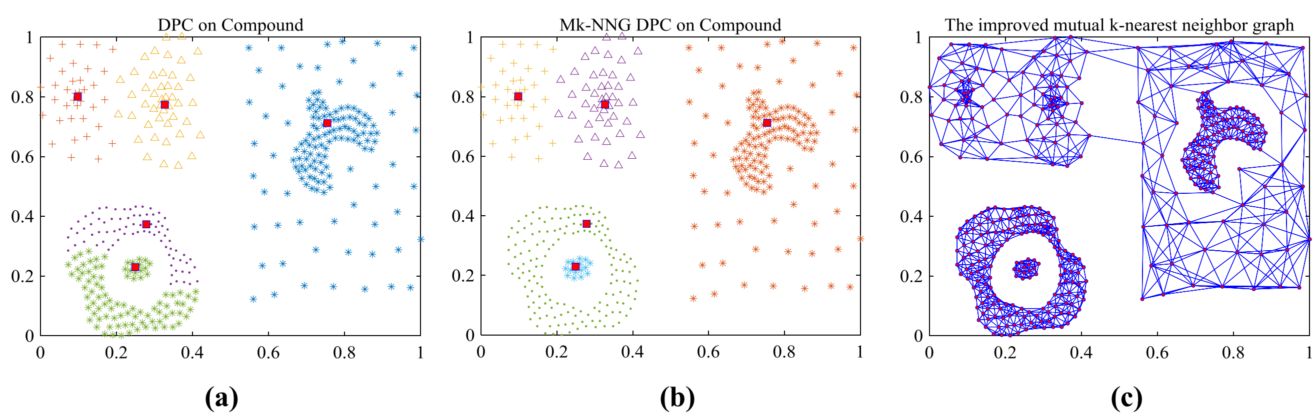


Fig. 10 Clustering results on the synthesis dataset Compound

standard DPC are not good enough (or has higher error rate). The main reason is that the density of this circle-shaped cluster is low but the standard DPC cannot solve this problem very well. Figure 8b shows the clusters clustered by $M_{k\text{-}NN}$G-DPC. There are edges between the circle-shaped cluster and other two high-density clusters, but the edges are sparse. This can help $M_{k\text{-}NN}$G-DPC generate arbitrary-shaped clusters with different densities. Figure 8c shows the clusters generated from $M_{k\text{-}NN}$G in (b).

Figure 9 shows the clustering results on the synthesis dataset Path-based1 which contains 300 data instances and 3 clusters. Figure 9a shows the results obtained from the standard DPC. Since the boundaries of the circle-shaped cluster and the other two clusters are not clear, the clusters generated by the standard DPC have many errors because the data instances in the boundaries are easily clustered erroneously. As shown in Fig. 9b, the clustering effect becomes much better when using the $M_{k\text{-}NN}$G-DPC than using the standard DPC. The result illustrated in Fig. 9c is generated from (b). The $M_{k\text{-}NN}$G-DPC algorithm focuses on the margin sections and there are few edges between different clusters, thereby obtaining very high accurate rate.

In Fig. 10a, c illustrate the clustering results generated by the standard DPC and the $M_{k\text{-}NN}$G-DPC algorithm, respectively, on dataset Compound which has 399 instances and 6 different shaped clusters with different densities. In Fig. 10a, there are two clusters in the left lower part of the coordinate, but the standard DPC cannot cluster them correctly because it only considers the factors of distance and density but does not consider the margin size between different clusters. The $M_{k\text{-}NN}$G-DPC algorithm produces the result illustrated in Fig. 10b, but there are only few edges in the margins of different clusters. Then we can obtain the clusters shown in Fig. 10c, from which we can see that the clustering effect is well.

From the above experiments on those synthesis data, our proposed algorithm is well-performed. Although the standard DPC and the $M_{k\text{-}NN}$G-DPC algorithms choose the same cluster centers, the effects are significantly different. This also validates the analysis for the $M_{k\text{-}NN}$G-DPC algorithm in Sect. 4. Due to the join of a constraint in the $M_{k\text{-}NN}$G-DPC algorithm, when data instances are assigned to a cluster, there is less chance for those instances with different local densities to be assigned to a same cluster. This alleviates the problem of the standard DPC and lifts the clustering performance.

5.3 Results and analysis on the benchmark datasets

In this section, we use the benchmark datasets, compared algorithms, and evaluation metrics listed in Sect. 5.1 to validate our proposed algorithm.

The comparison results on eight synthesis datasets are presented in Tables 3 and 4. These results are obtained by using five metrics and nine compared algorithms. From Table 3 the proposed algorithm $M_{k\text{-}NN}$G-DPC is better than most of the other algorithms.

For dataset zelnink6 and Path-based1, as illustrated in Figs. 8a and 9a, they contain multiple clusters with different shapes and densities. Obviously, $M_{k\text{-}NN}$G-DPC has the best clustering results. Because of the deficiencies of the standard DPC algorithm, as stated in Sect. 3, the lack of samples can limit the capability of allocation to the corresponding clusters, which increases the misclassification rates when there are clusters with low density. K-means, FCM and AP algorithms can cluster convex-shaped, especially spherical-shaped clusters, but are unsuitable for arbitrary-shaped ones. GMM is a Gaussian model-based algorithm, which can find those clusters with high density obeying normal distributions. However, for those sparse clusters, GMM has low clustering performances. Mean shift algorithm is a non-parametric feature-space clustering technique for finding the maxima of a density function by iteration. The mean shift algorithm has been widely used in many applications and has good clustering effect. However, there is no rigid proof for the convergence of the algorithm, especially in a high dimensional space.

The dataset Compound contains some different clusters with low-density and high-density, as illustrated in Fig. 10a. There are sparse clusters in the Compound, so $M_{k\text{-}NN}$G-DPC cannot perform very well because it uses the standard DPC to choose cluster centers, which is difficult to obtain the ideal centers. But $M_{k\text{-}NN}$G-DPC obtains the best result in F-score index. Nevertheless, the clustering results in all indexes are better than the standard DPC. The dataset Atom contains 3 features and 800 instances. Its shape looks like an atom, in the center of which there is the *nucleus*, that is, a cluster with the higher density than the surrounding *neutrons* (cluster). For this dataset, obviously, the distance-based clustering algorithms cannot perform well. Those non-distance-based algorithms, such as DBSCAN (density-based), Spectral clustering (graph-based), GMM (probability-based), and our proposed method, can obtain better results. Given a dataset in some space, DBSCAN groups together data points that are closely packed together with many nearby neighbors ("*nearby*" is measured by some parameters). Spectral clustering algorithm makes use of the eigenvalues (called spectrum) of the similarity matrix of the dataset to perform

Table 3 The comparison of Acc, F-Score and NMI for each clustering algorithms on synthetic datasets

Algorithm	Compound			Path-based1			Atom			Zelnik6		
	Acc	F-score	NMI	Acc	F-score	NMI	Acc	F-score	NMI	Acc	F-score	NMI
$M_{k\text{-}NN}$G-DPC	87.22	**79.37**	84.70	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
DPC	68.42	58.12	76.58	73.33	69.36	50.28	68.25	64.69	21.21	85.29	78.99	72.08
DBSCAN	**89.47**	73.16	80.20	65.33	35.02	76.97	**100**	**100**	**100**	76.47	57.66	49.91
K-MEANS	71.68	70.16	74.57	74.33	70.81	51.28	72.13	69.78	26.72	82.35	72.14	57.65
Spectral clustering	84.21	72.56	**88.70**	88.00	88.16	73.51	**100**	**100**	**100**	**100**	**100**	**100**
AP	71.43	69.80	74.83	74.00	70.33	50.92	65.63	61.02	17.72	82.35	73.05	57.69
FCM	65.66	58.69	71.13	74.67	71.29	51.58	74.00	72.19	28.09	82.77	73.94	58.39
GMM	70.18	58.67	75.72	72.33	72.35	54.65	99.63	99.63	96.81	98.74	98.48	94.91
Birch	76.69	49.42	72.15	73.33	71.63	47.42	64.13	63.31	6.45	80.67	69.18	55.44
Mean shift	84.46	56.63	77.26	69.67	63.66	46.68	69.75	47.74	38.13	81.51	71.13	56.58
Algorithm	Dim256			Dim1024			S3			S4		
	Acc	F-score	NMI	Acc	F-score	NMI	Acc	F-score	NMI	Acc	F-score	NMI
$M_{k\text{-}NN}$G-DPC	**100**	**100**	**100**	**100**	**100**	**100**	**85.60**	**85.55**	**79.80**	**80.18**	**80.52**	**73.49**
DPC	**100**	**100**	**100**	**100**	**100**	**100**	85.36	85.22	79.53	79.62	79.46	72.50
DBSCAN	**100**	**100**	**100**	**100**	**100**	**100**	35.42	27.73	45.37	28.02	17.42	64.20
K-MEANS	**100**	**100**	**100**	**100**	**100**	**100**	85.56	85.45	79.47	79.62	79.40	71.94
Spectral clustering	91.41	90.44	96.88	84.86	81.85	93.75	80.30	77.95	75.91	74.46	73.35	71.46
AP	**100**	**100**	**100**	**100**	**100**	**100**	**85.60**	85.48	79.48	79.22	79.00	71.59
FCM	62.50	52.34	78.90	68.75	59.38	83.34	79.32	76.86	77.15	79.70	79.49	71.89
GMM	N/A	N/A	N/A	N/A	N/A	N/A	64.72	62.00	70.67	57.28	56.60	62.85
Birch	43.75	38.64	48.81	**100**	**100**	**100**	61.32	56.99	70.41	60.16	53.99	64.39
Mean shift	**100**	**100**	**100**	**100**	**100**	**100**	82.00	84.49	76.36	74.36	71.72	69.26

Bold values indicate the best results

dimensionality reduction before clustering in fewer dimensions. $M_{k\text{-}NN}$G-DPC is also based on graph theory, as stated in the above section. The clustering results of these three algorithms are the best, that is, 100%.

The DIM256 and DIM1024 are two high-dimensional datasets, which contain 1024 and 10 clusters, 256 and 1024 dimensionalities, respectively. The clusters of these two datasets are separate in space and are all convex shapes, so most algorithms, including our proposed algorithm, obtain the highest values of all indexes. It indicates that $M_{k\text{-}NN}$G-DPC can deal with high-dimensional datasets. GMM algorithm cannot find the fittest parameters due to the high dimensionality, so there are no results on these two datasets. The S3 and S4 datasets all contains 5000 samples and 16 clusters. Although the cluster center of each cluster is obvious, the margins between the neighbored clusters are overlapped. K-means, FCM, and AP algorithms are fit for such datasets like S3 and S4, and they perform well on these data. The mean shift algorithm employs an iterative process to find those regions with high density, thereby having good performances on S3 and S4 data. However, the mean shift cannot effectively deal with the overlapping problems, so it is not the best. The GMM and Birch algorithms perform worse than other algorithms because the densities of many different clusters are similar and the density-based algorithms cannot distinguish these clusters well. $M_{k\text{-}NN}$G-DPC can overcome these disadvantages and improve the performance of the standard DPC, so it has the best clustering results on S3 and S4 data.

The experimental results on 10 real datasets are listed in Tables 5 and 6. The Iris dataset has 150 instances with 4 features and contains 3 clusters, two of which are nonlinear separable. The $M_{k\text{-}NN}$G-DPC algorithm is obviously more efficient on five indexes than all other algorithms. On the other hand, the standard DPC, K-means, FCM, spectral clustering, AP, and GMM can correctly allocate most of the instances to their clusters, so the clustering results are not very bad. The Seeds is a 7-dimensional dataset containing 210 instances which belong to three different clusters. Experimental results show that both the $M_{k\text{-}NN}$G-DPC and the standard DPC are able to find the correct number of clusters. Because the size of this dataset is small and the distribution of instances in space is too uniform, the results of $M_{k\text{-}NN}$G-DPC are slightly less than the standard DPC in Acc, Purity and F-score, but larger than the standard DPC in NMI and ARI indexes. The results of the other algorithms are almost similar except for

Table 4 The comparison of purity and ARI for each clustering algorithms on synthetic datasets

Algorithm	Compound		Path-based1		Atom		Zelnik6	
	Purity	ARI	Purity	ARI	Purity	ARI	Purity	ARI
$M_{k\text{-}NN}$G-DPC	86.97	85.31	**100**	**100**	**100**	**100**	**100**	**100**
DPC	68.92	58.42	80.67	46.42	81.75	14.85	89.08	67.35
DBSCAN	**92.73**	**86.32**	70.98	17.42	**100**	**100**	86.36	20.18
K-MEANS	82.21	72.62	82.67	46.13	74.22	19.51	82.35	64.45
Spectral clustering	85.71	70.42	88.67	78.54	**100**	**100**	**100**	**100**
AP	90.47	71.77	77.33	45.82	84.38	9.09	85.29	64.22
FCM	66.42	52.25	77.00	46.50	75.75	22.97	85.71	65.06
GMM	90.23	62.10	76.67	43,98	99.63	98.50	98.74	96.84
Birch	81.70	63.22	77.00	45.38	85.88	73.85	82.77	68.18
Mean shift	90.23	75.28	80.67	46.58	80.25	37.25	83.61	59.75
Algorithm	Dim256		Dim1024		S3		S4	
	Purity	ARI	Purity	ARI	Purity	ARI	Purity	ARI
$M_{k\text{-}NN}$G-DPC	**100**	**100**	**100**	**100**	**85.60**	72.16	**80.18**	**69.08**
DPC	**100**	**100**	**100**	**100**	85.36	72.19	79.62	63.21
DBSCAN	**100**	**100**	**100**	**100**	39.30	7.56	33.42	2.18
K-MEANS	**100**	**100**	**100**	**100**	85.56	72.52	79.62	63.06
Spectral clustering	91.41	82.60	84.86	78.85	80.30	71.60	74.46	68.28
AP	**100**	**100**	**100**	**100**	**85.60**	**72.69**	79.22	62.68
FCM	**100**	64.12	**100**	70.06	79.32	68.81	79.70	63.35
GMM	N/A	N/A	N/A	N/A	64.72	55.68	57.28	44.79
Birch	**100**	75.42	**100**	**100**	61.32	12.52	60.16	8.65
Mean shift	**100**	**100**	**100**	**100**	82.00	67.65	74.36	58.54

Bold values indicate the best results

DBSCAN that obtains the worst effect. The Art dataset has 300 instances with four attributes and contains three clusters. The clustering effect of $M_{k\text{-}NN}$G-DPC is clearly better than other algorithms in all indexes.

The Libras movement and Ionosphere are high-dimensional datasets in which the Ionosphere is an imbalanced dataset having 351 instances (containing one positive class and one negative class [31]). The $M_{k\text{-}NN}$G-DPC prefers to assign those smaller clusters to the larger ones. Therefore, for the imbalanced data, the positive instances are possibly merged to the negative clusters, which generates that the accurate rate (Acc) is slightly less than the standard DPC. But the clustering effects are better than other approaches for all indexes. The Libras movement is a time-sequential dataset. The spectral clustering algorithm achieves the best results on the F-score and NMI indexes because the spectral is fit for time sequence [32]. The $M_{k\text{-}NN}$G-DPC has the best value on the Acc and ARI indexes.

The WDBC is also a multi-dimensional dataset containing 30 features and has binary clusters. The $M_{k\text{-}NN}$G-DPC achieves the greater improvement than the standard DPC for all indexes, especially the NMI index. Some other algorithms obtain good performances with the Acc and F-score indexes, including K-means, FCM, spectral clustering, AP, and GMM algorithms. The Glass data contains 6 clusters and 214 instances with ten features. The $M_{k\text{-}NN}$G-DPC has the best effect on the Acc and NMI indexes and has similar result with DPC on F-score index. Some other algorithms have the similar performances on this dataset, including spectral clustering, AP, FCM, GMM, while the Mean shift method has the worst experimental values. The sizes of the Waveform and Image Segmentation datasets are larger than other datasets. The Waveform contains noisy values, which is a challenge for most of the algorithms. The graph-based techniques should be more suitable for this challenge due to the capacity of handling noisy data. From the experiments on the Waveform, the $M_{k\text{-}NN}$G-DPC, DPC, spectral clustering, and GMM have good performances because they are all graph-based algorithms, among which the $M_{k\text{-}NN}$G-DPC is the best. The Image Segmentation is a kind of image data on which our proposed algorithm achieves the best effect with the Acc, Purity, ARI and F-score indexes and the comparative result for the NMI, which is said to be able to deal with those complicated data. The Ecoli is a biological dataset containing protein localization sites. Particularly, the Ecoli is an imbalanced dataset that has several positive clusters,

Table 5 Comparison of Acc, F-score and NMI for 10 clustering algorithms on UCI datasets

Algorithm	Iris			Seeds			Art			Ionosphere		
	Acc	F-score	NMI	Acc	F-score	NMI	Acc	F-score	NMI	Acc	F-score	NMI
$M_{k\text{-}NN}$G-DPC	**97.33**	**97.33**	**90.09**	90.48	90.28	**72.73**	**95.00**	**95.00**	**83.91**	71.23	**71.21**	**25.20**
DPC	88.67	88.33	76.97	**90.95**	**90.76**	71.61	92.00	91.92	79.82	**73.22**	70.70	13.06
DBSCAN	66.67	56.65	57.52	62.38	59.78	42.34	67.33	57.01	58.36	72.08	59.24	13.49
K-MEANS	88.67	88.53	73.64	89.05	89.05	67.34	87.00	87.00	70.08	70.94	69.91	12.64
Spectral clustering	90.00	89.83	76.96	90.00	89.98	69.56	92.00	91.90	80.46	N/A	N/A	N/A
AP	90.67	90.63	75.92	89.52	89.57	68.81	86.67	86.66	69.66	70.94	69.97	12.84
FCM	89.33	89.26	74.04	90.00	90.02	69.04	87.00	87.00	70.08	70.94	69.91	12.64
GMM	96.67	96.66	89.83	89.52	89.40	72.14	73.33	68.25	62.48	N/A	N/A	N/A
Birch	84.00	83.02	71.71	89.05	89.33	70.55	84.00	83.09	70.46	64.67	41.47	4.73
Mean shift	68.67	58.42	59.23	89.52	89.62	67.55	66.67	55.56	58.15	64.39	39.92	0.45
Algorithm	Libras movement			WDBC			Glass			Waveform		
	Acc	F-score	NMI	Acc	F-score	NMI	Acc	F-score	NMI	Acc	F-score	NMI
$M_{k\text{-}NN}$G-DPC	**51.67**	48.54	62.32	**95.43**	**95.10**	**72.02**	**68.69**	38.50	**53.13**	**65.30**	**65.27**	27.82
DPC	41.94	40.03	51.37	82.95	79.19	37.32	57.94	38.90	48.49	58.80	57.73	30.69
DBSCAN	30.00	30.85	34.86	63.09	39.57	0.53	54.67	41.29	39.84	N/A	N/A	N/A
K-MEANS	48.89	42.54	58.59	92.79	92.11	61.15	55.61	**41.35**	40.48	50.14	49.95	36.32
Spectral clustering	51.39	**49.78**	**64.55**	94.02	93.42	67.23	44.86	32.07	40.91	51.06	51.00	**37.00**
AP	45.56	44.55	57.13	92.44	91.66	60.24	58.88	37.80	49.64	53.24	52.77	36.95
FCM	18.33	10.19	21.59	92.79	92.18	60.81	53.74	30.99	43.45	49.30	48.97	32.91
GMM	48.61	48.72	57.65	95.25	91.12	71.09	48.60	21.96	35.07	64.94	63.22	32.01
Birch	28.61	27.40	32.10	67.14	50.17	6.83	43.46	23.45	25.14	53.72	53.55	34.80
Mean shift	33.61	28.78	43.14	63.09	39.57	0.53	35.98	11.28	5.05	45.28	44.60	19.88

Algorithm	Image segmentation			Ecoli		
	Acc	F-score	NMI	Acc	F-score	NMI
$M_{k\text{-}NN}$G-DPC	**64.55**	**63.00**	53.78	**82.74**	**56.37**	**67.13**
DPC	62.77	60.19	**56.76**	79.76	46.34	64.10
DBSCAN	41.04	33.18	32.40	61.61	23.01	41.08
K-MEANS	58.66	56.62	51.60	60.28	54.87	56.20
Spectral clustering	N/A	N/A	N/A	76.49	46.21	65.61
AP	56.57	48.66	48.03	77.98	46.35	63.14
FCM	62.38	62.37	51.19	53.57	43.35	48.04
GMM	N/A	N/A	N/A	N/A	N/A	N/A
Birch	42.34	31.00	34.16	76.49	51.52	65.74
Mean shift	40.34	31.11	31.98	63.99	41.49	43.26

Bold values indicate the best results

each of which only contains a few instances. The index values of the $M_{k\text{-}NN}$G-DPC exceed the counterparts of other algorithms. It is indicated that our proposed approach is able to handle the imbalanced problem and biological data. For those imbalanced problems, K-means, AP, FCM, and spectral clustering algorithms easily assign the positive clusters to the negative ones, thereby obtaining fewer clusters than the actual number.

The running time of DPC algorithm and $M_{k\text{-}NN}$G-DPC algorithm on UCI datasets are listed in Table 7. Most of the time cost of $M_{k\text{-}NN}$G-DPC algorithm is used to establish the mutual neighbor graph and merge the 'negative classes'. For the same dataset, the time cost of the neighbor graph established by different parameters k *is* the same. If the value of parameter k is small, the probability that the samples in the nearest neighbor graph are near to each other will decrease, which leads to generate more 'negative classes', thereby spending more time to merge 'positive classes' and 'negative classes'.

Table 6 Comparison of Purity and ARI for 10 clustering algorithms on UCI datasets

Algorithm	Iris		Seeds		Art		Ionosphere	
	Purity	ARI	Purity	ARI	Purity	ARI	Purity	ARI
$M_{k\text{-}NN}$G-DPC	**97.33**	**90.38**	90.48	73.65	**95.00**	**85.22**	71.23	22.04
DPC	88.67	68.64	**90.95**	75.45	92.00	77.23	**73.22**	20.79
DBSCAN	66.67	43.28	62.38	20.38	67.33	39.47	72.08	**53.98**
K-MEANS	88.67	71.63	89.05	70.49	87.00	67.78	70.94	17.76
Spectral clustering	90.00	70.49	90.00	71.46	92.00	75.92	N/A	N/A
AP	90.67	75.65	89.52	71.51	86.67	67.79	70.94	17.27
FCM	89.33	72.87	90.00	72.66	87.00	68.40	70.94	17.27
GMM	96.67	90.37	89.52	72.34	73.33	60.38	N/A	N/A
Birch	84.00	59.94	89.05	74.98	84.00	67.47	64.67	40.93
Mean shift	68.67	49.73	89.52	**76.39**	66.67	42.43	64.39	39.02

Algorithm	Libras movement		WDBC		Glass		Waveform	
	Purity	ARI	Purity	ARI	Purity	ARI	Purity	ARI
$M_{k\text{-}NN}$G-DPC	51.67	**39.17**	**95.43**	**74.32**	**68.69**	**43.93**	**65.30**	**31.64**
DPC	60.00	30.12	82.95	42.04	68.22	22.56	58.80	26.52
DBSCAN	65.30	3.78	63.09	47.33	54.67	24.58	N/A	N/A
K-MEANS	**73.29**	30.75	92.79	73.02	55.61	27.44	50.14	25.35
Spectral clustering	63.29	34.38	94.02	72.38	68.34	10.84	64.92	20.19
AP	73.89	30.43	92.44	71.78	58.88	30.02	64.92	23.07
FCM	67.83	7.83	92.79	73.05	53.74	28.28	56.98	24.36
GMM	59.29	19.47	95.25	72.03	57.49	22.37	64.94	30.04
Birch	59.35	2.30	67.14	39.27	49.74	18.36	53.72	30.84
Mean shift	73.06	21.32	63.09	48.20	60.75	9.45	60.20	17.30

Algorithm	Image Segmentation		Ecoli	
	Purity	ARI	Purity	ARI
$M_{k\text{-}NN}$G-DPC	**73.77**	**58.89**	**92.56**	**71.58**
DPC	62.77	51.28	79.76	63.28
DBSCAN	69.23	2.93	61.61	35.67
K-MEANS	58.66	45.38	60.28	38.82
Spectral clustering	N/A	N/A	76.49	55.28
AP	56.57	24.23	77.98	63.98
FCM	62.38	41.60	53.57	41.19
GMM	N/A	N/A	N/A	N/A
Birch	64.39	17.95	76.49	50.51
Mean shift	69.30	44.86	63.99	41.38

Bold values indicate the best results

Table 7 The running time of DPC algorithm and $M_{k\text{-}NN}$G-DPC algorithm on UCI datasets

Dataset	Iris	Seeds	Art	Ionosphere	Ecoli	Waveform
DPC	0.048	0.048	0.049	0.053	0.051	2.875
$M_{k\text{-}NN}$G-DPC	0.085	0.094	0.106	0.189	0.121	3.371

Dataset	Libras movement	WDBC	Glass	Image segmentation
DPC	0.059	0.065	0.049	2.031
$M_{k\text{-}NN}$G-DPC	0.129	0.195	0.111	2.936

6 Conclusion and future work

As a widely used application field, clustering analysis is becoming more and more important, especially in the big data era. As the accumulation of data, the huge amounts of data mean that it is hard to obtain the training data because the enough labels and the actual number of classes might be hard to obtain, so that 'labeled' methods cannot be used normally. On the other hand, although there has been a great deal of clustering techniques, most of them are the specific data-oriented. That is, the generalization of many clustering algorithms is poor.

In this paper, we apply some classical and well-performed clustering algorithms, which can achieve good generalization, to perform the comparative analysis. In this study, we focus on DPC-based approach. We emply a mutual *k*-nearest-neighbor graph-based structure. The experimental analysis shows that, when the basic mutual *k*-nearest-neighbor graph is applied to DPC algorithm, the effects are not ideal and even much poor. Therefore, we improve the basic mutual *k*-nearest-neighbor graph to lift the DPC algorithm. This approach is to constrain the cluster assignment for data instances, which can distinguish the cluster membership of each instance more efficiently according to the densities of nodes in a graph. Typically, this technique can avoid such a case that the instances belonging to different densities of clusters are misclassified into the same cluster. It not only lifts the capacity of the DPC, but improve the performance of clustering the arbitrary shaped clusters.

The DPC-based algorithms are stable and robust in clustering many kinds of data. Our future work is to consider other complex data, such as Web data stream, video data, and DNA data, to be clustered by a series of the DPC-based algorithms.

Acknowledgements We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by Shandong Provincial Natural Science Foundation of China under Grant ZR2018MF009, the State Key Research Development Program of China under Grant 2017YFC0804406, the National Natural Science Foundation of China under Grant 61433012, and 61303167, the Special Funds of Taishan Scholars Construction Project, and Leading Talent Project of Shandong University of Science and Technology.

References

1. Han J, Kamber M, Pei J (2012) Data mining: concepts and techniques, 3rd edn. Morgan Kaufmann, Burlington
2. Bishop CM (2006) Pattern recognition and machine learning (information science and statistics). Springer, New York
3. Chifu AG, Hristea F, Mothe J, Popescu M (2015) Word sense discrimination in information retrieval: a spectral clustering-based approach. Inf Process Manag 51(2):16–31
4. Kaufman L, Rousseeuw P (2009) Finding groups in data: an introduction to cluster analysis. Wiley, Hoboken
5. Kearns M, Mansour Y, Ng AY (1999) An information-theoretic analysis of hard and soft assignment methods for clustering. In: Jordan MI (ed) Learning in graphical models. MIT Press, Cambridge, pp 495–520
6. Forgy EW (1965) Cluster analysis of multivariate data: efficiency versus interpretability of classifications. Biometrics 21:768–769
7. Bishnu PS, Bhattacherjee V (2013) A modified K-modes clustering algorithm. pattern recognition and machine intelligence, Volume 8251 of the series Lecture Notes in Computer Science, 2013, pp 60–66
8. Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: an efficient data clustering method for very large databases. In: Proceedings of ACM SIGMOD international conference on management of data, 1996, pp 103–114
9. Karypis G, Han E, Kumar V (1999) CHAMELEON: a hierarchical clustering algorithm using dynamic modeling. IEEE Comput 32(8):68–75
10. Fan J (2015) OPE-HCA: an optimal probabilistic estimation approach for hierarchical clustering algorithm. Neural Comput Appl. https://doi.org/10.1007/s00521-015-1998-5
11. Ester M, Kriegel H, Sander J, Xu X, Simoudis E, Han J, Fayyad UM (eds) (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the second international conference on knowledge discovery and data mining (KDD-96). AAAI Press, pp 226–231
12. Hinneburg A, Gabriel HH (2007) DENCLUE 2.0: fast clustering based on kernel density estimation. In: Proceedings of the 2007 international conference on intelligent data analysis (IDA'07), Ljubljana, Slovenia, 2007, pp 70–80
13. Banerjee A, Shan H (2010) Model-based clustering. In: Sammut C, Webb GI (eds) Encyclopedia of machine learning, pp 686–689
14. Ding S, Zhang N, Zhang J, Xu X, Shi Z (2017) Unsupervised extreme learning machine with representational features. Int J Mach Learn Cybern 8(2):587–595
15. Du M, Ding S, Xu X, Xue Y (2017) Density peaks clustering using geodesic distances. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-017-0648-x
16. Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. Science 344(6191):1492–1496
17. Frey BJ, Dueck D (2007) Clustering by passing messages between data points. Science 315(5814):972–976
18. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
19. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
20. Arias-Castro E, Chen G, Lerman G (2011) Spectral clustering based on local linear approximations. Electron J Stat 5(1):1537–1587
21. Székely GJ, Rizzo ML (2005) Hierarchical clustering via Joint between-within distances: extending ward's minimum variance method. J Classif 22(2):151–183
22. Figueiredo MAT, Jain AK (2002) Unsupervised learning of finite mixture models. IEEE Trans Pattern Anal Mach Intell 24(3):381–396
23. McLachlan GJ, Peel D (2000) Finite mixture models. Wiley, New Jersey
24. Fu G (1998) Optimization methods for fuzzy clustering. Fuzzy Sets Syst 93(3):301–309
25. Nayak J, Naik B, Behera HS (2014) Fuzzy C-means (FCM) clustering algorithm: a decade review from 2000 to 2014. Computational Intelligence in Data Mining-Volume 2, Volume 32

of the series Smart Innovation, Systems and Technologies, pp 133–149
26. Brito MR, Chávez EL, Quiroz AJ, Yukich JE (1997) Connectivity of the mutual *k*-nearest-neighbor graph in clustering and outlier detection. Stat Probab Lett 35(1):33–42
27. Sardana D, Bhatnagar R (2014) Graph clustering using mutual K-nearest neighbors. Active Media Technology, Volume 8610 of the series Lecture Notes in Computer Science, pp 35–48
28. Xie J, Gao H, Xie W (2016) K-nearest neighbors optimized clustering algorithm by fast search and finding the density peaks of a dataset. Sci Sin Inf 46(2):258–280
29. Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml. University of California, School of Information and Computer Science, Irvine
30. Cover TM, Thomas JA (2001) Elements of information theory. Wiley, Hoboken
31. Fan J, Niu Z, Liang Y, Zhao Z (2016) Probability model selection and parameter evolutionary estimation for clustering imbalanced data without sampling. Neurocomputing 211(10):172–181
32. Wang F, Zhang C (2005) Spectral clustering for time series. In: Proceedings of third international conference on advances in pattern recognition, ICAPR 2005, Bath, UK, August 22–25, 2005, pp 345–354
33. Xu X, Ding S, Du M, Xue Y (2018) DPCG: an efficient density peaks clustering algorithm based on grid. Int J Mach Learn Cybern 9(5):743–754
34. Du M, Ding S, Xue Y (2018) A robust density peaks clustering algorithm using fuzzy neighborhood. Int J Mach Learn Cybern 9(7):1131–1140
35. Bai X, Yang P, Shi X (2017) An overlapping community detection algorithm based on density peaks. Neurocomputing 226(2):7–15
36. Campello RJGB, Moulavi D, Sander J (2013) Density-based clustering based on hierarchical density estimates. In: Pei J, Tseng VS, Cao L, Motoda H, Xu G (eds) Advances in knowledge discovery and data mining. Lecture notes in computer science. Springer, Berlin Heidelberg, pp 160–172
37. Li J, Huang X, Selke C, Yong J (2007) A fast algorithm for finding correlation clusters in noise data. In: Proceedings of the 11th Pacific-Asia conference on knowledge discovery and data mining, pp 639–647
38. Zhang T-T, Yuan B (2018) Density-based multiscale analysis for clustering in strong noise settings with varying densities. IEEE Access 6:25861–25873
39. Zhang H, Wang S, Xu X, Chow TWS, Wu QMJ (2018) Tree2Vector: learning a vectorial representation for tree-structured data. IEEE Trans Neural Netw Learn Syst 29(11):5304–5318
40. Wang X, Xing H-J, Li Y et al (2015) A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning. IEEE Trans Fuzzy Syst 23(5):1638–1654
41. Wang R, Wang X, Kwong S, Chen X (2017) Incorporating diversity and informativeness in multiple-instance active learning. IEEE Trans Fuzzy Syst 25(6):1460–1475
42. Wang X, Wang R, Chen X (2018) Discovering the relationship between generalization and uncertainty by incorporating complexity of classification. IEEE Trans Cybern 48(2):703–715
43. Wang X, Zhang T, Wang R (2019) Non-iterative deep learning: incorporating restricted boltzmann machine into multilayer random weight neural networks. IEEE Trans Syst Man Cybern Syst 49(7):1299–1380
44. Lin JCW, Yang L, Fournier-Viger P, Hong TP (2018) Mining of skyline patterns by considering both frequent and utility constraints. Eng Appl Artif Intell 77:229–238
45. Fournier-Viger P, Lin JCW, Kiran RU, Koh YS, Thomas R (2017) A survey of sequential pattern mining. Data Sci Pattern Recognit 1(1):54–77
46. Chen CM, Xiang B, Liu Y, Wang KH (2019) A secure authentication protocol for internet of vehicles. IEEE ACCESS 7(1):12047–12057
47. Chen CM, Xiang B, Wang KH, Yeh KH, Wu TY (2018) A robust mutual authentication with a key agreement scheme for session initiation protocol. Appl Sci 8(10):1
48. Yang C, Huang L, Li F (2018) Exponential synchronization control of discontinuous non-autonomous networks and autonomous coupled networks. Complexity 1:1–10
49. Lian D, Xianwen F, Chuangxia H (2017) Global exponential convergence in a delayed almost periodic nicholsons blowflies model with discontinuous harvesting. Math Methods Appl Sci 41(5):1954–1965
50. Lian D, Lihong H, Zhenyuan G (2017) Periodic attractor for reactiondiffusion high-order hopfield neural networks with timevarying delays. Comput Math Appl 73(2):233–245
51. Huang C, Liu B, Tian X, Yang L, Zhang X (2019) Global convergence on asymptotically almost periodic SICNNs with nonlinear decay functions. Neural Process Lett 49(2):625–641
52. Huang C, Zhang H, Huang L (2019) Almost periodicity analysis for a delayed Nicholson's blowflies model with nonlinear density-dependent mortality term. Commun Pure Appl Anal 18(6):3337–3349
53. Huang C, Zhang H (2019) Periodicity of non-autonomous inertial neural networks involving proportional delays and non-reduced order method. Int J Biomath 12(02):1950016
54. Huang C, Cao J, Wen F, Yang X (2016) Stability analysis of SIR model with distributed delay on complex networks. PLoS One 11(8):e0158813
55. Li Y, Fan JC, Pan JS, Mao GH, Wu GK (2019) A novel rough fuzzy clustering algorithm with a new similarity measurement. J Internet Technol 20(4):1
56. Fan J-C, Li Y, Tang Lei-Yu, Geng-Kun W (2018) RoughPSO: rough set-based particle swarm optimisation. Int J Bio-Inspired Comput 12(4):245–253