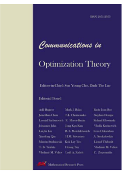
Commun. Optim. Theory 2021 (2021), Article ID 15 https://doi.org/10.23952/cot.2021.15

MATHRES

Communications in Optimization Theory

Available online at http://cot.mathres.org



DENSITY-BASED CORE-STRUCTURES EXPANSION FOR CLUSTERING DATA WITH VARYING DENSITIES GREATLY

LEI CHEN

Institute of Fundamental and Frontier Sciences,
University of Electronic Science and Technology of China, Chengdu, China

Abstract. The K-means clustering algorithm as the representation of the partition algorithms is efficient for convex data. Density-based clustering algorithms can be used to solve the nonconvex data, but they adopt the global threshold value, which makes them perform badly on varying densities. In particular, the densities of components are very different in datasets. Aiming at these problems, we propose a Density-Based Core-Structures Expansion algorithm (DBCSE) and use the Density-Based Core-Structures structured by density searching to replace the represent points of K-means. And we set a relatively large density threshold to detect the core structures and adopt border expanding way to cluster relatively low density area so that our algorithm can cluster different density components, arbitrary shape data, and detect noise. We also conduct experiments on two synthetic datasets and four UCI datasets to demonstrate our algorithm effectiveness.

Keywords. Cluster border expanding; Denoising; Density-Based Clustering; Partition clustering.

1. Introduction

The era of artificial intelligence (AI) raises a higher demand for data processing [1, 2]. A lot of valuable information need to be mining in all kinds of datasets. Clustering algorithms [3, 4, 5] are efficient tools to divide unlabeled data into some different components according to the similarity drawn by certain mathematical principles [6, 7, 8]. Clustering was proposed half a century ago. Since then, various results were obtained. Classification for unlabeled data is now a hot study in the field of data mining [4, 9]. K-means [10], which is a most fundamental clustering algorithm, is still one of the most popular and widely used clustering algorithms due to the efficiency and simpleness. K-means, including its evolutionary algorithms [11, 12], selects K points as the initial centers (represent points) of each cluster according to a certain strategy and then assigns all the other points to the center in accordance with the proximity principle [13, 14]. Then it calculates the mean of each cluster and regard the mean as the new center points. Repeat the above steps along the direction that the value of the cost function reduces. The principle decides that K-means just deals with the convex data. DBSCAN makes

E-mail addresses: rocky_cl@126.com, rock@std.uestc.edu.cn.

Received October 26, 2021; Accepted November 9, 2021.

up the shortcoming of K-means for just clustering convex datasets [15, 16, 17] since it takes advantage of the density to cluster. Observe that it just sets a global density threshold value to detect the core points and border points. So, it performs badly on the data with varying densities. Being the same with DBSCAN [18], DPC [19] suffers from the similar problem especially faced the component density varying greatly. A new approach, Local gap density(LGD) [20], had recently been proposed to address the problem of varying densities. It is executed on the basis of K-NN graph. And it considers the average distance, which is from this point to the all points connected to the point in the K-NN graph as a supplement. By this way, LGD has the ability to address the problem of varying density to some extent. However, the density is a relative concept. LGD is still faced with the challenge of varying densities when the densities of components (clusters) are very different and vary greatly. In a word, LGD cannot detect the noise.

In this paper, we propose a feasible clustering algorithm, the Density-Based Core-Structures Expansion (DBCSE), to address above issues. The DBCSE is based on the K-means, the DBSCAN, and the LGD. Like K-means, DBCSE also finds the center of each cluster. However, it is not a point but a block (structure), which is a core-structure containing many points in high density areas by the density searching way of the DBSCAN and the LGD. In this way, we will obtain c (c is the number of clusters in a dataset) biggest core-structures, i.e., there will be more than c core-structures, but we just select the biggest c core-structures as the centers. And we take these c biggest core-structures as the initial c clusters and mark them. We take the way of border expansion to fuse the remain points progressively on the principle of proximity. Besides, the step-length of the border expansion is based on the average weight of the links in the core-structures to set as the percentage relation. In the end, the points that have not been absorbed by the expansion process will be considered as noise according to the actual situation. In this way, we replace the center points with core-structures and use the density algorithm just in high density areas. Our algorithm absorbs the advantages of density clustering and partition clustering. Hence, we can obtain better results, and the experiment demonstrates that our density-based core-structures expansion method can cluster data with varying densities greatly and detect the noise.

The rest of the paper is organized as follows. We discuss the related works and evaluate them simply in Section 2. In Section 3, we define a new center structures and progressive assignment way, and develop a density-based and partition clustering algorithm. In section 4, we performe some experiments on relevant datasets. Section 5, the last section, concludes with a summary and some directions for future research.

2. Related Works

Some well-known clustering algorithms [21, 22, 23] based on partition and density have been proposed. We select some canonical and closely related works with us and give them a brief description in this section.

2.1. K-means.

K-means finds randomly k points as the initial cluster centers. Then it assigns the other points to the cluster centers on the principle of proximity. Finally, K-means calculates the means of every cluster and regards them as the new cluster centers. K-means repeats the process to obtain the better clustering results. The evaluation criterion of the repetitive operation is the objective function

$$J = \sum_{i=1}^{k} \sum_{v_j \in U_i} ||v_j - u_i||_2^2, \tag{2.1}$$

where k is the total number of clusters, U_i is the point set of a cluster, u_i is the centroid of this cluster, and v_j is the any point in this cluster. Limited by the centroid and the principle of proximity, K-means just clusters the convex data and cannot detect the noise because the centroid is just a point and it cannot reflect the actual structure of a cluster at all. The original K-means algorithm has been improved and optimized. For example, Mao and Jain [24] used the Mahalanobis distance metric to measure the similarity instead of the Euclidean distance. It extends the data shape from sphericity to hyper-ellipsoid. Another variant algorithm, K-medoid, replaces the mean with the actual point in this cluster. Compared with K-means, it makes the cluster more compact. But, it is still subject to this restriction that they cannot deal with the arbitrary shape data.

2.2. **DBSCAN.** The DBSCAN generates clusters by a serious of searching from a arbitrary core point, which is like the expansion of a neural node. This expansion can only proceed if certain conditions are met, i.e., the density-connected. some relevant definitions are elaborated in the following.

δ-neighborhood of p: $N_\delta(p) = \{q \in X | \text{dist}(p,q) \le \delta\}$, where δ is a radius, p,q are the points in the dataset X, and $\text{dist}(p,q)$ is the distance between p and q.

Core point p: $|N_\delta(p)| \ge$ MinPts, where MinPts is a threshold value.

Directly density-reachable: A point q is directly density-reachable from a core point p if $q \in N_\delta(p)$.

Density-reachable: A point p is density-reachable from a point q if there is a chain of points $p_1, \cdots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density-reachable from p_i.

Density-connected: A point p is density-connected to a point q if there is a point k such that p and q are both density-reachable from k.

DBSCAN clustering algorithm requires users to give the parameters δ and MinPts prior. Then it starts to cluster from a arbitrary core point and regards it as a seed. Next DBSCAN retrieve all points, which are density-reachable from the seed to generate a cluster. A cluster is a set of density-connected points, which is maximal. DBSCAN will find the next core points as the seed to generate the another cluster till all clusters are found and the points remained points will be regarded noise.

DBSCAN adopts density links to cluster instead of partition such that it can achieve clustering of arbitrary shape. But it completes the whole clustering process with two global parameters, which lead that DBSCAN cannot cluster data whose densities of component vary considerably [25, 26, 27]. As a result, it is also hard to detect the noise for DBSCAN in varying densities datasets.

2.3. **LGD.** LGD is a algorithm aiming at data high-dimensional data with varying densities. It argues the density of a point should not be reflected only by the number of surrounding locations but includes the average weight of the edges, which link to this point. First, the local density is defined as follows

$$\rho_i = \sum_{j=1}^{n} \lambda(d_{i,j} - d_\sigma), \tag{2.2}$$

where n is the number of points in the dataset, $d_{i,j}$ is the distance between p_i and p_j, d_σ is a threshold, and

$$\lambda(d) = \begin{cases} 1, & d < 0, \\ 0, & \text{otherwise}. \end{cases} \tag{2.3}$$

Second, LGD algorithm takes average weight of the edges, which link to this point into account. It is described as follows

$$\rho_i' = \frac{|\mathscr{N}_i^k|}{\bar{\mathscr{V}}_i^k}, \tag{2.4}$$

where $\mathscr{N}_i^k$ is the set of the points, which are linked to p_i in the K-NN graph. By the same token, $|\mathscr{N}_i^k|$ is the quantity of points, $\bar{\mathscr{V}}_i^k = \sum\limits_{p_j \in \mathscr{N}_i^k} w_{i,j}/|\mathscr{N}_i^k|$ is the average weight of the edges, which are linked to p_i and $w_{i,j} = d_{i,j}^2$.

Finally, LGD gives the definition of Local Gap Density as

$$\varphi_i = \frac{\rho_i'}{\max\{\rho_j' | p_j \in \mathscr{N}_i^k\}}. \tag{2.5}$$

LGD can cluster data with varying densities, but the concept of density is relative. When the densities of components (clusters) are very different and vary greatly, it perform badly. The most important thing is that LGD cannot deal with noise points.

3. Density-Based Core-Structure Expansion for Clustering

In order to address the ambiguous assignments caused by the global density in the intersection of clusters in the datasets with varying greatly density, we propose a Density-Based Core-Structures Expansion (DBCSE) way on the basis of LGD and K-means. By this DBCBE, we develop an effective clustering algorithm to process data whose densities of components (clusters) are very different and vary greatly.

3.1. Density-Based Core-Structure.

K-means clustering algorithm takes the mean as the representative point of a cluster. However, it is clear that a single point does not reflect the actual structure of a cluster. We take the advantage of LGD by reinforcing the conditions to construct the core-structure.

Give a set $\mathscr{A} = \{e_{i,1}, e_{i,2}, ..., e_{i,k}\}$, whose elements are the edges that point i to its k nearest points. The maximal edge $e_{i,m}(e_{i,m} \in \mathscr{A})$ would influence the local density of point i, specially, when the points are on the boundary of the clusters. To reduce the influence, we use relatively stable local density (see 3.1) to substitute local density used by LGD.

Definition 3.1. (Relatively Stable Local Density). The relatively stable local density of point i is defined as

$$\rho_r = \frac{|\mathscr{N}_i^k \setminus \{p_m\}|}{\bar{\mathscr{V}}_i^{k-1}}, \tag{3.1}$$

where p_m is the farthest from point i, and $\bar{\mathscr{V}}_i^{k-1} = \sum\limits_{p_j \in \mathscr{N}_i^k \setminus \{p_m\}} w_{i,j}/\left|\mathscr{N}_i^k \setminus \{p_m\}\right|$

In order to see that the relationship of individual and local points more accurately in the data with the densities of components(clusters) are very different and vary greatly, we normalize the relatively stable local density in k-neighborhood.

Definition 3.2. (Complete Local Gap Density). Complete local gap density is the global normalization in the k-neighborhood including the point i itself. The complete local gap density of p_i is defined as

$$\varphi_{c_i} = \frac{\rho_{r_i}}{\max\{\max\{\rho_{r_n} | p_n \in \mathcal{N}_i^k\}, \rho_{r_i}\}}, \tag{3.2}$$

where ρ_{r_n} is the relatively stable local density of point n.

Like LGD, we set a artificial density threshold γ to distinguish the core points and the border points. Differently, we take the complete local gap density as the measure, so the $\gamma \in (0,1]$ but LGD is not. Otherwise, finite interval is very helpful for us to find the appropriate threshold. The operation of the creating core structure is executed in the K-NN graph. To further impact the core structure, we adopt the stricter conditions to delete these cross-cluster edges.

Definition 3.3. (Ambiguity Cross-Cluster Edges). Let $e_{i,j}$ be the edge connecting point i to point j in the K-NN graph. The edge $e_{i,j}$ is defined the ambiguity cross-cluster edges when it satisfies one of the following two conditions

(1)$p_i \in \mathscr{B}$ or $p_j \in \mathscr{B}$ (The same to LGD);

(2)$p_i \notin \mathscr{B}, p_j \notin \mathscr{B}, \mathcal{N}_i^k \cap \mathscr{B} \neq \varnothing, \mathcal{N}_j^k \cap \mathscr{B} \neq \varnothing$ and $w_{i,j} \geq min\{w_{i,u}, w_{j,v}\}$,

where $\mathscr{B}$ is the set of border points, and $w_{i,u} = min\{w_{i,m} | p_m \in \mathcal{N}_i^k \cap \mathscr{B}, w_{j,v} = min\{w_{j,m} | p_m \in \mathcal{N}_j^k \cap \mathscr{B}\}$.

Now, the enhanced definitions have been introduced and we take advantage of foundation framework of LGD to create the core structure all the same. The algorithm of creating the core structure is described as follows.

Algorithm 1 The algorithm of creating the core structure

```
Input: A dataset $X = \{x_i\}_{i=1}^n$; the parameters $k$ and $\gamma$; the number of clusters $c$.
Output: $c$ core structures.
Build a k-NN graph from X.
Calculate complete local gap density of the points in X by (3.2)
Detect the core points and border points by comparing all complete local gap density with $\gamma$.
Delete the ambiguity Cross-Cluster Edges in the K-NN graph to obtain some subclusters.
Put these subclusters in order from the largest to the smallest according the number of their points.
Pick the first $c$ subclusters as the $c$ core structures.
```

3.2. **Expansion of core structures.** The expansion process is taking the core structures as the centers and outspreading in arbitrary directions according to the step-size until some radius.

Definition 3.4. (Expansion step size) $\mathscr{C}$ is the set of points and it includes the all points in the core structures. l_i is an edge weight in the core structure, which are inherited from the K-NN graph. The expansion step size is defined as

$$s = \frac{\sum_{i=1}^k l_i}{\xi |\mathscr{C}|}, \tag{3.3}$$

where k is the total number of edges, ξ is a coefficient, and $\xi = 100, 110, 120.......$

Definition 3.5. (Expansion radius) The expansion radius is defined as

$$\mathrm{R} = \alpha \frac{\sum_{i=1}^{k} l_i}{|\mathscr{C}|}, \tag{3.4}$$

where α is a multiplying factor, and $\alpha = 3, 4, 5......$

The vertical distance that the Core-Structures extend outward once is equal to the step size s, and the distance from core structure to margin is equal to the expansion radius R. They are decided by the character of the dataset. Experimental results show that the algorithm perform well when $s = 150$ and $\mathrm{R} = 3$. When the expansion process is over, the process of clustering is over too. Every point but noise points will be absorbed by certain core structure. So our algorithm can denoise and deal with the dataset whose densities of components vary greatly and some clusters' margin is very close. The whole algorithm is described as follows.

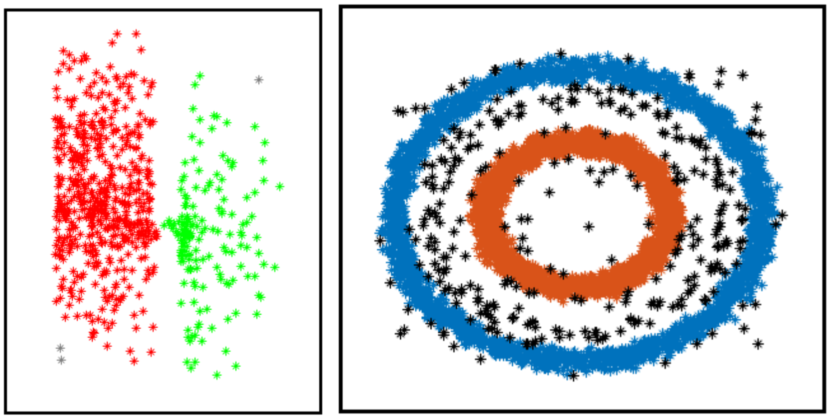
Algorithm 2 Density-based core-structure expansion for clustering

```
Input: A dataset $X = \{x_i\}_{i=1}^{n}$; the parameters $k$ and $\gamma$; the number of clusters $c$.
Output: Clustering result $\{\mathscr{C}_i\}_{i=1}^{c}$.
Obtain $c$ core structures by algorithm.1.
Expand the core structure as the step size $s$ till the radius $\mathscr{R}$
if Point $i$ doesn't belong arbitrary core structure do
    Regard point $i$ as a noise.
end if
```

4. Experiments

To test the effectiveness of our algorithm, the following datasets were selected for the experiments. We used a biological dataset ecoli, two image datasets gisette and USPS and a text dataset Vote. To be more precisely, we made two artificial datasets SyntheticNear and SyntheticCircle to give a visual analysis. The two artificial datasets can be see in Figure 4.1. We can see that the densities within a cluster and among the clusters vary greatly and the protruding parts of the two clusters are very close together in Figure 4.1(a). This is designed to prove that our algorithm can distinguish the clusters, which are very closed together and cluster the data whose densities vary greatly. And Figure 4.1(b) is designed to prove that our algorithm can distinguish the noise on the basis of varying densities. The above datasets are detailed in the following Table 1. Because our algorithm is based on partition clustering and density clustering, we select some relevant algorithm such as K-means, DBSCAN, ReCon-DBSCAN, DPC, SNN, GDL, GDPC, and LGD to compare.

4.1. **Parameter setting.** Because the result of K-means clustering varies with the initialization, we chose the one that worked best for comparison. For DBSCAN, we set the parameter $MinPts \in \{2, 3, 5, 7, 10, 15, 20, 40\}$, and the setting of parameter ε is as follows. Let D be the set of the Euclidean distances of pairwise points of the data. We sort the elements of the D from small to large, i.e., $d_1 < d_2 < d_3 < ... < d_{n^2}$, where n is the number of the data. Then, we set $\varepsilon = d_{\mathscr{Y}=(n^2 p/100)}$, where $\mathscr{Y}(.)$ is the round function, and p is selected from $[1 : 0.2 : 10]$, namely, p varies from 1 to 10, and each interval is 0.2. For ReCon-DBSCAN, we set the parameter $Ratio \in [1 : 0.1 : 2]$, the parameter $threshold \in [0.1 : 0.1 : 1]$, and the



(a)

(b)

FIGURE 4.1. Two artificial datasets: (a) SyntheticNear; (b) SyntheticCircle

TABLE 1. Detailed description of the datasets

Dataset	Instances	Dimensions	Classes
SyntheticNear	687	2	2
SyntheticCircle	5250	2	2
ecoli	336	343	8
gisette	7000	5000	2
Vote	435	16	2
USPS	9298	256	30

setting method of parameter ε is the same with the parameter ε in DBSCAN. For SNN, we set the parameter k as 15 or 25, $MinPts \in \{3,5,7,9,12\}$, and $\varepsilon \in \{6,8,10\}$. For GDL, we set the parameter $a \in 10^{[-3:0.5:3]}$. For DPC, we set the parameter d_c like the parameter ε in DBSCAN. For GDPC, we set $k \in \{3,5,7,10\}$, $\lambda \in \{10,40,80\}$,and $\gamma \in \{5,10,20\}$ For LGD, we set $\tau \in [0.25:0.2:0.8]$ and $k \in \{3,5,7,9,10,15\}$. If the size of the dataset is large, then $k \in \{10,20,30,40,...\}$. The last one is our algorithm, we set the τ and k as the same with LGD, and set the expansion radius to three times the expansion step.

4.2. **Experimental result and analysis.** We use datasets SyntheticNear and SyntheticCircle to verify our algorithm in the form of visualization, and give the accuracy of contrast algorithms. We use the clustering evaluation metric called clustering accuracy (ACC) to measure the clustering performance. Let q_i be the clustering results, and let p_i be the true label of x_i. ACC is defined as

$$ACC = \frac{\sum_{i=1}^{n} \delta\left(p_i, \operatorname{map}\left(q_i\right)\right)}{n}, \tag{4.1}$$

where $\delta(x,y) = 1$ if $x = y$; otherwise $\delta(x,y) = 0$. $\operatorname{map}\left(q_i\right)$ is the best mapping function that permutes clustering labels to match the true labels using the Kuhn–Munkres algorithm. In clustering, a high ACC means a good clustering result.

We select K-means, DBSCAN and LGD algorithms to give the visualized analysis in Figure 4.2 and Figure 4.3. K-means algorithms can only handle convex datasets, so it failed like Figure 4.2(a)(e). DBSCAN can only handle the uniform densities, so it failed, too. LGD has the ability

TABLE 2. Accuracy comparison of algorithms

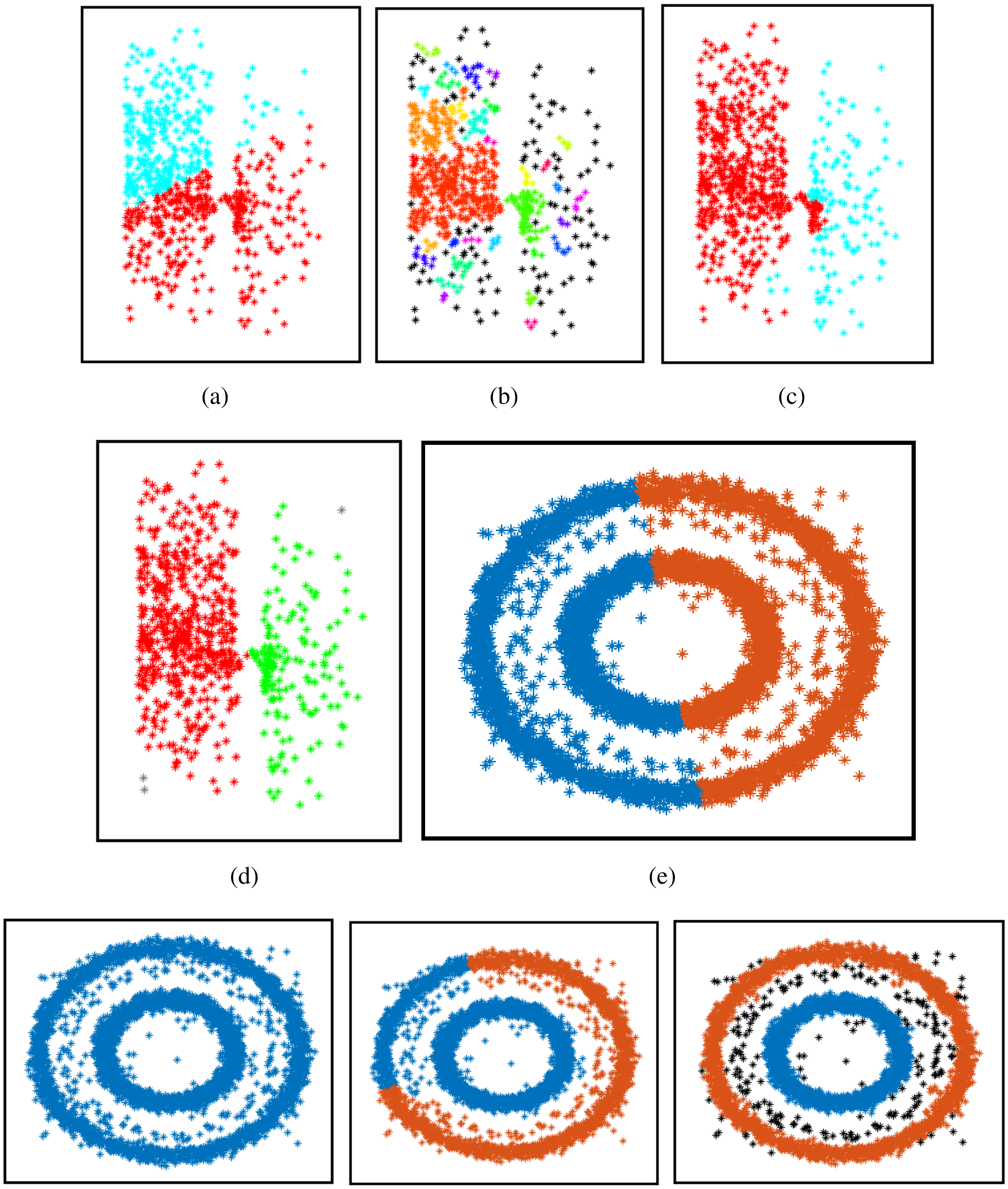
Dataset	k-means	DBSCAN	ReCon-DBSCAN	DPC	SNN	GDL	GDPC	LGD	DBCSE
SyntheticNear	0.422	0.201	0.562	0.466	0.628	0.633	0.706	0.788	**0.974**
SyntheticCircle	0.208	0.488	0.620	0.562	0.403	0.702	0.703	0.749	**0.968**
ecoli	0.674	0.606	0.671	0.552	0.700	0.649	0.698	0.765	**0.792**
gisette	0.401	0.558	0.713	0.505	0.512	0.500	0.784	0.835	**0.911**
Vote	0.806	0.506	0.701	0.856	0.728	0.623	0.870	0.634	**0.876**
USPS	0.633	0.322	0.440	0.418	0.499	0.742	0.865	**0.942**	0.938

to cluster the varying data, but like Figure 4.3(b), LGD choose the K-NN graph of the red area as the initial cluster instead of the black one. So the black area will be treated as the remained points. They will be clustered in the next turn of assignment of remained points. But the assignment of remained points is to cluster points to the nearest high density point (high density point means the point whose densities is higher than the remained points or they have the same densities). Because the densities of the black area are only less than the left area and more than the right area in Figure 4.3(b) and the distance between the black area and the left one is closer than the one that the distance between the black area and the right area , the black area will be clustered to the left area. The analysis coincides with the result in Figure 4.3(c). Just to make the results general, we change the $k = 7, \tau = 0.48$ in Figure 4.3(a)(b)(c) to $k = 30, \tau = 0.37$ in Figure 4.3(d)(e)(f). The initial cluster situation of the right portion is just the opposite, and the black area become the initial cluster. But the distance between red area and yellow area is closer because they have the semblable densities. So the red area is clustered to the yellow area. The analysis coincides with this result in Figure 4.3(f).

Our algorithm (DBCSE) starts to cluster by building core structures in a high-density area and expand the core structures to complete the remained clustering. The points which cannot be clustered by core structures expansion process will be detected noise. Experiments show that our algorithm is effective on clustering data with varying greatly.

5. CONCLUSIONS

In order to build the core density structures, we reinforce the density conditions of LGD. And then we take core density structures as the center to expand till all points are clustered. For the synthetic dataset or the UCI dataset, our algorithm performs better than the others. This also proves that our improvement is feasible. In the future, we will further investigate how to reduce the input parameter.



(a) (b) (c)

(d) (e)

(f) (g) (h)

FIGURE 4.2. Clustering result. (a)(e)k-means; (b)(f)DBSCAN; (c)(g)LGD; (d)(h)OURS

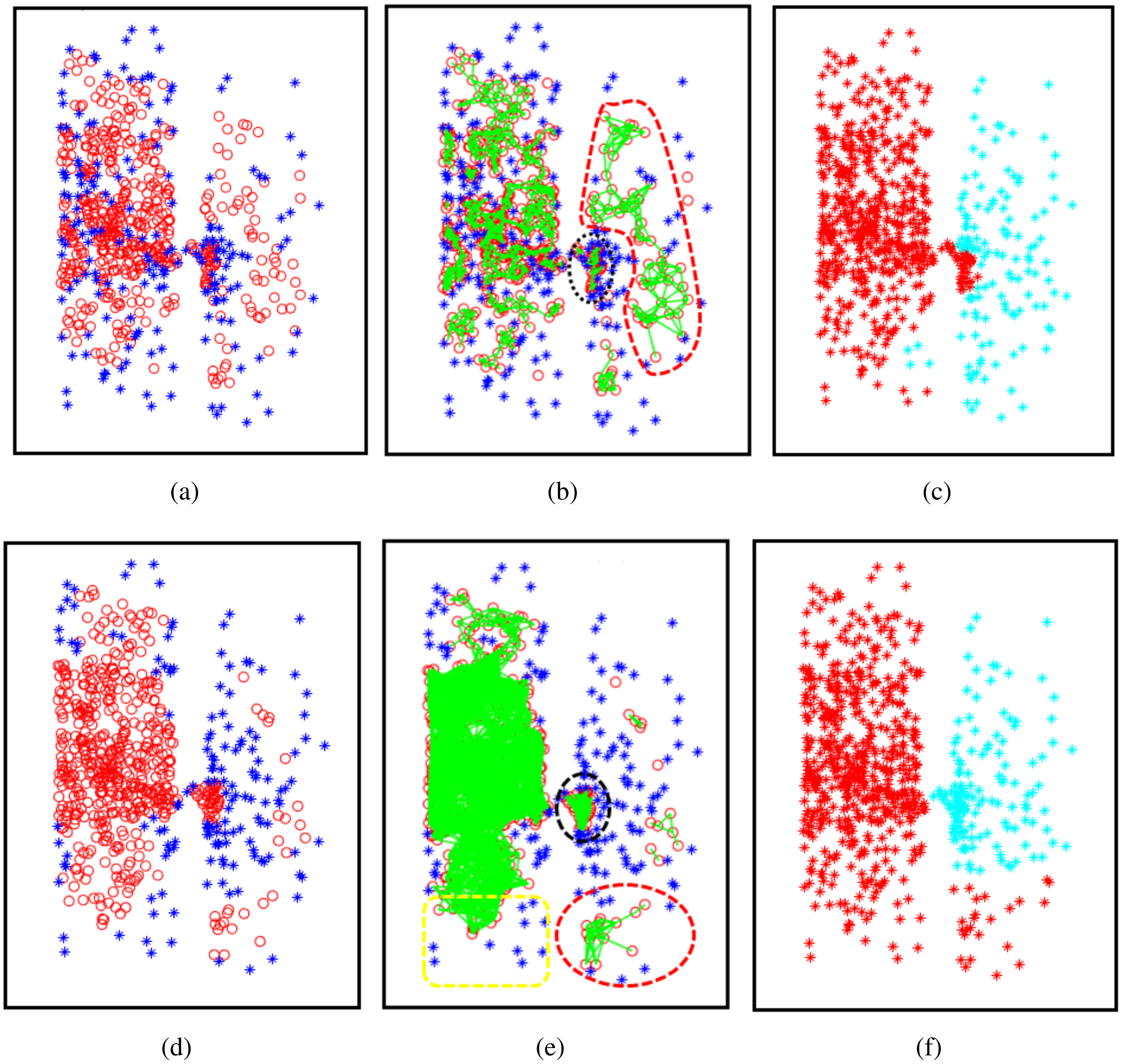


FIGURE 4.3. Analysis of clustering results

REFERENCES

[1] H. Q. Truong, L. T. Ngo, W. Pedrycz, Granular fuzzy possibilistic c-means clustering approach to dna microarray problem, Knowledge-Based Systems 133 (2017), 53-65.

[2] G. H. Ball, D. J. Hall. Isodata, A novel method of data analysis and pattern classification, Stanford Research Institute, 65 (1965), 79-94

[3] G. Andrade, G. Ramos, D. Madeira, et al. G-dbscan: A GPU accelerated algorithm for density-based clustering, Procedia Comput. Sci. 18 (2013), 369-378.

[4] D. Ruppert, The elements of statistical learning: Data mining, inference, and prediction, J. Amer. Stat. Ass. 99 (2004), 567-567.

[5] A. Hinneburg, D. A. Keim, An efficient approach to clustering in large multimedia databases with noise, Proceedings of the 4th International Conference on Knowledge Discovery and Datamining (KDD), pp. 58-65, 1998.

[6] W. R. Fox, Finding groups in data: An introduction to cluster analysis, J. Royal Statistical Soc. Series C 40 (1991), 486-487.

[7] J. E. Dunnage, Random polynomials-probability and mathematical statistics: a series of monographs and textbooks, Bull. London Math. Soc. 56 (1988), 220-243.

[8] Z. Cui, J. Zhang, Y. Wang, et al, A pigeon-inspired optimization algorithm for many-objective optimization problems, Sci. China Info. Sci. 62 (2019), 107-109.

[9] M. N. Tuma, R. Decker, S. W. Scholz, A survey of the challenges and pitfalls of cluster analysis application in market segmentation, Int. J. Market Res. 53 (2011), 391-400.

[10] J. Macqueen, Some methods for classification and analysis of multivariate observations, In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Vo. 1, pp. 281-297, University of California Press, Berkeley, 1967.

[11] S. J. Nanda, G. Panda, Design of computationally efficient density-based clustering algorithms, Data Knowledge Engineering 95 (2015), 23-38.

[12] C. Deng, X. He, J. Han, Document clustering using locality preserving indexing, IEEE Trans. Knowledge Data Engineering 17 (2005), 1624-1637.

[13] H. Frigui, R. Krishnapuram, A robust competitive clustering algorithm with applications in computer vision, IEEE Transactions on Pattern Analysis Machine Intelligence 21 (1999), 450-465.

[14] G. Karypis, E. H. Han, V. Kumar, Chameleon: hierarchical clustering using dynamic modeling, Computer, 32 (1999), 68-75.

[15] M. Iwayama, T. Tokunaga, Cluster-based text categorization: a comparison of category search strategies, International Conference on Research and Development in Information Retrieval, pp. 273-280, 1995.

[16] J. J. Hull, A database for handwritten text recognition research, IEEE Transactions on Pattern Analysis Machine Intelligence, 16 (2002), 550-554.

[17] J. Ho, M. H. Yang, J. W. Lim, et al., Chameleon: A hierarchical clustering algorithm using dynamic modeling, Computer, 32 (2008), 68-75.

[18] M. Ester, H.-P. Kriegel, J. Sander, et al, A density-based algorithm for discovering clusters in large spatial databases with noise, Knowledge Discovery and Data Mining, 96 (1996), 226-231.

[19] W. Zang, L. Ren, W. Zhang, et al., Automatic density peaks clustering using dna genetic algorithm optimized data field and gaussian process, Int. J. Pattern Recognition Artificial Intelligence, 31 (2017): 1750023.

[20] R. Li, X. Yang, X. Qin, et al, Local gap density for clustering high-dimensional data with varying densities, Knowledge-Based Systems, 184 (2019), 104905.

[21] B. Borah, D. K. Bhattacharyya, Ddsc: A density differentiated spatial clustering technique, J. Computers 3 (2018), 72-79.

[22] A. Amini, H. Saboohi, T. Herawan, et al, Mudi-stream: A multi density clustering algorithm for evolving data stream, J. Network Comput. Appl. 59 (2014), 370-385.

[23] M. Xu, Y. Li, R. Li, et al, Eadp: An extended adaptive density peaks clustering for overlapping community detection in social networks, Neurocomputing, 337 (2019), 287-302.

[24] J. Mao, A. K. Jain, A self-organizing network for hyper ellipsoidal clustering (HEC), IEEE Trans. Neural Networks, 7 (1996), 16-29.

[25] R. Ahmed, G. Dalkılıç, M. Erten, Dgstream: High quality and efficiency stream clustering algorithm, Expert Systems with Applications, 141 (2019), 104-123.

[26] X. Yang, Z. Cai, W. Zhu, GDPC: generalized density peaks clustering algorithm based on order similarity, International Journal of Machine Learning and Cybernetics, 12 (2020), 1-13.

[27] B. J. Frey, D. Dueck, Clustering by passing messages between data points, Science, 315 (2007), 972-976.