Hindawi
Complexity
Volume 2018, Article ID 5232543, 14 pages
https://doi.org/10.1155/2018/5232543

Research Article

A Data-Driven Parameter Adaptive Clustering Algorithm Based on Density Peak

Tao Du, Shouning Qu, and Qin Wang

School of Information Science and Engineering, University of Jinan, No. 336, West Road of Nan Xinzhuang, Jinan 250022, Shandong, China

Correspondence should be addressed to Tao Du; ise_dut@ujn.edu.cn

Received 31 May 2018; Accepted 6 August 2018; Published 21 October 2018

Academic Editor: Rafael Gómez-Bombarelli

Clustering is an important unsupervised machine learning method which can efficiently partition points without training data set. However, most of the existing clustering algorithms need to set parameters artificially, and the results of clustering are much influenced by these parameters, so optimizing clustering parameters is a key factor of improving clustering performance. In this paper, we propose a parameter adaptive clustering algorithm DDPA-DP which is based on density-peak algorithm. In DDPA-DP, all parameters can be adaptively adjusted based on the data-driven thought, and then the accuracy of clustering is highly improved, and the time complexity is not increased obviously. To prove the performance of DDPA-DP, a series of experiments are designed with some artificial data sets and a real application data set, and the clustering results of DDPA-DP are compared with some typical algorithms by these experiments. Based on these results, the accuracy of DDPA-DP has obvious advantage of all, and its time complexity is close to classical DP-Clust.

1. Introduction

Clustering is one of the most important methods in machine learning, and by clustering, data points are partitioned to several groups [1], and the ones in the same group are much similar, and points in different groups are much different [2–4]. Clustering algorithm can deal with data points without any labelled samples, so it is much fit for the fast-changing environment, in which the samples are hardly obtained [5]. Nowadays, with the development of big data, clustering has been more and more applied in Internet of things, environment monitoring, image processing, etc. [6].

There have been more and more researches focused in designing high efficient clustering algorithm, and these researches can be divided to four kinds: the partition-based methods, such as K-means [7] and K-medoids [8]; the hierarchy-based methods, such as BIRCH [9], ROCK [10], and Chameleon [11]; the density-based methods, such as DBSCAN [12] and OPTICS [13]; and grid-based methods, such as STING [14] and CLIQUE [15]. In classical partition-based algorithms, the number of clusters should be artificially defined before clustering, which much restricts the flexibility of clustering application, and they are not able to efficiently cluster the nonhypersphere data set [1]. In classical hierarchy-based algorithms, the threshold of merging microclusters or dividing macroclusters is the key parameter of clustering, and it is also set artificially before clustering [16], and these algorithms' time complexity is relatively large. In classical grid-based algorithms, grid granularity is the key parameter, and the clustering accuracy will be poor if it is set too large, otherwise, the time complexity will be much increased if it is set too little [17]. Density-based clustering algorithms can cluster arbitrary shapes of data sets and the clustering results are not influenced by noise points, so density-based algorithms have been the focus of clustering researches, and there have been many new algorithms proposed [18–20]. Density peak-based clustering (DP-Clust) is one of the important algorithms of these researches, and on the basis of the advantages of density based algorithms, DP-Clust improves the efficiency of clustering by detecting centers, borders, and outliers from all data points [21]. However, as

other density-based algorithms, DP-Clust needs to set the local field's radius of every point to accomplish clustering, and the thresholds of detecting centers and outliers are also set in advance, and then the performance is not good at dealing with sparse distribution data set.

Based on the above analysis, it can be seen that the artificial setting of clustering parameters has been the key factor of influencing the performance of clustering, so nowadays, some researchers have focused on optimizing parameters to improve clustering efficiencies: FEAC can adapt the number of clusters which was proposed by Silva to get rid of the defect K-means [22], however the complexities of time and memory are too large and it cannot efficiently cluster arbitrary shapes of data sets; Hou proposed a parameter independent hierarchy based algorithm named DSets-histeq [23], in which microclusters are merged according to the theory of dominant set, and it can automatically adjust parameters by establishing similarity matrices of every pair of microclusters, so the complexity of clustering is much increased; Myhre proposed a grid-based algorithm named KNN-MS, in which data points are partitioned to K grids, and by mode seeking theory, all grids would be adjusted and the result are not influenced by the value of K [24], but it is much influenced by noise points. And there are many other algorithms proposed to reduce the influence of the parameter's initial value; however, these ones have defects in dealing with arbitrary shapes of data sets or the efficiency in clustering. Then to use the advantages of high efficiency in clustering arbitrary data sets and relative simple clustering procedure of DP-Clust, many density peak-based algorithms are proposed to realize self-adapting parameters to improve clustering performance; however, these researches have more or less artificial factors when setting clustering parameters, and they cannot realize fully data-driven parameter adaptive clustering algorithm based on density peak. To improve the independence of parameters when clustering, we propose a fully data-driven parameter adaptive clustering algorithm based on density peak (DDPA-DP). In DDPA-DP, all parameters can be updated by the distribution of data points, and the procedure of adapting parameters is simple to be accomplished to reduce the time complexity of whole algorithm. The rest of this paper is organized as follows: in section two, the basic thought of DP-Clust is introduced, and related researches are analyzed; in section three, the thought of DDPA-DP is proposed, and the detail of this algorithm is designed; in section four, a series of experiments are simulated, and the other three algorithms are compared with DDPA-DP; and at last, the contribution of this paper is concluded.

2. Related Work

2.1. The Introduction of Density Peak Clustering. In 2014, Rodriguez proposed a density-based clustering algorithm named DP-Clust, and the basic thought of DP-Clust is that the centers of all clusters should be located at the peak of local density changing curve, and the borders will be located at the neighborhoods of centers, and outliers will be far away from high-density area [21]. To detect centers, borders, and outliers, two conceptions are defined in DP-Clust:

Definition 1. Local density is an attribute to measure the density station of point i by computing the influence of other points in point i neighborhood to point i, and it can be computed as (1) or as (2).

$$\rho_i = \sum X \times (d(i,j) - r), \quad X = \begin{cases} 1, d(i,j) > r, \\ 0, d(i,j) < r, \end{cases} \tag{1}$$

$$\rho_i = \sum \exp\left(\frac{d(i,j)^2}{r^2}\right). \tag{2}$$

In (1) and (2), r is a cutoff distance, and the radius of point i's neighborhood is r and the center is i. Then $d(i,j)$ is the distance from i to its neighbor j which is located in i's neighborhood. By (1) and (2), just the points in i's neighborhood can influence its local density. After all points' local densities are obtained, a list L will be established, and in L all points will be rearranged with the descending order of their local densities as $\{\rho_{q1}, \rho_{q2}, \rho_{q3}, \cdots, \rho_{qn}\}$.

Definition 2. The distance from the nearest neighbor with larger local density than i is defined as (3), and this is an attribute to measure the point whether be located in the center of a high-density field.

$$\delta_{qi} = \begin{cases} \min_{j<i}\{d_{qi,qj}\}, i \geq 2 \\ \max_{j\geq 2}\{d_{qi,qj}\}, i = 1 \end{cases}. \tag{3}$$

According to (3), if point i is the first one in list L, the value of δ_{qi} is set as the distance to the farthest point from i; otherwise, the value of δ_{qi} is set as the distance from i to the nearest point whose position in L is in front of i.

After obtaining the ρ and δ of every point, the one has both larger ρ and δ can be detected as centers, because larger ρ means this point located in a high-density area, and larger δ means there are not any points in the same high-density area with larger ρ than it, and then it can be seemed as the center of this area. Otherwise, if the point has less ρ and larger δ, it can be detected as outlier, because less ρ means this point is located in a sparse area, and meanwhile, larger δ means this point is far away any high-density area, and then it can be seemed as be out of all clusters. At last, all remaining points can be detected as borders, and these points have larger ρ and less δ, which means every border is located in a high-density area, but there is at least one point in the same area located nearer to the center. After all points' roles are being obtained, every border will join the nearest center to format cluster.

Because just local density instead of global density needs to be computed, DP-Clust has obvious advantage in clustering nonuninform density fields comparing to DBSCAN, and its clustering procedure is simple to be deployed in

application. Now, DP-Clust has dropped much attention, and many density peak-based clustering algorithms have been proposed [25–27]; however, same as DP-Clust, these algorithms should set three main thresholds as the radius of local field r, the standards of larger or less of local density, and the standards of larger or less of the nearest distance to neighbor with larger local density, and these settings restrict the clustering performance especially in sparse and changeable environments.

2.2. Existing Researches of Parameter Optimized Density Peak-Based Clustering. To reduce the influence of initial setting of parameters of DP-Clust, there are two problems to be resolved: one is how to determine the thresholds of local density ρ and the distance to the nearest neighbor with larger local density δ; the other is how to select optimizing radius of local field r.

In ref [28], Chen and He proposed an algorithm named ACC-FSFDP, in which a curve fitting method is adopted to automatically find the points with both larger ρ and δ to determine centers. ACC-FSFDP designed a variable $\gamma = \rho * \delta$, and centers will be detected by finding the points with obvious larger γ than the value predicting by the curve fitting function. Although ACC-FSFDP can automatically obtain centers, it did not take outliers into account, which leads the accuracy is much influenced by noise. In ref [29], Saki and Kehtarnavaz used a histogram to reflect the distribution of all points' ρ and δ, and centers and outliers will be detected by data-driven method; however, this algorithm time complexity is large, because it takes too many calculations when establishing histogram. In ref [30], Xu et al. proposed a FNLT algorithm, in which two different linear-regression analysis functions are established to, respectively, detect centers and outliers, and clusters will be stored as Leading Tree and centers as Fat Nodes of these trees, and by merging trees to optimize the distribution of clusters. Although FNLT can detect points' roles by data driven, the clustering procedure is too complex to be deployed because of the complex structure of FNLT, and the linear-regression analysis method is less accurate in predicting the change tendency of ρ and δ.

Besides the defects of setting thresholds of centers and outliers, the algorithms mentioned above adopt fixed and preset radius of local field to accomplish clustering, which much restricts the performance in complex environments. Nowadays, there are two kinds of effective algorithms that focus on optimizing the local field radius of density peak clustering: one adopts K nearest neighbors-based method to divide the local field instead of by radius; the other directly optimizes the local field's radius to reduce the affection of initial setting.

DPC-KNN is a classical KNN-based algorithm [26], and in DPC-KNN, the points' local density is computed as (4).

$$\rho_i = \exp\left(-\frac{1}{K}\sum_{j\in KNN_i} d_{ij}^2\right). \tag{4}$$

FKNN-DPC [31] is another KNN-based one, and the local density in FKNN-DPC is computed as (5).

$$\rho_i = \sum_{j\in KNN_i} \exp\left(-d_{ij}\right). \tag{5}$$

Comparing (1) and (2) with (4) and (5), in the KNN-based algorithms, a point will obtain its local density by computing the distances to its K neighbors, and the parameter K needs to be input in advance. The advantage of KNN-based method is that clustering complexity will be much less than DP-Clust if all points' KNN have been known. However, in most application environments, the operation of obtaining KNN of every point will be so hard that the performance is not obviously improved. Liu et al. proposed an adaptive KNN-based algorithm ADPC-KNN [32], and in this algorithm, the local density is computed by combination of DP-Clust and DPC-KNN as (1), in which r is deduced by K as (6) and (7), and the value of K can be adjusted by evaluating the distribution of clusters.

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

$$\mu^K = \frac{1}{N}\sum_{i=1}^{N}\delta_i^K, \tag{7}$$

where N is the number of all points, and $\delta_i^K = \max_{j\in KNN_i} d_{ij}$ is the distance from i to its Kth nearest neighbor. Based on (6), ADPC-KNN can optimize the value of r by all data points; however, the calculation complexity is much increased. Besides the clustering efficiency, the performances of KNN-based algorithms are relatively poor in nonuniform fields, because in these fields, density in a point's KNN will be much different with others, which leads to the centers in sparse area cannot be well detected. And the value of K will influence the result of clustering, which does not well satisfy the demand of parameter independence.

In ref [33], a DP-Clust-based algorithm named DCore was proposed by Chen et al. DCore uses a concept of density core to find high-density fields and to determine centers and borders, in which the clusters in sparse area can be detected by mean shift thought. In Dcore, data-driven thought is used to adjust the clusters' distribution; however, the value of is r fixed, and the threshold of determine centers is artificially set which restrict the DCore's performance in nonuniform fields. Based on Dcore, DCNaN was proposed by Xie et al. [34], and in DCNaN, every point will compute its local field's radius by (8).

$$r_i = \frac{\sum_{j\in NaN(i)} d(i,j)}{b(i)}. \tag{8}$$

In (8), $NaN\,(i)$ is the natural neighbors' set of point i, and $b\,(i)$ is the number of natural neighbors of point i, and the concept of natural neighbors was introduced in ref [ccc]

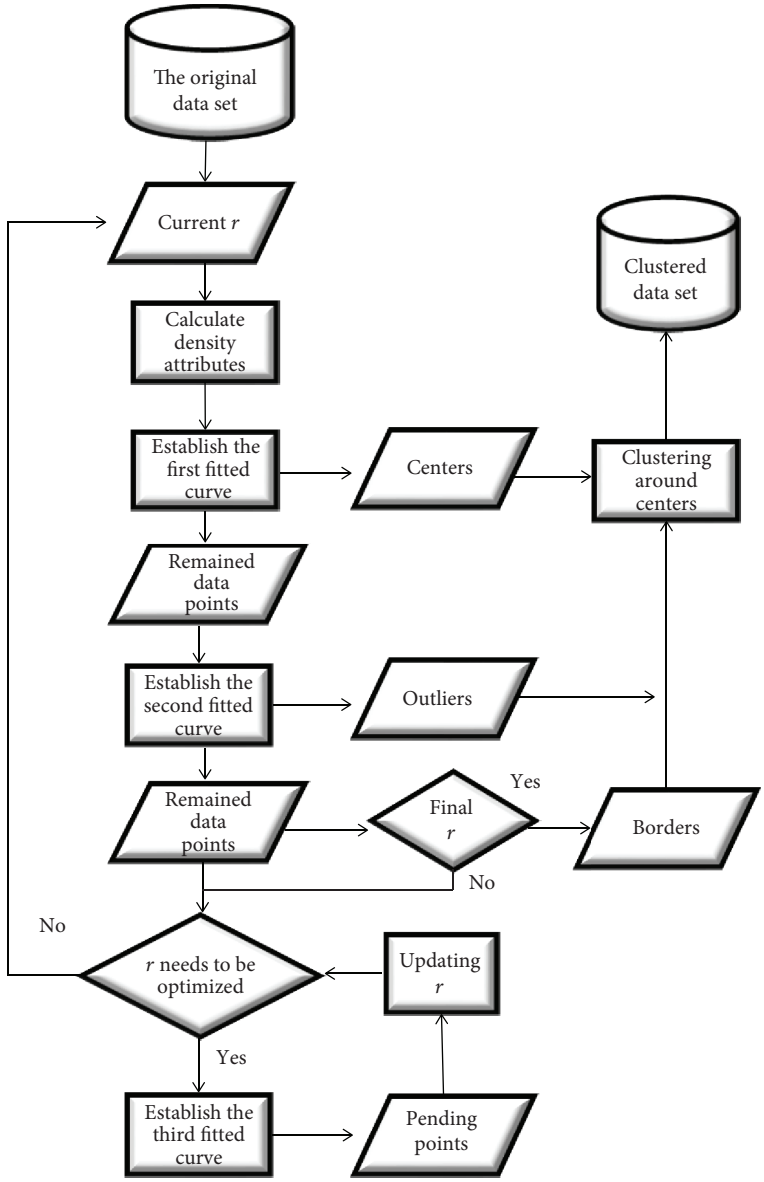

Figure 1: The flow of DDPA-DP.

and [ddd]. Then, a list of sorted scanning radiuses r is established, and by computing the variation of adjacent r in this list, centers and outliers are detected. In this algorithm, the data-driven method in dynamical adjusting local field's radius improves the clustering performance in sparse area; however, in DCNaN, the thresholds of judging natural neighbors, centers, and outliers are preset and fixed, and the procedure of adjusting the local field's radius needs too many iterative computations to much increase the complexity of clustering.

Based on the application of clustering, it can be concluded that there are three key problems that should be resolved when designing clustering algorithm: the accuracy in clustering arbitrary data set, the parameter independence when clustering, and the complexity of time and memory. However, based on the above analysis, existing researches have much or less defects so that these problems are not well resolved, and now these problems have been the major obstacles of restricting the clustering application.

3. The Design of DDPA-DP

To obtain the target of improving clustering performance, we propose a fully data-driven parameter adaptive clustering algorithm based on density peak (DDPA-DP), and in DDPA-DP, the parameter of the local field's radius r can be dynamically adjusted, and the thresholds of detecting points' roles are determined by data distribution, and the complexity of this algorithm is also better than classical density-based ones. In DDPA-DP, there are three steps: density attributes are computed by initial value of r, and then points' roles are automatically detected, and a self-adaptive procedure will be called to optimize r. The flow of DDPA-DP is shown in Figure 1.

In the first step of DDPA-DP, r will be set an initial value, and then all points' local density ρ and the distance to the nearest higher density neighbor δ will be computed with the current value of r; secondly, according to the current points' values of ρ and δ, a series of fitting curves will be established, and the points whose ρ and δ are obviously different to most of the others will be detected, and their roles will be determined based their ρ and δ; at last, by the distribution of points' roles, the value of r will be evaluated and optimized. Repeat these three steps until r is convergent.

3.1. Automatically Detecting Points' Roles. According to the thought of DP-Clust, we propose a series of fitted curves to predict the combination value of ρ and δ, and based on the distribution of the difference between predicting value and real value of every point, a point's ρ and δ are larger or less one among all points can be automatically determined, and then by the density attributes of all points, their roles can be detected. Meanwhile a new kind of points named "pending point" is defined as Definition 3. To better illustrate the algorithm, a simple model as Figure 2 is established.

Definition 3. When a point's ρ is less than most points and meanwhile its δ is less than most points too, it is hard to determine that this point belongs to borders or outliers, so we call these points "pending point." This role does exist in data points, but existing algorithms do not research it.

Assuming $D\{d_1, d_2, d_3, \dots, d_N\}$ is the target data set with N points, and every point has n attributes. Centers' ρ and δ are both larger than most of the other points, so a variable $\gamma = \delta \times \rho$ is defined to establish the fitted curve as (9).

$$\gamma = a_0 + a_1 \times I_c + a_2 \times I_c^2 + \cdots + a_N \times I_c^N. \quad (9)$$

In (9), I_c $\{i_{c1}, i_{c2}, \dots, i_{cM}\}$ is the index of N data points, which is used to act as independent variable, and $\{a_0, a_1, a_2, \dots, a_N\}$ are the coefficients of fitted curve. By this curve, if a data point's index is known, the value of γ can be predicted. The difference between the real value of γ^* and its predicted value γ is $\Delta\gamma$, and then both the mean and the variance of $\Delta\gamma$ can be obtained. The frequency histogram of the distribution of $\Delta\gamma$ is shown in Figure 3. In Figure 3, ε is the mean of $\Delta\gamma$ and its value is 0, which means that the predicting values of γ of most points are very close to their real value, and σ is the variance of $\Delta\gamma$. From Figure 3, most of the points are distributed in the value range $\{-\sigma \le \Delta\gamma \le \sigma\}$. When a point's $\Delta\gamma > \sigma$, it means that this point has larger γ than most of the other points, and it can be seemed as candidate center, and the corresponding relations are shown in Figure 4. Based on this procedure, the thresholds of judging centers are determined by the distribution of points and they need not be artificially set, which reflects the advantages of data-driven thought.

After detecting centers, outliers should be detected from remained points. So a variable $\gamma' = \delta \div \rho$ is defined to find the points with larger δ and less ρ, and a fitted curve as (10) is established to predict the value of γ'.

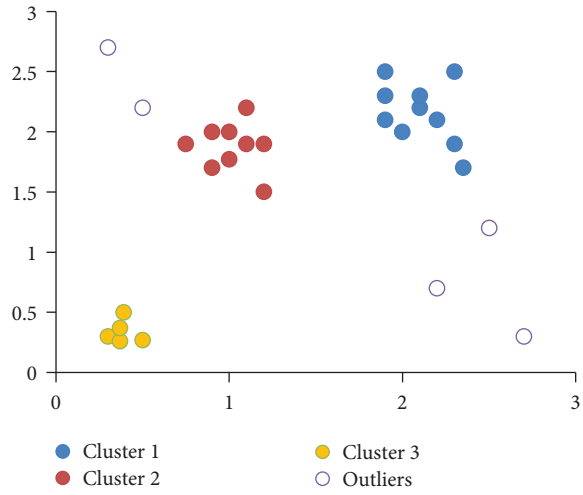

Figure 2: A simple model of clusters.

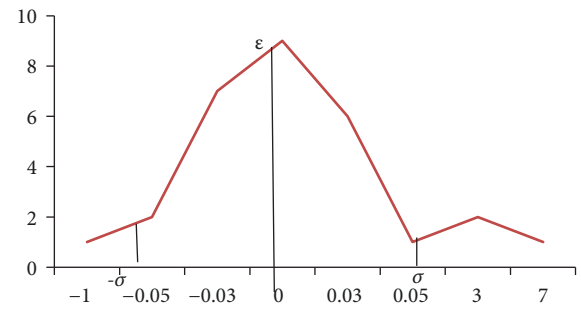

Figure 3: The distribution of $\Delta\gamma$.

$$\gamma' = b_0 + b_1 \times I_c + b_2 \times I_c^2 + \cdots + b_m \times I_c^m. \quad (10)$$

By (10), $\Delta\gamma'$ is obtained which is the difference between $\gamma' *$ and γ', and the frequency histogram of $\Delta\gamma'$ is shown in Figure 5, in which most of the points' $\Delta\gamma'$ are distributed in the value range $\{\Delta\gamma' \le \sigma\}$. Then, outliers can be automatically detected by finding the points whose $\Delta\gamma'$ is larger than σ, and the corresponding relations between outliers and their values of $\Delta\gamma'$ are shown in Figure 6.

As the operations of detecting centers and outliers, pending points can be detected by finding the points with less δ and less ρ, so a variable $\omega = \delta \div \rho$ is defined, and a fitted curve as (11) is established to predict the value of ω. When the difference between real value and predicted value of a point is larger than the variance, it can be seemed as pending points, which is shown in Figure 7.

$$\omega = c_0 + c_1 \times I_c + c_2 \times I_c^2 + \cdots + c_m \times I_c^m. \quad (11)$$

After centers, outliers and pending points are detected, remained points can be seemed as borders of clusters, and

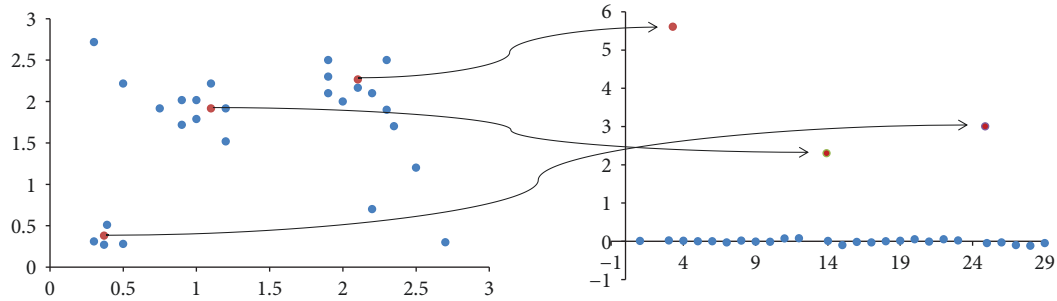

Figure 4: The corresponding points of centers.

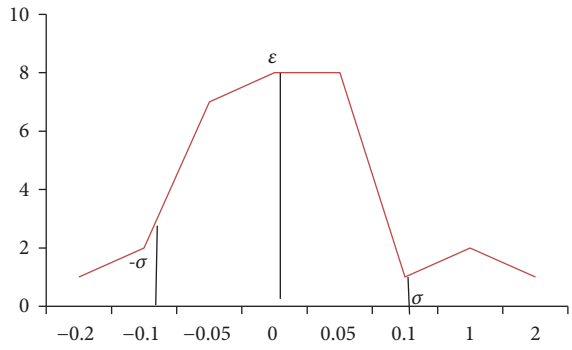

Figure 5: The distribution of $\Delta\gamma'$.

then every border will join the nearest center and form the cluster, and the result is shown in Figure 8.

3.2. Optimizing the Radius of Local Field. In Figure 8, there are two pending points, and these points have less ρ which means they are located in sparse area; however they have less δ which means every pending point has at least one neighbor with larger local density than it. As in Figure 8, the pending points' local field connects a relative dense area and a relative sparse area, and it is hard to determine the role of pending point in cluster. So the distribution of pending point means that the radius of points' local field is not well set, and then we propose a self-adaptive method to optimize the value of local field radius as (12).

$$\Delta r = (r_{i-1} - r_{i-2}) \times (-1)^x \div \rho_{i-1} \times e^{-|\rho_{i-2}-\rho_{i-1}|/\rho_{n-1}}, i \geq 2,$$
$$\Delta r = (r_{i-1} - r_{i-2}) \div (\rho_{i-1} - \rho_{i-2}), \quad \rho_{i-1} = \rho_{i-2}. \tag{12}$$

In (12), Δr is the adjustment quantity of local field radius, and r_{i-1} is the current value of radius which is used to detect points' roles in the last round of computing, and r_{i-2} is the last value of radius; p_{i-1} is the number of pending points in the last round of computing, and p_{i-2} is the number of pending points in the penultimate computing, x is an accommodation coefficient as $(r_{i-1} - r_{i-2}) \times (p_{i-2} - p_{i-1})$, in which $r_{i-1} - r_{i-2}$ is used to obtain the change tendency of radius, and $p_{i-2} - p_{i-1}$ is used to quantify the effect of adjustment: $(r_{i-1} - r_{i-2}) > 0$ means that the last adjustment of r is increased, if $(p_{i-2} - p_{i-1}) > 0$ means pending points are reduced and the tendency of adjustment should be maintained, otherwise if $(p_{i-2} - p_{i-1}) < 0$ means the adjustment should be turned; when $(r_{i-1} - r_{i-2}) < 0$, if pending points are reduced, $(p_{i-2} - p_{i-1}) > 0$ and then r will be reduced continually, otherwise if $(p_{i-2} - p_{i-1}) > 0$, the value of r will be increased to turn the tendency.

The points' distribution is shown in Figure 8 where the initial value of r is set as 0.25, and there are two pending points detected; then, by (12), the optimized value of r is 0.29, and the result of computing is shown in Figure 9, in which there are three pending points; then, the value of r is reduced to 0.21 by (12), and the result of computing is shown in Figure 10, in which there is no pending point, and the procedure of optimizing is complete, and all points are well clustered.

Although the result of Figure 10 is a particular case, DDPA-DP has a suspension method to avoid increasing the time complexity: if there are C continuous rounds of computing with same number of pending points or the changing range of pending points is less than $1/C$, the optimizing procedure will be completed, in which C is the number of centers. If there are still some pending points after optimizing r, they will be analyzed to be divided to borders or outliers by the next two principles: if a point is a pending point and its nearest neighbor with larger ρ is an outlier, this pending point is also an outlier; if a point is a pending point, and its nearest neighbor point with larger ρ is a center or border, this point can be seemed as a border point.

3.3. The Complexity of DDPA-DP. Time complexity is an important performance in designing clustering algorithm because there are a large number of data sets to be computed. In DDPA-DP, n points are used to accomplish initial local density computing by initial parameter, and the fitted curves are established, so the complexity of this step is O (n^2); and then in the local field radius optimizing step, just pending points should be redetected and its average number assumes p, and the complexity of this step is O (p^*k), where k is the average computing rounds' number; then, at the last step, the optimized r is used to compute final points' roles and clustering, and the complexity is O (n^2) too. Because the numbers of p and k are much less than n, the complexity of whole DDPA-DP is O (n^2), which is the same with DP-Clust. Based on this analysis, it can be concluded that DDPA-DP can maintain relative high performance in complexity with parameter independence.

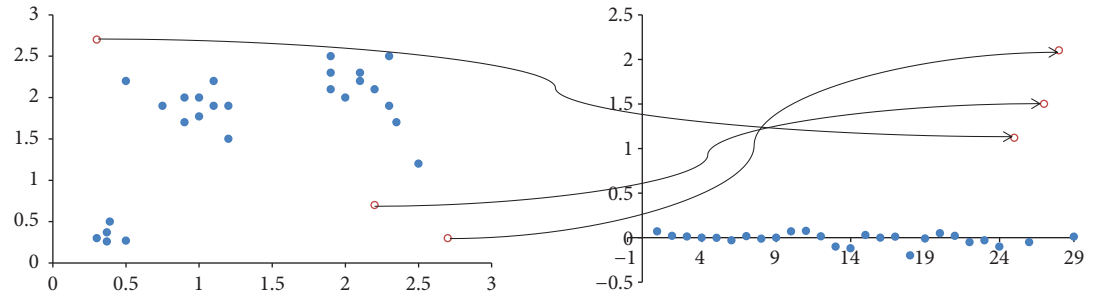

Figure 6: The corresponding points of outliers.

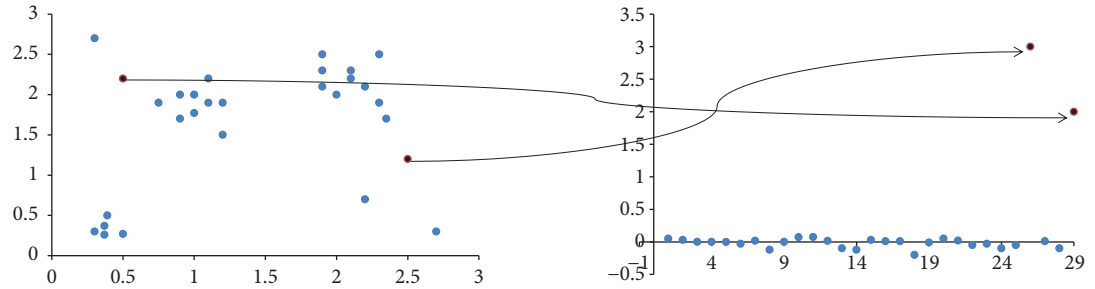

Figure 7: The corresponding points of pending points.

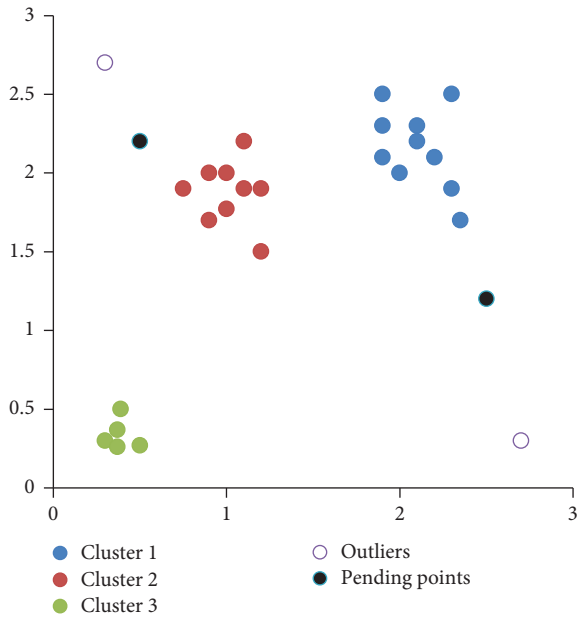

Figure 8: The distribution of points' roles when r is 0.25.

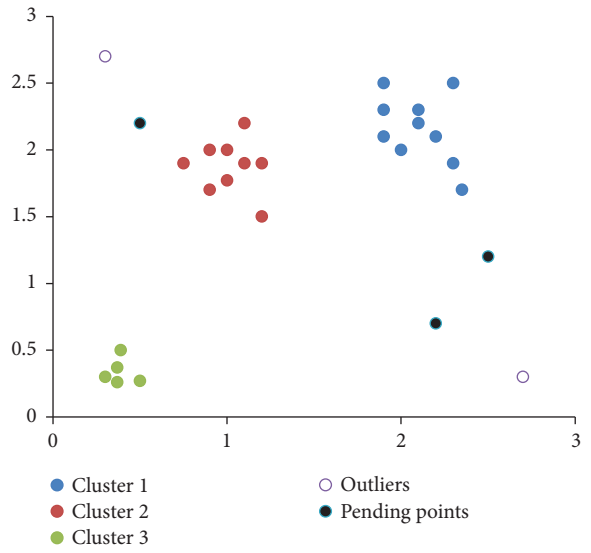

Figure 9: The distribution of points' roles when r is 0.29.

4. Experiments and Results

To prove the advantages of DDPA-DP, a series of experiments are designed and simulated, and three typical clustering algorithms DBSCAN [12], DP-Clust [21], DPC-KNN [26], FKNN-DPC [31], Dcore [32], and DCNaN [33] are compared with DDPA-DP in these experiments. In this section, experiments are simulated by MATLAB 2015b, and two main performances are analyzed: the accuracies of all clustering algorithms are compared and analyzed in Section 4.1, and the real-time performances of these algorithms are compared and analyzed in Section 4.2. Six artificial data sets with arbitrary shapes of distribution and one real data set GL1 are used to be simulated, which are listed in Table 1. In Table 1, N means the number of data points, K means the number of clusters, and D means the number of dimensions. The distributions of 2-D data sets are shown in Figures 11–14.

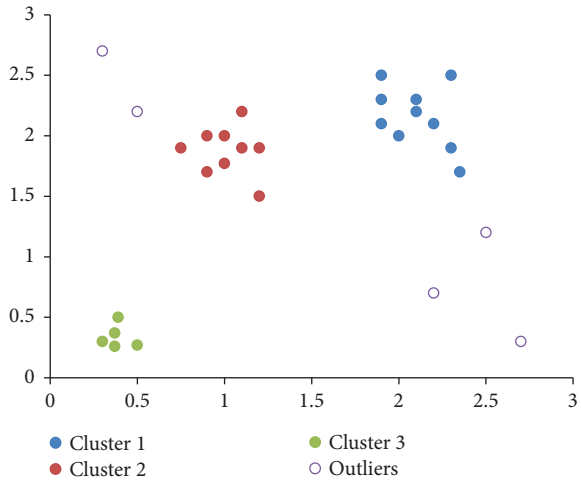

Figure 10: The distribution of points' roles when r is 0.21.

Table 1: The parameters of experiment data set.

Data set	N	K	D
Aggregation	788	7	2
Pathbased	300	3	2
Spiral	312	3	2
Jain	373	2	2
DIM	1024	16	32
KDDCUP04Bio	145751	2000	74
GL1	280307	16	18

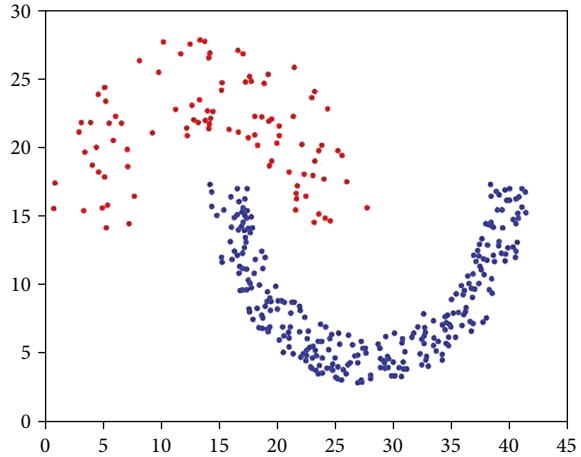

Figure 11: The distribution of JAIN.

4.1. The Accuracy of Clustering. Accuracy is one of the most important performances of clustering algorithms, and to compare different algorithms' clustering accuracy, the clustering purities of all algorithms in the same data set are calculated, and clustering purity has been used in most researches to judge clustering accuracy [24, 26, 27, 32]. purity = $\sum_{i=1}^{k}(|C_i^d|/|C_i|/K)$ in (13).

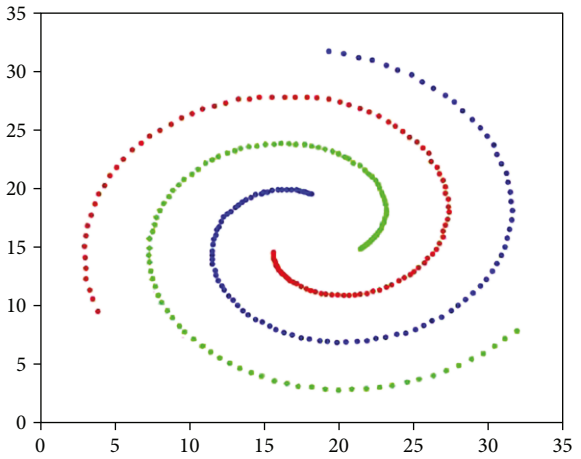

Figure 12: The distribution of Spiral.

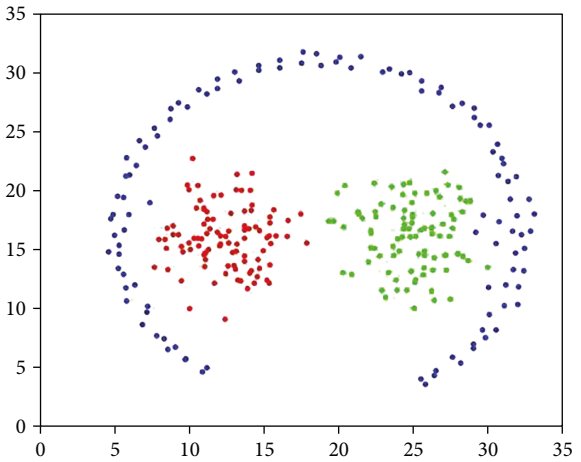

Figure 13: The distribution of Pathbased.

$$\text{purity} = \sum_{i=1}^{k} \frac{C_i^d/|C_i|}{K}. \tag{13}$$

In (13), K is the number of clusters, and $|C_i^d|$ represents the number of data points correctly distributed to cluster i, and $|C_i|$ represents the number of all data points in cluster i. Then, purity is the ratio of correctly clustered to all points, and its value is between 0 and 1. The ratio is higher, the clustering result is more accurate, so it can be used to directly illustrate the different algorithms' performance in the same data set, and the experiments in this section are compared based on clustering purity.

Before clustering, initial parameters should be preset to deal with different data sets, and in DBSCAN, DP-Clust, DCore, DCNaN, and DDPA-DP, the initial parameter should be set is the radius of local field, and in DPC-KNN

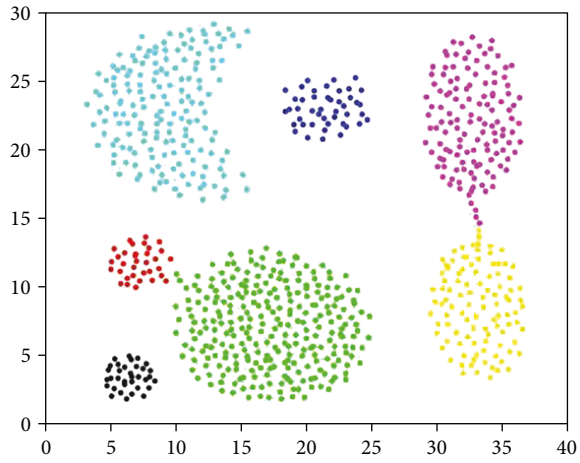

Figure 14: The distribution of Aggregation.

and FKNN-DPC, the initial parameter should be set is the neighbors' number of every point. In this paper, the initial parameters are defined also by data driven instead of by experience, and to overall simulate different applications, three initial states are designed in this section to better illustrate the parameter independence and accuracy of different algorithms. The first parameter state is set that the local field's radius $r1$ is as (14), and the neighbors' number K1 is by as (15).

$$r1 = \sum_{x=1}^{C} \sum_{i=1,j=1,i\neq j}^{i=n_x,j=n_x} \frac{2d_{ij}}{n_x C}, \tag{14}$$

$$\mathrm{K1} = \sum_{j=1}^{C} \sum_{i=1}^{N} \frac{x_i}{C}, \quad x_i = \begin{cases} 1, d(i,c_j) < r1, \\ 0, d(i,c_j) > r1. \end{cases} \tag{15}$$

In (14), C is the number of clusters in the data set, and n_x is the number of points in xth cluster, and it can be seen that $r1$ is computed by the average distance between points in the same clusters. Meanwhile in (15), N is the number of all points, and c_j is the center of cluster j, and if point i is located in the local field of c_j, x_i is set 1, otherwise it is set 0. It can be seen that K1 is computed by the average number of points in the centers' local fields.

The second parameter state is as the local field's radius $r2$ is as (16), in which the parameters' means are the same as (14), and it can be seen that $r2$ is computed by the average distance among all points in the data set. The neighbors' number K2 is as (17), and it can be seen that K2 is decided by the average number of points in all points' local fields.

$$r2 = \sum_{i=1,j=1,i\neq j}^{N} \frac{2d_{ij}}{N}, \tag{16}$$

$$\mathrm{K2} = \sum_{j=1}^{N} \sum_{i=1,i\neq j}^{N} \frac{x_i}{N}, \quad x_i = \begin{cases} 1, d(i,j) < r2, \\ 0, d(i,j) > r2. \end{cases} \tag{17}$$

The third parameter state is as the local field's radius $r3$ is decided by the average distance among the points in different clusters as (18), and the neighbors' number K is decided by the average number of points in all points' local fields too.

$$r3 = \sum_{C_x=1}^{C} \sum_{C_y=C_x+1}^{C} \sum_{i\in c_x, j\in c_y} \frac{2d_{ij}}{mn}. \tag{18}$$

In (18), m is the points' number in cluster c_x and n is the e points' number of c_y. It can be seen that $r3$ is computed by the average distance among the points in different clusters.

Among three initial local field's radiuses, $r1$ is the least because the distances between points in the same cluster are obviously less than the distances between different clusters as $r3$, and $r2$ is at the middle of $r1$ and $r3$, so by these three initial radiuses, DBSCAN, DP-Clust, DCore, DCNaN, and DDPA-DP can be relatively overall simulated and compared. Meanwhile, the initial values of K are also divided to three levels: K1 is the largest one because it is decided by the neighbors of centers, and centers have obviously more neighbors than other points; then, K2 is decided by the average neighbors of all points, and borders and outliers have less neighbors than centers, so it is obviously less than K1; K3 is at the middle of K1 and K2, because when computing K3, the points' local field is expanded, so it is larger than K2, however, it is also decided by all points' neighbors, and then it is less than K1. By these three levels of K, DPC-KNN and FKNN-DPC are also able to be overall compared.

In Figure 15, the clustering results of these algorithms for JAIN are shown. JAIN 1 means the state the local field's radius r in DBSCAN, DP-Clust, DCore, DCNaN, and DDPA-DP is set as $r1$ with value 2 computed by (14), and the neighbors' number K in DPC-KNN and FKNN-DPC is set as K1 with value 70 by (15); JAIN 2 means the state the local field's radius is set as set as $r2$ with value 2.5 by (16), and the neighbors' number is K2 with value 50 by (17); JAIN 3 means the state the local field's radius is set as $r3$ with value 2.75 by (18), and the neighbors' number is set as K3 with value 60 by (17) with $r2$. By Figure 15, it can be concluded that DDPA-DP has obvious advantage in the accuracy of clustering no matter what initial states set, and its accuracy is not less than 0.96; DCore and DCNaN have relative stable accuracy, but they have no advantage over DPC-KNN and FKNN-DPC; DPC-KNN and FKNN-DPC are influenced by the value of K, and the larger accuracy will be obtained with larger K; DP-Clust is much influenced by the value of r and it is just advanced than DBSCAN.

In Figure 16, the clustering accuracy of these algorithms in Spiral is shown, and because the distribution of Spiral is much unbalance and there are some sparse areas, the clustering accuracies of most algorithms are declined; however, DDPA-DP can still maintain the accuracy is not less than 0.95, and it is not much influenced by initial set of r. The results of DP-Clust, DPC-KNN, and FKNN-DPC are influenced by initial parameter much obviously, and the results of DCore and DCNaN are also obviously influenced by

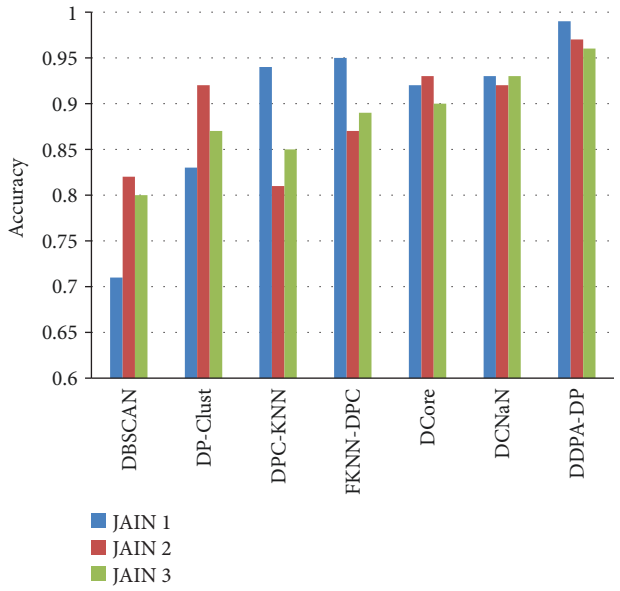

Figure 15: The simulation results in JAIN.

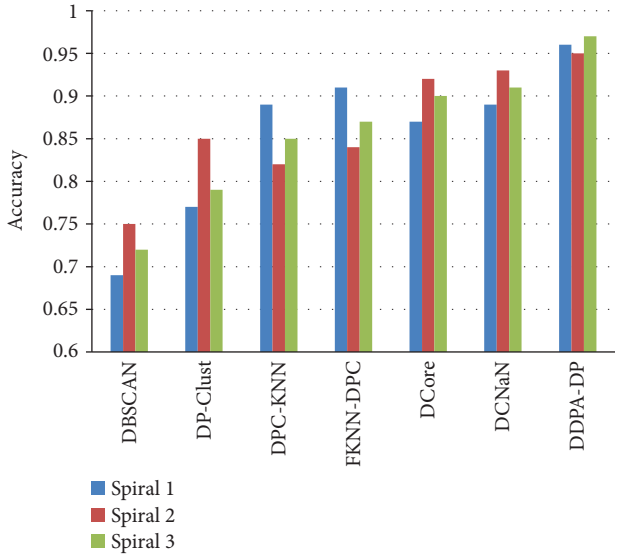

Figure 16: The simulation results in Spiral.

parameter. The initial parameters in Spiral 1 is that the local field's radius $r1$ is set as 1.5, and the neighbors' number K1 is set as 75; Spiral 2 means the state $r2$ is set as 1.7 and K2 is set as 50; Spiral 3 means the state $r3$ is set as 2 and K3 is set as 65.

In Figures 17 and 18, the clustering accuracies in Pathbased data set and Aggregation data set of these algorithms are shown. Although the density of these two data sets are relatively uniform, but the shapes of clusters are arbitrary. It can be seen that the accuracies of DDPA-DP are both stable and in a relative high level; and other algorithms' performances are not stable especially in Aggregation. According to (14), (15), (16), (17), and (18), in the initial parameters, Pathbased 1 means the local field's radius $r1$ is set as 5, and the neighbors' number K1 is set as 60; Pathbased 2 means $r2$ is set as 3, and the neighbors' number K2 is set as

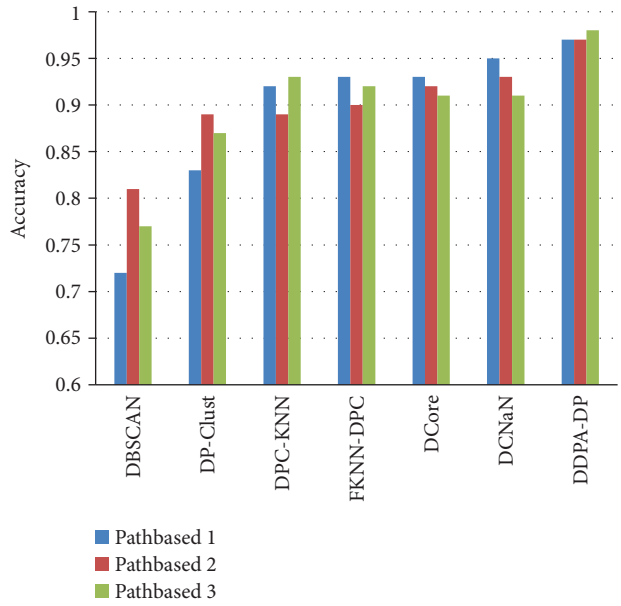

Figure 17: The simulation results in Pathbased.

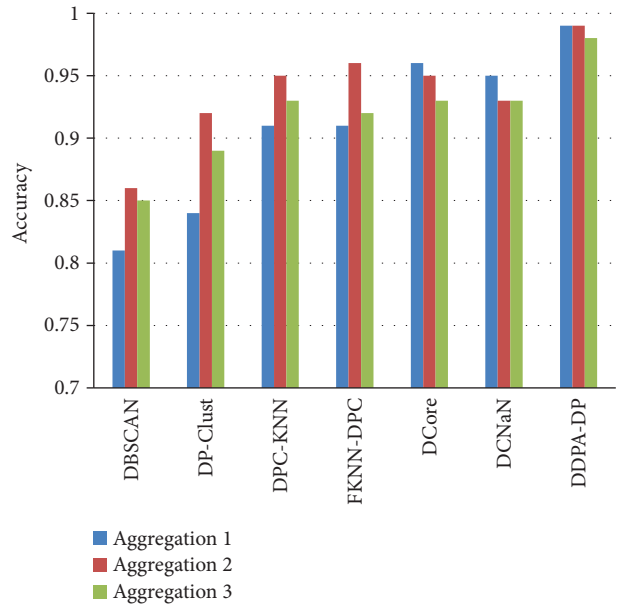

Figure 18: The simulation results in Aggregation.

50; Pathbased 3 means $r3$ is set as 4, and the neighbors' number K3 is set as 55. Then, the initial parameters in Aggregation 1 are that the local field's radius r is set as 2, and the neighbors' number K is set as 100; in Aggregation 2, the local field's radius r is set as 2.5 and the neighbors' number K is set as 85; in Aggregation 3, the local field's radius r is set as 3 and the neighbors' number K is set as 75.

In Figures 19–21, the clustering results of these algorithms in high-dimension data sets DIM, KDDCUP04Bio, and GL1 are shown, and in these data sets, the accuracies of all algorithms are declined obviously. In DIM, the accuracies of DBSCAN and DP-Clust are less than 0.75 in all

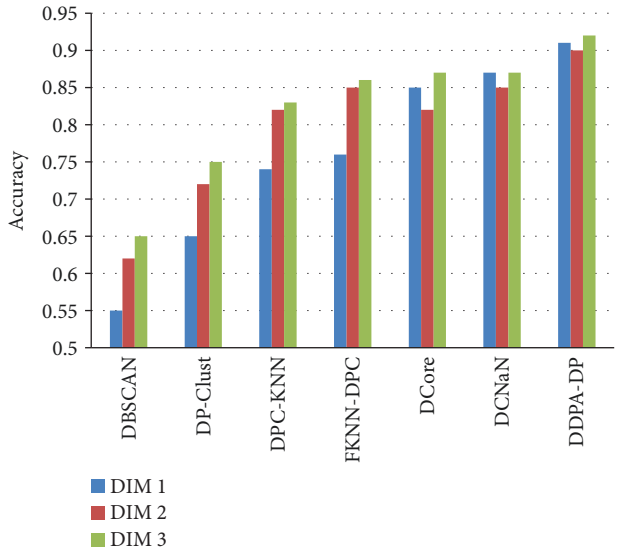

Figure 19: The simulation results in DIM.

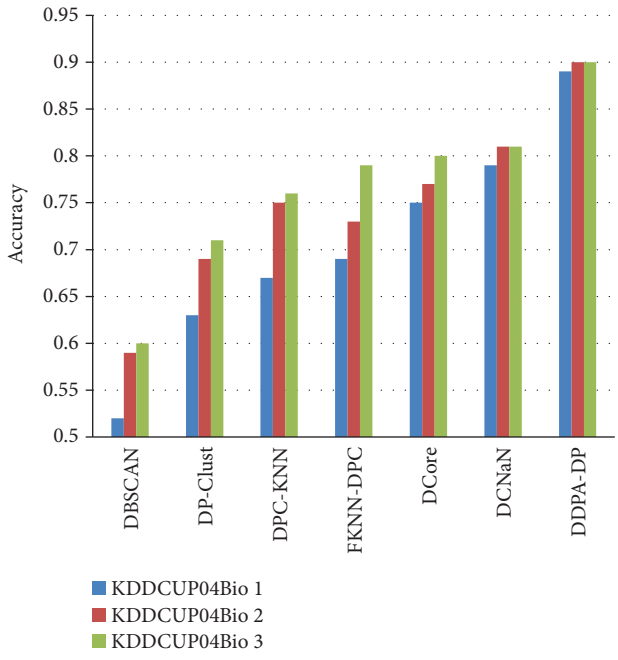

Figure 20: The simulation results in KDDCUP04Bio.

states; and although DPC-KNN and FKNN-DPC have relative large accuracies, but they are much influenced by the value of K, and the range abilities are larger than 10%; although DCore and DCNaN have stable accuracies, they are still less than DDPA-DP, and just DDPA-DP can obtain the clustering accuracy larger than 0.9. In DIM 1, the local field's radius r is set as a 32-dimension vector with value 5 and K is set as 100; in DIM 2, the local field's radius r is set as 6, and K is set as 120; in DIM 3, r is set as 7, and K is set as 150.

In Figure 20, the accuracies of clustering are further down because of the large number of data points in KDDCUP04Bio

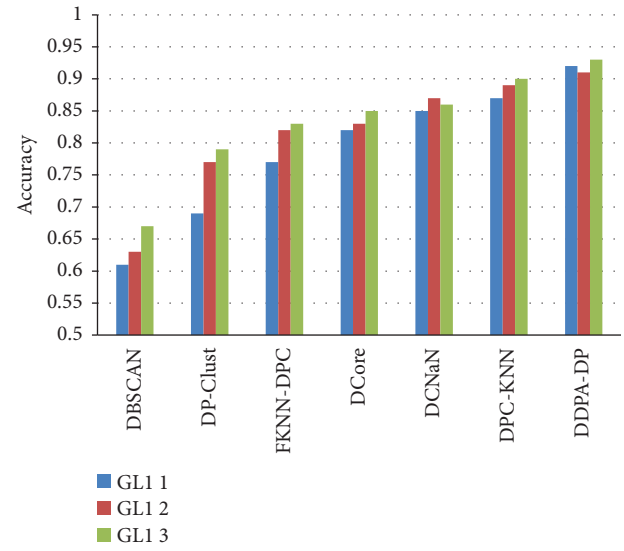

Figure 21: The simulation results in GL 1.

with high dimensions. By the results of Figure 20, the advantage of DDPA-DP is more obvious with the increase of points, and there are no any algorithms that can obtain the clustering accuracies larger than 0.8 except DDPA-DP, and DDPA-DP can maintain the accuracy as 0.9 in every state. In this data set, there are too many points to obtain the exact parameters by the method as the used in last five data sets, so we define the parameters by random sampling 20% points from all. Then by (14) to (18), in KDDCUP04Bio 1, the local field's radius $r1$ is set as a 74-dimension vector with value 20, and K1 is set as 2000; in KDDCUP04Bio 2, $r2$ is set as 27, and K2 is set as 1500; in KDDCUP04Bio 3, $r3$ is set as 35, and K3 is set as 1200.

In Figure 21, the data set GL1 is a real data collected from a thermal power plant, and the distribution of GL1 is more uniform than KDDCUP04Bio although it has more points. By Figure 21, all algorithms' accuracies are improved than KDDCUP04Bio, and the advantage of DDPA-DP is not obvious as KDDCUP04Bio, but it is still the most acute and stable one. The parameters in GL1 are set by the same method as in KDDCUP04Bio. In GL1 1, $r1$ is set as an 18-dimension vector with value 17, and K1 is set as 1750; in GL1 2, $r2$ is set as 21, and K2 is set as 1395; In GL1 3, $r3$ is set as 27, and K3 is set as 1535.

Based on the simulated results in this section, it can be concluded that DDPA-DP has obvious advantage in clustering accuracy, because the parameters in DDPA-DP are continuously adapted by the data-driven method, by which the parameters are optimized to improve the clustering accuracy, and then the optimized parameters can reduce the influence by initial set values which ensures the clustering accuracy is stable at high level. The advantages of DDPA-DP are more obvious with more complex data set, so DDPA-DP is fitter for the big data applications.

4.2. The Runtime of Clustering. Runtime is also an important standard to estimate the performance of clustering algorithm, and it can be used to estimate the time complexity of

clustering. In this section, the runtime of DDPA-DP is compared with DBSCAN, DP-Clust, DPC-KNN, FKNN-DPC, DCore, and DCNaN, and the results of experiments are shown in Figures 22 and 23. These results are obtained by computing the average runtime of every algorithm in the three states introduced in Section 4.1.

In Figure 22, the results of these algorithms simulated in relative small data sets are shown. From these results, it can be concluded that DP-Clust has the best time complexity of all, because the calculation procedure of DP-Clust is the simplest; among other algorithms, the runtimes have little differences, because in small data sets, the calculation procedure of iterating, optimizing, and searching neighbor points in local field can be accomplished in a short time.

In Figure 23, the runtimes of these algorithms in two large data sets are shown. Based on these results, it can be seen that the runtime of DDPA-DP has become obvious less than other ones except DBSCAN and DP-Clust. In DBSCAN and DP-Clust, all clustering operations are executed one time, which reduces their time complexity; in DCore and DCNaN, large data set means there will be many "false peaks" when detecting points' roles, and the discovery of density core needs many comparing operations, so much iteration will be executed in these two algorithms, which leads the runtime of DCore and DCNaN are the longest ones; in DPC-KNN and FKNN-DPC, the K neighbors should be used to judge the local fields for every point, and to obtain high clustering accuracy, the value of K is generally large in large data sets, and meanwhile, the iteration should be executed to optimize the choose of neighbors, which leads the time complexities of these two algorithms are just less than DCore and DCNaN and larger than others; in DDPA-DP, the optimization of local field radius r is determined by the distribution of "pending points," and these points are small in number among all points especially in large data sets, and the calculation of detecting pending points is much less than detecting other roles, so the time complexity in iteration is not much increased, and its runtime will be close to DP-Clust with high clustering accuracy.

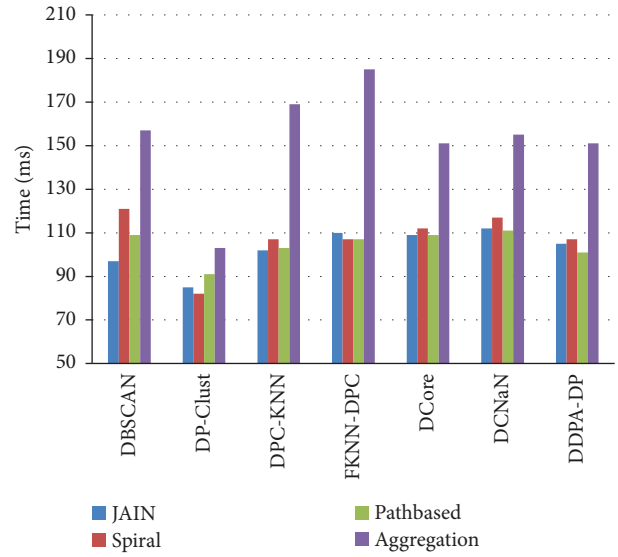

Figure 22: The runtimes in small data sets.

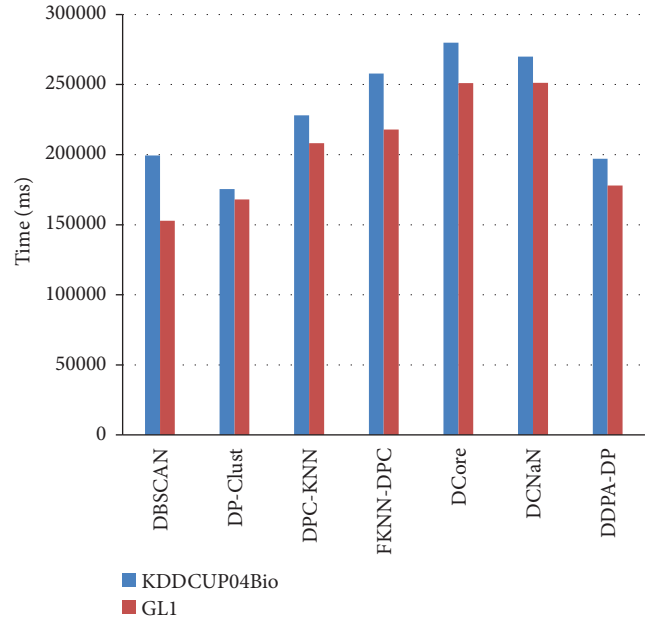

Figure 23: The runtimes in large data sets.

5. Conclusion

Based on the classical density-based clustering algorithm DP-Clust, we proposed a parameter adaptive clustering algorithm named DDPA-DP in this paper. The data-driven thought goes through the design of DDPA-DP: at first, a series of fitted curves are established to automatically detect points' roles by points' density attributes instead of any artificial thresholds; meanwhile, a new point's role "pending point" is defined, and then by the change of pending points' number, the local field's radius can be adaptively optimized.

DDPA-DP improves the flexibility of clustering by avoiding the influence of artificial parameters, and the time complexity of DDPA-DP is not significantly increased comparing with DP-Clust because there is little extra calculation added to optimize parameters. A series of experiments are designed to compare DDPA-DP with some existing clustering algorithms, and in these experiments, some typical synthetic data sets and a real-world data set from thermal power industry are simulated with different initial conditions to overall estimate these algorithms. By the results of experiments, it can be concluded that DDPA-DP has advantage in the performance of clustering accuracy and time complexity.

Data Availability

All artificial data sets can be downloaded from the following website: http://cs.uef.fi/sipu/datasets/.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by Natural Science Foundation of China under Contract no. 60573065 and Science and Technology Development Plan of Shandong Province under Contract no. 2014GGX101039, and it is partially supported by Natural Science Foundation of China under Contract no. 60903176.

References

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[2] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, no. 2-3, pp. 293–306, 1985.

[3] C. L. Philip Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: a survey on big data," *Information Sciences*, vol. 275, pp. 314–347, 2014.

[4] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[5] A. Cornuéjols, C. Wemmert, P. Gançarski, and Y. Bennani, "Collaborative clustering: why, when, what and how," *Information Fusion*, vol. 39, pp. 81–95, 2018.

[6] G. Bello-Orgaz, J. J. Jung, and D. Camacho, "Social big data: recent achievements and new challenges," *Information Fusion*, vol. 28, pp. 45–59, 2016.

[7] J. B. Mac Queen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5-th Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, University of California Press, Berkeley, 1967.

[8] T. Velmurugan and T. Santhanam, "Computational complexity between K-means and K-medoids clustering algorithms for normal and uniform distributions of data points," *Journal of Computer Science*, vol. 6, no. 3, pp. 363–368, 2010.

[9] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," *ACM SIGMOD Record*, vol. 25, no. 2, pp. 103–114, 1996.

[10] S. Guha, R. Rastogi, and K. Shim, "ROCK: a robust clustering algorithm for categorical attributes," *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, 1999, pp. 512–521, Sydney, NSW, Australia, 1999.

[11] G. Karypis, Eui-Hong Han, and V. Kumar, "Chameleon: hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.

[12] M. Ester, H. P. Kriegel, J. Sander, and X. W. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of Second International Conference on Knowledge Discovery and Data Mining*, vol. 96, pp. 226–231, Portland, OR, USA, 1996.

[13] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," in *SIGMOD '99 Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, vol. 28, no. 2, pp. 49–60, Philadelphia, Pennsylvania, USA, May-June 1999.

[14] W. Wang, J. Yang, and R. R. Muntz, "STING: A Statistical Information Grid Approach to Spatial Data Mining," in *Proceedings of International Conference on Very Large Data Bases*, pp. 186–195, Athens, Greece, 1997.

[15] M. Ankerst, M. Breuning, and H. Kriegel, "OPTICS: Ordering points to identify the clustering structure," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 49–60, ACM Press, New York, USA, 1999.

[16] C. M. M. Pereira and R. F. de Mello, "PTS: Projected Topological Stream clustering algorithm," *Neurocomputing*, vol. 180, pp. 16–26, 2016.

[17] E. W. M. Ma and T. W. S. Chow, "A new shifting grid clustering algorithm," *Pattern Recognition*, vol. 37, no. 3, pp. 503–514, 2004.

[18] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 133–142, San Jose, California, USA, 2007.

[19] L. Wan, W. K. Ng, X. H. Dang, P. S. Yu, and K. Zhang, "Density-based clustering of data streams at multiple resolutions," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 3, pp. 1–28, 2009.

[20] R. M. Aliguliyev, "Performance evaluation of density-based clustering methods," *Information Sciences*, vol. 179, no. 20, pp. 3583–3602, 2009.

[21] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[22] J. de Andrade Silva, E. R. Hruschka, and J. Gama, "An evolutionary algorithm for clustering data streams with a variable number of clusters," *Expert Systems with Applications*, vol. 67, pp. 228–238, 2017.

[23] J. Hou and W. Liu, "Parameter independent clustering based on dominant sets and cluster merging," *Information Sciences*, vol. 405, pp. 1–17, 2017.

[24] J. Nordhaug Myhre, K. Øyvind Mikalsen, S. Løkse, and R. Jenssen, "Robust clustering using a kNN mode seeking ensemble," *Pattern Recognition*, vol. 76, pp. 491–505, 2018.

[25] R. Mehmood, G. Zhang, R. Bie, H. Dawood, and H. Ahmad, "Clustering by fast search and find of density peaks via heat diffusion," *Neurocomputing*, vol. 208, pp. 210–217, 2016.

[26] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowledge-Based Systems*, vol. 99, pp. 135–145, 2016.

[27] C. Jinyin, L. Xiang, Z. Haibing, and B. Xintong, "A novel cluster center fast determination clustering algorithm," *Applied Soft Computing*, vol. 57, pp. 539–555, 2017.

[28] J.-Y. Chen and H.-H. He, "A fast density-based data stream clustering algorithm with cluster centers self-determined for mixed data," *Information Sciences*, vol. 345, pp. 271–293, 2016.

[29] F. Saki and N. Kehtarnavaz, "Online frame-based clustering with unknown number of clusters," *Pattern Recognition*, vol. 57, pp. 70–83, 2016.

[30] J. Xu, G. Wang, T. Li, W. Deng, and G. Gou, "Fat node leading tree for data stream clustering with density peaks," *Knowledge-Based Systems*, vol. 120, pp. 99–117, 2017.

[31] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted *k*-nearest neighbors," *Information Sciences*, vol. 354, pp. 19–40, 2016.

[32] L. Yaohui, M. Zhengming, and Y. Fang, "Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy," *Knowledge-Based Systems*, vol. 133, pp. 208–220, 2017.

[33] Y. Chen, S. Tang, L. Zhou et al., "Decentralized clustering by finding loose and distributed density cores," *Information Sciences*, vol. 433-434, pp. 510–526, 2018.

[34] J. Xie, Z.-Y. Xiong, Y.-F. Zhang, Y. Feng, and J. Ma, "Density core-based clustering algorithm with dynamic scanning radius," *Knowledge-Based Systems*, vol. 142, pp. 58–70, 2018.