Journal of Computers Vol. 31 No. 6, 2020, pp. 38-51
doi:10.3966/199115992020123106004

A Peak Density Clustering Algorithm Based on the Automatic Selection of the Cluster Center Points

Shi-Qi Cui, Bing Liu[*], Yong Li, Hui Liu

School of Computer Science &. Engineering, Changchun University of Technology, Changchun 130012, China
csqmvp@163.com;liubing@ccut.edu.cn;liyong@ccut.edu.cn;Liuhaa@163.com

Received 26 July 2019; Revised 23 December 2019; Accepted 20 January 2020

Abstract. The fast searching clustering algorithm of the density peak is a simple and efficient density-based clustering algorithm. However, there are shortcomings such as the setting of the truncation distance d_c is too sensitive, the similarity measure is too simple, and the artificial selection of the cluster center points is subjective. To deal with these problems, this paper proposes a new density peak clustering algorithm KE-DPC (KNN-ESD-density-peak-cluster) that can automatically select the cluster center points. First, the algorithm uses the near information to adjust the distribution of data samples, and optimizes the similarity measurement criterion in combination with Euclidean distance. Then the local density calculation formula is redefined according to the number of neighbor samples, thereby avoiding the setting of the sensitive d_c. Finally, the sample distribution on the decision map is fitted by linear regression to obtain the Residual set, and the cluster center point is automatically obtained according to the characteristics of the Residual analysis in ESD anomaly detection, removing the subjectivity of artificial selection. The experimental results of the artificial data set and UCI standard set show that the KE-DPC algorithm is better than K-means, DBSCAN, DPC, A-DPC and other algorithms.

Keywords: density peak, ESD anomaly detection, linear regression, near information

1 Introduction

With the rapid development of information technology and intelligent industry, Data shows exponential growth. The way to extract valuable information from massive data has become a concern of all circles. Data mining, as a technical means to obtain effective information, has been widely concerned in recent years. Cluster analysis, as an important means in the field of data mining, has developed rapidly, and has been widely used in the field of life science, image segmentation, pattern recognition [1], and so on.

The clustering algorithm is an unsupervised learning method that does not require class marking in advance. Its principle is the process of class clustering based on the similarity between data samples. The ultimate goal is to make the similarity of different cluster samples low, and make the similarity of the same cluster samples high [2]. At present, there are five most popular clustering algorithms: the partition-based clustering algorithm, the hierarchical-based clustering algorithm, the density-based clustering algorithm, the grid-based clustering algorithm and the model-based clustering algorithm [3]. The partition-based clustering algorithm is one of the first proposed algorithms, and the most typical of which are K-means [4] and K-medoids algorithm [5]. There are two typical hierarchy-based clustering algorithms, namely CURE algorithm [6] and CHAMELEON algorithm [7]. The grid-based clustering algorithm and the model-based clustering algorithm are relatively complicated, and the corresponding classical algorithms include CLIQUE [8] and GMM algorithm [9]. The density-based clustering algorithm is discussed in this paper, and the most representative of which is DBSCAN. The principle of the DBSCAN algorithm [10] is to find the neighborhood of the data through the spatial search technique,

[*] Corresponding Author

so that the samples of any shape can be effectively clustered. However, the DBSCAN algorithm has a strong dependence on input parameters [11], so it cannot objectively reflect real data. In 2014, Alex Rodriguez [12] published an article entitled "Clustering by Fast Search and Find of Density Peaks" in "Science". The algorithm proposed in this article can find clusters of arbitrary shapes without input parameters with strong dependence. The DPC (Density Peaks Clustering) algorithm described in this paper can find clusters of arbitrary shapes without inputting highly dependent parameters, which overcomes the shortcomings of K-means and DBSCAN algorithm.

Although the DPC algorithm has made great progress compared to the previous density clustering algorithm, but there are still some shortcomings: (1) the choice of the cutoff distance d_c can be sensitive, which will badly influence the density estimate of samples. (2) the use of Euclidean distance to define similarity is too simple and has limitations in many complex data sets. (3) during the process of finding the cluster center points, the artificial selections of cluster center points in the decision diagram should be used, which are subjective and can easily make mistakes on data sets with low discrimination, leading to bad clustering effects.

To deal with the problems above, a new KE-DPC algorithm is proposed. The innovation of this algorithm mainly includes the following points: (1) The structure of the sample distribution is adjusted by the combination of KNN's neighbor information and Euclidean distance, and the similarity measurement formula is redefined to solve the problem of simple similarity measurement criteria. (2) Redefine the calculation formula of the local density according to the new similarity of the sample and the K-nearest neighbor information, thereby replacing the d_c setting and eliminating the hidden danger of the d_c setting. (3) The cluster center is automatically selected based on the method of univariate linear regression and anomaly detection. It is data-driven and avoids the subjectivity of manually selecting cluster center points.

The other parts of this paper are composed as follows. The second part introduces the related work and basic principles of the DPC algorithm, the third part introduces the KE-DPC algorithm proposed in this paper, the fourth part is the comparison experiment, and the fifth part is the summary.

2 Related Work

The DPC algorithm has attracted widespread attention once it was proposed in 2014. Nowadays, many scholars have optimized it. The main optimization methods are divided into the following types: 1. Optimization of the cutoff distance d_c: Mehmood et al. [13] accurately selects the cutoff distance d_c based on the technology of thermal diffusion in the infinite field, and can estimate the probability distribution of the sample's density in a nonparametric state. Wang et al. [14] proposed a method to firstly establish a data field and then adaptively select d_c based on the potential energy relationship and information entropy between samples in the data field. It can solve the limitations of d_c selection, but it introduces the data field so the algorithm's calculation increases the redundancy. Zhou et al. [15] proposed an algorithm for finding the optimal cutoff distance d_c based on the fruit flies foraging method, which can replace the method of choosing d_c with experience to optimize the clustering effect, But the search process is very complicated. 2. Optimization of local density calculation and allocation method: Xie et al. [16] redefined the density calculation formula based on the KNN ideas, and then changed into the two-step allocation strategy based on the neighbor information to optimize the clustering effect. Zhang et al. [17] proposed an E_CFSFDP algorithm combined with the CHANELELEON algorithm. This algorithm uses the characteristics of hierarchical clustering to selectively merge the formed clusters, thereby optimizing the allocation strategy and improving the clustering effect. 3. Optimization for manual selection of cluster center points: Bie et all. [18] proposed an algorithm called Fuzzy-DPC based on fuzzy rules, which can firstly establish cluster center candidate sets, then adaptively merge according to fuzzy rules, and finally get the cluster center points. Li et all [19] proposed an A-DPC algorithm that automatically selects the cluster center points based on the slope, which can automatically determine the cluster center points based on the slope difference. Ma et all [20] proposed a method for automatically selecting cluster center points based on the trend of weight changes between samples. It can determine the maximum inflection point based on the relationship between weights, and then consider the point before the maximum inflection point as the cluster center point.

Compared with the above literatures, this paper has the following advantages: Xie et all. Optimized the density measurement formula by just using the K-nearest neighbor information. While this paper adjusted the distribution of the data based on the neighbor information, which can more accurately determine the similarity between samples to get more accurate local density. Li et all. automatically determines the cluster center points based on the change in slope, and Ma et all. automatically determines the cluster center points based on the change in weight. This paper proposes a new method for determining the cluster center points, The process of automatically selecting cluster center points is considered as the process of outlier measurement. The cluster center points are selected by using univariate linear regression and ESD anomaly detection.

DPC Algorithm

At first, the sample set are supposed to be $S = \{x_1, x_2, \cdots x_n\}, x_1, \cdots x_n$ means there being n samples. For any sample point *i*, $x_i = [x_{i1}, x_{i2}, \cdots x_{im}] 1 \le i \le n, x_{i1} \cdots x_{im}$ means there being m features. The process of clustering is the process of dividing the sample *S* into the sample sets $C = \{c_1 \cdots c_L\}$.

The DPC algorithm measures the similarity by taking the distance between any two points in the sample as a measure, which can be adapted to clusters of any shape. The algorithm has 2 prerequisites of assumption: (1) the cluster center points are surrounded by points with low local density. (2) the distance between different cluster centers is relatively far. Therefore, in order to satisfy the above prerequisites, the DPC algorithm introduces two quantities, namely the local density ρ and the distance δ. Among them, the local density has two calculation methods, which are calculated by using the Cutoff kernel function and the Gaussian kernel function. Their calculation formula is as follows:

$$\rho_i = \sum_{j \neq 1} x(d_{ij} - d_c) \begin{cases} x(e) = 1, e < 0 \\ x(e) = 0, e \ge 0 \end{cases}. \quad \textbf{(1)}$$

$$\rho_i = \sum_j \exp(-(d_{ij}{}^2 - d_c{}^2)). \quad \textbf{(2)}$$

$$d_{ij} = \| x_i - x_j \| = \sqrt{\sum_t^m (x_{it} - x_{jt})^2}. \quad \textbf{(3)}$$

d_{ij} is the Euclidean distance between sample point *i* and point *j*, and d_c is the cutoff distance (the specified coverage d_c is between 1% and 2% of the total data set [12]). Formula (1) will be used in sample sets of large scale and formula (2) will be used in sample sets of small scale. The δ distance is the closest distance from point *i* to its higher density points, and its formula is as follows:

$$\delta_i = \begin{cases} \min_j(d_{ij}), \rho_{ij} > \rho_i \\ \min_j(d_{ij}), \ other \end{cases}. \quad \textbf{(4)}$$

If the point *j* is the highest density, the farthest distance from its own is taken as its value of δ. Next, the decision diagram can be drawn with the density ρ as the horizontal coordinate and the distance δ as the vertical coordinate. The decision diagram is just as follows:

According to the Fig. 1, the two greater points on the upper right can be selected to be cluster center points. The literature [12] also pointed out another method, that plots decision diagram γ of the product of ρ and δ in order to find the cluster center points, so that the cluster center can be found more accurately. Each of the remaining points is assigned to the cluster in which the density is higher than it and the data point is the closest to it. Finally, the noise points can be found and marked, and the clustering is completed.

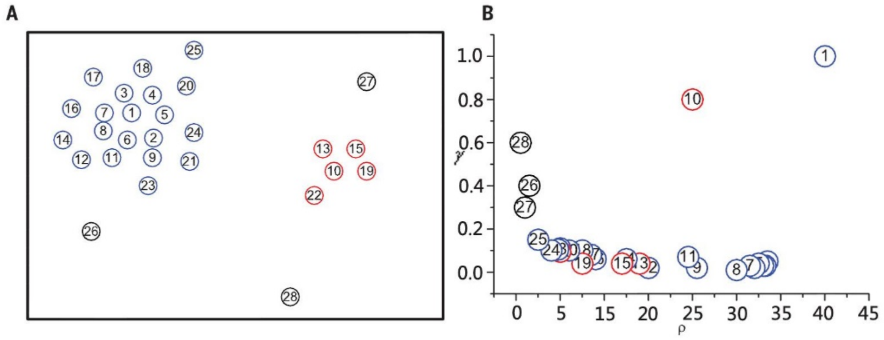

Fig. 1. Decision diagram

3 Detailed Explanation of KE-DPC Algorithm

The following introduces two parts of the KE-DPC algorithm. The first is the introduction of the optimization method of similarity, and the second is the introduction of how to automatically select the cluster center points

3.1 Similarity Optimization Based on K-nearest Neighbors

Fig. 2 shows two cluster diagrams and $d_{ba} = d_{bc}$. Due to the large difference in the density of the sample distribution, if the Euclidean distance is used to define the similarity, it is difficult to distinguish the similarity between *b* and *a* and the similarity between *b* and *c*.

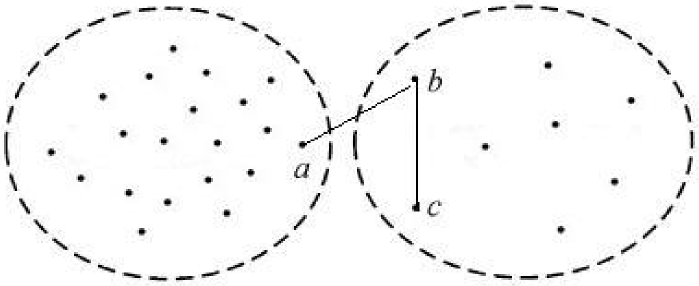

Fig. 2. Cluster diagram

However, it is clear from the distribution of the data that *b* and *c* are in the same area where the sample points are sparse, while sample point *a* is in a dense area. So *b* is similar to *c*. In view of the above problem, this paper uses KNN algorithm to consider the characteristics of local sample, combines the neighbor information into the process of similarity calculation, and uses the local characteristics of the neighbor to adjust the distance between the sample points within the class and the sample points between the classes, so that the sample points within the class are more similar and the sample points between the classes are more different from other classes. The new similarity definition formula is as follows:

Definition 1: Suppose the data set as *S*, and the similarity of any two points in the data set is:

$$sim(i, j) = \exp(-\frac{d(i, j)^2}{2\partial^2}(1 + \frac{|\sigma_i - \sigma_j|}{\sigma_{\max} - \sigma_{\min}})). \quad \textbf{(5)}$$

$$\sigma_i = \sqrt{\sum_t^m (x_{it} - x_{kt})^2} \begin{cases} 1 \le i \le n \\ 1 \le t \le m \end{cases}. \quad \textbf{(6)}$$

$$\hat{\sigma} = \frac{1}{n}\sum_i^n \sigma_i. \quad \textbf{(7)}$$

$$\sigma_{\max} = \max\{\sigma_i \cdots \sigma_n\}. \tag{8}$$

$$\sigma_{\min} = \mathrm{mix}\{\sigma_i \cdots \sigma_n\}. \tag{9}$$

Among them, $\sigma_1 = (i, k)$ is the Euclidean distance from point i to its nearest Kth neighbor, which can reflect the distribution of data around point i. $\sigma_{\max}$ is the maximum value of the distance set $\{\sigma_1, \sigma_2, \cdots \sigma_m\}$ is the minimum value and $\hat{\sigma}$ is the average value. $\dfrac{|\sigma_i - \sigma_j|}{\sigma_{\max} - \sigma_{\min}}$ is the weight adjustment coefficient, which can auto-adaptively adjust the distance between the intra-cluster sample points and inter-cluster sample points. The smaller the value of $|\sigma_i - \sigma_j|$ is, the greater the similarity between the sample points is, and vice versa.

Since point *b* and point *c* in Fig 2 are located in the sparse area, Point *a* is located in a dense area alone, So the value of $|\sigma_b - \sigma_c|$ is less than that of $|\sigma_b - \sigma_a|$. Because of $d_{ba} - d_{bc}$, other variables are counted into the formula (5), and $sim(b, c) > sim(b, a)$ can be worked out. Therefore, the new similarity measure can effectively determine the similarity between samples.

Because the optimized similarity measuring criteria adjusts the data distribution of the samples, the samples within the same cluster are closer, and the samples between different clusters are more sparse. Therefore, based on this feature and the ideas of literature [21], the calculation formula of the local density is redefined. The specific formula is as follows:

Definition 2: suppose *i* to be the data set as any point in *S*. $KNN(i) = \{x_1 \cdots x_k\}$ is the K points that are the most similar to *i*. The local density of point *i* is the sum of the similarities of *k* points.

$$\rho_i = \sum_{j \in KNN(i)} sim(i, j). \tag{10}$$

The redefined local density avoids the setting of d_c and better reflects the differences between samples, and make the distribution of sample points in the decision diagram clearer.

3.2 Judging the Cluster Center Points Based on Linear Fit and ESD Algorithm

Fig.3 reflects the mapping relationship of the data set in the decision diagram. The artificial selection of cluster center points is to select the point where the γ value is larger in Fig. 3. Because the artificial selection is subjective and the cluster center points and the non-cluster center points are too dense in some complex data sets with the less large discrimination, artificial selections will lead to improper selections.

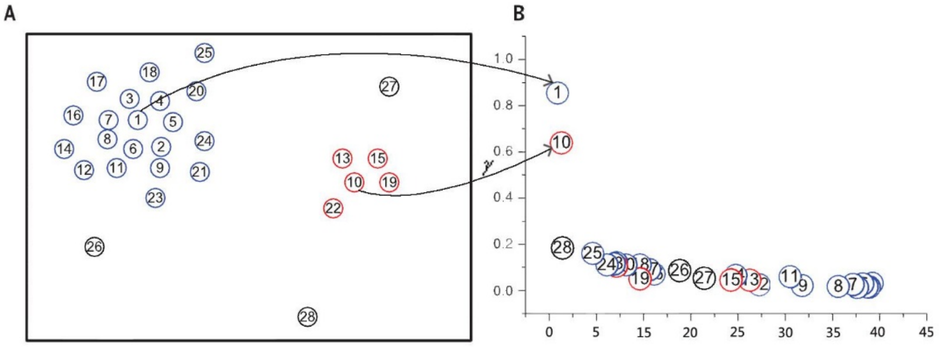

Fig. 3. γ decision diagram

This paper proposes a new automatic selection method of cluster center points, which mainly uses unary linear regression and the ideas of ESD anomaly detection. First of all, based on the mapping relationship in the Fig. 3, it can be clearly seen that most of the sample points are approximated on a straight line except for a few outliers in the decision diagram. If we use linear regression analysis to fit the sample points in the decision diagram, a set of residuals can be got, which is the candidate set of cluster center points. Then, the cluster center points can be obtained by the test in the ESD algorithm.

3.2.1 The Process of Linear Fit

Regression analysis is a statistic analysis method commonly used in mathematics [22], usually describing the relationship between two or more variables. When establishing a linear regression analysis, the quantitative relationship between two or more variables can be determined firstly, which is the process of setting mathematical models and judging unknown parameters, then using the unknown parameters to create a new linear fit. The assumed equation of linear regression is as follows:

$$y = b_1 x + b_0 + \varepsilon. \tag{11}$$

b_0 and b_1 are as the unknown parameters of the linear equation, and ε is a random error term of the linear equation.

The random error term of the linear function has the following characteristics: (1) ε is an average value or a random variable with an expected value of 0. (2) ε is independent and obeys a normal distribution. (3) ε is also the main reason why the model cannot accurately fit the data.

Assuming that the sample set on the decision graph is $\{(x_i, \gamma_i) \mid i = 0, 1, \cdots n\}$, by estimating the least squares (b_0, b_1), the estimator $(\hat{b}_0, \hat{b}_1)$ of the unknown parameter (b_0, b_1), can be obtained, and the sum of squared errors $y_i - \hat{y}_i$ can be minimized. The formulas are as follows:

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \min \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2. \tag{12}$$

$$\hat{y}_i = \hat{b}_1 x - \hat{b}_0. \tag{13}$$

Among them, $\hat{y}_i$ is the linear regression function. If

$$f(b_0, b_1) = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2. \tag{14}$$

By solving $\partial F / \partial b = 0$, the estimated parameters $\hat{b}_0$ and $\hat{b}_1$ can be obtained, which are the parameters of the linear regression model. The linear function in the decision diagram is as follows:

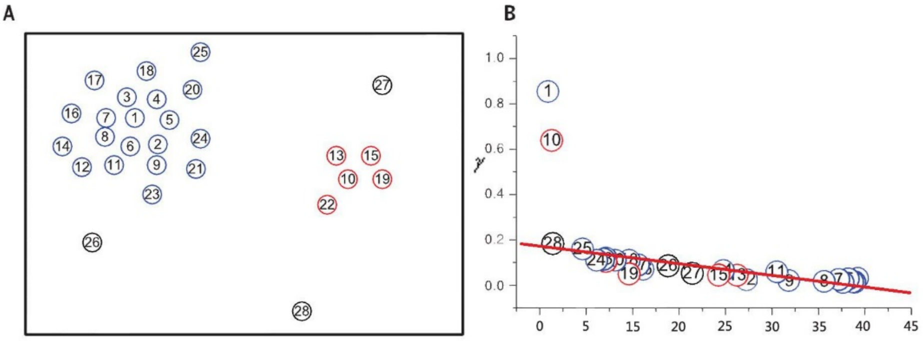

Fig. 4. Linear regression sample diagram

3.2.2 Judging the Center Points Based on the Thoughts of ESD Anomaly Detection

The ESD algorithm is a statistical-based anomaly detection method [23], which can judge more than one ab-normal point in real data. It is mainly based on the principle of hypothesis testing. Firstly, calculate the difference between the arbitrary point and the average of the entire data set. Then combine the standard deviation of the data set to find the test statistics. Finally, the test threshold is obtained based on the knowledge of the T-distribution. The point larger than the inspection threshold is the abnormal point.

Since most of the sample points in the decision graph are approximately fitted on a straight line. According to the idea of the ESD algorithm, this paper changes the difference between the mean of any point and the data set into the difference between the arbitrary point and the fitted value. In this way, the distribution of the samples can be considered more effectively, and a more accurate set of test statistics can be acquired. Finally, the test threshold λ_r is calculated by using the knowledge of the T distribution, and the cluster center points can be automatically determined. The specific process is as follows:

$$R_i = \frac{y_i - \hat{y}_i}{s}. \quad \textbf{(15)}$$

Among them, R_i is the standard residual, y_i is the y coordinate in the decision diagram γ , $\hat{y}_i$ is the fitting value of the linear regression, s is the standard deviation of the sample. among them, the larger R is, the larger the probability of the outlier is. The final standard residual diagram is as follows:

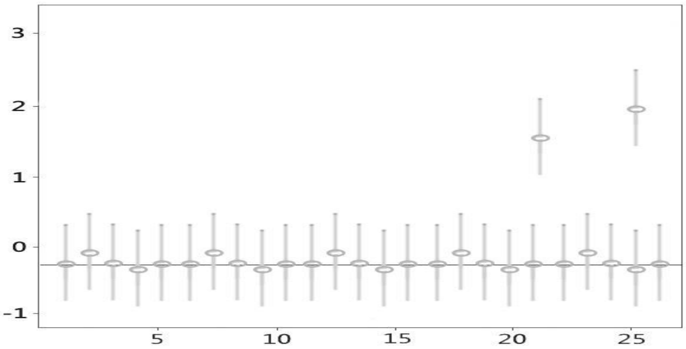

Fig. 5. Standard residual plot

From the inference of Gauss-Markov's theorem, there is a test threshold that shows the similarity between the true value and the predicted value for any standard residual R_i within the confidence coefficient. The residual calculated by linear regression is in accordance with the normal distribution, so the test threshold can be calculated under a confidence coefficient of $1-\alpha$. The formula is as follows:

$$\lambda_r = \frac{(n-r)t_p, n-r-1}{\sqrt{(n-r-1+t^2{}_p, n-r-1)(n-r+1)}}. \quad \textbf{(16)}$$

Where in, $t_{p,\,n-r-1}$ is the 100p percentile of the t-distribution with the DOF is *n-r-1*, n is the number of samples, and r is the number of outliers. The confidence coefficient level is calculated as follows:

$$p = 1 - \frac{a}{2(n-r-1)}. \quad \textbf{(17)}$$

Among them, α is the confidence factor of the confidence interval, referring to the interval test's require-ments of general confidence [24], where the confidence factor is 0.05, and then the 100p percentile is obtained according to the inverse function of the T distribution function. The formula is as follows:

$$f(t) = \frac{\Gamma(\frac{n-r}{2})}{\sqrt{\pi-(n-r-1)\Gamma\frac{n-r-1}{2}}}(1+\frac{r^2}{n-r-1})^{-(\frac{n-r}{2})}. \quad \textbf{(18)}$$

$$t_{p,\,n-r-1} = qt(p,(n-r-1)). \quad \textbf{(19)}$$

Among them, $f(t)$ is the density function of T-distribution, $qt(\,)$ is the inverse function of the T-distribution function.

According to the formulas above, when the value of r meets the condition that the number of the test statistic is greater than that of the test threshold $(R_i > \lambda_r)$, it is the number of outliers. then the distribution's way of DPC algorithm can be used and the cluster of sample point can be finished.

3.3 Clustering Steps for the KE-DPC Algorithm

The following mainly describes the components of the KE-DPC algorithm, which is mainly divided into three parts. The first part is the optimization of the similarity measurement formula with KE-DPC algorithm. the second is the linear fit part based on the estimation of the least squares. the final part is based on the ESD algorithm idea automatically selecting the cluster center. Specific steps are as follows:

Algorithm 1. KE-DPC

```
Input: data set $S = \{x_1, x_2, \cdots x_n\}$, nearest neighbor number $K$ and confidence factor $\alpha$.
Output: clustering results $c = \{c_1 \cdots c_L\}$.
Initialize dataset $S$
Calculate the Euclidean distance matrix $S^{n \times n} = \{d_{ij}\}^{n \times n}$ according to formula (3), among them, $d_{ii} = 0$.
Calculate the similarity matrix $sim\{ij\}^{n \times n}$ by parameter $K$ and formulas (5)~(9).
Calculate the local density $\rho_i$ according to formula (10).
Calculate the distance $\delta$ according to formula (4).
Plot the decision diagram $\gamma$ according to the calculated $\rho_i$ and $\delta$.
Calculate the linear fit according to Algorithm 2.
Calculate the cluster center points by Algorithm 3.
Assign the remaining sample points to the cluster where the cluster center is closest to itself, calculate the outliers, and complete the clustering.
```

Algorithm 2. Linear Fit based on the Estimation of Least Squares

```
Input: sample points on the decision diagram $\{(x_i, \gamma_i) \mid i = 0, 1, 2, \cdots n\}$
Output: the equation of the linear fit $\hat{y}_i = \hat{b}_1 x_i + \hat{b}_0$.
1. Initialize the $\gamma$ value of the sample point on the decision diagram.
2. Assume that the random variable obeys the following linear relationship: $y = b_1 x + b_0 + \varepsilon$.
3. Calculate the minimum value of the error squared of all sample points according to formula (12).
4. Find $(\hat{b}_0 + \hat{b}_1)$ from the first-order partial derivative $\partial F / \partial b = 0$.
5. Get the linear equation $\hat{y}_i = \hat{b}_1 x_i + \hat{b}_0$.
```

Algorithm 3. Automatically Select Cluster Centers based on ESD Algorithm

```
Input: data set $S = \{x_1, x_2, \cdots x_n\}$, nearest neighbor number $K$ and confidence factor $\alpha$.
Output: number of cluster center points.
1. Calculate the standard residual $R_i$ according to formula (15).
2. Calculate the confidence level $p$ based on the confidence factor $\alpha$ and equation (17)
3. Calculate the 100p percentile based on the inverse function of the formula (18) and formula (19).
4. According to formula (16), the test threshold $\lambda_r$ can be obtained. When $R_i > \lambda_r$, the points is the
   cluster center point.
```

4 Experimental Simulation and Analysis

The software environment of this experiment is python3. The hardware environment of the operating system is Windows7. The memory is 8G. The CPU is Inter Core i5-3230M. The main frequency is 2.60GHz. In order to verify the advantages of the KE-DPC algorithm, this paper selects 4 individual work sets and 6 UCI standard sets as the data sets of experimental test. The specific data is shown in Table 1. Then, DBSCAN, DPC, K-means, ADPC and other algorithms are compared in each performance index, and all the results are averaged for 10 experiments.

Table 1. Experimental data set

Dataset	Record	attributes	clusters
Jain	373	2	2
Aggregation	788	2	7
D31	3100	2	31
R15	600	2	15
Seeds	214	7	3
Iris	150	4	3
Vehicle	846	18	4
Wine	178	13	3
Waveform	5000	21	3
Waveform (noise)	5000	40	3

4.1 Comparative Analysis of Cluster Center

the following analyzes the KE-DPC algorism's and DPC algorism's accuracy of judging cluster center on artificial sets. the specific comparison diagram of the experiments is as follows:

Fig. 6 and Fig. 7 show the clustering effect diagram on the Jain dataset. It can be seen from the comparison of the two diagrams that the KE-DPC algorithm can effectively determine the cluster center points like artificial selection. As can be seen from the left half of Fig. 6 and Fig. 7, the KE-DPC algorithm is more discrimination than the DPC algorithm. which can more effectively determine the cluster center points. Then, as can be seen from the right half of Fig. 6 and Fig. 7, the DPC algorithm does not find the cluster center points of the sparse region, resulting in poor clustering effect. The KE-DPC algorithm considers the sample distribution of the data set, and combines the local structure of the sample points to adjust the distance between the sample points within the class and between the classes. it more accurately selects the cluster center points of the cluster-distributed sparse regions, and effectively complete clustering.

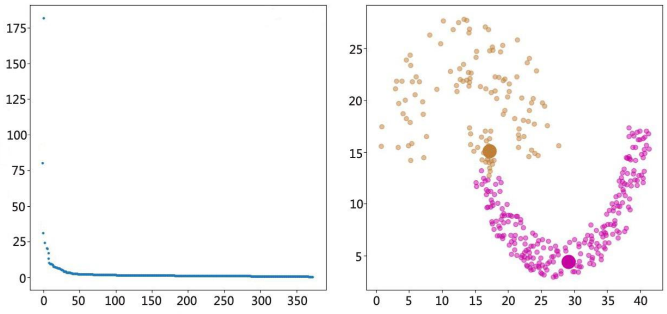

Fig. 6. Jain (DPC)

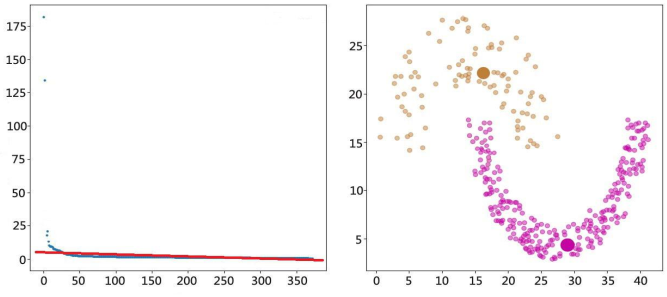

Fig. 7. Jain (KE-DPC)

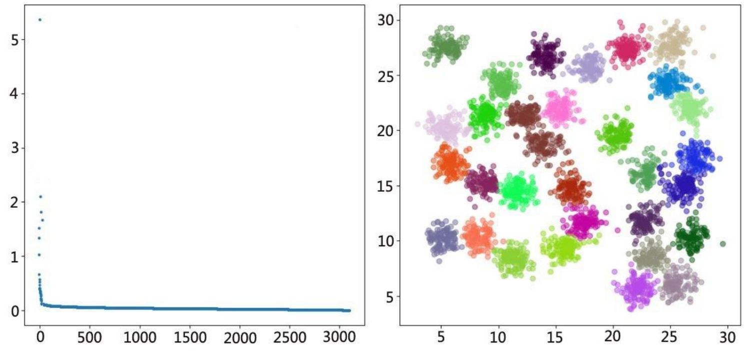

Fig. 8. D31 (DPC)

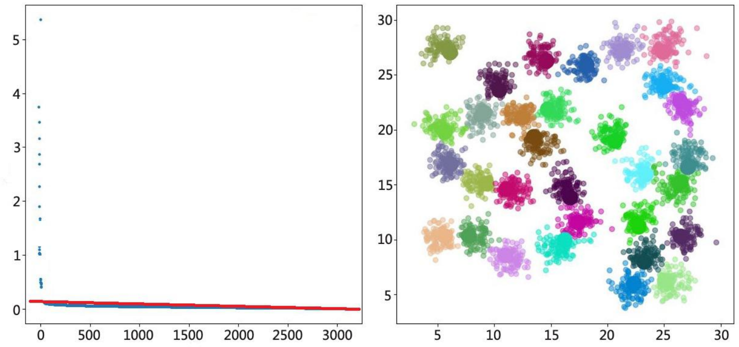

Fig. 9. D31 (KE-DPC)

By comparing the KE-DPC and DPC on the D31 dataset, it can be seen that the DPC algorithm is difficult to manually select the cluster centers, because the D31 dataset is larger and the categories are much more, the discrimination of the sample points on the decision graph is not obvious. It is easy to choose more or miss. On the decision graph constructed by KE-DPC algorithm, it can be seen that the discrimination of sample points is more obvious than the DPC algorithm's, and the candidate set of the cluster center points can be judged more reliably. Then linear regression is used to find the standard residual set. The standard residual set is as follows:

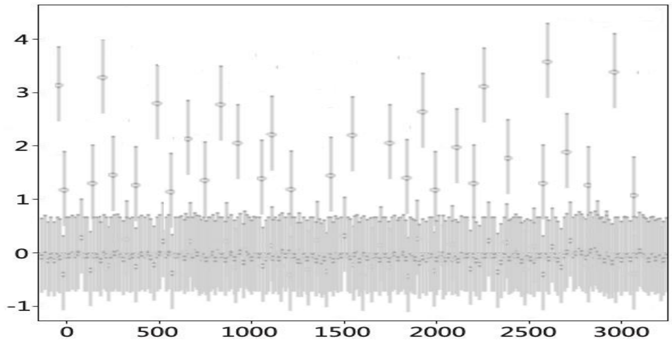

Fig. 10. D31 residual map

After the residual set being obtained, the test threshold λ_r can be obtained by using the ESD algorithm, and finally the cluster center points can be accurately determined.

Fig. 11 and Fig. 12 are the clustering's effect diagrams of KE-DPC on Aggregation dataset and R15 dataset respectively. Since the dataset distribution is relatively simple, it can be seen from the figure that the KE-DPC algorithm can effectively judge the number of cluster center points with a good clustering effect.

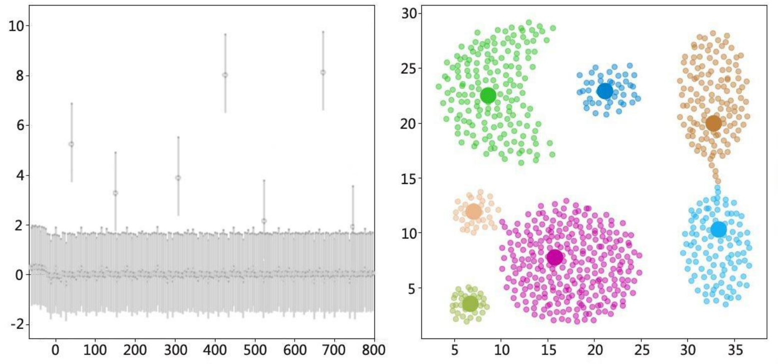

Fig. 11. Aggregation (KE-DPC)

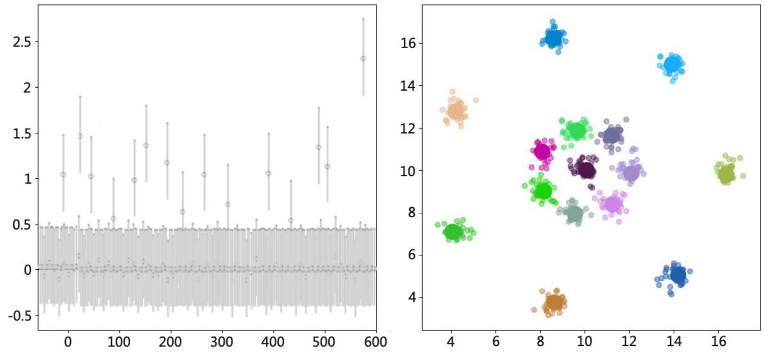

Fig. 12. R15 (KE-DPC)

In summary, compared with the DCP algorithm, KE-DPC can accurately determine the cluster center on a simple data set, and can also adjust the intra-class structure and the inter-class structure on the complex data set, which more accurately determines the cluster center. It avoids the subjectivity and the contingency of artificial selection, and realizes the function of automatically selecting the cluster center.

4.2 Analysis of Experimental Result

In order to clearly verify the validity of the proposed KE-DPC, In this paper, three evaluation indicators are used to evaluate and analyze the clustering results on the artificially synthesized set and the standard UCI standard set. The three evaluation indicators are F-measure, recalling rate (NMI) and accuracy (ACC). The accuracy rate represents the proportion of the samples with the correct clustering result. The higher the value is, the better the clustering effect has. Its calculation formula is as follows:

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n} h(\gamma_i, c_i). \tag{19}$$

Among them, γ_i represents the real class of the sample *i*, and c_i represents the clustered class of the sample *i*. If the class of γ_i and the class of c_i are the same, $h(\gamma_i, c_i) = 1$, otherwise it is 0. The NMI reflects the degree of matching between the class label of clustering result and the class label of real sample, which can reflect the robustness of the clustering result. Its calculation formula is as follows:

$$NMI = \frac{I(R,C)}{\sqrt{H(R)H(C)}}. \tag{20}$$

$$I(R,C) = H(R) - H(R \mid C). \tag{21}$$

$H(R)$ represents the entropy of the correct classification R, and $H(C)$ represents the entropy of the algorithm's clustering result C. The F-measure indicator is the weighted harmonic mean of Precision and Recall [25], It is a commonly used evaluation standard, which can effectively explain the clustering effect. The larger the F-measure is, the better the clustering effect will have; the smaller the F-measure is, the worse the class effect will have. P_j will be treated as artificially labeled known-cluster and C_i as a cluster formed after clustering. The specific calculation formula is as follows:

$$\text{Precision: } P(P_j, C_i) = \frac{|P_j \cap C_i|}{|C_i|}. \tag{22}$$

$$\text{Recall: } R(P_j, C_i) = \frac{|P_j \cap C_i|}{|P_j|}. \tag{23}$$

$$\text{F-measure: } F(P_j \cap C_i) = \frac{2 \cdot P(P_j, C_i) \cdot R(P_j, C_i)}{P(P_j, C_i) + R(P_j, C_i)}. \tag{24}$$

Table 2 shows the experimental results of three evaluation indexes of K-means, DBSCAN, DPC, ADPC, KE-DPC and other clustering algorithms on the UCI dataset. All the results use the average value after 10 experiments. The specific results are as follows:

Table 2. Performances of different clustering algorithms on UCI datasets

Algorthm	Iris			Seeds			Vehicle		
	ACC	NMI	FMI	ACC	NMI	FMI	ACC	NMI	FMI
K-mean	0.765	0.696	0.752	0.877	0.405	0.752	0.365	0.103	0.325
DBSCAN	0.667	0.305	0.398	0.404	0.005	0.398	0.420	0.167	**0.418**
DPC	0.695	0.653	0.814	0.890	0.724	0.710	0.363	0.142	0.358
ADPC	0.705	0.696	0.883	0.894	0.756	0.803	0.365	0.138	0.387
KE-DPC	**0.894**	**0.757**	**0.907**	**0.905**	**0.757**	**0.857**	**0.421**	**0.170**	0.399
Algorthm	Wine			Waveform			Waveform (noise)		
	ACC	NMI	FMI	ACC	NMI	FMI	ACC	NMI	FMI
K-mean	0.509	0.350	0.550	0.501	0.335	0.535	0.512	0.256	**0.501**
DBSCAN	0.516	0.432	0.532	0.412	0.452	0.481	0.389	0.331	0.577
DPC	0.702	0.517	0.677	O.574	O.474	O.544	0.535	**0.434**	0.458
ADPC	0,704	0.524	0,704	0.589	**0.519**	0.557	0.531	0.401	0.489
KE-DPC	**0.723**	**0.603**	**0.753**	**0.622**	0.502	**0.602**	**0.566**	0.405	0.500

Overall, the KE-DPC algorithm is superior to other algorithms in ACC indexes. Especially on the Iris dataset, it is 19.9% higher than the DPC algorithm, and 5.8% higher on the higher-dimensional Vehicle data. Under the NMI index, the KE-DPC algorithm achieves the best results in the Iris, Seeds, and Wine datasets, but it didn't obtain the optimal results under the higher-dimensional datasets like Vehicle, Waveform, and Wave-form (noise) datasets, which indicates that the robustness of this algorithm under the dimensional data set is not good. Under the FMI index, KE-DPC has a significant improvement over the DPC algorithm. Especially on the Iris dataset, it increased by 9.3% and on the Seeds dataset increased by 14.3%. However, it is slightly lower than the DBSCAN and K-means algorithms on the Vehicle and Waveform (noise) data sets, which shows that the KE-DPC algorithm is not the most optimal on some data sets with high dimensions or complex distribution.

In summary, according to the comparison experiment of UCI data set, it can be concluded that the KE-DPC algorithm has a better optimizing effect than the DPC algorithm's, and the accuracy, the optimization of the clustering effect and other aspects of judging the cluster centers are improved a lot.

5 Conclusion

Although the traditional DPC algorithm has made great progress, it still has some defects in the similarity measure, the artificial selection of cluster center points and other aspects. To deal with these shortcomings, combined with some ideas from K-Nearest Neighbor, ESD algorithm and other algorithms, this paper proposes a new KE-DPC algorithm which can automatically select the cluster center points. It can adjust the local distribution of the samples through the neighbor information, and then optimize the similarity's measurement and the calculation formula of the local density. Finally, it automatically select the cluster center points accurately based on the theory of linear regression and anomaly detection. Experiments on the synthetic set and the UCI standard set show that the clustering effect of the KE-DPC algorithm is better than that of the DPC algorithm. The clustering effect of the KE-DPC algorithm will be better especially in the data set with less sample discrimination. However, it can be seen that the NMI indexes are very low and the robustness is poor in the high-dimensional data set Waveform, and it doesn't deal with the shortcomings in distribution. Therefore, the next research is to solve how to deal with high-dimensional data sets and the optimization of distribution.

Acknowledgements

Foundation items: Jilin Province Education Department Science and Technology Research and Planning Project, Clinical Big Data Analysis and Application for Diabetes, No: JJKH20181046KJ; Jilin Province Education Department Science and Technology Research and Planning Project, Energy-efficient and efficient cloud data center resource scheduling strategy research, number: JJKH20191310KJ.

References

[1] C.C. Aggarwal, Data Classification: Algorithms and Applications. CRC Press, Boca Raton, 2015.

[2] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, Annals of Data Science 2(2)(2015) 165-193.

[3] R. Liu, W. Huang, Z. Fei, K. Wang, J. Liang, Constraint-based clustering by fast search and find of density peaks, Neurocomputing 330(2019) 223-237.

[4] S. Khanmohammadi, N. Adibeig, S. Shanehbandy, An improved overlapping k-means clustering method for medical applications, Expert Systems with Applications 67(2017) 12-18.

[5] D.-H. Yu, M.-Z. Guo, Y. Liu, S.-J. Ren, X.-Y. Liu, G.-J. Liu, The K-medoids clustering algorithm based on distance inequality, Journal of software (12)(2017) 3115-3128.

[6] D. Saravanan, CURE clustering technique suitable for video data retrieval, in: Proc. 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2016.

[7] U. Gupta, N. Patil, Recommender system based on hierarchical clustering algorithm chameleon, in: Proc. 2015 IEEE International Advance Computing Conference (IACC), 2015.

[8] Y. Fan, N. Li, C. Li, Z. Ma, L.-J. Latecki, K. Sun, Restart and random walk in local search for maximum vertex weight cliques with evaluations in clustering aggregation, in: Proc. 26th International Joint Conference on Artificial Intelligence, 2017. DOI: 10.24963/ijcai.2017/87.

[9] M. Heck, S. Sakriani, S. Nakamura, Feature optimized DPGMM clustering for unsupervised subword modeling: a contribution to zerospeech 2017, in: Proc. 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2017.

[10] J. Hou, H.-J. Gao, X.-L. Li, DSets-DBSCAN: a parameter-free clustering algorithm, IEEE Transactions on Image Processing 25(7)(2016) 3182-3193.

[11] D. Ienco, G. Bordogna, Fuzzy extensions of the DBScan clustering algorithm, Soft Computing 22(5)(2018) 1719-1730.

[12] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344.6191 (2014) 1492-1496.

[13] R. Mehmood, G. Zhang, R. Bei, H. Dawood, Clustering by fast search and find of density peaks via heat diffusion, Neurocomputing 208(2016) 210-217.

[14] S. Wang, D. Wang, C. Li, Y. Li, G. Ding, Clustering by fast search and find of density peaks with data field, Chinese Journal of Electronics 25(3)(2016) 397-402.

[15] R. Zhou, Q. Liu, Z. Xu, L. Wang, X. Han, Improved fruit fly optimization algorithm-based density peak clustering and its applications, Tehnicki vjesnik 24(2)(2017) 473-480.

[16] J. Xie, H. Gao, W. Xie, X. Liu, P.-W Grant, Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors, Information Sciences 354(2016) 19-40.

[17] W. Zhang, J. Li, Extended fast search clustering algorithm: widely density clusters, no density peaks. <https://arxiv.org/abs/1505.05610>, 2015 (accessed 21.05.15).

[18] R. Bie, R. Mehmood, S. Ruan, Y. Sun, H. Dawood, Adaptive fuzzy clustering by fast search and find of density peaks, Personal and Ubiquitous Computing 20(5)(2016) 785-793.

[19] T. Li, H. Ge, S. Su, Density peaks clustering by automatic determination of cluster centers, Journal of Frontiers of Computer Science and Technology 10(11)(2016) 1614-1622.

[20] C.-L. Ma, H. Shan, T. Ma, Improved density peaks based clustering algorithm with strategy choosing cluster center automatically, Computer Science 43(7)(2016) 255-258.

[21] R. Liu, H. Wang, X. Yu, Shared-nearest-neighbor-based clustering by fast search and find of density peaks, Information Sciences 450(2018) 200-226.

[22] R.-F. Gunst, Regression Analysis and Its Application: A Data-oriented Approach, CRC Press, Boca Raton, 2018.

[23] J. Hochenbaum, O.-S. Vallis, A. Kejariwal, Automatic anomaly detection in the cloud via statistical learning. <https://arxiv.org/abs/1704.07706>, 2017 (accessed 24.04.17).

[24] J.-Y. Chen, H.-H. He, Research on density-based clustering algorithm for mixed data with determine cluster centers automatically, Acta Automatica Sinica 41(10)(2015) 1798-1813.

[25] A. Borji, M.-M. Chen, H. Jiang, J. Li, Salient object detection: a benchmark, IEEE transactions on image processing 24(12)(2015) 5706-5722.