International Journal of Machine Learning and Cybernetics (2021) 12:2825–2841
https://doi.org/10.1007/s13042-021-01369-7

ORIGINAL ARTICLE

A novel density peaks clustering algorithm based on K nearest neighbors with adaptive merging strategy

Xiaoning Yuan[1] · Hang Yu[1] · Jun Liang[2] · Bing Xu[2]

Received: 21 October 2020 / Accepted: 22 June 2021 / Published online: 3 August 2021

Abstract

Recently the density peaks clustering algorithm (DPC) has received a lot of attention from researchers. The DPC algorithm is able to find cluster centers and complete clustering tasks quickly. It is also suitable for different kinds of clustering tasks. However, deciding the cutoff distance d_c largely depends on human experience which greatly affects clustering results. In addition, the selection of cluster centers requires manual participation which affects the efficiency of the algorithm. In order to solve these problems, we propose a density peaks clustering algorithm based on K nearest neighbors with adaptive merging strategy (KNN-ADPC). A clusters merging strategy is proposed to automatically aggregate over-segmented clusters. Additionally, the K nearest neighbors are adopted to divide data points more reasonably. There is only one parameter in KNN-ADPC algorithm, and the clustering task can be conducted automatically without human involvement. The experiment results on artificial and real-world datasets prove higher accuracy of KNN-ADPC compared with DBSCAN, K-means++, DPC, and DPC-KNN.

Keywords Density peaks · Clustering algorithm · K nearest neighbors · Adaptive merging

1 Introduction

Clustering algorithm is one of the most important machine learning algorithms which has been widely applied in many fields, such as data mining and chemical industry. The basic idea of clustering algorithm is that those data with high similarity should be divided into the same cluster while data with low similarity should be divided into different clusters [1].

There are mainly three classic clustering algorithms: density-based clustering [2], partition-based clustering [3] and hierarchical clustering [4]. In recent years, many new clustering algorithms have been proposed, such as spectral clustering [5], multi-kernel clustering [6], multi-view clustering [7], subspace clustering [8], ensemble clustering [9], and deep embedded clustering [10]. The drawback of these new clustering algorithms is that both complexity and computation costs are larger than classical clustering algorithms.

Among all clustering algorithms, K-means [11] and DBSCAN [12] are the most classic methods. K-means is one of the most famous partition-based methods. The clustering process begins with selecting K initial center points, then iteratively assigning the remaining points into its nearest cluster. The compact idea of K-means allows it to complete clustering tasks quickly. However, the clustering result is vulnerable to the selection of initial center points while K-means++ [13] can partially solve this problem. Besides, both K-means and K-means++ are inadequate in dealing with the non-spherical cluster. DBSCAN is one of the most popular density-based clustering algorithms. The density-based clustering algorithm is suitable for the non-spherical clustering tasks. The basic idea of DBSCAN is that clusters are decided according to the density connection relationship.

✉ Bing Xu
bingxu@zju.edu.cn

Xiaoning Yuan
yuanxiaoning@cgnpc.com.cn

Hang Yu
yuhang@cgnpc.com.cn

Jun Liang
jliang@zju.edu.cn

[1] State Key Laboratory of Nuclear Power Safety Monitoring Technology and Equipment, China Nuclear Power Engineering Co., Ltd, Shenzhen 518172, Guangdong, China

[2] State Key Lab of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, Zhejiang, China

In DBSCAN, points are divided into the core objects and the noise points. Then core objects are aggregated to the same cluster if they are density reachable. Nevertheless, using DBSCAN, researchers need to predefine two hyperparameters for screening core objects, and the optimal hyperparameters are usually difficult to define in practice.

The density peaks clustering algorithm (DPC) [14] was proposed by Rodriguez A in 2014, which attracts great attention from plenty of researchers. DPC can deal with clusters of different shapes. It is mainly based on two basic assumptions: (1) the cluster center is surrounded by other low density points; (2) the cluster center is far from other cluster centers. With these two basic assumptions, it is easy and fast for DPC to find cluster centers and complete clustering task. The core idea of DPC is calculating the local density ρ_i and the distance from the higher density points δ_i to draw a decision graph. Cluster centers can be selected according to decision graph. The remaining points are assigned to the cluster to which its nearest higher density point belongs. Although DPC is very concise and efficient, there are still some shortcomings. For example, the selection of cluster centers depends on human experience which greatly limits its autonomy. Furthermore, DPC can not deal with the clustering task that one cluster has more than one high density center points. In addition, although the assignment rules in DPC are very efficient, the domino effect will occur when there occurs misclassification in the process.

Aiming at drawbacks in DPC, a lot of improved clustering algorithms based on DPC are proposed. FKNN-DPC [15] improves two assignment strategies to overcome the drawbacks in the assignment rules of DPC. However, the selection of cluster centers in FKNN-DPC is still the same as in DPC, and it also requires manual participation. CFSFDP+A [16] is proposed to accelerate the calculation process of distance between points. Nevertheless, the clustering process remains unchanged. In order to solve the disadvantage of one-step allocation strategy in DPC, a shared-nearest-neighbor-based clustering algorithm (SNN-DPC) is proposed. SNN-DPC [17] uses a two-step allocation strategy to ensure the correct assignment of points. However, SNN-DPC is far more complex than DPC and also needs manual participation. In DPC-KNN [18], an allocation strategy using K nearest neighbors is proposed. The calculation of the distance from higher density points is more reasonable, but it is still lacks autonomy.

In this paper, we propose a density peaks clustering algorithm based on K nearest neighbors with adaptive merging strategy (KNN-ADPC). The strategy of K nearest neighbors is used to calculate the distance δ_i and data points assignment. In addition, a novel adaptive merging strategy is proposed to solve the potential problem of over-segmentation. The main innovations of KNN-APDC algorithm are listed as follows:

1. The K nearest neighbors are introduced to the assignment rules which is more reasonable than the original assignment rules in DPC.
2. The KNN-ADPC only has one hyperparameter which boosts the efficiency of determining parameters to a large extent.
3. The KNN-ADPC has a high degree of autonomy without losing accuracy. No need for human involvement, the adaptive merging strategy we proposed still has a good performance in non-spherical clustering tasks.
4. A creative and effective cluster automatic merging strategy is proposed to solve the over-segmentation problem and correct clustering results.

The rest of this paper is organized as follows. The details of DPC and KNN-DPC are described in Sect. 2. Section 3 introduces how KNN-ADPC algorithm works. The experiment results are given in Sect. 4. Then discussions are made in Sect. 5 according to the experiment results. Finally, the paper ends with some conclusions and perspectives in Sect. 6.

2 Related works

In this section, we briefly review the original DPC and DPC-KNN algorithm.

2.1 DPC: density peaks clustering

DPC is a novel density peaks clustering algorithm that finds cluster centers quickly and has good adaptability to a variety of clustering tasks. For each point p_i.in dataset X, DPC computes the local density ρ_i and the distance δ_i from points with higher density to build a decision graph for cluster centers selection. The calculation of ρ_i and δ_i is defined by Eqs. (1) and (2).

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

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

where d_{ij} is the distance between the two different points p_i and p_j, d_c is the cutoff distance parameter given by user. $\chi(\bullet)$ is the indicative function where $\chi(\bullet) = 1$ if $(\bullet)<0$ and $\chi(\bullet) = 0$ otherwise. For the points with highest local density, its δ_i is specified as the maximum distance between two points which can be written as $\delta_i = \max_j(d_{ij})$. However, the original calculation of local density in Eq. (1) will be confusing in some situations when d_c is not given properly. For instance, as shown in Fig. 1, the red point in case 1 and case

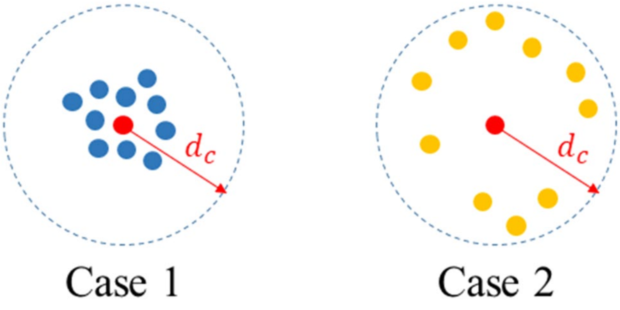


Fig. 1 Density contrast of two points

2 both have 10 points within d_c range. If we calculate the local density follow Eq. (1), we will get a conclusion that their local densities are the same. However, the local density of the red point in case 1 is obviously higher than in case 2. Although this problem can be solved by constantly adjusting d_c, the probability of its occurrence is still very high.

In order to solve this problem, another local density calculation method is used in DPC which is defined by Eq. (3).

$$\rho_i = \sum_{j\in I_S\setminus\{i\}} \exp(-\frac{d_{ij}^2}{d_c^2}) \tag{3}$$

where I_S is the set of all points whose distance from point p_i is less than d_c.

2.2 DPC-KNN: density peaks clustering based on k nearest neighbor

In order to solve the problem of misclassification of original DPC on some circular clustering tasks, DPC-KNN adopts K nearest neighbor method to compute the distance δ_i as Eq. (4). Compared with the calculation of δ_i in DPC, Eq. (4) expands the scope of distance calculation which treats the set of K nearest neighbors of each point as a whole, considering that the K nearest neighbors of each point can represent its internal structure more comprehensively than a single point.

$$\delta_i = \begin{cases} \max\limits_j(d_{ij}), & if \quad \rho_i = \max(\rho) \\ \min\limits_{\substack{k\in\{KNN_i, x_i\},\\ j:\,\rho_j>\rho_i}}(d_{kj}), & otherwise \end{cases} \tag{4}$$

where KNN_i is the K nearest neighbor points set of point p_i defined as Eq. (5). K is the number of nearest neighbors. In DPC-KNN, the calculation of local density ρ_i is still the same as in the original DPC. However, the cluster centers selection method and points assignation strategy in DPC-KNN remain the same, which means when using DPC-KNN, researchers still need to manually deicide the parameter d_c and select cluster centers.

$$KNN_i = \{x_j | \min_K(d_{ij}),\ x_i, x_j \in \mathrm{X},\ x_i \neq x_j\} \tag{5}$$

3 Methods

Although there are many improved clustering algorithms based on DPC, there are still some drawbacks like requiring manual participation and low clustering accuracy. In order to solve these problems, we proposed a density peaks clustering algorithm based on KNN-ADPC. KNN-ADPC consists of three main steps: (1) cluster centers selection; (2) remaining points assignation; (3) clusters merging.

3.1 Cluster centers selection

As mentioned above, the choice of parameter d_c greatly affects clustering results. Aiming at this problem, a new approach of calculating d_c based on the internal structure of the data [19] is raised.

As suggested in DPC, d_c can be chosen so that the average number of neighbors of each point is approximately 1–2% of the whole dataset. Inspired by it, we can choose d_c according to the original data structure in which the tightness around points can be described by its K nearest neighbors. The calculation method of d_c introduces the concept of K nearest neighbors defined as Eq. (6).

$$d_c = \mu^K + \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\delta_i^K - \mu^K)^2} \tag{6}$$

where N is the number of all sample points in the data, δ_i^K is the distance between the points p_i and its Kth nearest neighbor points defined as Eq. (7), μ^K is the average value of each δ_i^K defined as Eq. (8).

$$\delta_i^K = \max_{j\in KNN_i}(d_{ij}) \tag{7}$$

$$\mu^K = \frac{1}{N}\sum_{i=1}^{N}\delta_i^K \tag{8}$$

In Eq. (6), μ^K represents the average degree of dispersion of all points. The calculation of the second part is similar to the standard deviation which can reflect the volatility between δ_i^K of each point and μ^K. Thus merging these two parts can help us appropriately estimate the structure of the whole data so that we can choose a proper d_c for cluster centers.

After calculating the cutoff parameter d_c, we can compute the local density ρ_i.for every point based on Eq. (3). Considering that the calculation method of the distance δ_i in original DPC leading to chain misclassification easily, we also adopt the K nearest neighbors to calculate δ_i based on Eq. (4).

Considering that d_c is mainly used to determine the local density of each point. In order to verify the rationality and effectiveness of d_c, we conduct comparative experiments with different local density calculation methods on Jain [20] which has uneven density. The calculation of ρ_i in this paper [21] is free from human efforts and d_c setting. The local density is defined as Eq. (9).

$$\rho_i = \exp(-(\frac{1}{k}\sum_{x_j \in KNN(x_i)} d(x_i, x_j)^2)) \tag{9}$$

The clustering results of our proposed KNN-ADPC using two different density calculation methods are shown in Fig. 2. With Eq. (9), we prefer to use dense points as cluster centers which are more likely to cause sparse clusters to be merged into dense clusters. Additionally, we can identify the center points of both low-density clusters and high-density clusters because that d_c comprehensively considers the internal structure information of the entire data.

To ensure that all cluster centers can be filtered, a looser selection strategy is implemented. For each point, we select the points whose δ_i is larger than the cutoff distance d_c as initial cluster centers. In this step, neither drawing decision graph nor manual participation is necessary.

3.2 Remaining points assignation

Here we demonstrate how chain misclassification can be caused in original DPC. For example, as shown in Fig. 3, point 2 and point 3 are the higher density points of point 1. This distance between point 1 and point 3 is closer than the distance between point 1 and point 2. So if the original assignation strategy is adopted, point 1 will be assigned to the cluster to which point 3 belongs. And then chain misclassification is likely to occur. Hence we adopt the same calculation method to compute δ_i as in DPC-kNN. The new calculation method guarantees the K nearest neighbors to be considered when assigning points. After selecting initial cluster centers, the remaining points will be assigned to their nearest higher density points according to δ_i.

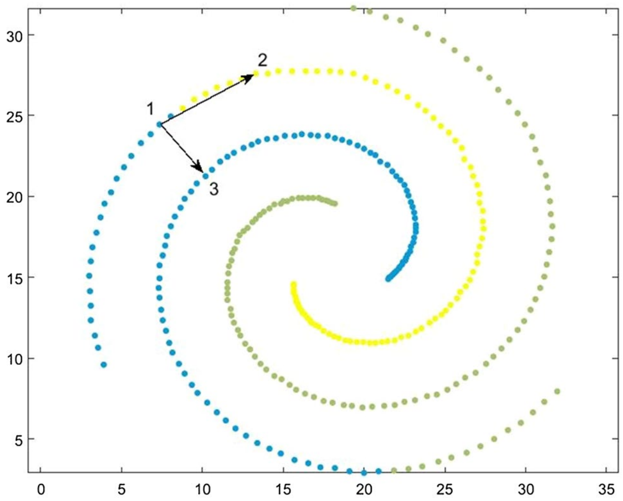


Fig. 3 Misallocation diagram

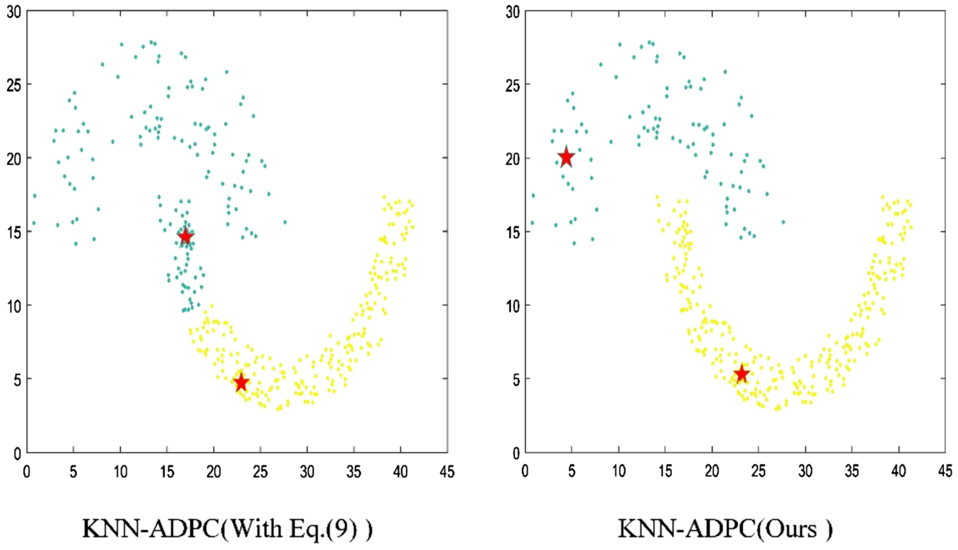


Fig. 2 Comparison of KNN-ADPC with different ρ_i calculation

3.3 Clusters merging

To avoid over-segmentation caused by loose initial cluster centers selection strategy, an adaptive merging strategy according to the density difference and distance between two clusters is brought forward. By analyzing clusters shape, we find that two clusters that need to be merged generally have two characteristics: (1) two clusters have close neighbor border. (2) there is usually less density difference between two clusters. In order to further clarify the relationship between clusters, we make some definition declarations in advance.

Definition 1 (Adjacent regions between two clusters) *The adjacent regions between two clusters $border(p,q)$ can be denoted as Eq.* (10).

$$border(p,q) = \{(x_i, x_j)|d_{ij} < d_c, x_i \in S_p, x_j \in S_q), \ \ p \neq q\} \tag{10}$$

where S_p and S_q are two different clusters which can be defined as $S_p = \{x_i|c|\left(x_i\right) = p, \quad x_i \in X\}$ and $S_q = \{x_j|c|(x_j) = q, \quad x_j \in X\}$.

Definition 2 (The border region of one single cluster) *The border region of one single cluster B refers to the set of all points in the cluster who have adjacencies with other clusters. B can be denoted as Eq.* (11).

$$B(p) = \cup_{p\neq q}\{x_i|x_i \in border(p,q), \ \ x_i \in S_p\} \tag{11}$$

Definition 3 (The local density and boundary density of one single cluster) *The local density of one single cluster ρ_C is the average local density of all points in the cluster which can be denoted as Eq.* (12). *And the boundary density of one single cluster $\rho_B(p)$ is the average local density of all points who is in the border region of the cluster which can be denoted as* Eq. (13).

$$\rho_C(p) = mean(\rho_i|x_i \in S_p) \tag{12}$$

$$\rho_B(p) = mean(\rho_i|x_i \in B(p)) \tag{13}$$

Definition 4 (The density difference between two clusters) *Considering that high density cluster generally merges the lower one, we define the density difference between the boundary density of the high density cluster and local density of the low density cluster as density difference $diff(p,q)$ of these two clusters S_p and S_q. The calculation method can be denoted as Eq.* (14).

$$diff(p,q) = \frac{(\rho_B(p) - \rho_C(q))^2}{\max(\rho_B(\rho), \ \ \rho_C(q))}, \ \ if \ \ \rho_C(p) > \rho_C(q) \tag{14}$$

As for clusters that have no adjacent regions, their density difference $diff(p,q)$ are set as $+\infty$.

Definition 5 (Density directly-reachable) *These density directly-reachable clusters should satisfy two constraints: (1) There should be at least one pair of adjacent points between two clusters (2) The density difference between two clusters should be less than the average value of the density difference between all clusters. These two rules for S_p and S_q can be denoted as* Eqs. (15) *and* (16).

$$border(p,q) \geq 2 \tag{15}$$

$$diff(p,q) < mean(diff(u,v)|diff(u,v) \neq +\infty) \tag{16}$$

Under this circumstance, we consider cluster S_p and S_q density directly-reachable. In addition, the density directly-reachable relationship is symmetrical.

Definition 6 (Density reachable) *We consider two clusters S_p and S_q density reachable if there exists $S_1 = S_p, S_2, \ldots, S_n = S_q$ where S_{i+1} is density directly-reachable to S_i.*

Similar to the assumption proposed in the paper [22] that "points in the same high-density area or the same structure are likely to have the same label". We think that the average density of each cluster can represent the internal structure itself. On one hand, the prerequisite of merging is that two clusters should have border points with each other. On the other hand, according to our observation, densities of close points in the same cluster are generally continuous and smooth. Therefore, we propose a method to calculate the density difference between each two clusters whose local density are different. Besides, when the threshold of the parameter diff(p,q) being set as the average density difference between two clusters, it obtains the best performance in our experiments.

After performing assignation in Sect. 3.2, a preliminary clustering result is given. Firstly, we can determine the adjacent region between each two clusters and the border region of each cluster according to Eqs. (10) and (11). Secondly, the local density and boundary density of each cluster is calculated with Eqs. (12) and (13). Thirdly, the density difference between two clusters is obtained by Eq. (14). Finally, the density reachable relationship between two clusters can be judged. Cluster pairs that are density reachable to others will be merged.

Table 1 Pseudocode of KNN-ADPC

```
Input: dataset $X$(Containing $N$ points), parameter $K$(K nearest neighbors)
Calculate the Euclidean distance matrix and sorting the distance vector based on fast sorting:
    Calculate the cutoff distance $d_c$:
        $\delta_i^K \leftarrow \max_{j\in KNN_i}(d_{ij})$
        $\mu^K \leftarrow \frac{1}{N}\sum_{i=1}^{N} \delta_i^K$
        $d_c \leftarrow \mu^K + \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\delta_i^K - \mu^K)^2}$
    Calculate the local density $\rho_i$ and the distance $\delta_i$ for each point and sorting the density array in descending order:
        $\rho_i \leftarrow \sum_{j\in I_S\{i\}} exp(-\frac{d_{ij}^2}{d_c^2})$
        $\rho_{sorted} = sort(\rho,'descend')$
        $\delta_i \leftarrow \{ \begin{array}{l} \max_j(d_{ij}), if\ \rho_i = max(\rho_i) \\ \min_{K\in\{KNN_i,x_i\},j:\rho_j>\rho_i}(d_{kj}), otherwise \end{array}$
    Select the points whose $\delta_i$ is larger than the cutoff distance $d_c$ as the initial cluster centers and points tags:
    Initialize the clusters num and point tags: $centers = 0,\ tags = -ones(1,\ N)$
    $\boldsymbol{for}\ i = 1,2,\ldots,N\ \boldsymbol{do}$
        $\boldsymbol{if}\ \delta_i > d_c\ \boldsymbol{then}$
            $centers \leftarrow centers + 1$
            $tags_i \leftarrow centers$
        $\boldsymbol{end\ if}$
    $\boldsymbol{end\ for}$
    Assign the remaining points to the cluster to which its nearest higher density points belongs based on our new $\delta_i$:
    $\boldsymbol{for}\ i = 1,2,\ldots,N\ \boldsymbol{do}$
        $\boldsymbol{if}\ tags(\rho_{sorted}(i) == -1)\ \boldsymbol{then}$
            $\boldsymbol{for}\ x_j\ in\ \{x_j|\rho_j > \rho_i\}\ \boldsymbol{do}$
                $\boldsymbol{if}\ dist_{ij} = mi\ n(dist_{ij}|\rho_j > \rho_i)\ \boldsymbol{then}$
                    $tags(\rho_{sorted}(i)) \leftarrow tags(j)$
                $\boldsymbol{end\ if}$
            $\boldsymbol{end\ for}$
```

Table 1 (continued)

```
        end if
end for
Merge all the density reachable clusters according to the proposed adaptive merging strategy:
Determine the adjacent region between every two clusters and the border region of each single cluster according to Eq. (9) and Eq. (10)
Calculate the local density and boundary density of each single cluster according to Eq. (11) and Eq. (12)
Calculate the density difference between every two clusters according to Eq. (13)
for $cl_1 = 1,2,\ldots, centers$ then
    for $cl_2 = cl_1 + 1,\ldots, centers$ then
        if $(border(cl_1, cl_2) \geq 2$ and $diff(cl_1, cl_2) < mean(diff(u,v)|diff(u,v)) \neq +\infty)$ then
            $tags_{x_i \in \{cl_1, cl_2\}} \leftarrow min(cl_1, cl_2)$
        end if
    end for
end for
Merge the density reachable clusters according to definition 6
Output: point tags $tags$
```

3.4 Algorithm flow

The algorithm flow of the proposed KNN-ADPC is shown in Table 1.

3.5 The time complexity analyses of KNN-ADPC

Assuming that there are N points in dataset X and C denotes the number of clusters. The time complexity of KNN-ADPC mainly determined by the following steps: (1) computing the distance matrix ($O(N^2)$); (2) sorting the distance vector with fast sorting ($O(NlogN)$); (3) computing the cutoff distance d_c ($O(N)$); (4) computing the local density ρ_i ($O(N)$); (5) Computing the distance δ_i based on K nearest neighbors ($O(KN^2)$), here exists$K \ll N$, so the time complexity of this step is$O(N^2)$; (5) selecting the initial cluster centers and assigning the remaining points ($O(N^2)$); (6) determining the adjacent region between clusters ($O(N^2)$). The complexity reaches to top when the data is divided into multiple clusters with single point in each cluster (i.e., N clusters); (7) computing the boundary density for each cluster ($O(N^2)$).

According to the above analysis, the overall time complexity of KNN-ADPC is $O(N^2)$ which is the same as DPC.

3.6 The space complexity of KNN-ADPC

For the proposed KNN-ADPC, there are some major steps requiring storage space. When computing the local density ρ_i and the distance δ_i of each point, we need $2N$ spaces. And we also need KN space to store the K nearest neighbors of each point. During merging step, spaces of storing adjacent region between two clusters are required. Theoretically, N points can generate maximum N^2 pairs when there are N clusters. Since the number of border points is usually far less than N, the space complexity of this step is at most $O(N^2)$.

In conclusion, the space complexity of the proposed KNN-ADPC is $O(N^2)$, which is the same as DPC.

Table 2 The details of the datasets

Datasets	Points	Dimensions	Clusters
Flame [23]	240	2	2
Jain [20]	373	2	2
Aggregation [24]	788	2	7
Spiral [25]	312	2	3
R15 [26]	600	2	15
D4 [27]	1268	2	4
D9 [27]	1427	2	4
D31 [28]	3100	2	31
Wine [29]	178	13	3
Breast_wpbc [30]	699	10	2

4 Results

In this section, we conduct comparison experiments on artificial datasets and real-world datasets to evaluate the effectiveness of KNN-ADPC. Experiments are conducted on a desktop computer with a core i5 4210U-Intel 1.7 GHz processor and 8 GB RAM running MATLAB R2016A. The performance of KNN-ADPC is compared with DBSCAN, K-means++, DPC, and DPC-KNN. The details of the dataset are shown in Table 2. From first one to the eighth one are the artificial datasets and the last two are representatives of the real-world datasets.

4.1 Clustering evaluation metrics

We adopt clustering accuracy (ACC) to measure the performance of each algorithm. In addition, considering that there may be no labels in some actual clustering tasks, the performance can be evaluated according to whether the clustering result is consistent with similar points clustered into the same cluster while low similarity points are divided into different clusters. Therefore we also implement three evaluation metrics which are independent of the absolute values of labels: adjusted mutual information (AMI) [31], adjusted Rand index (ARI) [31], and Fowlkes-Mallows index (FMI) [32]. The upper limit and the lower limit of the above four evaluation indexes are 1 and − 1 respectively. And the larger these indexes are, the higher clustering accuracy these algorithms obtain.

4.2 Experiment on artificial and real-world dataset

Before conducting clustering task, we do preprocess for these real-world datasets. We replace missing values with the mean value of all valid values of same dimension. In

Table 3 Accuracy evaluation

Datasets	DBSCAN	K-means++	DPC	DPC-KNN	KNN-ADPC
Flame	0.3542	0.8417	**1.0000**	**1.0000**	**1.0000**
Jain	0.9351	0.7748	0.7400	**1.0000**	**1.0000**
Aggregation	0.7919	0.8084	**0.9987**	**0.9987**	**0.9987**
Spiral	1.0000	0.3397	**1.0000**	**1.0000**	**1.0000**
R15	0.9308	**0.9967**	**0.9967**	**0.9967**	**0.9967**
D4	**1.0000**	0.5213	**1.0000**	**1.0000**	**1.0000**
D9	0.9237	0.3315	0.3714	0.5452	**1.0000**
D31	0.96296	0.88097	0.96742	0.96548	**0.97452**
Wine	0.6487	**0.6910**	0.6854	0.6854	0.6742
Breast_wpbc	0.7703	0.6032	0.8038	**0.8551**	**0.8551**

Table 4 Adjusted mutual information evaluation

Datasets	DBSCAN	K-means++	DPC	DPC-KNN	KNN-ADPC
Flame	0.3542	0.8417	**1.0000**	**1.0000**	**1.0000**
Jain	0.7962	0.3236	0.1681	**1.0000**	**1.0000**
Aggregation	0.7974	0.8207	**0.9956**	**0.9956**	**0.9956**
Spiral	**1.0000**	-0.0050	**1.0000**	**1.0000**	**1.0000**
R15	0.9594	0.8853	**0.9938**	**0.9938**	**0.9938**
D4	**1.0000**	0.5031	**1.0000**	**1.0000**	**1.0000**
D9	0.8509	0.1553	0.2456	0.5626	**1.0000**
D31	0.93125	0.93343	0.95468	0.95317	**0.96503**
Wine	0.1904	0.3400	0.3383	0.3383	**0.3395**
Breast_wpbc	0.0424	0.0034	0.2924	**0.3882**	**0.3882**

Table 5 Adjusted rand index evaluation

Datasets	DBSCAN	K-means++	DPC	DPC-KNN	KNN-ADPC
Flame	0.3542	0.8417	**1.0000**	**1.0000**	**1.0000**
Jain	0.9484	0.3004	0.2464	**1.0000**	**1.0000**
Aggregation	0.8089	0.7744	**0.9978**	**0.9978**	**0.9978**
Spiral	**1.0000**	-0.0054	**1.0000**	**1.0000**	**1.0000**
R15	0.9244	**0.9928**	**0.9928**	**0.9928**	**0.9928**
D4	**1.0000**	0.3614	**1.0000**	**1.0000**	**1.0000**
D9	0.9668	0.0003	0.0650	0.4303	**1.0000**
D31	0.96282	0.87004	0.93454	0.93063	**0.96687**
Wine	0.2440	0.3510	0.3583	0.3583	**0.3960**
Breast_wpbc	0.0739	0.0219	0.3679	**0.4893**	**0.4893**

Table 6 Fowlkes-Mallows index evaluation

Datasets	DBSCAN	K-means++	DPC	DPC-KNN	KNN-ADPC
Flame	0.3542	0.8417	**1.0000**	**1.0000**	**1.0000**
Jain	0.9800	0.6894	0.8026	**1.0000**	**1.0000**
Aggregation	0.8652	0.8222	**0.9983**	**0.9983**	**0.9983**
Spiral	**1.0000**	0.3281	**1.0000**	**1.0000**	**1.0000**
R15	0.9318	**0.9933**	**0.9933**	**0.9933**	**0.9933**
D4	**1.0000**	0.5317	**1.0000**	**1.0000**	**1.0000**
D9	0.9872	0.4589	0.4914	0.7026	**1.0000**
D31	0.96825	0.87496	0.93664	0.93285	**0.97253**
Wine	0.6025	0.5683	0.5737	0.5737	**0.6577**
Breast_wpbc	**0.8077**	0.5862	0.6990	0.7968	0.7968

Table 7 Optimal parameters of each algorithm

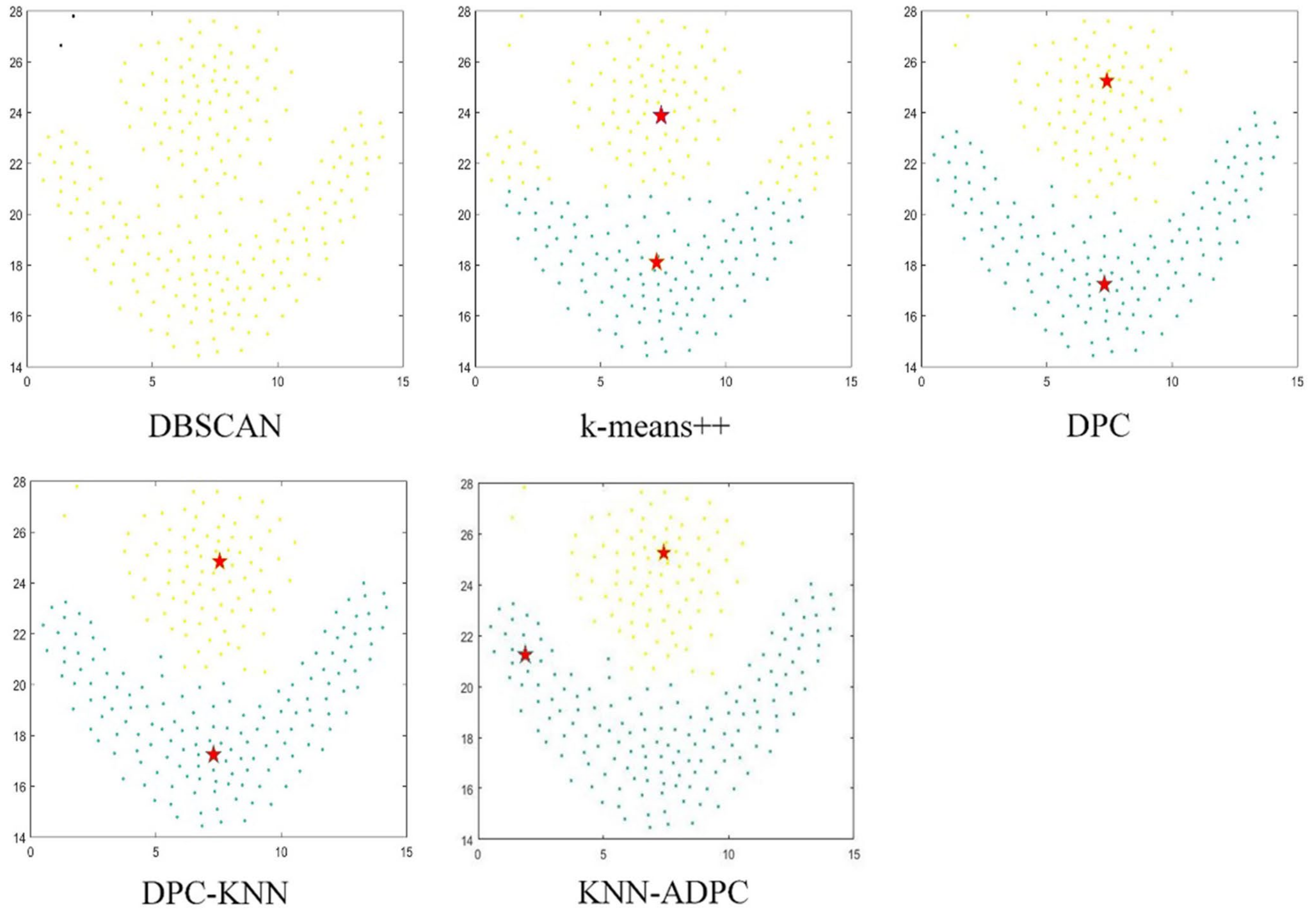
Datasets	DBSCAN	K-means++	DPC	DPC-KNN	KNN-ADPC
Flame	$MinPts = 4$ $\varepsilon = 0.93$	$K = 2$	$pc = 3$	$pc = 6$ $K = 4$	$K = 4$
Jain	$MinPts = 4$ $\varepsilon = 2.5$	$K = 2$	$pc = 13.0124$	$pc = 60$ $K = 9$	$K = 13$
Aggregation	$MinPts = 4$ $\varepsilon = 1$	$K = 7$	$pc = 3.1185$	$pc = 3.1185$ $K = 7$	$K = 4$
Spiral	$MinPts = 4$ $\varepsilon = 2$	$K = 3$	$pc = 3.6041$	$pc = 13.6041$ $K = 7$	$K = 8$
R15	$MinPts = 5$ $\varepsilon = 0.34$	$K = 15$	$pc = 0.6$	$pc = 4.5$ $K = 8$	$K = 4$
D4	$MinPts = 8$ $\varepsilon = 15$	$K = 4$	$pc = 2$	$pc = 4.5$ $K = 8$	$K = 55$
D9	$MinPts = 6$ $\varepsilon = 0.1$	$K = 4$	$pc = 2$	$pc = 4.5$ $K = 5$	$K = 29$
D31	$MinPts = 3$ $\varepsilon = 0.07$	$K = 30$	$pc = 2$	$pc = 2$ $K = 2$	$K = 2$
Wine	$MinPts = 10$ $\varepsilon = 0.1$	$K = 3$	$pc = 2$	$pc = 60$ $K = 4$	$K = 16$
Breast_wpbc	$MinPts = 4$ $\varepsilon = 0.5$	$K = 2$	$pc = 3$	$pc = 2$ $K = 5$	$K = 70$

addition, "min–max normalization" [33] for each feature is performed to eliminate the problem of magnitude difference. The "min–max normalization" can be denoted as Eq. (16).

$$x_{ij}^{'} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{17}$$

where x_j represents one feature. The clustering results are shown in Tables 3, 4, 5 and 6 in detail. For each clustering algorithm, we carefully adjust the parameters to get the best performance. Results in Tables 3, 4, 5 and 6 illustrate the proposed KNN-ADPC obtains good results compared with the other clustering algorithm under various evaluation indexes. In addition, the numbers in bold in the table represent the best-performing results on the dataset. Specifically, KNN-ADPC is superior to these classic clustering algorithms like DBSCAN and K-means++. It also surpasses original DPC and KNN-DPC on multiple artificial datasets. Besides, on the two real-world datasets, KNN-ADPC still gets the best performance under most metrics. Moreover, there is no need for human involvement in KNN-ADPC which can overcome the problem of misclassification caused by insufficient human experience. In conclusion, KNN-ADPC clustering algorithm can achieve satisfactory performance on both artificial and real-world datasets.

The optimal parameters of each algorithm on datasets are shown in Table 7. For DBSCAN, the maximum radius ε and the minimum points *MinPts* need to be decided. For K-means++, the number of clusters is necessary. For DPC and DPC-KNN, we need to input the cutoff distance d_c and manually select cluster centers. And we can choose d_c to make sure the number of neighbors of each point is within a certain percentage range of the total number of points.



DBSCAN

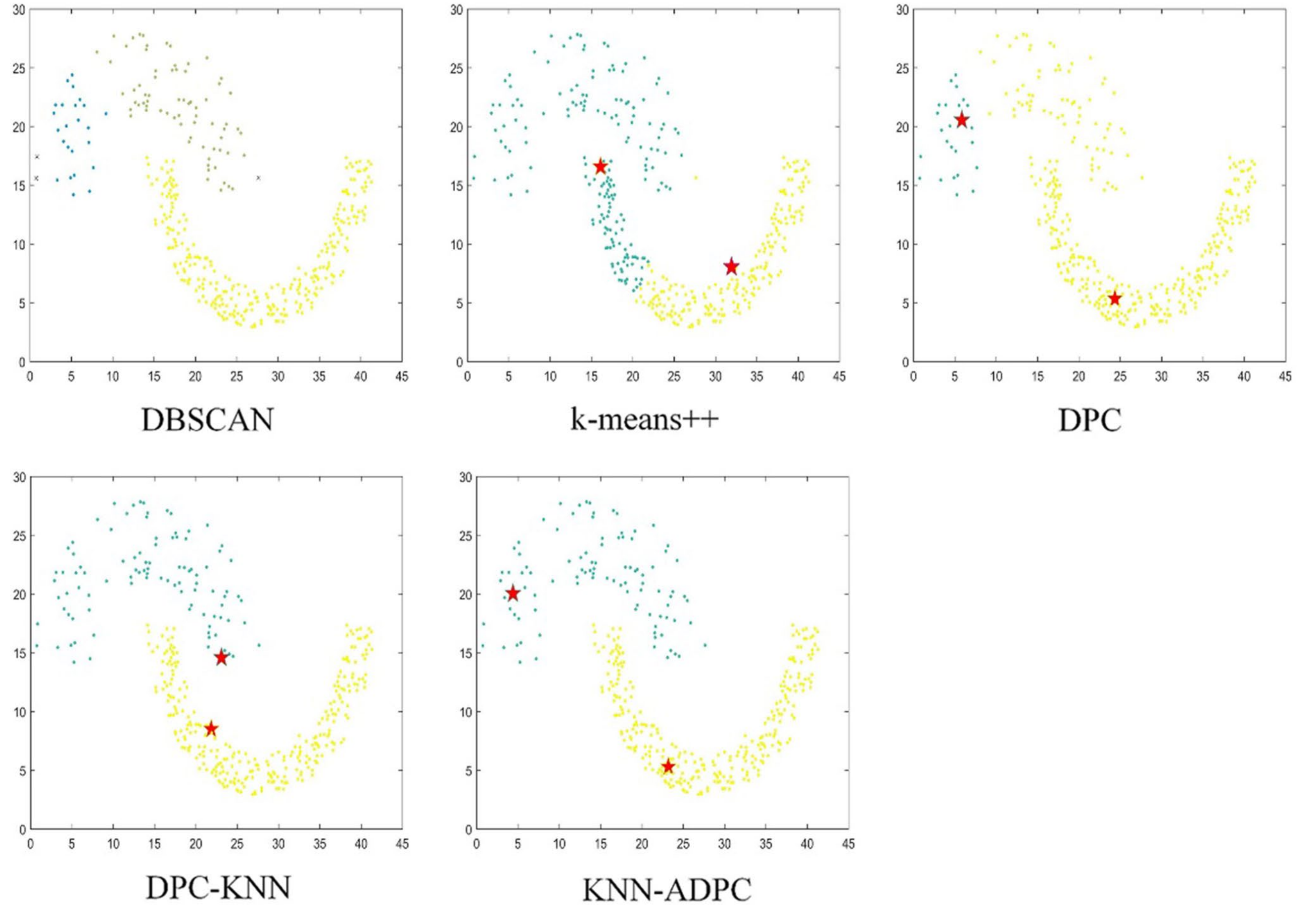
k-means++

DPC

DPC-KNN

KNN-ADPC

Fig. 4 The performance of each algorithm on Flame



DBSCAN

k-means++

DPC

DPC-KNN

KNN-ADPC

Fig. 5 The performance of each algorithm on Jain

In this paper, we determine the cutoff distance d_c by giving parameter pc. Besides, DPC-KNN requires the number of nearest neighbors K. For the proposed KNN-ADPC, the number of nearest neighbors K is the only needed parameter.

5 Discussion

Clustering results of each clustering algorithm on these artificial datasets are visualized in Figs. 4, 5, 6, 7, 8, 9 and 10.

As shown in Fig. 4, dataset Flame has two clusters that are very close to each other. In addition, the edge of one cluster is close to the other cluster which is prone to causing misclassification. From the result, we can see that the density-based clustering algorithm DBSCAN is powerless for the reason that the distance between two clusters is too close. The K-means++ is hard to deal with the edge of the curved cluster well while these DPC based clustering algorithms get correct results.

As shown in Fig. 5, the two clusters in dataset Jain embedded in each other, which leads to misclassification

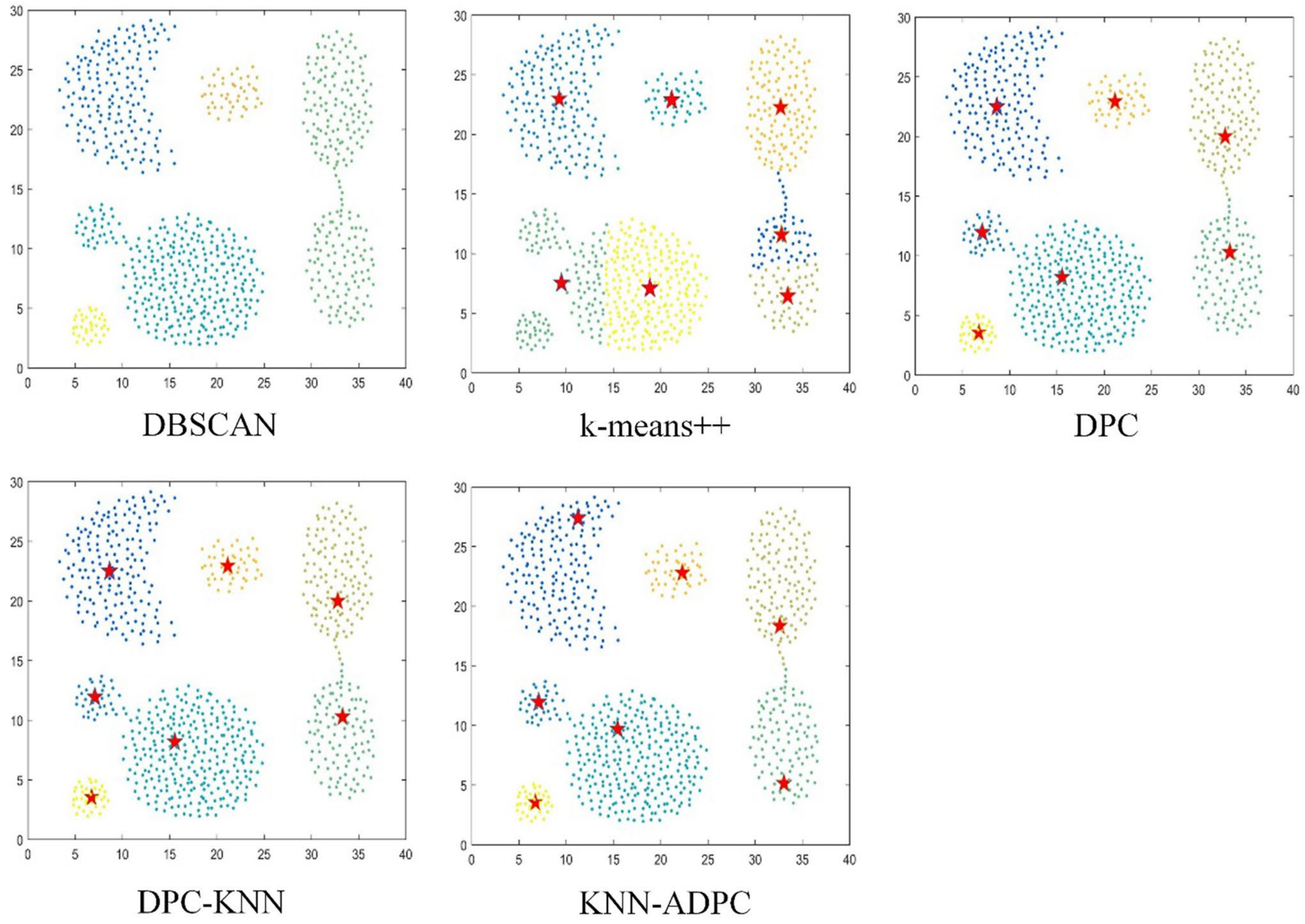
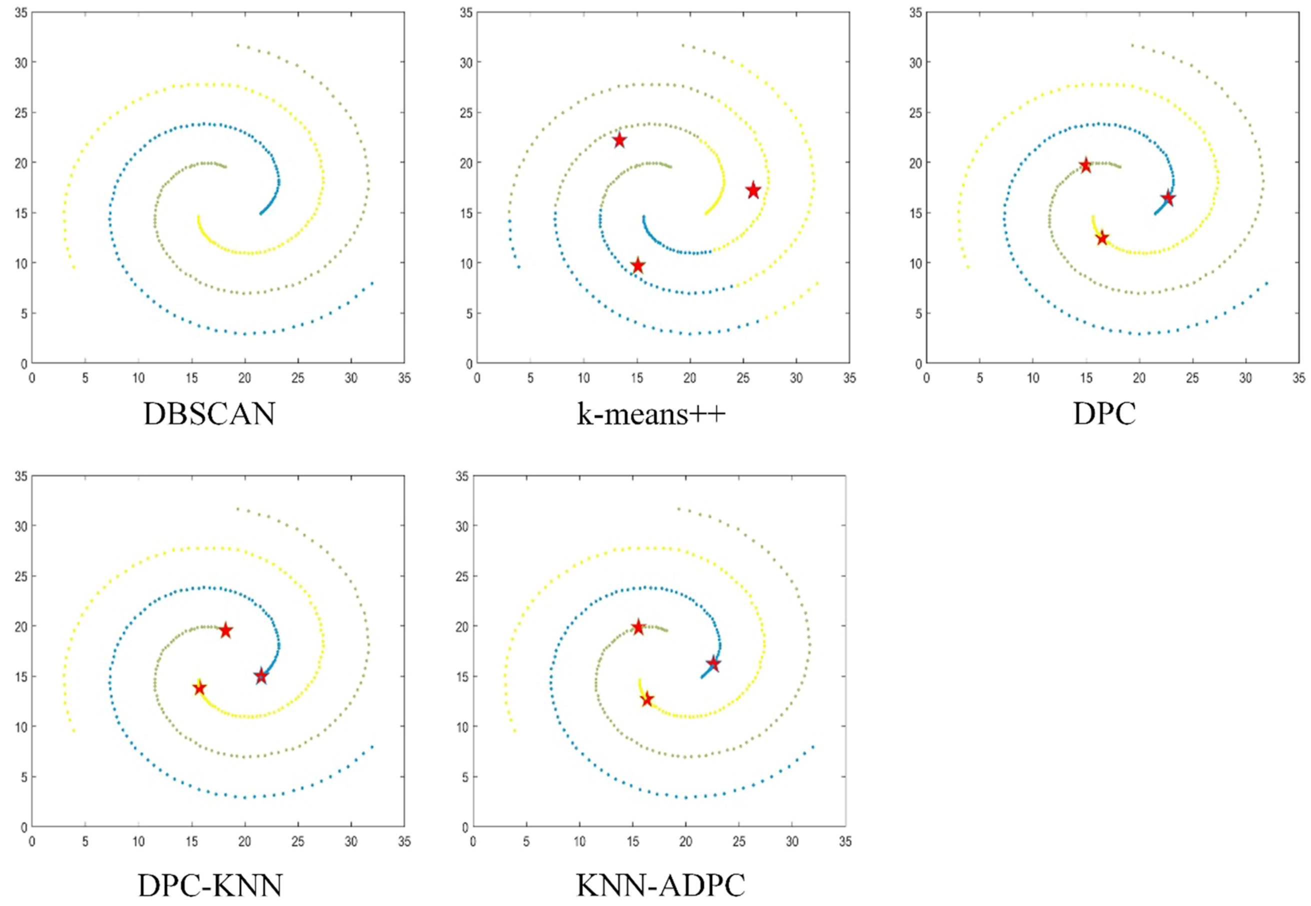


Fig. 6 The performance of each algorithm on Aggregation

in DBSCAN, K-means++, and original DPC. Only DPC-KNN and the proposed KNN-ADPC can properly deal with this embedded dataset. DPC-KNN and KNN-ADPC is able to assign points correctly because δ_i is computed with its K nearest neighbors.

As shown in Fig. 6, there are seven clusters and a few connections between partial clusters in the dataset Aggregation. DPC based clustering algorithm can outperform both DBSCAN and K-means++, while the latter are unsuitable for dealing with this type of clustering task.

As shown in Fig. 7, three curved clusters exist in the dataset Spiral. Apart from the partition-based clustering algorithm K-means++, which is unable to deal with the non-spherical clustering task, all other algorithms can obtain correct results.



DBSCAN

k-means++

DPC

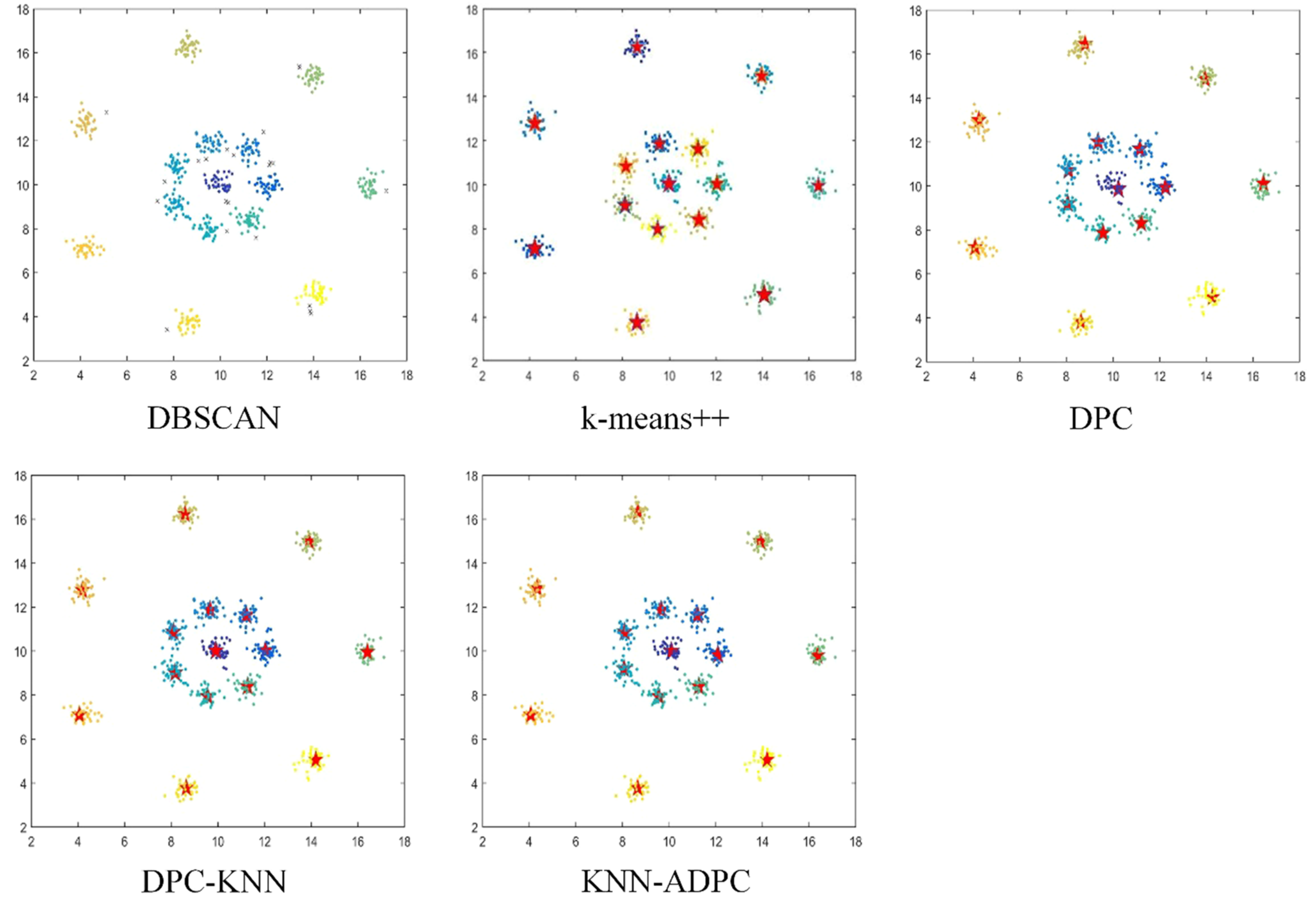
DPC-KNN

KNN-ADPC

Fig. 7 The performance of each algorithm on Spiral

As shown in Fig. 8, dataset R15 contains fifteen clusters. The misclassification occurs only in DBSCAN because there are some discrete points between clusters that are very close. Other clustering algorithms can assign points correctly.

For the discrete clusters in Fig. 9, all algorithms perform well except K-means++. And as shown in Fig. 10, dataset D9 has four clusters which contain spherical and curved clusters. In addition, there are many discrete points between clusters which easily lead to misclassification. Due to lack of merging strategy, DPC-KNN can not properly deal with the cluster with large curvature. Only the proposed KNN-ADPC can complete this kind of clustering task accurately.



DBSCAN

k-means++

DPC

DPC-KNN

KNN-ADPC

Fig. 8 The performance of each algorithm on R15

6 Conclusion

In this paper, we propose a novel density peaks clustering algorithm based on KNN-ADPC. The K nearest neighbors are adopted to calculate the cutoff distance d_c which can solve the problem of misclassification caused by giving parameter d_c unreasonably, In addition, The calculation of the distance δ_i also takes how discrete its K nearest neighbors are into consideration. The adaptive merging strategy allows us to automatically merge some over-segmentation clusters. The proposed KNN-ADPC is free of human involvement which can enhance the computing efficiency of the clustering algorithm to a large extent. At last, the outstanding performance of KNN-ADPC is

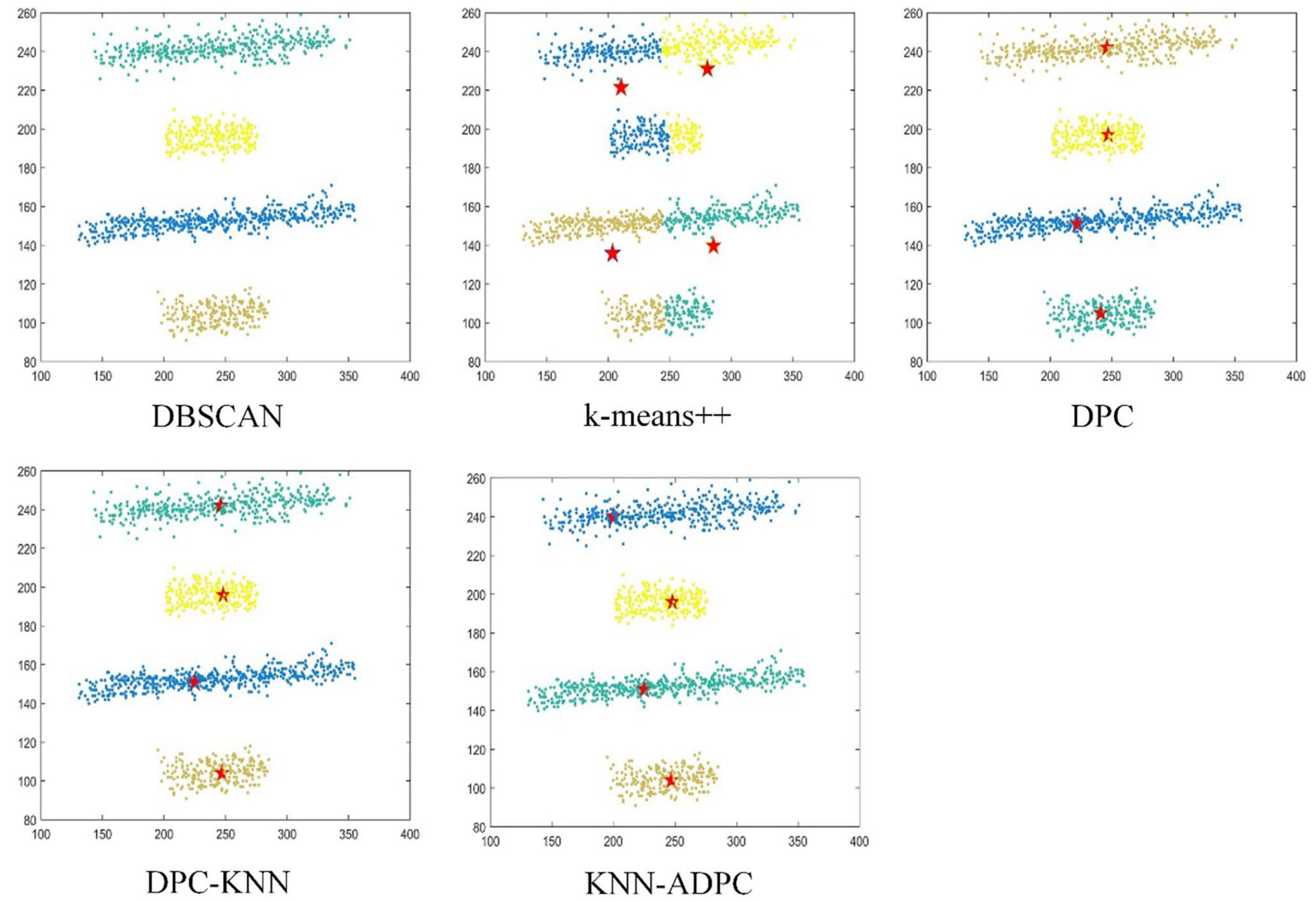
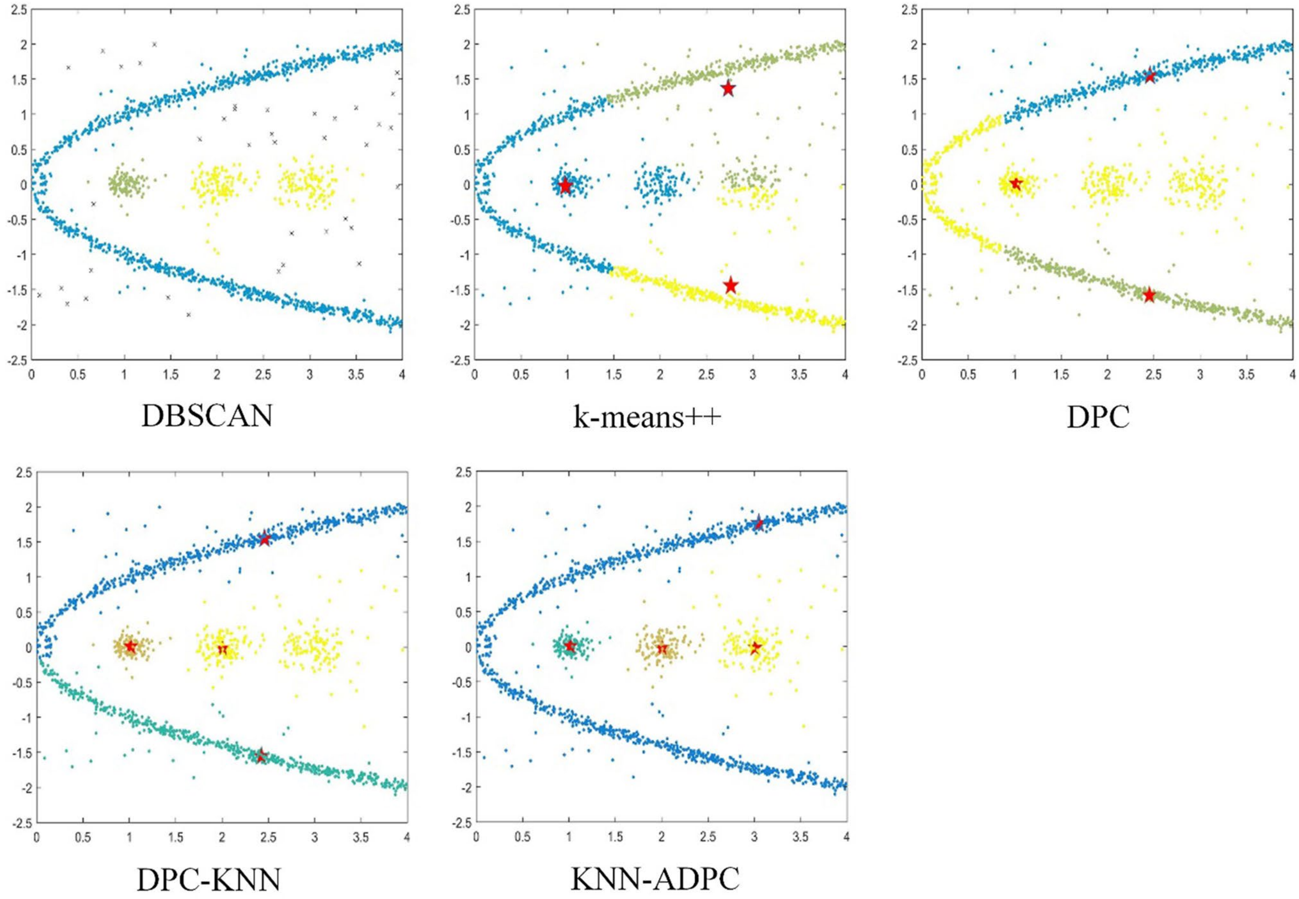


Fig. 9 The performance of each algorithm on D4

demonstrated in the experiment results compared with other clustering algorithms. On artificial datasets, the proposed KNN-ADPC can achieve the best performance with almost all 100% accuracy and other evaluation metrics like ARI, AMI, and FMI. And for the higher-dimensional and more complex real-world dataset, KNN-ADPC can still automatically complete clustering tasks and obtain excellent performance.

However, when implementing KNN-ADPC, it is still inevitable to decide the parameter K with expertise. In future work, efforts will be devoted to determining the parameter K automatically.



DBSCAN

k-means++

DPC

DPC-KNN

KNN-ADPC

Fig. 10 The performance of each algorithm on D9

Acknowledgements This work was supported in part by the National Key R&D Program of China under Grant 2019YFB1600500.

References

1. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv (CSUR) 31(3):264–323
2. Lotfi A, Moradi P, Beigy H (2020) Density peaks clustering based on density backbone and fuzzy neighborhood. Pattern Recognit 107:107449
3. Zhou K, Yang S (2020) Effect of cluster size distribution on clustering: a comparative study of k-means and fuzzy c-means clustering. Pattern Anal Appl 23(1):455–466
4. Cheung Y, Zhang Y (2018) Fast and accurate hierarchical clustering based on growing multilayer topology training. IEEE Trans Neural Netw Learn Syst 30(3):876–890
5. Jia H, Ding S, Du M, Xue Y (2016) Approximate normalized cuts without Eigen-decomposition. Inf Sci 374:135–150
6. Lu H, Liu S, Wei H et al (2020) Multi-kernel fuzzy clustering based on auto-encoder for fMRI functional network. Expert Syst Appl 159:113513
7. Chen MS, Huang L, Wang CD et al (2020) Multi-view clustering in latent embedding space. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, no 04, pp 3513–3520
8. Ji P, Zhang T, Li H et al (2017) Deep subspace clustering networks. arXiv:1709.02508
9. Huang D, Wang CD, Wu JS et al (2019) Ultra-scalable spectral clustering and ensemble clustering[J]. IEEE Trans Knowl Data Eng 32(6):1212–1226
10. Xie J, Girshick R, Farhadi A (2016) Unsupervised deep embedding for clustering analysis. In: International conference on machine learning, pp 478–487
11. Xu J, Lange K (2019) Power k-means clustering. In: International conference on machine learning. PMLR, pp 6921–6931

12. Ester M, Kriegel HP, Sander J et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. KDD 96(34):226–231
13. David A, Sergei V (2007) K-means++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7–9, 2007. ACM, 2007
14. Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. Science 344(6191):1492–1496
15. Xie J, Gao H, Xie W et al (2016) Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors. Inf Sci 354:19–40
16. Bai L, Cheng X, Liang J et al (2017) Fast density clustering strategies based on the k-means algorithm. Pattern Recognit 71:375–386
17. Liu R, Wang H, Yu X (2018) Shared-nearest-neighbor-based clustering by fast search and find of density peaks. Inf Sci 450:200–226
18. Jiang J, Chen Y, Meng X et al (2019) A novel density peaks clustering algorithm based on k nearest neighbors for improving assignment process. Phys A 523:702–713
19. Yaohui L, Zhengming M, Fang Y (2017) Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy. Knowl Based Syst 133:208–220
20. Jain AK, Law MHC (2005) Data clustering: a user's dilemma. In: International conference on pattern recognition and machine intelligence. Springer, Berlin, pp 1–10
21. Seyedi SA, Lotfi A, Moradi P et al (2019) Dynamic graph-based label propagation for density peaks clustering. Expert Syst Appl 115:314–328
22. Du M, Ding S, Xue Y et al (2019) A novel density peaks clustering with sensitivity of local density and density-adaptive metric. Knowl Inf Syst 59(2):285–309
23. Fu L, Medico E (2007) FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. BMC Bioinform 8(1):3
24. Gionis A, Mannila H, Tsaparas P (2007) Clustering aggregation. ACM Trans Knowl Discov Data 1(1):4-es
25. Chang H, Yeung DY (2008) Robust path-based spectral clustering. Pattern Recognit 41(1):191–203
26. Veenman CJ, Reinders MJT, Backer E (2002) A maximum variance cluster algorithm. IEEE Trans Pattern Anal Mach Intell 24(9):1273–1280
27. Dai QZ, Xiong ZY, Xie J et al (2019) A novel clustering algorithm based on the natural reverse nearest neighbor structure. Inf Syst 84:1–16
28. Su Z, Denoeux T (2018) BPEC: belief-peaks evidential clustering. IEEE Trans Fuzzy Syst 27(1):111–123
29. Dua D, Karra Taniskidou E (2017) UCI machine learning repository. University of California. School of Information and Computer Science, Irvine, CA. Available at http://archive.ics.uci.edu/ml. Accessed 21 Apr 2019
30. Jossinet J (1996) Variability of impedivity in normal and pathological breast tissue. Med Biol Eng Comput 34(5):346–350
31. Vinh NX, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J Mach Learn Res 11:2837–2854
32. Fowlkes EB, Mallows CL (1983) A method for comparing two hierarchical clusterings. J Am Stat Assoc 78(383):553–569
33. Jain A, Nandakumar K, Ross A (2005) Score normalization in multimodal biometric systems. Pattern Recognit 38(12):2270–2285

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.