Applied Soft Computing 57 (2017) 539–555

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A novel cluster center fast determination clustering algorithm

Chen Jinyin*, Lin Xiang, Zheng Haibing, Bao Xintong

College of Information Engineering, Zhejiang University of Technology, Hangzhou, China

ARTICLE INFO

Article history:
Received 1 November 2016
Received in revised form 4 March 2017
Accepted 18 April 2017
Available online 23 April 2017

Keywords:
Data mining
Clustering algorithm
Rapid determination of cluster centers
Density based clustering

聚类中心是具有较高密度且彼此远离的数据点，引入一个决策图（距离*密度），设置一个置信区间，置信区间外的奇异点被视为聚类中心，然后时间扫描聚类按密度设计其余点，以完成聚类。

ABSTRACT

As one of the most important techniques in data mining, cluster analysis has attracted more and more attentions in this big data era. Most clustering algorithms have encountered with challenges including cluster centers determination difficulty, low clustering accuracy, uneven clustering efficiency of different data sets and sensible parameter dependence. Aiming at clustering center determination difficulty and parameter dependence, a novel cluster center fast determination clustering algorithm was proposed in this paper. It is supposed that clustering centers are those data points with higher density and larger distance from other data points of higher density. Normal distribution curves are designed to fit the density distribution curve of density distance product. And the singular points outside the confidence interval by setting the confidence interval are proved to be clustering centers by theory analysis and simulations. Finally, according to these clustering centers, a time scan clustering is designed for the rest of the points by density to complete the clustering. Density radius is a sensible parameter in calculating density for each data point, mountain climbing algorithm is thus used to realize self-adaptive density radius. Abundant typical benchmark data sets are testified to evaluate the performance of the brought up algorithms compared with other clustering algorithms in both aspects of clustering quality and time complexity.

1. Introduction

Clustering refers to the process of the collection of physical or abstract objects into a number of classes, which are composed of similar objects. With the development of big data technology, cluster analysis is widely used in the fields of finance, marketing, information retrieval, information filtering, scientific observation and engineering [1]. There are many different clustering methods, such as partitioning, hierarchical, density-based, grid-based, and their combinations [2,3].

Partition based clustering algorithms include k-means [4], K-medoids [5] and PAM [6]. The similarity calculation in K-means algorithm is based on the mean value of objects in a cluster. The goal of this algorithm is to divide the data set into K clusters, where K is a predefined parameter. In each round, each cluster is composed of the points nearest to the corresponding reference point, and the centroid of each cluster is taken as the reference point of the next round. Such iterations make the selection of the reference point closer to the true cluster centroid, so the clustering effect is getting better and better. K-medoids algorithm is similar to the k-means algorithm. Firstly, an initial reference point is selected for each cluster. Secondly, the remaining points are assigned to the nearest cluster according to the distance from each reference point. Then, the reference points are repeatedly replaced by non-reference points to improve the clustering effect. PAM algorithm is used to analyze all the objects and the object of each class is viewed as the central point. For various possible combinations, the quality of the clustering results is estimated. PAM method can work well on small-sized data sets, while the effect is not ideal for large data collection.

Level based clustering algorithms includes BIRCH [7], CURE [8] and ROCK [9]. BIRCH is a comprehensive hierarchical clustering method. Data set is stored in a compact compressed format and is clustered directly on the compressed form. Its I/O cost is linear with the size of the data set. BIRCH is particularly suitable for large data sets, and supports incremental clustering or dynamic clustering. Algorithm scan data set can be used to generate a better clustering and increasing the number of scans can be used to further improve the quality of clustering. Experiments show that the algorithm has a linear expansion of the number of objects and better clustering quality. However, if the data cluster is not spherical, BIRCH cannot achieve satisfied results because the concept of radius or diameter is adopted to control the boundary of clustering. CURE algorithm adopts a novel hierarchical clustering algorithm, which uses the

* Corresponding author.
E-mail addresses: chenjinyin@zjut.edu.cn, chenjinyin@163.com (C. Jinyin).

http://dx.doi.org/10.1016/j.asoc.2017.04.031
1568-4946/

intermediate strategy based on the center of mass and the representative object method. Instead of using a single centroid or object to represent a cluster. A cluster is represented by a number of representative points in the data space and makes them shrink to the cluster center with a shrinkage factor. Thus CURE cannot identify clusters with large scale nor non spherical shape, and it inhibits the effect of isolated point.

Density based clustering algorithm includes DBSCAN [10], PTICS [11] and DENCLUE [12]. DBSCAN is a typical density clustering method. It defines core points whose density is greater than threshold by introducing the concept of density reachable. Adjacent core points can be directly reachable from each other. And all reachable points form a cluster. The points belonging to no class are regarded as noise. DBSCAN algorithm with no preprocessing directly clusters the whole data set. When the amount of data is quite large, big memory is needed to support algorithm running and Input and output consumption. The time complexity of DBSCAN(Density based Spatial Clustering of Applications with Noise) is relatively high, mainly determined by the query operation. DBSCAN algorithm is very sensitive to the parameters of Eps and Minpts which are difficult to determine in prior. OPTICS algorithm is an extension of DBSCAN. Instead of generating a data set of clustering explicitly by generating a parameterized sort which can represent the density based clustering structure. The results of clustering can be represented by graphs or other visualization techniques. DENCLUE is a clustering algorithm based on a set of density distribution function. This algorithm is mainly based on the following ideas: (1) The impact of each data point can be simulated with a mathematical function which describes the impact of a data point to its neighborhood, called "influence function". (2) The overall density of the data space can be modeled as the sum of the influence functions of all data points. (3) The clustering can be obtained by determining the density attraction point (attractor density). The density attraction here is the local maximum of the global density function.

In order to cluster data with mixed attributes, Huang proposed k-prototypes [17] which combine k-means and k-mode algorithms. Considering the uncertainly character of data, KL-FCM-GM [18] extends k-prototypes algorithm. KL-FCM-GM is an extension of Gath-Geva, which is designed for the Guss-Multinomial distributed data. EKP [19] is developed based on an evolutionary algorithm framework to help k-prototypes improve global search capability. Distance-based Agglomerative Clustering algorithm (SBAC)[16] was proposed adopting the distance measure defined by Goodall. CAVE [27] is designed for clustering mixed data based on the variance and entropy. However, CAVE needs to build the distance hierarchy for each categorical attribute, while the determination of distance hierarchy requires the domain expertise. Another k-means type algorithm [28] is implemented to deal with mixed data by using co-occurrence of categorical values to calculate the significance of attribute and the distance between categorical values. Pathak A and Pal N R develop a fuzzy (soft) clustering framework to find a partition of a mixed dataset by exploiting the common cluster substructure present in both categorical and numeric data [23]. IWKM [15] combines the mean value of all distribution centroids to represent the prototypes of the cluster and takes into account the significance of each attribute towards the clustering process. WFK-prototypes [20] combines the mean value of fuzzy centroids to represent the prototypes of the cluster and adopt the significance concepts proposed by A. Amir [28] to extend k-prototypes [22,24].

Rodriguez team proposed an algorithm [13] on SCIENCE journal which is based on main idea like a center point and have a high density ρ while assuming a large distance between the center point and the point of high density (RLM algorithm). If the algorithm could make following unsettled problem solved, it would be faster and more applicable. (1)In this algorithm, the sketch of density and distance varies with the density radius d_c which makes d_c directly influence effect of cluster. (2) This algorithm is unable to automatically define the clustering center points. We need to observe the density-distance sketch and decide clustering center points through observation [14]. (3) The similarity mechanism adopted by the algorithm cannot deal with mixed data sets. For the first question, RLM algorithm proves that when the average density of data objects is 1%∼2% of the data set; we can get better clustering results. However, it doesn't give the method of determining the optimal d_c to achieve the best clustering effect; For the second question, the RLM algorithm calculates the γ values for all data points and sorts them from large to small, and determines the number of clusters by empirically setting the truncation thresholds for γ, but this method is somewhat opportunistic, and needs to set different truncation threshold for different data sets. Obviously it is not universal for all kinds of data sets.

We will design a mixed data similarity calculation based on three types' analysis. Based on the density cluster, we adopt the density-distance cluster theory which is presented in the SCIENCE journal. Improved mountain climbing algorithm and the iterative method are applied to find the optimal value of d_c. After defining the value of d_c, we adopt the method of normal curve fitting, set the confidence interval to filter the singular points out of the confidence interval, and it can automatically define the clustering center points fatly. Then we divide and cluster the other point by the way of being divided with the nearest high density point, finally we obtain the clustering result. Compared with the original algorithm, this new algorithm has the following three characteristics: (1) The new algorithm adds pre-processing and dominance analysis of the data set, and can deal with the mixed attribute data set. (2) The new algorithm realizes the parameter adaptation of d_c, and can calculate the optimal density radius automatically.(3) The new algorithm can automatically select the cluster center in the clustering process, and the whole clustering process does not need manual participation.

This article include four sections?:(1) Similarity metric for mixed data sets. (2) Fast determination of the clustering center points for clustering algorithm. (3) Finding the optimal value of d_c based on climb hill algorithm. (4) Simulation and analysis of classic clustering algorithms on different data sets for comparison of clustering performances.

2. Definitions of distance

Distance or similarity is the precondition of cluster analysis. Table 1 lists the various definitions of distance in several classic clustering algorithms.

K-means algorithm uses the popular Euclidean distance to deal with pure numerical attribute data, the K-modes algorithm uses a simple matching distance to deal with pure classification data. K-prototypes algorithm combined the two to deal with mixed attribute data. EKP, WFK-prototypes, and some other algorithms add fuzzy factor and weight coefficient to the original distance formula to improve their efficiency, since such distances may measure the similarity between objects more accurately [25,26].

However, these algorithms always use the same definition of distance for all kinds of data sets, without considering the differences between them. For example, some data sets are category dominated data sets, while some others are numerical dominated data. It is reasonable to choose different definitions of distance according to different situations.

Chen [14] proposed DC-MDACC algorithm in which the data set is divided into the numerical dominant type, the classified dominant type, and the mixed attribute type. According to different types of data sets, they use different definitions of distance to get the distance matrix. The main purpose of this algorithm is to pro-

Table 1
Various definitions of distance in some clustering algorithms.

Algorithm	Distance	Numerical distance	Classification distance
K-means [4]	$\mathrm{d}\left(X_i, X_j\right) = \sqrt{\sum_{p=1}^{m}\left(X_i^p - X_j^p\right)^2}$	$\mathrm{d}\left(X_i^p, X_j^p\right) = \left(X_i^p - X_j^p\right)^2$	None
K-modes [21]	$\mathrm{d}\left(X_i, X_j\right) = \sum_{p=1}^{m}\delta\left(X_i^p, X_j^p\right)$	None	$\delta\left(X_i^p - X_j^p\right) = \begin{cases} 0, X_i^p = X_j^p \\ 1, X_i^p \neq X_j^p \end{cases}$
K-prototypes [17]	$\mathrm{d}\left(X_i, Q_l\right) = \sum_{j=1}^{p}\left(X_{ij}^r - q_{ij}^r\right)^2 + \mu_l\sum_{j=p+1}^{m}\delta\left(X_{ij}^c, q_{ij}^c\right)$	$\mathrm{d}\left(X_i, Q_l\right) = \left(X_i^p - Q_j^p\right)^2$	$\mathrm{d}\left(X_i^p, Q_j^p\right) = \begin{cases} 0, X_i^p = Q_j^p \\ 1, X_i^p \neq Q_j^p \end{cases}$
EKP [19]	$\mathrm{d}\left(X_i, Q_l\right) = \sum_{j=1}^{p}\left(X_{ij}^r - q_{ij}^r\right)^2 + r\sum_{j=p+1}^{m}\delta\left(X_{ij}^c, q_{ij}^c\right)$	$\mathrm{d}\left(X_i, Q_l\right) = \left(X_i^p - Q_j^p\right)^2$	$\mathrm{d}\left(X_i^p, Q_j^p\right) = \begin{cases} 0, X_i^p = Q_j^p \\ 1, X_i^p \neq Q_j^p \end{cases}$
WFK-prototypes [20]	$\mathrm{d}\left(X_i, Q_l\right) = \sum_{l=1}^{p}\left(s_l\left(X_{ij}^r - q_{ij}^r\right)^2\right) + \sum_{l=p+1}^{m}\phi\left(X_{ij}^c, v_{ij}^c\right)^2$	$\mathrm{d}\left(X_i, Q_l\right) = s_l\left(X_i^p - Q_j^p\right)^2$	$\phi\left(X_i^p, Q_j^p\right)^2$

cess the mixed attribute data sets. For mixed attribute data sets, it uses conditional probability to calculate the influence of the other attributes to the focused one in the distance calculation and then add them together. In this algorithm, we can get a more accurate distance matrix, but it's quite time consuming, therefore, this algorithm is unable to handle large data sets efficiently.

条件概率

In our paper, dominant analysis of data sets is adopted first to analyze data. Aiming at processing large mixed attribute data sets, we incorporate them into numerical or categorical data sets by using attribute weights, then calculate the distance matrix by adopting the corresponding definition of distance. Compared to other clustering algorithms, our method can reduce the time complexity of computing the distance matrix of the mixed attribute data sets.

本文算法的优势

2.1. Dominant analysis of data and calculation of distance

文章可引用

For the data set D, it contains n samples and each sample has d attributes: p numerical attributes and q classification attributes, satisfying $d = p + q$. Then, the dominant analysis of the data set D based on the size of q and p is as following:

(1) If $p > q$, the data set is a numerical dominant data set.
(2) If $p < q$, the data set is a classified dominant data set.
(3) If $p = q$, the data set is a balanced attribute data set.

For the different data sets, we adopt different definitions of distance to reduce the effect of non-dominant attributes, and thus may improve the clustering accuracy.

For a data set that contains n data, denoted by $D = \left\{A_1, A_2, \ldots, A_n\right\}$, the sample A_i has d attributes denoted by $\left\{A_i^1, A_i^2, \ldots, A_i^d\right\}$. Using $d\left(A_i, A_j\right)_n$ said the numerical attributes from the part and using $d\left(A_i, A_j\right)_c$ said the classification attribute part of the distance.

(i) If the data set D is the numerical attribute dominant data set, for any two objects, A_i and A_j:

$$d(A_i, A_j)_n = \sqrt{\sum_{k=1}^{p}(A_i^k - A_j^k)^2} \quad \text{前p维用欧几里得距离计算} \tag{1}$$

For categorical attribute data:

$$d(A_i^k, A_j^k) = \begin{cases} 0, & (A_i^k = A_j^k) \\ 1, & (A_i^k \neq A_j^k) \end{cases} \tag{2}$$

Then, the total distance of categorical attributes is:

$$d(A_i, A_j)_c = \sum_{k=1}^{q} d(A_i^k, A_j^k) \quad \text{每一维上的距离之和} \tag{3}$$

(ii) If the data set D is the classification attribute dominant data set, for the k dimensional numerical attributes: 混合属性

$$d(A_i^k, A_j^k) = \left|\frac{A_i^k - A_j^k}{A_{max}^k - A_{min}^k}\right| \tag{4}$$

Where A_{max}^k and A_{min}^k represent the maximum and minimum values, respectively, of the k dimensional sample data.

The total distance of numeric attributes is:

$$d(A_i, A_j)_n = \sum_{k=1}^{p} d(A_i^k, A_j^k) \tag{5}$$

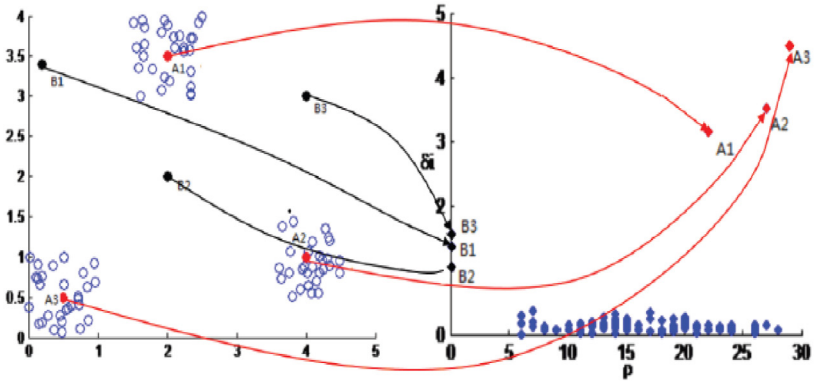
Its distance definition of categorical attributes is the same as numerical attribute dominant data set. (2), (3)

(iii) If the data set D is the balanced attribute data set, we need to first determine whether the data set is numerical attribute dominant data set or classification attribute dominant data set, then calculate the distance according to the distance calculation formula.

For the numerical attribute dominant data sets, this method can effectively reduce the impact of classification attribute distance on the overall similarity calculation when there are a large number of numerical attributes. And for the classification attribute dominant data sets, the contribution of all numerical attributes on the distance is limited to the interval [0,1], and thus diminishes the effect of large numerical attribute value on the overall distance.

2.2. Case verification

For a data set containing n data, d dimensional attributes, the time complexity to calculate the distance matrix is $O(n \times (n-1) \times d)$. As for DC-MDACC algorithm, the time



(a) DataSet1's point distribution. (b)Decision graph for the data in(a).

Fig. 1. The algorithm in two dimensions about DataSet1.

complexity of computing the distance matrix of numerical or categorical data set is alsoO(n × (n − 1) × d). But for mixed attribute data set, the time complexity of computing the distance matrix is much higher, equal to O $\left(n^2 \times (n-1) \times d^3\right)$.

Now taking *Heart* data set [23] as an example, this data set contains 8 dimensional categorical attributes and 5 dimensional numerical attributes. The algorithm in this paper treats it as a classified dominant data set. But the DC-MDACC algorithm considers it as a mixed attribute data set, and utilizes conditional probability to calculate the distance matrix.

We calculate two distance matrices by these two methods, respectively. Utilizing the distance matrix obtained by CH-CCFDAC algorithm as input data, we find that the clustering accuracy is 0.842 and the cluster purity is 0.84. While the distance matrix is obtained by DC-MDACC algorithm, the clustering accuracy and the cluster purity are 0.848 and 0.833, respectively. It can be seen that the two methods with different distance matrices have quite similar clustering effect. However, by comparison, our algorithm has much lower time complexity, and thus is more suitable when dealing with large data sets.

3. Cluster center fast determination algorithm (CCFD)

3.1. Main idea of the algorithm

The fast clustering algorithm of RLM is based on the distribution of density and distance of the data object, and determines the cluster center quickly by constructing the normal distribution function. There are two hypotheses in clustering method proposed by RLM algorithm:

Definition 1. For any data point i, the local density is calculated by

$$\rho_i = \sum f\left(d_{ij} - d_c\right) \tag{6}$$

$$f(x) = \begin{cases} 1x = d_{ij} - d_c < 0 \\ 0x = d_{ij} - d_c > 0 \end{cases} \tag{7}$$

Where d_{ij} represents the distance between points i and j, and d_c represents the density radius parameter.

Our algorithm is based on assumptions that the cluster centers are surrounded by neighbors with lower density while at relatively larger distance from objects with higher density [14]. Noise objects have comparatively larger distance and smaller density. Therefore for a given data object i, only density and distance need to be calculated. The density ρ_i can be calculated according to the formula (6)–(7). δ_i is calculated by minimum distance between data object i and any other objects with higher density in Definition 2.

Definition 2. For any point, the distance between this object to another object whose local density is larger than this object is:

$$\delta_i = min(d_{ij}) \quad (\rho_j \geq \rho_i) \quad ? \tag{8}$$

For the point with highest density, we conventionally take:

传统的

$$\delta_i = \max(\delta_j) \quad (i \neq j) \quad ? \tag{9}$$

Note that δ_i is much larger than the typical nearest neighbor distance only for objects that are local or global maxima in density. Thus cluster centers are recognized as objects of larger δ_i. This core observation could be illustrated by the simple example in Fig. 1. The data original distribution is shown in Fig. 3(a), while relationship of density ρ and distance δ distribution is shown in Fig. 3(b). Specific corresponding relationships are illuminated by lines.

As we can see, A1, A2, A3 are the clustering centers in Fig. 1(a), These points have high ρ and high δ in decision graph; B1, B2, B3 are the noise points in Fig. 1(a), and they have high δ but low ρ in Fig. 1(b); as for other points, we call them boundary points, they belong to a cluster and have low δ.

We then define a new variable γ for each point i as following:

$$\gamma_i = \rho_i \times \delta_i \tag{10}$$

According to the probability distribution graph of γ, it is found that the shape of the curve is similar to that of a normal distribution curve. Now we have given a confidence interval, and use this confidence interval to find singular points. After this, we set the parameter k, and remove the points of larger relative deviation of ρ and δ from the singular point set; the remaining k singular points are clustering centers.

According to the above ideas, the flow chart of CCFD algorithm is as shown in Fig. 2, and the steps of CCFD algorithm are in the following.

Step1: Given the parameter d_c, use formula (6)–(10) to calculate ρ, δ and γ for each data point.

Step2: Set ρ-δ decision graph and density distribution map of γ.

Step3: Calculate the mean value and the variance of γ, and make use of the normal distribution curve of γ through these two values.

Step4: Given a confidence interval, and use this confidence interval to find singular points.

Step5: Set parameter k, remove the points of larger relative deviation of ρ and δ from the singular point set, and set the remaining k singular points as cluster centers.

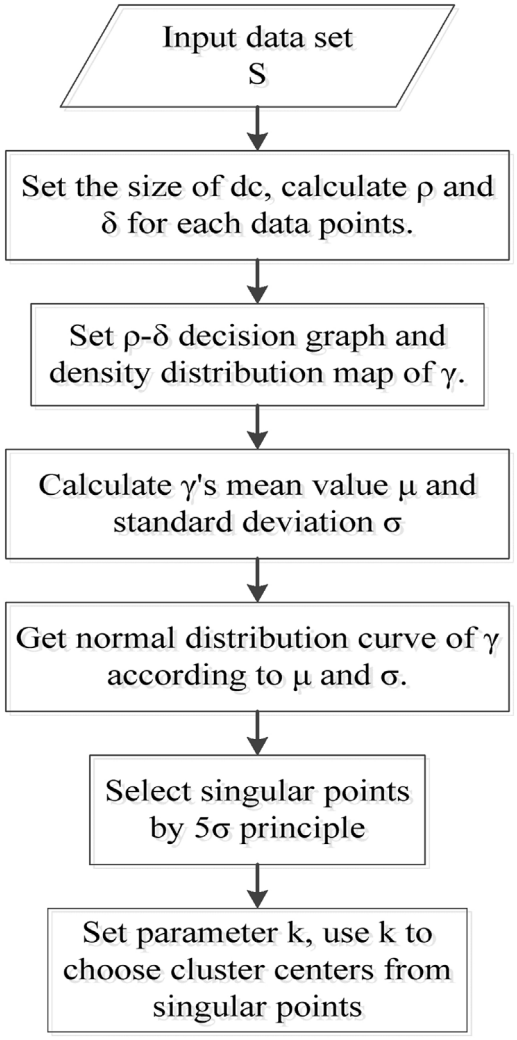


Fig. 2. Flow chart of CCFD algorithm.

3.2. Fast determination of cluster centers based on normal distribution analysis

Definition 3. For a random variable X following a normal distribution, if its mathematical expectation and variance are μ and σ^2, respectively, we write it as $X \sim N(\mu,\sigma^2)$.

The probability that the variable X falls into the interval(μ-5σ,μ+5σ)is about 99.99999999% $\approx$ 1. When the sample size is not sufficiently large, we can say that almost all the sampled values of X are contained in(μ-5σ, μ+5σ), the interval(μ-5σ, μ+5σ)is thus called confidence interval. The points that don't fall into the confidence interval are singular points.

3.2.1. The way to get sample mean $\bar{x}$ and sample variance S

Suppose, for each data point i, γ_i follows a normal distribution $\gamma_i \sim N\left(\mu, \sigma^2\right)$. Denoting $\bar{x}$ the mean value of these data points and S the variance, then we can make the following estimation:

$$\mu = \bar{x}, \quad \sigma = \sqrt{\frac{N-1}{N}}s \tag{11}$$

Moreover, we have that γ_i is not negative for any data point, as a result, their distribution is not strictly normal distribution due to the missing data in negative. Thus, we make the following adjustments for the calculation of μ and σ.

Firstly, calculate the mean value for all the data points, denoted by $\overline{x_1}$. Then, set the threshold $n\overline{x_1}$. For any data point i, if the value of γ_i is greater than the threshold, we remove the data point. Finally, we calculate the mean value $\overline{x_2}$ and variance S for the remaining data points, and use formula (11) to calculate μ and σ. The specific steps are as follows:

Step1: calculate the mean value $\overline{x_1}$ for all the data points.

Step2: Setting the parameter n, carrying out the following test:

```
For(i=1 ; i≤N ; i++){
    if(γi>nx̄1)
        εi = 0;        // If εi=0, data point i was excluded.
    else
        εi = 1;
}
```

Step3: Selecting the points with $\varepsilon_i = 1$, then calculate their mean value $\overline{x_2}$ and variance S.

Step4: Using formula (11) to calculate μ and σ.

In order to solve the problem of missing data points on the $\gamma < 0$ side, this method sets the parameter n to remove some points with large γ. According to the principle of symmetry, we set n=2 in this case. In order words, only the points in $[0,2\overline{x_1}]$ will be used to calculate μ and σ. And this method effectively improves the accuracy to estimate μ and σ.

3.2.2. Cluster center fast determination

After calculating μ and σ, we can get a normal distribution curve. Now according to the 5σ principle, we find singulars points as follows:

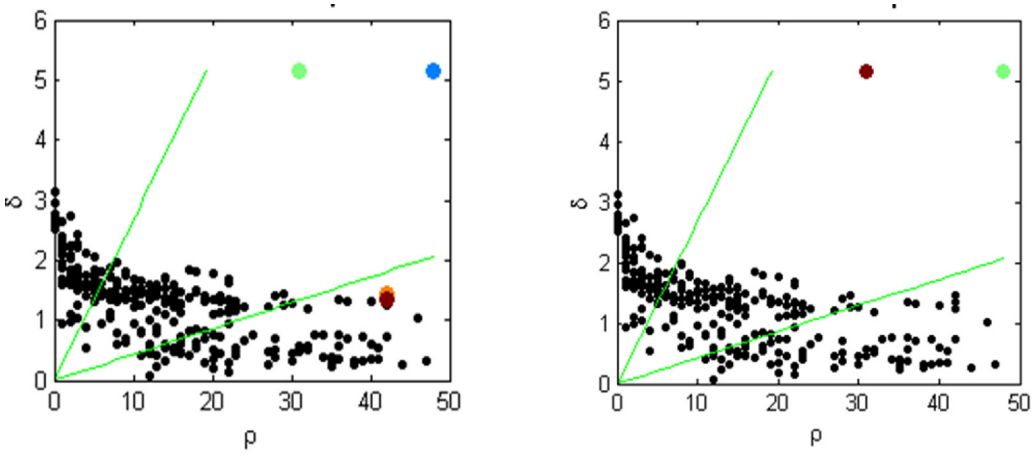
Set boundary value Wide=μ+5σ, then, for each data point i, if $\gamma_{i>}$ Wide, point i is considered as a singular point.

Obviously, the data points marked as singular points have greater value than normal data points. After several experiments, it is found that the number of singular points selected by this method is always larger than that of real clusters in data set. Further analysis of these singular points shows that some singular points are distributed on both sides of the ρ-δ graph, with large relative deviation of ρ and δ. According to the physical meaning of the ρ-δ graphs, it is easy to know that these points are of high density but not far enough away from other high density points (such as a relatively dense portion of a cluster) or have low density but are far from dense cluster (outliers). For these points, although they have large γ values, they are not suitable to be clustering centers. We call these points pseudo-clustering centers. We can find a true clustering center and eventually complete a high-accuracy clustering, only when we remove these points from the singular points collection.

In this paper, the screening method of singular points is:

Normalize singular points' ρ and δ to get ρ^* and δ^*, and set parameter k. For singular point i, if $\rho_i^*/\delta_i^* < k$ and $\delta_i^*/\rho_i^* < k$, it is selected as one of cluster centers. 被选为聚类中心的条件

Take the data set Heart as an example, and set k = 3. After finding the singular points by the normal distribution curve fitting method, the positions of the singular points on ρ-δ plot are shown in Fig. 3(a). It can be seen that there are four singular points, two of which fall below $\rho_i^*/\delta_i^* = 3$, indicating that the other two singular points only represent the density concentration region in a cluster, not true clustering centers. If all the four singular points are set to clustering centers, the clustering accuracy is only 62.3%; the color dots in Fig. 3(b) represent the singular points after screening, in other words, the true clustering center points. If only these two points are set to be clustering centers, the clustering accuracy is 84.2%. Comparing the accuracy of two clustering results, it can be seen that it is effective and necessary to perform the screening operation on singular points.



(a) Singular points on ρ-δ plot **(b) Clustering centers on ρ-δ plot**

Fig. 3. Use the constraints $\rho_i^*/\delta_i^* < k$ and $\delta_i^*/\rho_i^* < k$ to filter the singular points.

3.2.3. Case verification

Taking DataSet1 as an example, Fig. 4 shows the main process of using the CCFD algorithm to determine the cluster center automatically. Firstly, determine the density radius d_c, calculate ρ and δ at this density radius, and calculate γ for all data points by the formula (10). According to the γ values of all data points, the density distribution of DataSet1 is estimated (Fig. 4(a)); Then the normal distribution curve to fit the γ density distribution is obtained, and the confidence interval is determined according to the normal distribution curve (Fig. 4(b)); Screen out those points (A1, A2, A3) which fall outside the confidence interval from density distribution map and take them as singular points (Fig. 4(c)); Finally make two straight lines with slope of k and $1/k$ in the decision graph, select singular points between the two lines as the real cluster centers (Fig. 4(d)). We let k=3 here, because through the analysis of some experimental data sets, we find when k=3, we can choose the true cluster centers more accurately.

4. The selection of optimal d_c

4.1. Method of determining d_c

In the above, we can see that when the parameter d_c is determined, cluster centers can be extracted effectively by CCFD algorithm. Therefore, the selection of optimal d_c is very important. In order to select appropriate d_c, we first obtain the maximum distance d_{max} and the minimum distance d_{min} from all data points. Introducing a parameter percent, and set

$$dc = d_{min} + (d_{max} - d_{min}) \times percent/100 \quad (12)$$

In fact, the algorithm is a density-based clustering algorithm. In the clustering process, it needs to set the density radius d_c to calculate the density value for each data point. The change in d_c will directly change the density of the data points, resulting in density fluctuations of all data points, thus affecting the final clustering results. In order to study the effect of density fluctuation on clustering results, it is necessary to study the relationship between d_c and accuracy.

Taking the Statlog Heart data set as an example, the clustering accuracy as a function of density radius d_c is shown in Fig. 5.

In Fig. 5, 20% of the maximum distance between two objects is the boundary of d_c, based on which the values of d_c are divided into two parts. When d_c is smaller than the boundary value, the density fluctuation caused by the change of d_c has little effect on the accuracy rate, and the accuracy rate is relatively high, containing the global optimal value of the accuracy rate; When d_c is larger than the boundary value, the accuracy rate is affected by the density fluctuation, and the overall accuracy rate is relatively low, and thus the clustering is not satisfied.

For other datasets, the same method was used for analysis and it is found that the analysis results were similar to the Heart dataset and that the density fluctuation boundaries of these datasets were also centered on 20% of the maximum distance between two objects. In addition, when the d_c of these data sets is in the vicinity of the density fluctuation boundary, the average density of the data object is 2% ~ 3% of the data set, which is consistent with the range of d_c given by the RLM algorithm.

According to the above analysis, in the adaptive process of density radius, we can narrow the range of values of d_c to less than 20% of the maximum distance between pairwise objects. In this range, although the accuracy rate is not a single-peak function of d_c, but the density fluctuation caused by the change of d_c in the range of the segment is relatively small, and the local optimal accuracy rate is not much different from the global optimal accuracy rate. So we can use the improved climbing algorithm to find the optimal solution within this d_c range.

After the selection of d_c, clustering centers are selected by the CCFD algorithm automatically. Then we can classify all data points by the method that a point belongs to another point whose density is larger and it is also the nearest point for target object.

4.2. Fitness function design

When d_c is selected and the clustering results are obtained, in order to compare the effect of clustering for different d_c, a Fitness function is designed as an evaluation index.

Fitness function is composed of two parts:

$$Fitness1 = \frac{\sum_{j=1}^{m}[\sum_{x_i \in C_j} d(x_i, C_j)/|C_j|]}{m} \quad (13)$$

$$Fitness2 = \frac{\sum_{j=1}^{m}[\sum_{i=1, i\neq j}^{m} d(C_i, C_j)/(m-1)]}{m} \quad (14)$$

where m is the number of clusters, C_i and C_j denote the cluster centers of i and j, respectively, and $|C_j|$ represents the number of data objects in cluster j.

From formula (13) and (14), we can find that Fitness1 represents the global average intra cluster distance and Fitness2 represents the global average inters cluster distance. Based on the most essential

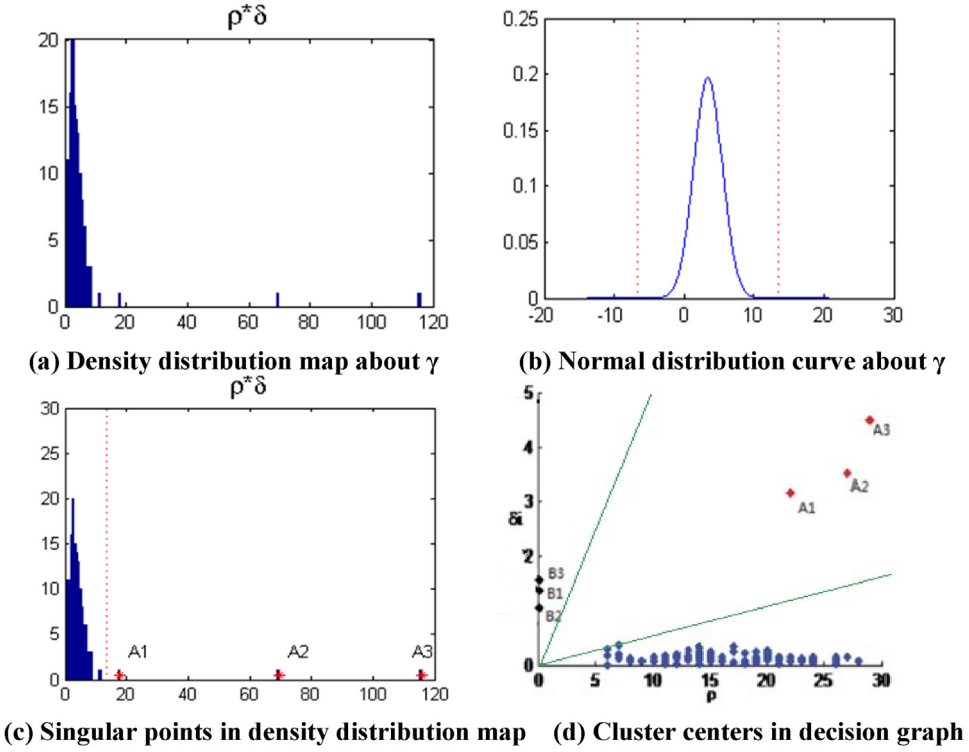


Fig. 4. Graphs about determine cluster centers.

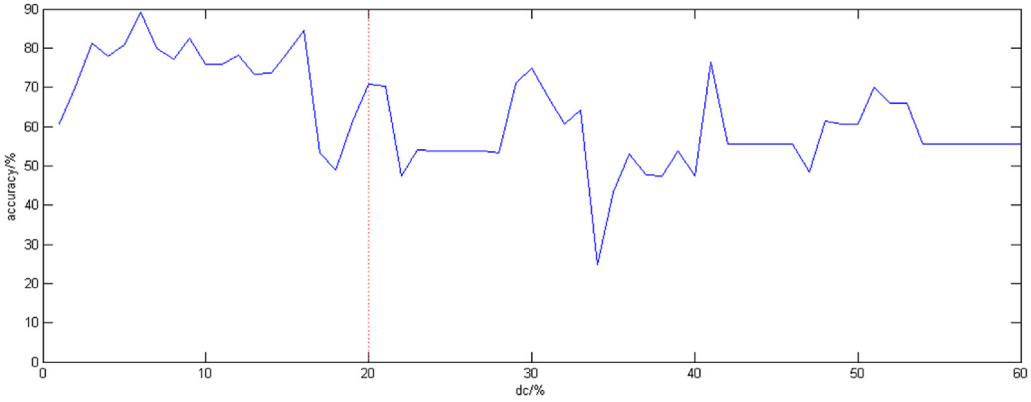


Fig. 5. The accuracy of d_c changes about Statlog Heart.

definition of clustering effect——we need smaller distance within the cluster, but larger distance between clusters, thus we define

$$\text{Fitness} = \frac{\text{Fitness2}}{\text{Fitness1}} \tag{15}$$

For a given d_c, the greater the value of the Fitness function, the better the clustering effect is.

4.3. The selection of optimal d_c

Now we can determine $d_c$by setting the percent parameter, with the range of percent value being 1%–20%. Then, we use the improved mountain climbing algorithm to find the best d_c.

The improved climbing algorithm uses 10% of the maximum distance between pairwise objects as a starting point and extends to the two sides with a certain iteration radius. The clustering accuracy is determined according to the *Fitness* function value in the stretching process; the iteration radius is gradually reduced until the iteration radius reaches 0, then the best d_c can be found.

The flow chart of the algorithm is shown in Fig. 6.

The specific steps are:

Step1: Inputting data set's distance matrix.

Step2: Set P_o as the initial value of percent, set the iteration radius r, calculate the initial value of d_c.

Step3: Calculate the Fitness values when*percent*=P_o, *percent*=P_o+*r* and *percent*=P_o-*r*.

Step4: Select the percent which has maximum Fitness value as the temporary optimal percent (P_{tbest}), set *r*=*r*-0.5.

Step5: Determine whether r = 0. If $r \neq 0$, jump to Step2, and use P_{tbest} to replace P_o; If *r* = 0, stop iteration, set the global optimal percent $P_{best} = P_{tbest}$, then the best d_c is found.

The number of iterations is ξ:

$$\xi = \frac{r}{0.5} \times 2 + 1 \tag{16}$$

loop count is ζ:

$$\zeta = \frac{r}{0.5} + 1 \tag{17}$$

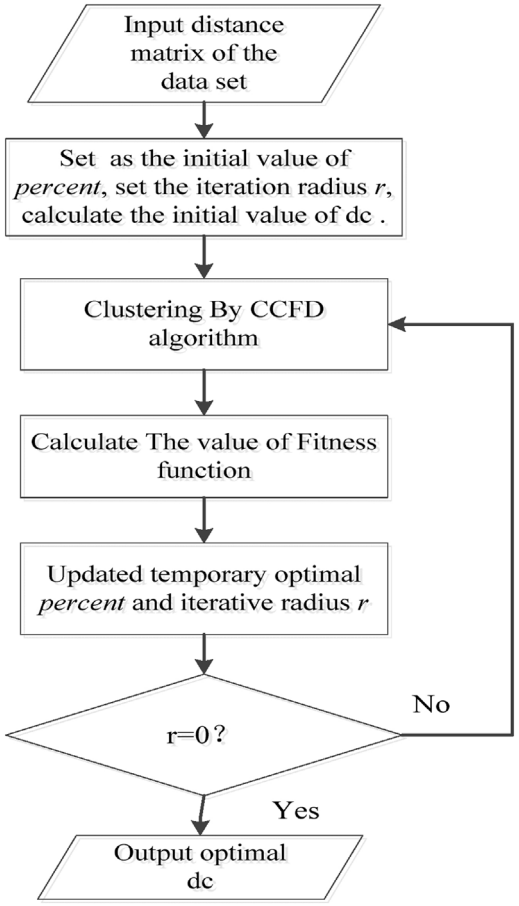


Fig. 6. Algorithm flow chart about to find the optimal d_c.

According to the scope of *percent*, in the process, to ensure that the iterative range can traverse the entire interval and the optimal d_c is much accurate, we set $P_o = 10$, r=3. In this case, ξ=13, ζ=7.

Using the improved climbing algorithm, we can avoid that d_c in the adaptive process falls into the local optimal, and this method can find out the optimal solution or approximate optimal solution by few iterations.

5. CH-CCFDAC algorithm

5.1. Main ideas of CH-CCFDAC algorithm

In Section 2 to Section 4 of this paper, we discuss the distance calculation method for different types of data, and the cluster center fast determination Algorithm(CCFD) under given density radius and the method of density radius adaptation. CH-CCFDAC algorithm is a clustering algorithm for mixed attributes data that concatenates these three pieces of content. When the CH-CCFDAC algorithm is used to process a data set, it is necessary to analyze the data set first. According to the result of the dominance analysis, the distance matrix of the data set is calculated according to the corresponding definition of distance. Then the algorithm uses the distance matrix as input, set the initial value of the d_c and find cluster centers automatically by CCFD algorithm. After the clustering is completed, the *Fitness* value under certain density radius is calculated to measure the effect of clustering. In order to find the optimal density radius, CH-CCFDAC uses the improved climbing algorithm to calculate the *Fitness* value under different density radius. Finally, the density radius of the maximum *Fitness* value is chosen as the optimal d_c.

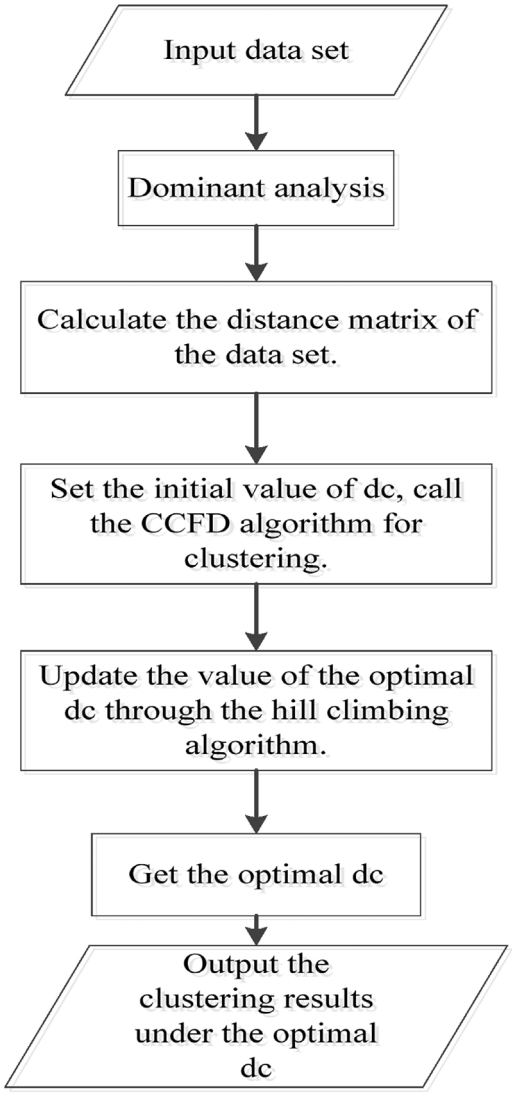


Fig. 7. Flow chart of CH-CCFDAC algorithm.

CH-CCFDAC takes the clustering result under the optimal d_c as the output and completes the whole clustering process .

5.2. Flowing chart and step description of CH-CCFDAC

Algorithm flow chart is shown in Fig. 7; the specific steps are as follows:

Step1: Do dominant analysis for the input data sets, and according to the result of the dominant analysis, calculate the distance matrix of the data set by using the corresponding distance calculation formula.

Step2: Set the initial value of d_c, and call the CCFD algorithm for the first fast clustering.

Step3: Using mountain climbing algorithm for d_c's iteration, through the formula (13)–(15) to calculate the corresponding Fitness value of each d_c, and update the optimal d_c by comparing the value of the *Fitness*.

Step4: Obtain optimal d_c, and output clustering results in optimal d_c.

Table 2
Ten data sets' information.

Name	Dimension number	Data set's classification	Amount of data
Aggregation	2	7	788
Spiral	2	3	312
Jain	2	2	373
Flame	2	2	240
Iris	4	3	150
Soybean	35	4	47
Zoo	15	7	101
Acute	7	2	120
Heart	13	2	270
KDD-99	41	5	1000

6. Experimental test and performance analysis

The operating system in the experiment is Windows 7. Integrated development environment is Matlab2013a. CPU is Radeon R5; Memory is 4GB.

In order to verify the effect of this algorithm, we select some data sets to test. The following table lists specific information about these data sets (Table 2).

6.1. Clustering result evaluation

In this paper, the clustering accuracy rate proposed by Gan [29] is used as the evaluation criteria.

(1) Clustering accuracy

The definition of clustering accuracy is:

$$\omega = \frac{\sum_{i=1}^{m} \alpha_i}{N} \tag{18}$$

where α_i represents the number of samples which to be correctly classified; m represents the number of clusters; N represents the number of samples in the data set. The clustering is perfect if $\omega = 1$.

(2) Average cluster purity:

$$\mathrm{Pur} = \sum_{i=1}^{m} \frac{|C_i^d|}{|C_i|} / m \tag{19}$$

where m represents the number of clusters;$|C_i^d|$ represents the number of data points of most important class label in cluster i. $|C_i|$ represents the number of data points included in cluster i. The range of Pur is [0,1], the higher the purity value is, the better the clustering quality is.

Clustering accuracy reflects the clustering effect of the data, while cluster purity can reflect the quality of cluster. They complement each other.

6.2. Experimental result analysis

6.2.1. Experimental data sets

The data sets used in this paper include Aggregation, Spiral, Jain, Flame, Iris, Soybean, Zoo, Inflammatins Acute, Heart Statlog and KDD-CUP 99. These data sets all come from the UCI database.

Aggregation, Spiral, Jain and Flame are two-dimensional numerical data, Fig. 8 shows the two-dimensional scatter plots of these data sets.

It can be seen from Fig. 8 that this algorithm can find the clusters of arbitrary shape accurately and get great clustering results. The disadvantages of traditional clustering algorithms can be overcomed. And the superiority of density based clustering is reflected.

6.2.2. Experimental results and performance analysis

According to the hill climbing algorithm, we use Fitness function as the index to select the optimal d_c. The relationship between iter(ζ in the formula (17)) and Fitness is shown in Fig. 9. It can be seen from the figure that the value of the Fitness function is gradually increased in the iterative process, indicating that the algorithm will push the d_c step by step in the iterative process.

According to CH-CCFDAC algorithm, we can find the cluster centers and complete the clustering process in the optimal d_c. Figs. 10–14 shows the overall process of clustering, and the color points in Fig.14 are the final cluster centers.

Compare this algorithm with IWKM [15], SBAC [16], K-prototypes [17], KL-FCM-GM [18], EKP [19], WFK-prototypes [20], IPC [23], RLM [14] and DC-MDACC [14], the result is shown in the table.

For the above six data sets, cluster purity of CH-CCFDAC are 0.98, 0.82, 0.93, 1.00↘0.88 and 1.00, respectively.

Table 3 summarizes the clustering accuracy of the nine algorithms for six real data sets. As we can see, our proposed algorithm (CH-CCFDAC), IPC, RLM, WFK-prototypes and DC-MDACC do better than other algorithms in these datasets. So we should pay attention to the performance of these five algorithms.

Acute Inflammatins data was created by a medical expert as a data set to test the expert system, which will perform the presumptive diagnosis of two diseases of urinary system. It belongs to classified dominant data set with one numeric and six categorical attributes. It consists of 120 instances belonging to two classes. Table 3 summarizes the clustering accuracy of the nine algorithms. From this table we can see that our proposed algorithm(CH-CCFDAC), IPC, RLM and DC-MDACC do better than other algorithms. The accuracies of these algorithms are higher than 80%, while our proposed algorithm's accuracy is even 3.8% and 4.0% higher than those of the IPC and DC-MDACC algorithms, respectively.

The Statlog Heart dataset is a mixed dataset. It contains 270 patient instances described by five numeric attributes and eight categorical attributes. We list the clustering accuracies of the nine algorithms in Table 3. The proposed algorithm(CH-CCFDAC), IPC, RLM, WFK-prototypes and DC-MDACC give accuracy of 0.842, 0.822, 0.842, 0.835 and 0.848, respectively. DC-MDACC has the highest accuracy. But combined with the analysis of section 2.2, we have known that, while it only has 0.5% accurate improvement compared with our method, this algorithm has much higher time complexity. Therefore, when dealing with huge data sets, our method is the more efficient one.

Iris dataset is a purely numeric dataset, and it consists of 150 data objects described by 4 numeric attributes, and has three classes. On this dataset we compare the performance of our algorithm with that of the IPC algorithm, RLM algorithm, and DC-MDACC algorithm on clustering numeric data. The proposed algorithm, IPC, RLM and DC-MDACC give clustering accuracy of 0.968, 0.903, 0.968 and 0.960, respectively. Clustering results show that our algorithm is as high as the accuracy of the RLM algorithm and 6.5% and 0.8% more accurate than IPC and DC-MDACC, respectively. Therefore our algorithm is suitable for dealing with numeric dataset.

The Soybean dataset is a purely categorical dataset. It consists of 47 data objects described by 35 categorical attributes, and has four classes. On this dataset we compare the performance of our algorithm with that of the DC-MDACC, IPC, and KL-FCM-GM on clustering categorical data. The proposed algorithm, DC-MDACC, and IPC give clustering accuracy of 1.000, 0.957, and 0.903 respectively; KL-FCM-GM gives the best clustering accuracy of 0.903 when its fuzziness coefficient α equals 1.8. Clustering results show that our algorithm is 4.3%, 9.7%, and 9.7% more accurate

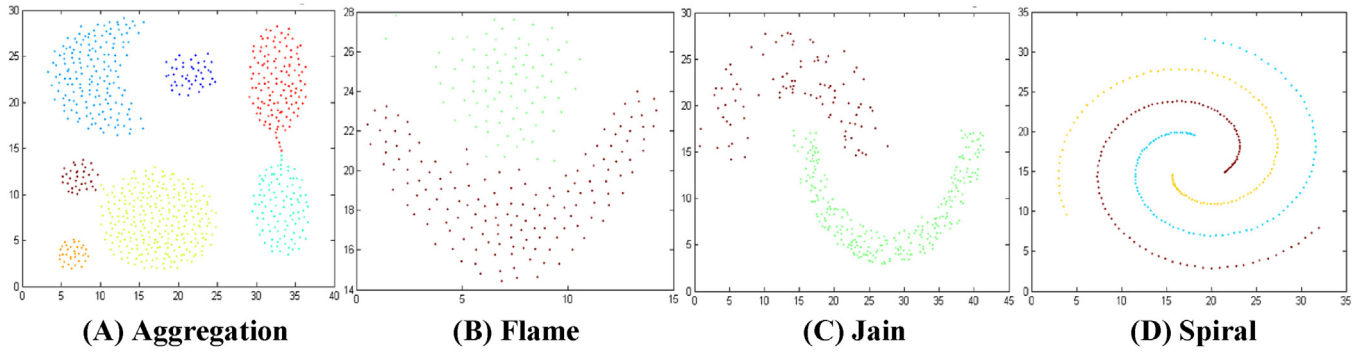


Fig. 8. 2D scatter plots and their classification.

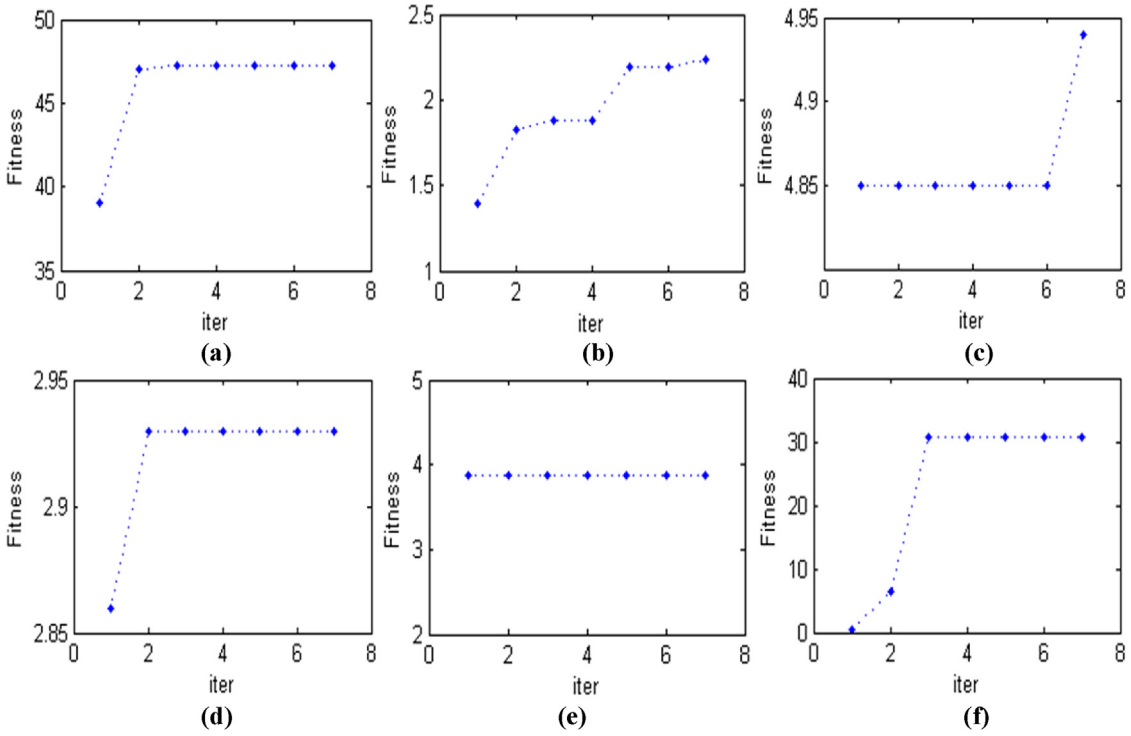


Fig. 9. Relationship of Fitness value and iter:(a)Acute (b)Heart (c)Iris (d)Soybean (e)Zoo (f)KDD-CUP 99.

Table 3
Comparison of the accuracy of various algorithms for testing different data sets.

Algorithm/Data sets	Acute	Heart	Iris	Soybean	Zoo	KDD-CUP 99
K-prototypes	0.61	0.577	0.819	0.856	0.806	0.539
SBAC	0.508	0.752	0.426	0.617	0.426	0.672
KL-FCM-GM	0.682	0.758	0.335	0.903	0.864	0.779
EKP	0.508	0.545	0.762	0.786	0.629	0.501
IWKM	0.543	-	0.822	0.908	0.703	-
WFK-prototypes	0.710	0.835	0.880	0.891	0.908	0.701
IPC	0.883	0.822	0.880	0.903	0.853	-
RLM	0.921	0.842	0.968	1.000	0.921	0.992
DC-MDACC	0.917	0.848	0.960	0.957	0.892	0.916
CH-CCFDAC	0.921	0.842	0.968	1.000	0.921	0.992

than DC-MDACC, IPC, and KL-FCM-GM, respectively. Our algorithm therefore is suitable to deal with categorical dataset.

Zoo data set consists of 101 data objects, each of which is described by one numeric attributes and 16 categorical attributes. The last categorical attribute is the classes attribute which has 7 values. We list the clustering accuracy of our proposed algorithm, WFK-prototypes, KL-FCM-GM, IPC, and DC-MDACC algorithm. The proposed algorithm, IPC, and DC-MDACC give clustering accuracy of 0.921, 0.853 and 0.892, respectively; WFK-prototypes gives the best clustering accuracy of 0.908 when its fuzziness coefficient α equals 2.1, KL-FCM-GM gives the best clustering accuracy of 0.864 when its fuzziness coefficient α equals 1.3, respectively.

KDD Cup 1999 Data is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition,

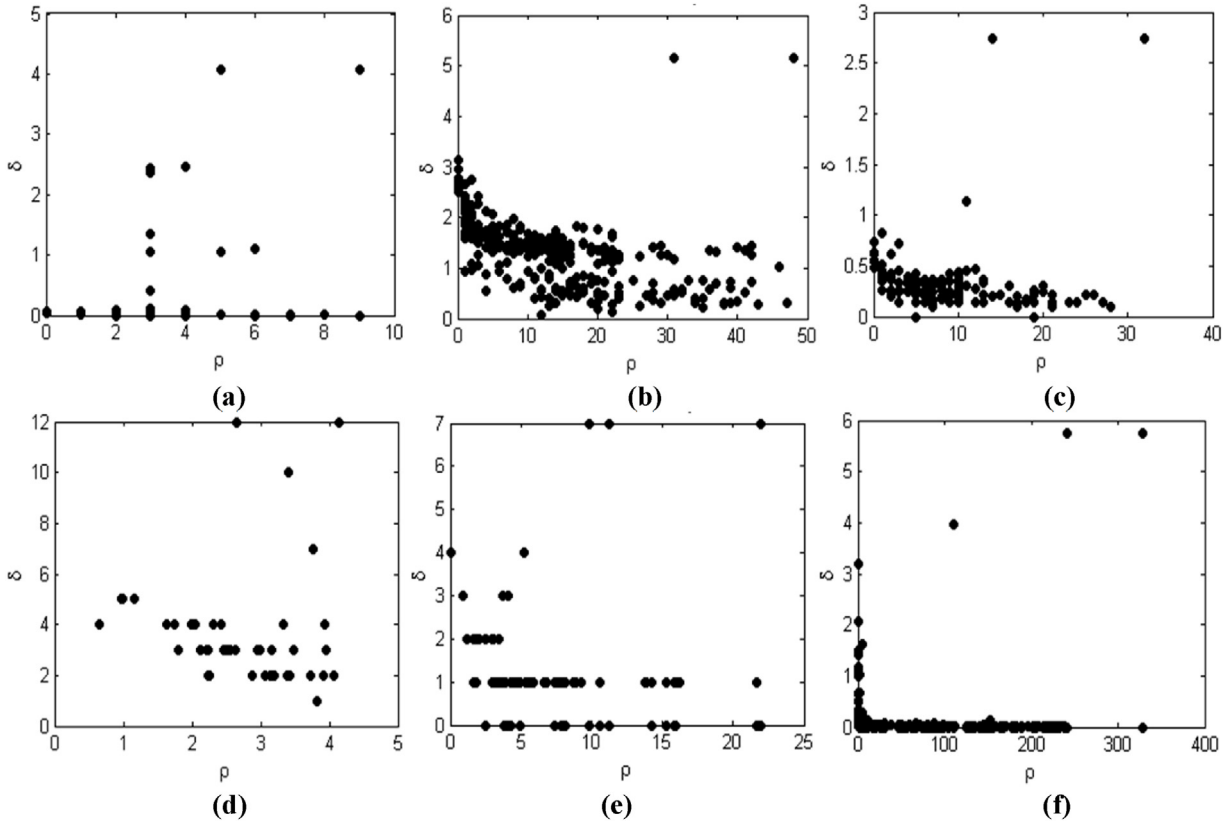


Fig. 10. Six data sets' decision graphs in optimal d_c: (a)Acute (b)Heart (c)Iris (d)Soybean (e)Zoo (f)KDD-CUP 99.

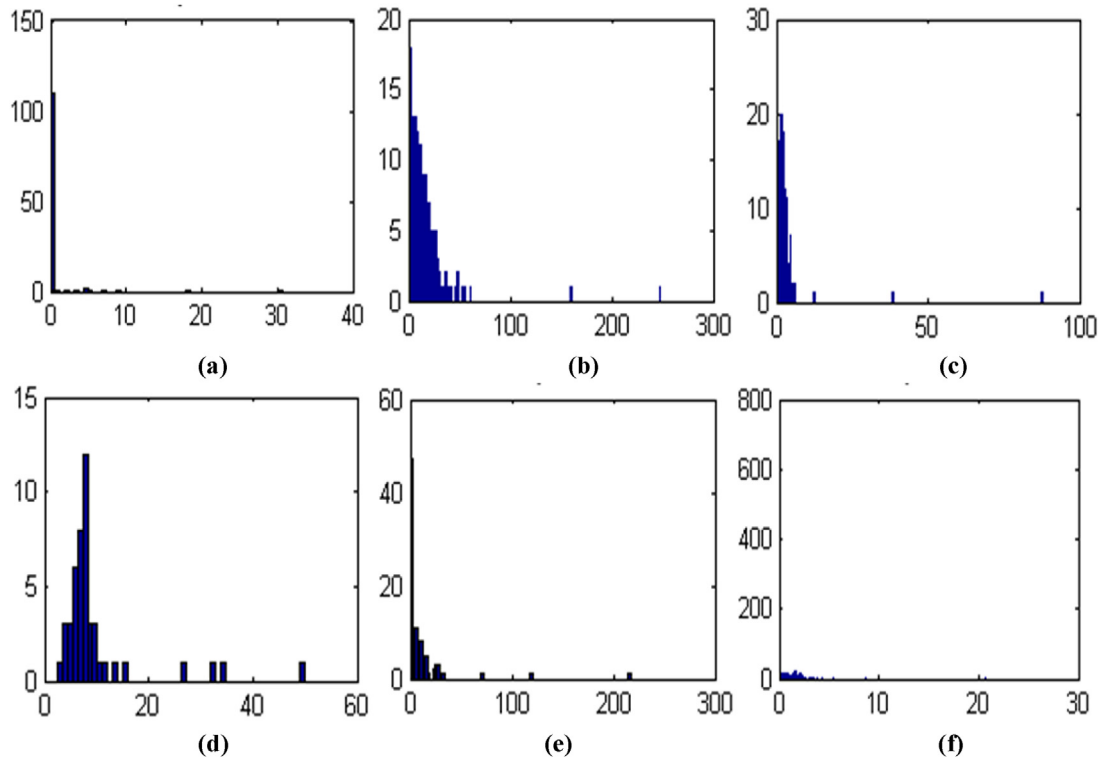


Fig. 11. The γ's density distribution of six data sets in optimal d_c: (a)Acute (b)Heart (c)Iris (d)Soybean (e)Zoo (f)KDD-CUP 99.

which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. Each record of the data set contains 41-dimensional attributes, including 34-dimensional numerical attributes and 7-dimensional classification attributes, a total of five categories and 24 sub-categories, the experiment from the overall data set randomly selected 1000

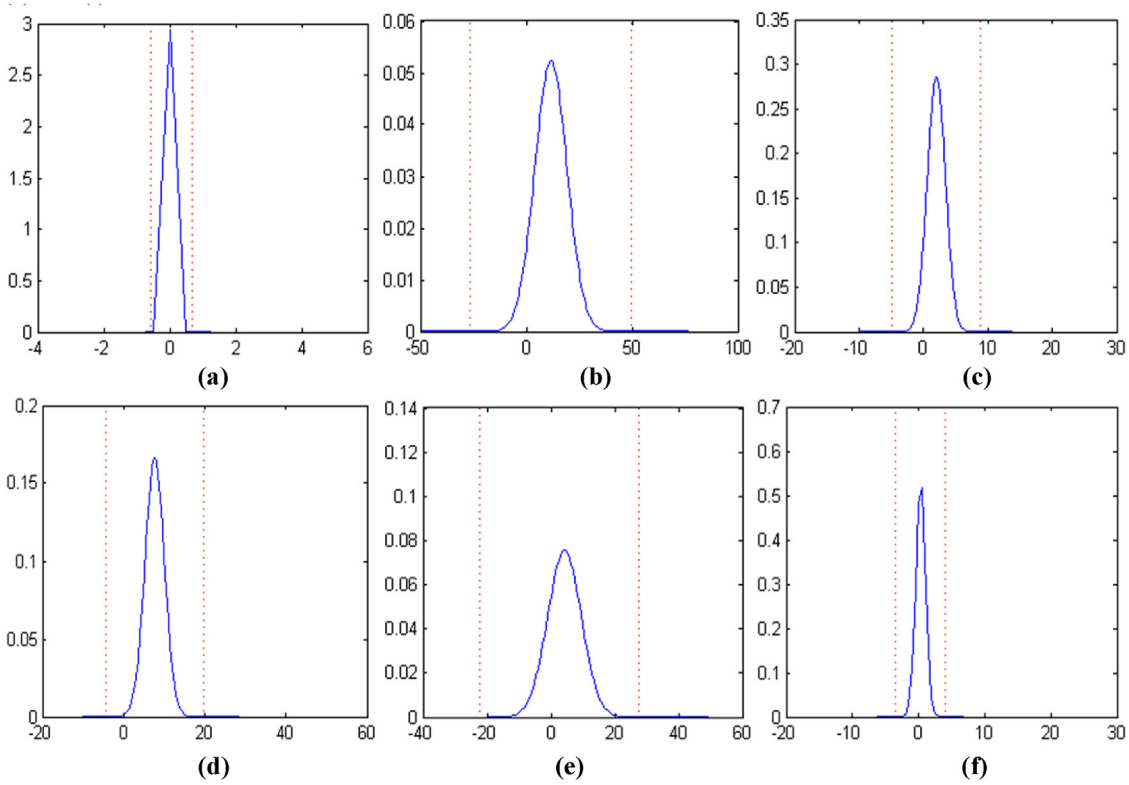


Fig. 12. The normal distribution curves fitted according to γ's density distribution of six data sets: (a)Acute (b)Heart (c)Iris (d)Soybean (e)Zoo (f)KDD-CUP 99.

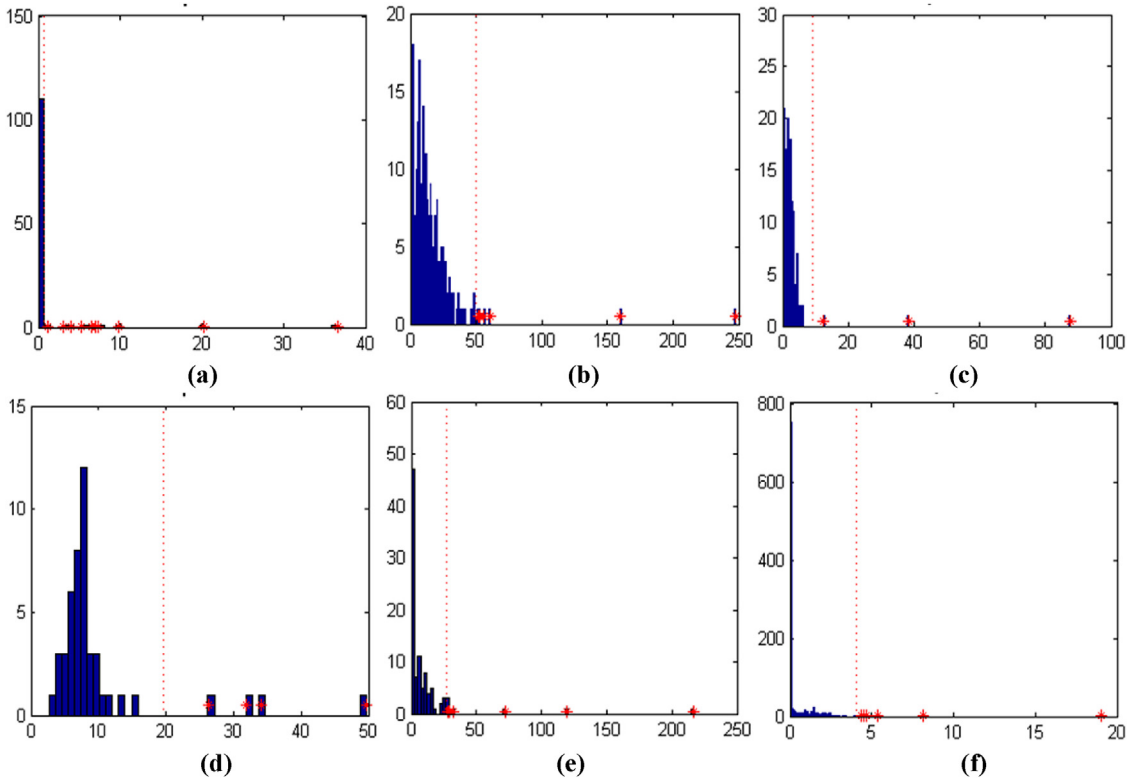


Fig. 13. Singular points which are chosen by 5σ principle of normal distribution curve: (a)Acute (b)Heart (c)Iris (d)Soybean (e)Zoo (f)KDD-CUP 99.

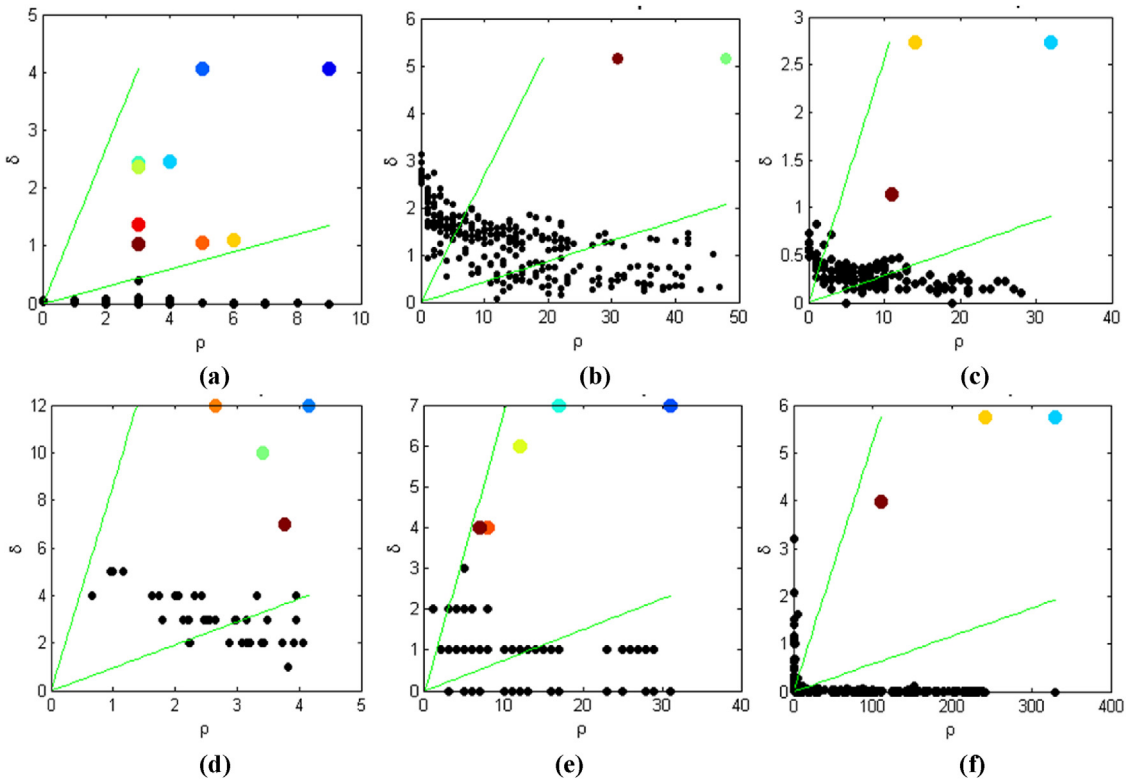


Fig. 14. Final cluster centers are identified of six data sets:(a)Acute (b)Heart (c)Iris (d)Soybean (e)Zoo (f)KDD-CUP 99.

samples as test data. The clustering accuracy rate is the average accuracy of the clustering obtained when clustering the different 1000 samples.

We list the clustering accuracy of our proposed algorithm, WFK-prototypes, KL-FCM-GM and DC-MDACC. The proposed algorithm and DC-MDACC give clustering accuracy of 0.992 and 0.916; WFK-prototypes gives the best clustering accuracy of 0.701 when its fuzziness coefficient α equals 1.7, KL-FCM-GM gives the best clustering accuracy of 0.779 when its fuzziness coefficient α equals 1.9. Clustering results show that our algorithm is 7.8%, 29.1%, and 21.3% more accurate than DC-MDACC, WFK-prototypes, and KL-FCM-GM. Our algorithm therefore is suitable to deal with mixed dataset.

There are some shortcomings about WFK-prototypes and DC-MDACC. The WFK-prototypes algorithm needs to adjust the parameter value in the clustering process. DC-MDACC algorithm uses conditional probability to calculate the distance matrix of the mixed attribute data and uses PSO algorithm to select 10 initial points and iteration 10 times to find the optimal d_c in iteration. The time complexity of this algorithm is high, and it cannot deal with the large data sets efficiently. Our CH-CCFDAC algorithm eliminates the dependence of the parameters, and has the function of automatic clustering. The time complexity of this algorithm is also lower than DC-MDACC. Comparing these two algorithms, the difference of the accuracy of clustering results is less than 3%. This shows that our algorithm has some advantages in the processing of the real data set.

It can also be seen from Table 3 that the clustering accuracy of the RLM algorithm on all data sets is the same as that of the CH-CCFDAC algorithm. This is because in experiment, the distance matrix processed by RLM algorithm is the same as that in CH-CCFDAC algorithm, and in determination of the density radius, RLM algorithm adjusts the d_c value several times, and finally determines the optimal d_c value; CH-CCFDAC algorithm uses the improved climbing algorithm to automatically find the optimal d_c value. The experimental results show that the final d_c results obtained by these two algorithms are the same, so the clustering accuracy is the same.

6.3. Execution time comparison

The time complexity of the algorithm mainly consists of four parts:

1. The number of iterations of the algorithm
2. Calculate each data point's ρ, δ and γ
3. Calculate the mean value μ and variance σ^2 of γ. Then use μ and σ^2 to find out the clustering centers and partition other data points.
4. Calculation of *Fitness* function value

When we use the mountain climbing algorithm to find the optimal d_c, we should set the iteration radius r, and we can calculate the number of iterations by formula (11). When d_c is given, the time complexity of the algorithm is mainly determined by the calculation of ρ and δ. If a data set is composed of n data points, the asymptotic time complexity for the calculation of ρ and δ is $O\left(n^2 - n\right)$, the asymptotic time complexity for calculating μ and σ^2, finding cluster centers and partitioning other data points is $O(8n)$. *For the calculation of Fitness function, if there are m clusters, the time complexity is* $O(m \times n)$. Therefore, the overall time complexity of this algorithm is:

$$O\left(\left(\frac{r}{0.5} \times 2 + 1\right) \times \left(2n^2 + (m+6) \times n\right)\right) \tag{20}$$

Table 4 lists the calculation formulas of the time complexity of some algorithms. Through the analysis of this table, we can find that compared with RLM algorithm and the partition based clustering algorithms (such as K-prototypes, EKP and IWKM), the time complexity of this algorithm is a little bit higher. Its time complexity is mainly determined by the search of the optimal d_c using iter-

Table 4
Time complexity statistics of various algorithms.

Algorithm	Time complexity
K-prototypes	$O((s+1)\times k\times n)$
SBAC	$O\left(n^2+m_c^2\log\left(m_c^2\right)\right)$
KL-FCMGM-GM	$O(T\times(d\times l\times k+(c+2)\times k))$
EKP	$O(T\times k\times n)$
IWKM	$O(k\times(m+p+N\times m-N\times p)\times n\times l)$
WFK-prototypes	$O\left(m^2\times n+m^2\times S^3+k\times(m+p+N\times m-N\times p)\times n\times s\right))$
IPC	$O(T\times(2n+(s+2)\times k))$
RLM	$O\left(\left(2n^2+(m+6)\times n\right)\right)$
DC-MDACC	$O\left(\text{iter}\times m\times\left(\frac{3}{2}n^2+n\times\log(n)\right)\right)$
CH-CCFD	$O\left(\left(\frac{r}{0.5}\times 2+1\right)\times\left(2n^2+(m+6)\times n\right)\right)$

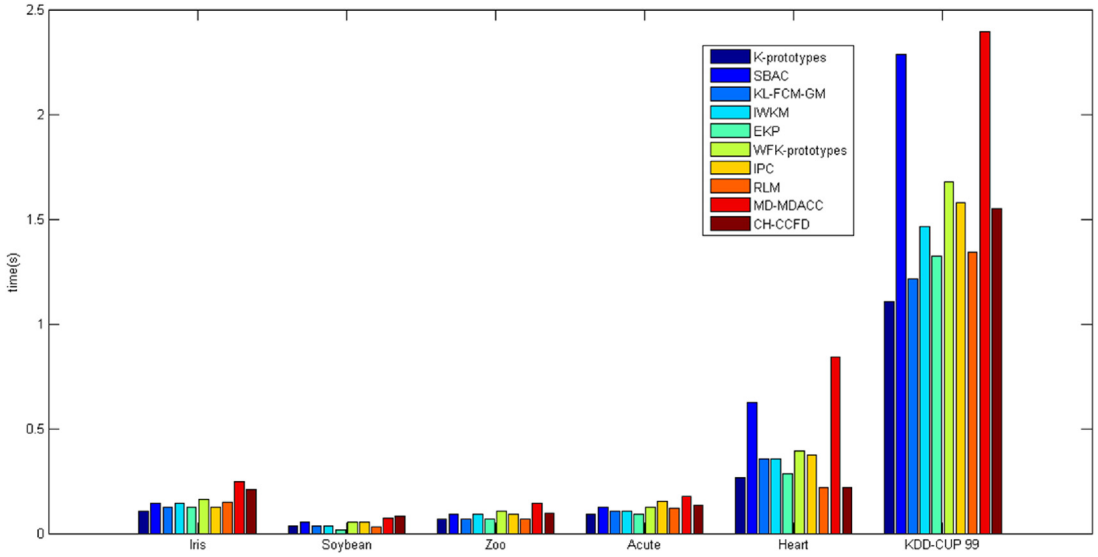


Fig. 15. Comparison of the execution time of all kinds of algorithms.

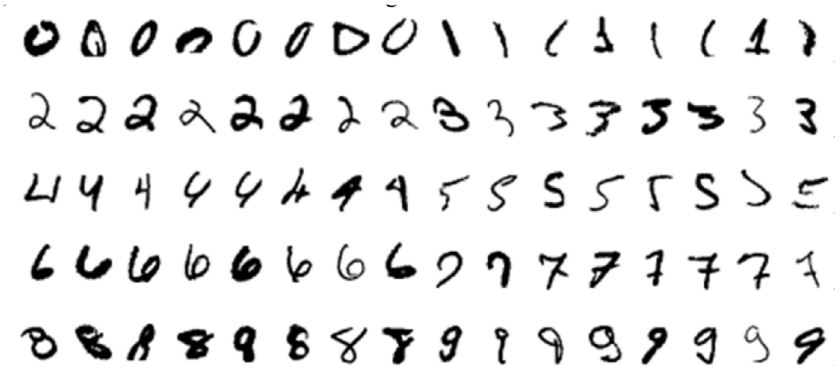


Fig. 16. Examples of handwritten image of different numbers.

ative method. By this way, the clustering process is automated, and the sensitivity of parameters is eliminated. Compared with other algorithms, this algorithm has higher accuracy and can deal with arbitrary shape clusters. It means that this algorithm could get better clustering result. To a certain extent, it makes up for the shortcomings of high time complexity.

Fig. 15 shows the comparison of different algorithms in test data sets. For this algorithm, the execution time is mainly related to the number of iterations, the amount of data points and the data dimension.

It can be seen from Fig. 15 that the data of Iris, Soybean, Zoo, Acute data set is small, so the algorithm performs fast. But the KDD-CUP 99 data set is relatively large in quantity and dimension, therefore, the algorithm execution consumes a long time. The K-prototypes, EKP, IPC and IWKM algorithms are based on the partition of the clustering algorithm, their algorithm execution time is comparable to CH-CCFDAC algorithm. RLM algorithm does not need to adapt to the density radius d_c, so the algorithm execution time is slightly shorter than CH-CCFDAC. And the SBAC algorithm's exe-

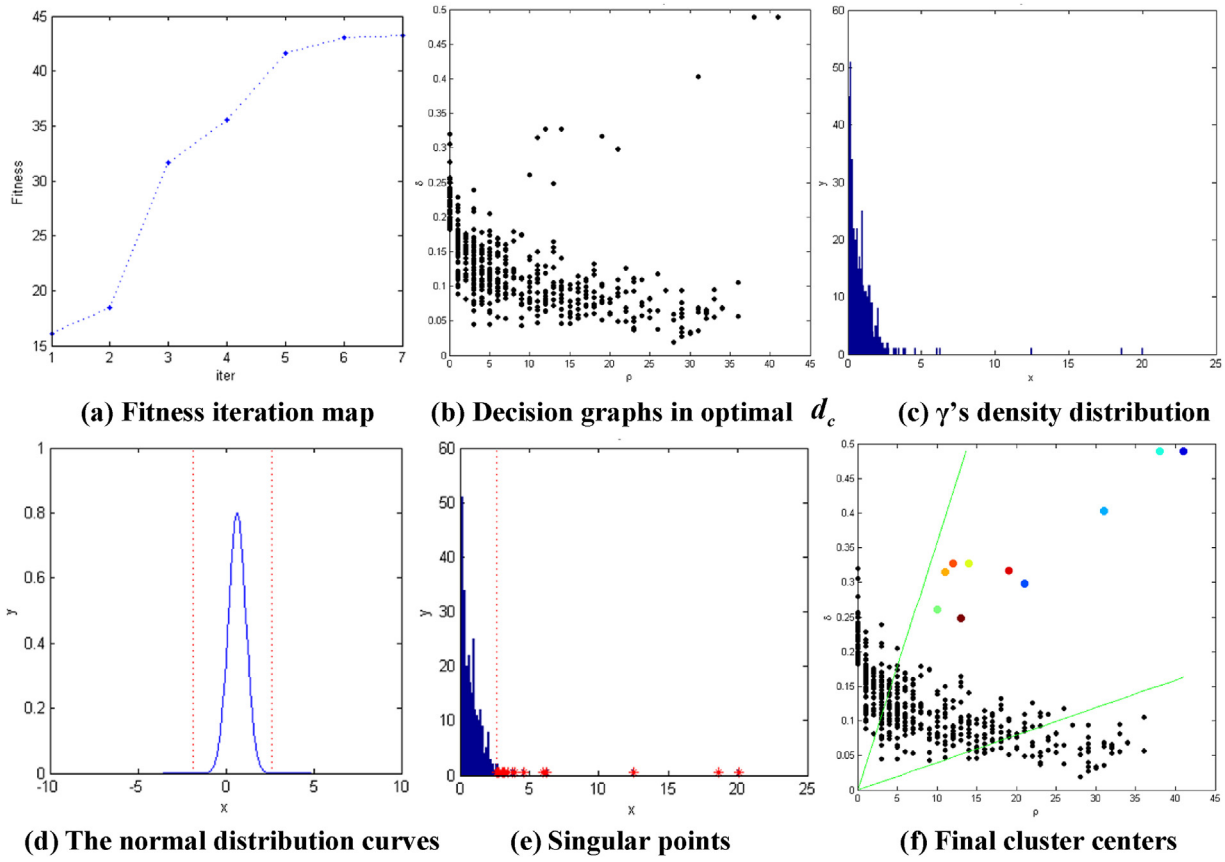


Fig. 17. Clustering process of MNIST Data Set.

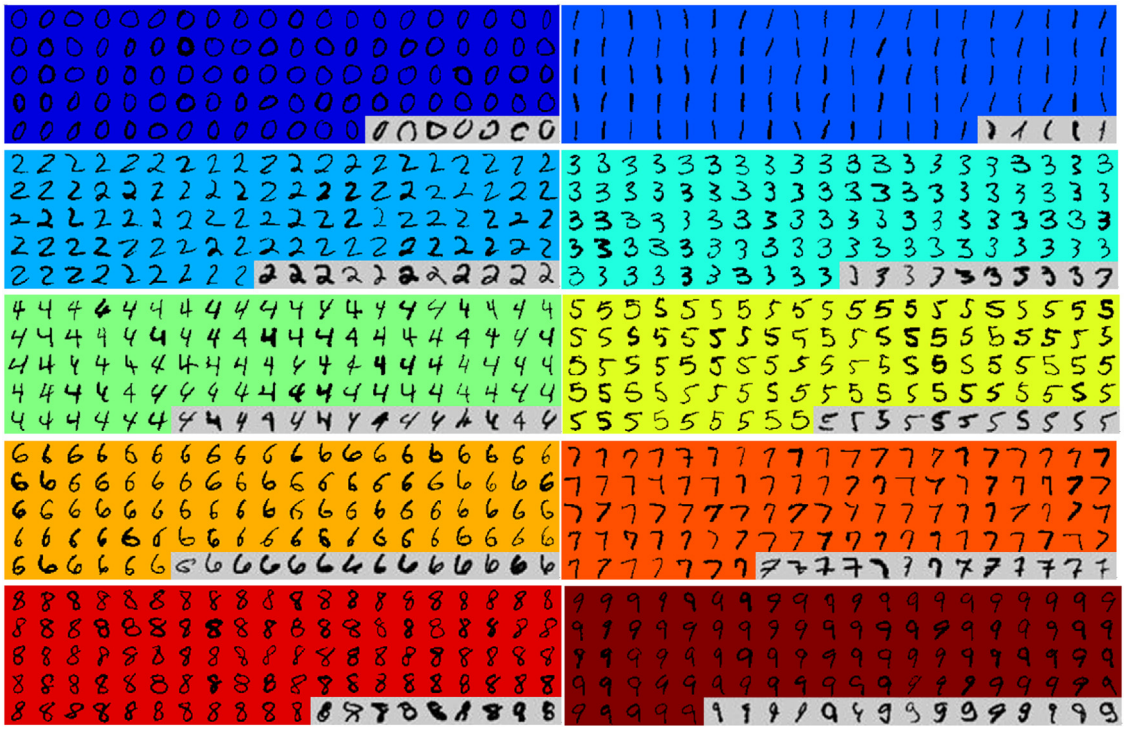


Fig. 18. Clustering result of MNIST dataset(The images of the same color belong to the same cluster, and the gray image represents the misplaced image).

cution time and DC-MDACC algorithm's execution time is slightly higher than CH-CCFDAC algorithm.

6.4. The experiment in real data set

6.4.1. Experimental data set

MNIST data set is applied to testify the performance of CH-CCFDAC algorithm, which is consisted of 60000 number handwriting images. Number from 0 to 9 are all collected for classifier stored as binary file, each of which is 28×28 shown as Fig. 16.

We randomly selected 1000 images from the MNIST data set as training samples, including 100 digital handwritten images for each number, and then use CH-CCFDAC algorithm to test these 1000 samples and analyze the clustering effect of the algorithm on this data set by accuracy and purity.

6.4.2. Similarity distance calculation method for MNIST data set

CW-SSIM (Complex Wavelet Structural Similarity) is applied to evaluate the similarity of number handwriting images. In the process of complex wavelet transform, assume $C_x = \{C_{x,i} | i = 1, \ldots, N\}$ and $C_y = \{C_{y,i} | i = 1, \ldots, N\}$ respectively represent two coefficient sets of different images to be compared, which are extracted from same wavelet sub-band and same spatial location.

$$\tilde{S}(C_x, C_y) = \frac{2|\sum_{i=1}^{N} C_{x,i} C_{y,i}^*| + K}{\sum_{i}^{N} |C_{x,i}|^2 + \sum_{i=1}^{N} |C_{y,i}|^2 + K} \tag{21}$$

Where C^* and C are complex conjugate, K is positive constant with small value, which is used to improve robustness of $\tilde{S}$ at low ratio of signal to noise.

In order to better understand CW-SSIM, right part of the equation is multiplied by an equivalent factor, whose value is 1.

$$\tilde{S}(C_x, C_y) = \frac{2\sum_{i=1}^{N} |C_{x,j}||C_{y,j}| + K}{\sum_{i}^{N} |C_{x,i}|^2 + \sum_{i=1}^{N} |C_{y,i}|^2 + K} \cdot \frac{2|\sum_{i=1}^{N} C_{x,i} C_{y,i}^*| + K}{2|\sum_{i=1}^{N} C_{x,i} C_{y,i}^*| + K} \tag{22}$$

In the first part of right-hand side, each factor is constant or mode of complex wavelet coefficient. For two given images, complex wavelet coefficient corresponds to a certain value. If the condition of $|C_{x,i}| = |C_{y,i}|$ for all i is met, then the first part of right-hand side is maximum value of 1, and the value of second part is related to phase change of C_x and C_y. If the condition that phase change of C_{xi} and C_{yi} is constant for all i is met, then the second part is maximum value of 1. And we use this part as an image structure similarity index.

The range of CW-SSIM is [0,1]. The larger the value is, the higher the image similarity is. So we convert the structural similarity as follows:

$$dist(C_x, C_y) = 1 - \tilde{S}(C_x, C_y) \tag{23}$$

We use $dist(C_x, C_y)$ to represent the distance between C_x and C_y. According to this formula, the smaller the distance, the higher degree of similarity.

Using (21)–(23) can compute the distance values of any two of the 1000 images, that is, we can get the similarity distance matrix of the 1000 images by this method.

6.4.3. Clustering MNIST datasets by CH-CCFDAC algorithm

The process of clustering the MNIST dataset by CH-CCFDAC algorithm is shown in Fig. 17. And the specific clustering steps are:

Step1: Using the improved mountain climbing algorithm to determine the optimal d_c by the iteration value of *Fitness* (as shown in Fig. 17(a)).

Step2: Calculate the ρ and δ of all data points according to the optimal density radius, and make decision graph the density distribution graph of γ (as shown in Fig. 17(b), (c)).

Step3: Calculate the mean value and variance of γ, make a fitting curve for the γ density distribution, and plot the confidence interval(as shown in Fig. 17(d)).

Step4: Select the singular points that fall outside the confidence interval (as shown in Fig. 17(e)).

Step5: Observe the location of singular points on decision graph, delete pseudo-clustering centers in singular points, and finally determine the real cluster centers (as shown in Fig. 17(f)).

Step6: Divide all data points from cluster centers, get the results.

6.4.4. Clustering result

It can be seen from Fig. 17(f) that CH-CCFDAC finally determines 10 clustering centers. And the experimental results show that the 10 clustering centers represent 10 images with different numbers. This shows that CH-CCFDAC can accurately find clustering centers. In addition, the accuracy of the clustering results is 0.892 and the clustering purity is 0.910. The final clustering results are shown in Fig. 18.

7. Conclusion

Based on the density based clustering method proposed by Alex Rodriguez and Alessandro Laio in the journals Nature and Science, we use the normal distribution method to analyze and extract the information of the data points. According to the confidence interval principle, the singular points are selected, then the cluster centers are determined automatically. We also eliminate the parameters' sensitivity of the algorithm by using mountain climbing algorithm to find the optimal d_c and get great clustering result by experimental verification. But this algorithm cannot analyze and deal with the noise points of the data set effectively and cannot get great clustering results for data sets with different levels of density. This will also be our next research direction.

Acknowledgments

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions. This work was supported by a grant from the National Natural Science Foundation of China (No. 61502423), Zhejiang Provincial Natural Science Foundation (Y14F020092).

References

[1] Lixiongwei, Study of clustering algorithm based on density control, Sci. Eng (2010) 27–30.
[2] Huizhuanni, Multidimensional data clustering algorithm research and GPU acceleration based Global K-means, 2012.
[3] Sunlingyan, Research of clustering algorithm based on density, App. Math. TP301 (2009) 6.
[4] Jiawei Han, Micheline Kambr. Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001, pp. 21–25.
[5] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: an Introduction to Cluster Analysis, John Wiley & Sons, NY, 1990.
[6] R. Ng, J. Hart, Efficient and effective clustering method for spatial data mining, in: Proceedings of the International Conference on Very Large DataBases(VLDB' 94, San Francisco : Morgan Kaufmann Publishers, 1994, pp. l44–155.
[7] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: an efficient data clustering method for very large databases, in: Proceedings Ofthe 1996 ACM SIGMOD International Conference on Management of Data, Montreal: ACM Press, 1996, pp. 103–114.
[8] S. Guha, R. Rastogi, K. Shim, CURE: an efficient clustering algorithm for large databases, in: ACM SIGMOD International Conference on Management of Data, Washington: ACM Press, 1998, pp. 73–84.
[9] S. Guha, R. Rastogi, K. Shim, Rock: a robust clustering algorithm for categorical attributes, in: Proceedings of the l5th International Conference On Data Engineering, Washington, DC, USA: IEEE Computer Society, 1999, pp. 512–521.
[10] Martin Ester, H.-P. Kriegel, J. Sander, X. Xu, A DensityBased algorithm for discovering clusters in large spatial databases with noise, Proceedings of KDD (1996).

[11] M. Ankerst, et al., OPTICS: ordering points to identify clustering structure, in: Proceedings of the ACMSIGMOD Conference on Management of Data, Philadelphia: ACM Press, 1999.
[12] A. Hinneburg, D.A. Keim, An efficient approach to clustering in large multimedia databases with noise, ACM SIGKDD (1998) 58–65.
[13] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[14] J.Y. Chen, H.H. He, Research on density-based clustering algorithm for mixed data with determine cluster centers automatically, J. Autom. (10) (2015).
[15] J.C. Ji, T. Bai, C.G. Zhou, C. Ma, Z. Wang, An improved K-prototypes clustering algorithm for mixed numeric and categorical data, Neurocomputing 120 (2013) 590–596.
[16] C. Li, G. Biswas, Unsupervised learning with mixed numeric and nominal data, IEEE Trans. Knowl. Data Eng. 14 (4) (2002) 673–690.
[17] Z.X. Huang, Clustering large data sets with mixed numeric and categorical values, in: Proceedings of the 1st PacificAsia Conference on Knowledge Discovery and Data Mining, Singapore: World Scientific Publishing, 1997, pp. 21–34.
[18] S.P. Chatzis, A fuzzy C-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional, Expert Syst. Appl. 38 (7) (2011) 8684–8689.
[19] Z. Zheng, M.G. Gong, J.J. Ma, L.C. Jiao, Q.D. Wu, Unsupervised evolutionary clustering algorithm for mixed type data, in: Proceedings of the 2010 IEEE Congress on Evolutionary Computation, Barcelona: IEEE, 2010, pp. 1–8.
[20] J.C. Ji, W. Pang, C.G. Zhou, X. Han, Z. Wang, A fuzzy Kprototype clustering algorithm for mixed numeric and categorical data, Knowl.-Based Syst. 30 (2012) 129–135.
[21] Z.X. Huang, A fast clustering algorithm to cluster very large categorical data sets in data mining, in: Research Issues on Data Mining and Knowledge Discovery, ACM Press, Arizona, 1997, pp. 1–8.
[22] A. Ahmad, S. Hashmi, K-Harmonic means type clustering algorithm for mixed datasets, Appl. Soft Comput. 48 (2016) 39–49.
[23] A. Pathak, N.R. Pal, Clustering of mixed data by integrating fuzzy, probabilistic, and collaborative clustering framework, Int. J. Fuzzy Syst. 18 (3) (2016) 339–348.
[24] D. Lam, M. Wei, D. Wunsch, Clustering data of mixed categorical and numerical type with unsupervised feature learning, Access IEEE. 3 (2015) 1605–1613.
[25] R.S. Sangam, H. Om, Hybrid data labeling algorithm for clustering large mixed type data, J. Intell. Inf. Syst. 45 (2) (2015) 273–293.
[26] A. Foss, M. Markatou, B. Ray, et al., A semiparametric method for clustering mixed data, Mach. Learn. (2016) 1–40.
[27] C.H. Chung, C.C. Yu, Mining of mixed data with application to catalog marketing, Expert Syst. Appl. 32 (1) (2007) 12–23.
[28] A. Amir, D. Lipika, A k-mean clustering algorithm for mixed numeric and categorical data, Data Knowl. Eng. 63 (2) (2007) 503–527.
[29] G. Gan, J. Wu, Z. Yang, A genetic fuzzy k k mathContainer Loading Mathjax −Modes algorithm for clustering categorical data, Expert Syst. Appl. 36 (2) (2009) 1615–1620.