Relative Density Clustering Algorithm Based on Density Fluctuation

Cai Zhao GAN, Hong Bin HUANG
School of Information Systems and Management, National University of Defense Technology, Changsha, Hunan 410073, China
Email: 438184836@qq.com

Abstract: With strong ability of noise resistance and discovery clusters of arbitrary shapes makes, the traditional density-based clustering algorithm has been widely used in clustering analysis. However, the quality of clustering results would be reduced significantly when the traditional density clustering algorithm was used to analyze uneven density datasets. Though the classical OPTICS algorithm can identify the clusters with uneven density, it is useless to the clusters whose boundaries are adjacent or mixed. To overcome this problem, a density fluctuation-based relative density clustering algorithm, named DFRDC (Relative Density Clustering Algorithm Based on Density Fluctuation) is proposed. The experimental results show that DFRDC can effectively identify multiple clusters with different densities and outperform of the state-of-the-art density clustering algorithms.

1. Introduction
Clustering analysis is an important algorithm for machine learning, which has been widely used in many fields such as pattern recognition [1], data analysis [2], image processing [3] and so on. There are many clustering algorithms existed in the literature. Although all the clustering algorithms can cluster datasets, the effect of each clustering algorithm is different for different categories of datasets. At present, the mainstream clustering algorithms can be divided into the following categories: the division-based method, the hierarchical-based method, the density-based method, the grid-based method and the model-based method. Among those methods, the density-based method has the advantages of strong ability of noise resistance and discovering clusters of arbitrary shapes. However, the general density-based clustering algorithms use the absolute density indicators, which make the results obtained unsatisfactory when clustering the clusters with different densities. To solve this problem, this paper proposes a new density-based clustering algorithm, named DFRDC (Relative Density Clustering Algorithm Based on Density Fluctuation). The DFRDC algorithm not only has the advantages of the general density-based clustering algorithm, but also can effectively distinguish multiple clusters with uneven density because of using the indicators of relative density.

2. Related research
DBSCAN and OPTICS are both typical density-based clustering algorithms. The DBSCAN algorithm [4] implements clustering by setting two parameters: the radius of the object's neighborhood $\varepsilon$ and the minimum number of objects $MinPts$ in the $\varepsilon$-neighborhood which searches for clusters by examining the $\varepsilon$-neighborhood of each point of the dataset. If there are more than $MinPts$ points in the $\varepsilon$-neighborhood of a point $p$, create a cluster with $p$ as the core object. Then DBSCAN repeatedly looks for objects that are density-reachable from these core objects which may involve the merge of some density reachable clusters. When no new points can be added to any cluster, this
It can be found that the high-density clusters are completely included in the connected low-density clusters for a neighborhood radius $\epsilon$ and a minimum number of objects $MinPts$ which are globally constant [5]. That is, the DBSCAN algorithm cannot find multiple clusters with different densities at the same time. In order to solve this problem, Ankerst et al. proposed another well-known density-based clustering algorithm OPTICS [6]. OPTICS calculates a clustering order for automatic and interactive clustering analysis, which represents the density-based clustering structure of the data. The information contained is equivalent to the density-based clustering obtained from a wide range of parameter settings, but it does not explicitly create a clear data clustering result for the user. And when the boundaries between clusters with different densities are close, OPTICS cannot completely distinguish all clusters from any parameter values.

According to the above analysis, it can be found that using absolute density indicators to clustering which making it difficult to distinguish between multiple clusters with different densities is the biggest drawback of the classical density clustering algorithms. And the essence of clustering is to make the objects in the same cluster have great similarity, and the objects between different clusters have great dissimilarity. That is, the difference of density between different clusters should be relative, not absolute [7]. Therefore, this paper proposes a clustering algorithm based on the relative density, named DFRDC, which breaks down the entire dataset and then reorganizes it to find clusters with different densities according to the variation of density between different blocks. The DFRDC can well identify the clusters with arbitrary shapes and effectively solve the problem that the traditional density clustering algorithm cannot simultaneously identify multiple clusters with uneven density whose boundaries are adjacent to each other.

3. Related concepts
This chapter will introduce some concepts used in this paper.

**Definition 1.** The $n$-distance of an object $p$ [8].

Given any positive integer $n$ and a set $D$, the $n$-distance of an object $p$ is defined as the distance from the object $p$ to its $n$th nearest neighbor $o$, where $p, o \in D$. The $n$-distance of an object $p$ is denoted by $n\_distance(p)$.

**Definition 2.** The $n$-neighbors of an object $p$ [8].

Given any positive integer $n$ and a set $D$, the $n$-neighbors of an object $p$ is defined as the set constituted by the nearest $n$ objects of the object $p$ (include the object $p$), where $p \in D$. The $n$-neighbors of an object $p$ is denoted by $n\_near(p)$, the set representation is as follows:

$$n\_near(p) = \{q \in D | dist(p, q) \leq n\_distance(p)\}$$  \hspace{1cm} (1)

where $dist(p, q)$ stands for the Euclidean distance between the object $p$ and the object $q$.

**Definition 3.** The $n$-density around an object $p$.

Given any positive integer $n$ and a set $D$, the $n$-density around an object $p$, where $p \in D$, is defined as follows:

$$n\_density(p) = \frac{n}{n\_distance(p)^{dimen(p)}}$$  \hspace{1cm} (2)

where $dimen(p)$ represents the vector dimension of the object $p$.

**Definition 4.** The $n_1-n_2$-amplitude of the density fluctuation of an object $p$.

Given any positive integer $n_1, n_2$ and a set $D$, the $n_1-n_2$-amplitude of the density fluctuation of an object $p$, where $p \in D$, is defined as follows:

$$n_1-n_2\_Fluct(p) = \frac{n_2\_density(p)}{n_1\_density(p)}$$  \hspace{1cm} (3)

**Definition 5.** The entity class: Circle.
Given a set $D$, the Circle object $C_p$ which considers the object $p$ as the center has four properties: $Cpt$ (the object of center point), $r$ (the radius), $dens$ (the density) and $neighbors$ (the set of neighbors), where $p \in D$.

From the perspective of two-dimensional data, a Circle object corresponds to the whole constituted by the data points covered by the round with the $Cpt$ as the center of the round and the $r$ as the radius of the round.

**Definition 6.** The amplitude of the density fluctuation between the Circle objects.

Given a set $D$ and two Circle objects $C_1, C_2$, where $C_1.cpt, C_2.cpt \in D$, the amplitude of the density fluctuation between the Circle objects is defined as follows:

$$C\_\text{Fluct}(C_1, C_2) = \frac{C_1.dens}{C_2.dens}$$

(4)

**Definition 7.** Density-reachable.

Given a set $D$ and two Circle objects $C_1, C_2$, where $C_1.cpt, C_2.cpt \in D$, when two Circle objects meet the following conditions, these Circle objects can be considered to be density-reachable.

$$|C\_\text{Fluct}(C_1, C_2) - 1| < C_2$$

(5)

$$C_1.r + C_2.r < \text{dist}(C_1.cpt, C_2.cpt)$$

(6)

where $C_2$ is the parameter controlling the constraint of density-reachable, $\text{dist}(p, q)$ stands for the Euclidean distance between the object $p$ and the object $q$.

**Definition 8.** The set of density-connected.

Given a Circle object $C_0$ and multiple other Circle objects $C_i, i = 1, 2, 3 \cdots$. If $C_0, C_i$ are density-reachable and $C_i, C_{i+k}, k = 1, 2, 3 \cdots$ are density-reachable, the set of density-connected which is started from $C_0$ is constituted by $C_0, C_i, C_{i+k}$.

As shown in Figure 1, there are five Circle objects. Assume that the Circle object A is the starting point, if B and A, C are density-reachable and D and A, E are density-reachable, the set of density-connected which is started from A is constituted by these five objects.

![Figure 1 The illustration of density-connected](image_url)

**4. The description of DFRDC algorithm**

The general idea of the DFRDC algorithm is to divide the entire dataset into a number of small pieces according to the density of the small area, and then gather those pieces with similar density whose boundaries are adjacent to each other to finish clustering. The DFRDC algorithm has three main steps described as follows.

Step 1: Divide the dataset into Circle objects.

First, given a small positive integer $n_0$, $n_0 \ll N$ ($n_0$ is the parameter used to determine the initial $n_0$-density around the data object, $N$ is the number of the data objects in the dataset), calculate the
initial $n_0$-density around each data object $n_0.density(p_i), i = 1, 2, \ldots n$ in the dataset. Second, take an data object with the largest value of the initial $n_0$-density around from those data objects which have not been accessed and find out the Circle object centered on this object. Third mark the data objects in the neighbors of this Circle object as being accessed. Repeat the second and third parts, when all the data objects in the dataset are accessed, this step ends and returns all the Circle objects $C_i, i = 1, 2, 3 \ldots$ as the results of step 1. The process of finding the Circle object centered on the object $p$ in the second part is as follows. Given a small positive integer $k$ ($k$ is the radius expansion each time, $k \ll N$), calculate the $n_i.n_0.Fluct(p), n_i = n_0 + i \times k, i = 1, 2, 3 \ldots$ in turn. When the value of $i$ meet $|n_i.n_0.Fluct(p) - 1| < C1$, where $C1$ is the threshold parameter, the Circle object centered on $p$ can be determined. The four attribute values of this Circle object are that the center object is $p$, the radius is $n_{i-1}.distance(p)$, the density is $n_{i-1}.density(p)$ and the set of neighbors is $n_{i-1}.near(p)$.

Step 2: Clustering all Circle objects.

All the Circle objects obtained in step 1 are used as the input of step 2. First, take a Circle object which have the largest density attribute value from the Circle objects which have not been accessed. Second, find out all Circle objects in the set of density-connected which is started from this Circle object and sort these Circle objects into one class. Third, mark these Circle objects as being accessed. Repeat the second and third parts, when all the Circle objects are accessed, this step ends, and returns the clustering results for these Circle objects as the results of step 2.

Step 3: Clustering the data objects of the dataset according to the clustering results of the Circle objects.

The clustering results obtained in step 2 are used as the input of step 3. First, take a cluster from the clustering results of step 2 in turn. Second, take all non-repeating data objects which have not been accessed from the neighbors of the Circle objects of this cluster and sort these data objects into one class. Third, mark these data objects as being accessed. Repeat the first, second and third parts, when all the data objects in the dataset are accessed, this algorithm ends. The results of step 3 is the final results. The pseudo code of DFRDC is described below.

1. **DFRDC** ($data, n_0, k, C1, C2$):
   // $data$ are the input of the data objects
   // $n_0$ is the parameter of calculating initial density around
   // $k$ is the parameter of the radius expansion each time
   // $C1, C2$ are the thresholds controlling the condition of clustering.
2. BEGIN
3. Calculate the initial $n_0$-density around of each object $n_0.density(p_i), i = 1, 2, 3 \ldots$
4. WHILE NOT (ALL DATA OBJECT BEEN VISITED):
5. Take out the data object $p$ which have the largest $n_0.density(p_i)$ from the data objects which have not been accessed.
6. let $j = 1, n_j = n_0 + k$
7. WHILE $|n_j.n_0.Fluct(p) - 1| < C1$:
8. let $j = j + 1, n_j = n_0 + j \times k$
9. END WHILE
10. According to the value of $j$, the Circle object $C_p$ centered on $p$ can be determined. The four attribute values of $C_p$ are that $C_p.cpt = p, C_p.r = n_{j-1}.distance(p), C_p.dens = n_{j-1}.density(p), C_p.neighbors = n_{j-1}.near(p)$
// The Circle objects $C_i, i = 1, 2 \ldots$ are constituted by the Circle object $C_p$ obtained each time.
11. THE DATA OBJECT IN $n_{j-1}.near(p)$ SET VISITED
12. END WHILE
13. WHILE NOT (ALL CIRCLE OBJECT BEEN VISITED):
14. Take out the Circle object $C$ which have the largest density attribute value from the Circle objects $C_i, i = 1, 2 \ldots$ that have not been accessed.
(11) Sort all the Circle objects in the set of density connected which is started from $C$ into one class $\text{Cluster}_C^i, i = 1, 2, \ldots$.

(12) ($C_i$ IN $\text{Cluster}_C^i$) SET VISITED

(13) END WHILE

(14) SET ALL DATA OBJECT UNVISITED

(15) WHILE NOT (ALL DATA OBJECT BEEN VISITED):

(16) $\text{Cluster}_C = \text{Pop (Cluster}_C^i)$

(17) Sort all non-repeating data objects which have not been accessed from the neighbors of the Circle objects in $\text{Cluster}_C$ into one class $\text{CLUSTER}_i, i = 1, 2, 3, \ldots$.

(18) Mark these data objects in $\text{CLUSTER}_i$ as being accessed

(/\text{CLUSTER}_i, i = 1, 2, 3, \ldots are the final clustering results.)

(19) END WHILE

(20) END

5. Results

The performance of the DFRDC algorithm is analyzed from the algorithm's time complexity and the clustering effect.

5.1 Time complexity

There is no cross in the implementation of the three steps of DFRDC. Therefore, it is only need to find the maximum value of time complexity in these steps when calculating the time complexity of the DFRDC algorithm. The time complexity of each step is calculated separately below.

In the step 1, the time complexity of calculating the initial $n_0$-density around can be divided into two parts. The first part is to calculate the Euclidean distance between each data object, let $N$ is the number of the data objects in the dataset, then the time complexity of this part is $O(N^2)$. The second part is to calculate the initial $n_0$-density around each data object in turn. While calculate the initial density around a data object each time, the Euclidean distances between the target data object and other data objects need to be traversed, so the time complexity of this part is also $O(N^2)$. For the process of finding the Circle objects, let the number of the Circle objects found is $K, K < N$, considering the worst case that only one data object in the neighbors of the Circle object which is found each time has not been accessed, then the time complexity of finding the Circle objects is $O\left(\sum_{k=1}^{N-K+1} k\right) = O(NK)$. According to the above analysis, the time complexity of step 1 is $O(N^2 + N^2 + NK)$, that is $O(N^2)$.

In the step 2, let $K$ is the number of the Circle objects found in step 1, $K < N$, considering the worst case that each Circle object is clustered into one class, then the time complexity of step 2 is $O\left(\sum_{k=1}^{K-1} (k-1) + (k-2) + \ldots + 1 + 0\right) = O(K^2)$.

In step 3, regardless of the number of the clusters of the Circle objects obtained in step 2, the number of Circle objects contained by these clusters is fixed to $K$. Considering the worst case that only one data object in the neighbors of the Circle object which is found each time has not been accessed in step 1, and it means that only one data object that has not been accessed can be found when visiting each Circle object, and the number of data objects that need to be visited each time in the worst case is $N - K + 1, N - k + 2, \ldots, N$ in turn. Therefore, the time complexity of step 3 is $O\left(\sum_{k=1}^{N-K+1} (N-k+1)\right) = O(KN)$.

According to the above analysis, the time complexity of the DFRDC algorithm is $\max(O(N^2), O(K^2), O(KN)) = O(N^2)$. The time complexity of the DBSCAN and the OPTICS are both $O(N^2)$. Therefore, there is no significant difference in the operation time between the DFRDC algorithm and the classical clustering algorithm such as DBSCAN and OPTICS.
5.2 Results and analysis

There are two types of datasets in the experiment, one of which is uniform density, which is called the general datasets. The other one is constituted by uneven clusters whose boundaries are adjacent to each other. The experiment will test the clustering performance of the DFRDC according to the clustering results of these two types of datasets. The first experiment used the general dataset. The data is taken from the sample data of the website [9] http://scikitlearn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py. Figure 2 shows four datasets belonged to the general datasets from the website.

![Figure 2 The general datasets](image)

Figure 2 The general datasets

Figure 3 shows the clustering results of the DFRDC for those datasets. It can be seen from the figure that the results obtained by DFRDC are as good as the other clustering algorithms. And it means that the DFRDC has the advantages of the general density-based clustering algorithm such as recognizing the clusters with arbitrary shapes and strong anti-noise ability.

![Figure 3 The clustering results of DFRDC for the general datasets](image)

Figure 3 The clustering results of DFRDC for the general datasets

The second experiment used the datasets with uneven density whose clusters’ boundary are adjacent. Figure 4 shows three datasets with uneven density. The datasets are made up by the annular clusters constituted by two dimensional random points which is generated by the random function of the computer, the boundary of each annular cluster is adjacent, and the width of each annular cluster in these datasets is 1. The Dataset1 contains two clusters, the number of the data points in each cluster is 4000 and 3000. The Dataset2 contains three clusters, the number of the data points in each cluster is 4000, 3000 and 2000. The Dataset3 contains four clusters, the number of the data points in each cluster is 4000, 3000, 2000 and 1000.

![Figure 4 The dataset constituted by the clusters with different densities](image)

Dataset1  Dataset2  Dataset3

Figure 4 The dataset constituted by the clusters with different densities
The DBSCAN, the OPTICS and the DFRDC were used for experimenting in the second experiment. Figure 5 shows the results of these algorithms obtained by clustering these datasets. The specific values of these three algorithm parameters in the experiment are as follows. The parameters of DBSCAN are all $\varepsilon = 0.1$, $MinPts = 15$. The parameters of OPTICS in the process of calculating the reachable distance are all $\varepsilon = 0.3$, $MinPts = 10$. The parameters $\bar{\varepsilon}$ of OPTICS used in the clustering process are 0.08, 0.12, 0.20, respectively. The parameters of DFRDC are all $n_0 = 10$, $k = 10$, $C1 = 0.15$, $C2 = 0.6$. The results show that the DFRDC algorithm is better able to identify such datasets than the other algorithms. On the contrast, the DBSCAN algorithm is unable to identify the clusters with different densities, and the OPTICS is faced with the problem of high-density clusters containing low-density clusters when clustering the clusters whose boundaries are adjacent to each other.

The accuracies of the algorithms are calculated by using the correct rate.

$$\text{accuracy} = \frac{\text{the number of the data clustered properly}}{\text{the number of the data in the dataset}}$$  \hspace{1cm} (7)

The experimental accuracies of the DBSCAN, the OPTICS and the DFRDC are shown in Table 1.

| Accuracy | Dataset1 | Dataset2 | Dataset3 |
|----------|----------|----------|----------|
| The number of clusters | 2 | 3 | 4 |
| DBSCAN | 54.84% | 45.66% | 41.23% |
| OPTICS | 55.50% | 43.05% | 51.36% |
| DFRDC | 96.37% | 93.77% | 90.55% |

According to the experimental results, the DFRDC algorithm not only has the advantages of the traditional density-based clustering algorithm, but also can get satisfactory clustering effect on datasets.
with uneven density. In addition, when the boundaries of clusters with uneven densities are adjacent, the OPTICS algorithm will have the problem of high-density clusters containing low-density clusters. And the DFRDC can also solve this problem effectively.

6. Conclusion

The traditional density-based clustering algorithm uses the absolute density index to clustering, which cannot get satisfactory clustering results for the clusters with uneven density. In order to solve this problem, this paper proposes a new clustering algorithm based on relative density, named DFRDC, which can effectively cluster the datasets with uneven density. The experimental results show that though the classical OPTICS algorithm can identify the clusters with different densities, the DFRDC algorithm can obtain more satisfactory clustering results compared with the OPTICS algorithm when the boundary of each cluster is adjacent or mixed.

References

[1] Tsai C W, Lin T Y, Chiang M C, et al. Continuous space pattern reduction for genetic clustering algorithm[C]// Conference Companion on Genetic and Evolutionary Computation. ACM, 2012:1475-1476.

[2] Son E J, Kang I S, Kim T W, et al. A spatial data mining method by clustering analysis[C]// Acm-Gis '98, Proceedings of the, International Symposium on Advances in Geographic Information Systems, November 6-7, 1998, Washington, Dc, Usa. DBLP, 1998:157-158.

[3] He J, Zheng J, He J, et al. A robust image segmentation method using hierarchical color clustering[C]// International Conference on Intelligent Information Processing. ACM, 2016:1.

[4] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// International Conference on Knowledge Discovery and Data Mining. AAAI Press, 1996:226-231.

[5] Liu Q B, Deng S, Lu C H, et al. Relative density based k-nearest neighbors clustering algorithm[C]. 2003 Intern-ational Conference on Machine Learning and Cybernetics. IEEE, 2003

[6] Ankerst M, Breunig M, K riegel H-P, et al. OPTICS: Ordering Points To Identify the Clustering Structure. In: Proc. ACM SIGMOD’99, Int Conf. on Management of Data, Philadelphia, PA, 1999

[7] LIU Qing Bao, Deng Su, ZHANG Wei Ming. Relative Density based Clustering Algorithm. Computer science, 2007, 2(34)

[8] Breunig M M. LOF: identifying density-based local outliers[J]. Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD '00), 2000, 29(2):93-104.

[9] Comparing different clustering algorithms on toy datasets [EB/OL] http://scikitlearn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py.