Optimization of Density Peak Clustering Algorithm Based on OpenMP

Anbo Qiu, Zhuowei Wang*
School of Computer, Guangdong University of Technology, Guangzhou 510006, China

*Corresponding author. Tel: 008618666625197; email: wangzhuowei0710@163.com
Manuscript submitted January 14, 2018; accepted February 26, 2018.
doi: 10.17706/jsw.13.3.168-179

Abstract: The density peak clustering algorithm(CFSFDP) is a new clustering algorithm that implements simple, clustering non-spherical data sets. The algorithm needs artificial selection of clustering center, it is difficult to get the actual clustering centers accurately and can not effectively deal with various data sets. And the density calculation process has nonlinear time complexity. In response to the above problems, a threshold-based parallel optimization CFSFDP (PT-CFSFDP) algorithm is proposed, which sets the threshold for the local density of samples and the distance to the points with higher local density, the sample point is selected as the cluster center when the parameter is greater than the threshold. The distance matrix is optimized in parallel with OpenMP. Experiments show that the PT-CFSFDP algorithm can get the clustering center accurately, the accuracy of the clustering results is up to 94% and the speedup of the algorithm is up to 4.25.

Key words: Density peaks, clustering centers, OpenMP, accuracy, speedup.

1. Introduction

Clustering analysis is referred to as clustering, which is the process of a data object is divided into subsets. Each subset is a cluster so that the objects in the cluster are similar to each other but not similar to the objects in other clusters. Its applications range from computer science to statistics, biology and economics and other disciplines. Currently clustering methods can be roughly divided into several categories: based on the division, based on the density-based, grid-based approach. In the partition-based K-means[1] and K-medoids[2] algorithms, each cluster consists of a set of nearest-neighbor data to their respective cluster centers. The objective function is used to assess the quality of the division, is repeated iterations until the best candidate cluster centers are found. However, because data points are always assigned to the nearest center, these methods are not suitable for non-spherical clusters[3] and they are very sensitive to outliers and noise. In the hierarchy-based clustering, the dataset is decomposed hierarchically according to a certain method, until the specified conditions are satisfied. According to the different classification principles, it can be divided into cohesion and splitting methods, AGNES[4] algorithm and DIANA[5] algorithm are one of the representatives. Such algorithms may not be well scalable because of poor or poorly selected clusters due to merging or splitting. In the density-based clustering algorithm, it is usually assumed that each cluster is generated by a different probability density function, and each sample point is subject to these probability distributions with different weights. The parameters are usually solved iteratively, whereas the EM algorithm is the most commonly used one. The most typical representative of this type of algorithm is the

Gaussian Mixture Model[6]. The accuracy of this method depends on whether the pre-defined probability distribution fits well to the training data.

Density-based clustering algorithms can easily detect clusters with arbitrary shape. Such algorithms assume that the clustering structure can be determined by the tightness of the sample distribution. Usually, the density clustering algorithm considers the connectivities of samples from the viewpoint of sample density and expands the clustering clusters continuously based on the connectable sample to get the final clustering result. In density-based spatial clustering of application with noise(DBSCAN)[7], users can connect adjacent areas with enough density to deal with abnormal data effectively, given the density threshold and area radius as parameters. However, when the density of spatial clustering is not uniform and the clustering distances vary greatly, the clustering quality is poor. In 2014, Alex Rodriguez proposed a simple and efficient clustering algorithm (CFSFDP)[8].The algorithm only needs to calculate the distance once between data points, without worrying about the setting of search radius and density threshold. The algorithm is based on the idea that the local density of clustering centers is higher than that of the surrounding samples and the distance from the points with higher density is far. Similar to DBSCAN, the clustering effect can detect non-spherical clusters. Like the mean-shift method[9], the cluster center is defined as the point with the highest local density in the data points, but it is not necessary to explicitly solve the point with the highest local density for each sample point in the space defined by the kernel function. The algorithm needs to artificially select the clustering center through the decision-making graph, which will make it difficult to obtain the actual clustering center accurately and bring a deviation to the clustering result. Density calculation has nonlinear time complexity $o(n^2)$. When the amount of sample data increases, it will consume a large amount of time to calculate the distance matrix.

Therefore, this paper proposed an improved method to set the threshold for the local density of samples and the distance to the points with higher local density in view of the shortcomings of the CFSFDP algorithm. When the parameters of the sample points are greater than the threshold, it is selected as the cluster center At the same time , the distance matrix is optimized in parallel with the combination of OpenMP. The proposed algorithm is called the parallel optimization of CFSFDP based on threshold setting (PT-CFSFDP). The experimental results show that the PT-CFSFDP algorithm proposed can get the clustering center exactly, and the speedup of the algorithm can reach 4.25 for data.dat.

The rest of this paper is organized as follows. In the next section we introduce the CFSFDP algorithm and its problems in details. And then we give the optimization process of the CFSFDP algorithm. The experimental evaluation of algorithms CFSFDP and PT-CFSFDP are given and the final section presents the conclusion of our work.

2. The CFSFDP algorithm

The CFSFDP algorithm is based on the assumption that the local density of sample points around the cluster center is lower than the cluster centers and the distance between the cluster centers and points with higher local density is relatively large. For each data point i, we calculate two quantities: its local density ρ_i and the distance δ_i from this point to a point with a higher local density. These quantities depend only on the distance d_{ij} between the data points .The local density ρ_i of the data point i is defined as

$$\rho_i = \sum \chi(d_{ij} - d_c) \tag{1}$$

Where d_c is a cutoff distance, d_{ij} is the distance between the data points. $\chi(x)$ is defined as

$$\chi(x)=\begin{cases}1 & \text{if } x<0\\ 0 & \text{otherwise}\end{cases} \tag{2}$$

In fact, ρ_i is equals the number of sample points that are closer than d_c. Calculating the distance between sample points requires calculating a total of $N(N-1)/2$ (N is the size of the data set) distance values. When dealing with large-scale data sets, calculating the distance matrix will consume a large amount of time. Therefore, we calculate the distance matrix in parallel with OpenMP to improve the efficiency of the algorithm.

δ_i is measured by computing the distance between x_i and the nearest sample point with higher density. If ρ_i is the maximum, then δ_i is the farthest distance from x_i.

$$\delta_i=\begin{cases}\min(d_{ij}) & \text{if } \exists j,\rho_j>\rho_i\\ \max(d_{ij}) & \text{otherwise}\end{cases} \tag{3}$$

For the sample points that are local or global mazima in the density, which δ_i is much larger than the δ_j value of other sample points. The former indicates the distance between the sample points with the highest local density and the latter indicates the sample points and their corresponding the distance between the sample points with the highest local density. Therefore, a sample point with a large δ value is most likely a cluster center.

Fig. 1 and Fig. 2 simply explain the core idea of the algorithm.

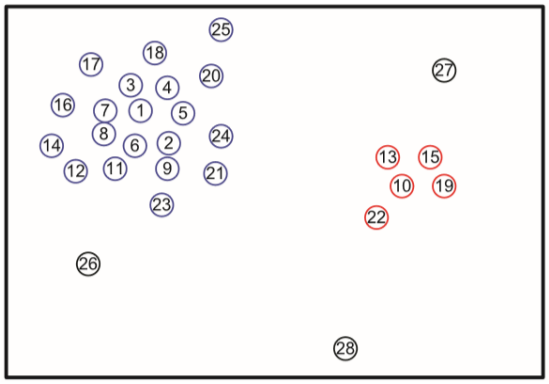


Fig. 1. Point distribution(Date points are ranked in order of decreasing density).

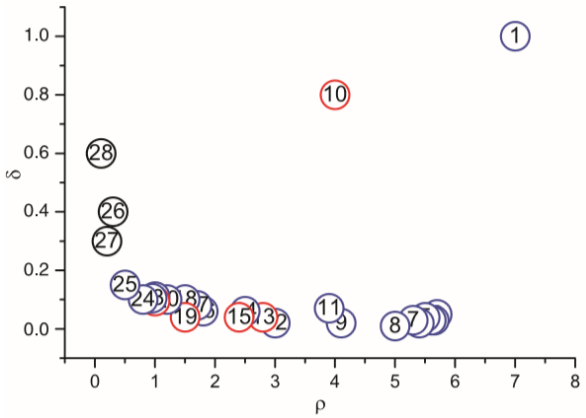


Fig. 2. Decision graph for the data in Fig1(different colors correspond to different clusters).

Fig. 1 shows 28 points are embedded in the two-dimensional space. we can see that the maximum

density is 1 and 10 sample points, which we identify as centers. Figure 2 shows the ρ and δ for the horizontal and vertical coordinates of the decision graph. The value of ρ for points 9 and 10 are very close, with the value of δ are quite different. Point 9 belongs to the cluster of point 1, and the other points with higher ρ are very close to it, while the nearest point with higher density point 10 belongs to another cluster center. Thus, the only points with relatively large ρ and δ are clustering centers. The points 26, 27, and 28 have relatively high δ and are low ρ because they are isolated. They can be thought of as clusters of single points, namely outliers. The author of the article draws the decision-making graph which is determined by the value ρ and the value δ, and then artificially judges the clustering center, so that it is very difficult to obtain the actual clustering center. Figure 3 and Figure 4 show the data decision graph and the different results after artificially choosing the cluster centers according to the algorithm. (Test data set example_distance.dat).

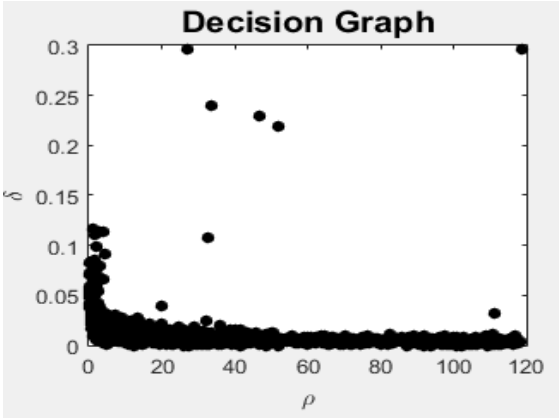


Fig. 3. Data decision graph.

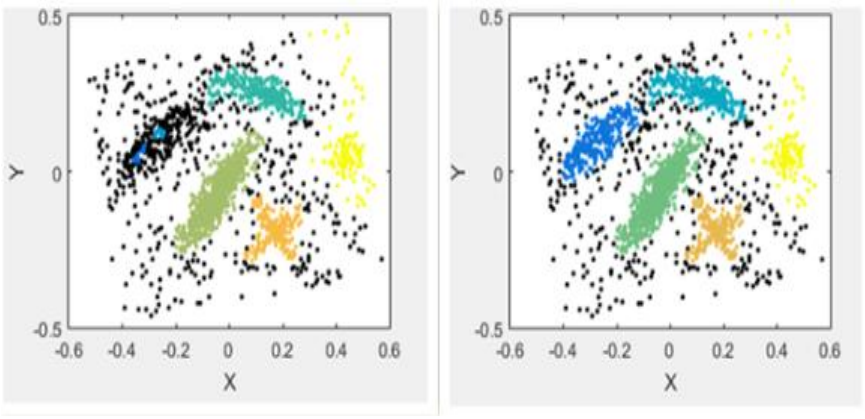


Fig. 4. Different clustering results after artificial selection of cluster centers.

As we can see from Fig. 4, when the clustering center is artificially selected according to the data decision graph, different clustering results will be obtained. In order to avoid this situation, an improved method is proposed to set the appropriate threshold for ρ and δ .It is considered as the clustering center when the parameters of the point are greater than the threshold at the same time, which avoids Artificially selected cluster centers.

After finding the cluster center, each remaining point is assigned to the cluster to which the nearest sample point with higher density that belongs. The choice of cutoff distance d_c will obviously have a great

impact on the clustering result. If d_c is too big, the density of each point will be very close, resulting in all the sample points are divided into the same cluster; if d_c is too small, each cluster contains points will be very small. It is possible that a cluster is divided into several parts. Therefore, it is necessary to select the appropriate d_c such that the average number of neighbors of the points to the total data set is m ($m \in (0,1)$).

3. CFSFDP Algorithm Optimization Process

3.1. Clustering Center Optimization

The CFSFDP algorithm may not get the actual clustering center by artificially selecting the clustering centers. Therefore, this paper introduces thresholds for ρ and δ.

$$th\rho = \lambda_1 * (\max(\rho) - \min(\rho)) + \min(\rho) \tag{4}$$

$$th\delta = \lambda_2 * (\max(\delta) - \min(\delta)) + \min(\delta) \tag{5}$$

where $\max(\rho)$ and $\min(\rho)$ are the maximum and minimum values of the sorted ρ, and $\max(\delta)$ and $\min(\delta)$ are the maximum and minimum values of the sorted δ. we set the appropriate λ_1 and λ_2 based on the actual data set. Table 1 shows different λ_1 and λ_2 based on six data sets. When the sample points meet :

$$\rho_i > th\rho \text{ and } \delta_i > th\delta \tag{6}$$

they are regarded as the center of clustering.

Table 1 λ_1 and λ_2 for different Data Sets

Data set	λ_1	λ_2
Aggregation	0.6	0.2
Flame	0.8	0.2
D31	0.75	0.05
R15	0.6	0.1
Compound	0.5	0.08
Pathbased	0.2	0.4

First of all, the distance between all the points will be sorted from small to large, and the distance between them in a specific position will be taken as the cutoff distance (m × len is selected as the truncation distance, where len is the length of the data set). This method avoids the complexity in the calculation. Then calculate ρ and δ based on equations (1) and (3). Finally, we calculate the threshold of ρ and δ according to the equations (4) and (5), find the cluster centers, and each remaining point is assigned to the same cluster as its nearest neighbor of higher density.

3.2. Parallel Optimization of Distance Matrix

After calculating the distance between points using the Euclidean distance, a distance matrix is produced. Calculating the distance matrix is a very important part of the clustering algorithm so that the density of each point can be calculated later. Because most of the density calculation is a simple matrix, the efficiency of the optimization is not greatly improved, and after testing, the calculation of the distance matrix consumes most of the time of the program. Therefore, this paper mainly focuses on the parallel optimization of the distance matrix. We use OpenMP to optimize the distance matrix function in parallel, assign a thread to each computation task to improve algorithm execution efficiency.

3.3. PT-CFSFDP Algorithm

Specific algorithm flow is as follows:

Step 1: Initialization and preprocessing

(1) Calculate the distance between points d_{ij}.

(2) Give m($m \in (0,1)$)for determining the cut-off distance d_c is given, the cut-off distance d_c is thus determined.

(3) Calculate distance matrix and optimize it in parallel with OpenMP.

(4) Calculate ρ and δ according to equations (1) and (3).

Step 2 Determine the cluster center

(1) Calculate the threshold of ρ and δ according to the equations (4) and (5).

(2) According to the equations (6) to determine the cluster center, and initialize the point classification attribute tag.

Step 3 Assign sample points

(1) Categorize the points in non-clustering centers.

(2) Determine to which the classification tag each point belongs.

The pseudo-code of the entire algorithm flow is shown in Table 2.

Table 2. Pseudocode for PT-CFSFDP Algorithm

PT-CFSFDP Algorithm
Input: clustering data set $S_1 = \{x_i\}_{i=1}^N$
Output: clustering results data set S_2
1.initialize and pretreatment through S_1
calculate the distances d_{ij} between data points
give m for determining the cutoff distance d_c
optimize the distance matrix use openmp
calculate the$\rho$and$\delta$by equations (1) and (3)
2. calculate the threshold by equations(4) and(5)
3. for each i in S_1 do
4. if $\rho_i > th\rho$ and $\delta_i > th\delta$ then
5. assign the sample point of i to the center
6. result[i] ← center
7. center ← center+1
8. initialize the point classification tag $\{c_i\}_{i=1}^N$
9. if $x_i \in result[k]$ then
10. c_i ← k
11. else
12. c_i ← -1
13. end

4. Experiment

4.1. Experimental Environment and Test Dataset

The experimental platform is provided by Guangzhou Supercomputer Center Tianhe II, the experimental environment for Red Hat 4.4.7-4, compiler gcc4.4.7.We use matlab2014 and python 2.6.6 for simulation experiments. The dataset properties for testing are shown in Table 3.

Table 3. Dataset Properties

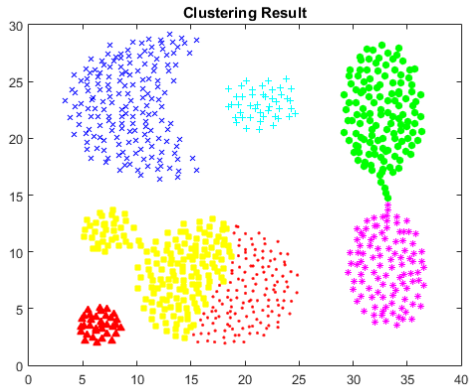
dataset	quantity	dimension	category

Aggregation	788	2	7
Flame	240	2	2
D31	3100	2	31
R15	600	2	15
Compound	399	2	6
Pathbased	300	2	3

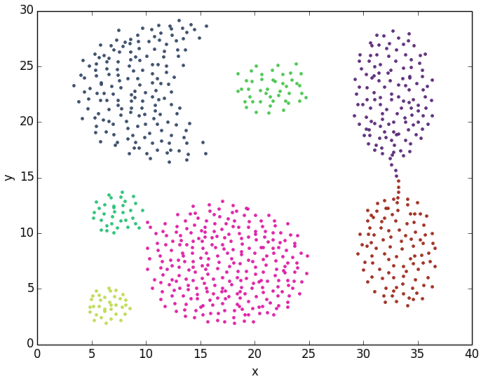
4.2. Accuracy Analysis

4.2.1. The Accuracy of the Selecting of Cluster Center

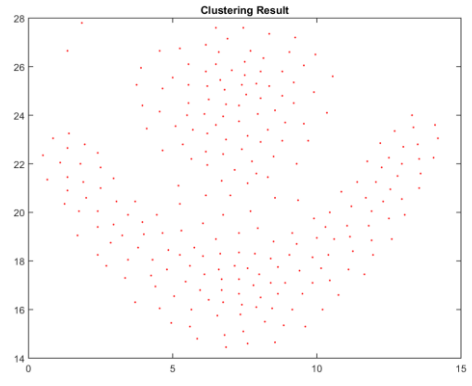
Figure 5 indicates the clustering results of CFSFDP algorithm and PT-CFSFDP algorithm for the above four data sets. Comparing (a) with (b), we can see that the two algorithms have got seven categories, the CFSFDP algorithm is not very accurate for the classification of non-clustering center sample points, but PT-CFSFDP algorithm is very good clustering effect. It can be seen from (c) and (g) and (i) and (k) that for the dataset Flame, R15, Compound and Pathbased only one category, eleven categories, two categories and two categories are obtained respectively using the CFSFDP algorithm, and the clustering result is not satisfactory. The PT-CFSFDP algorithm has been 2 categories ,15 categories, 2 categories and 2 categories in line with the actual situation. It can be seen that the proposed PT-CFSFDP algorithm has a better clustering effect than the CFSFDP algorithm.



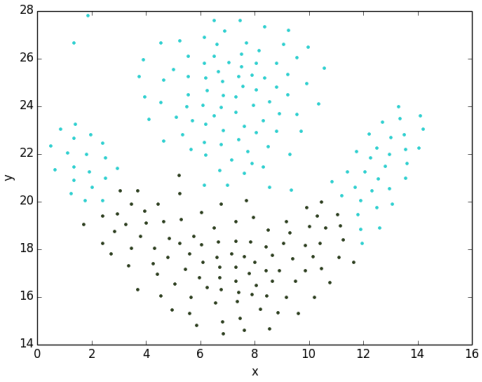
(a) CFSFDP algorithm results (Aggregation)



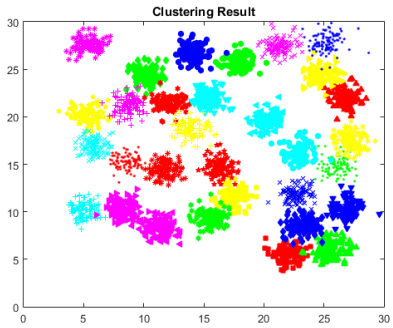
(b) PT-CFSFDP algorithm results (Aggregation)



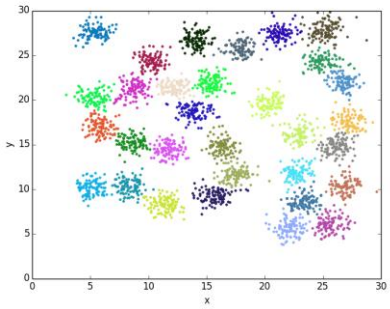
(c) CFSFDP algorithm results (Flame)



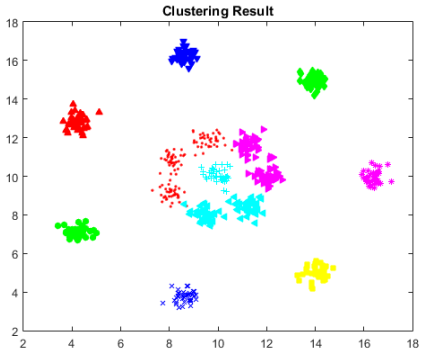
(d) PT-CFSFDP algorithm results (Flame)



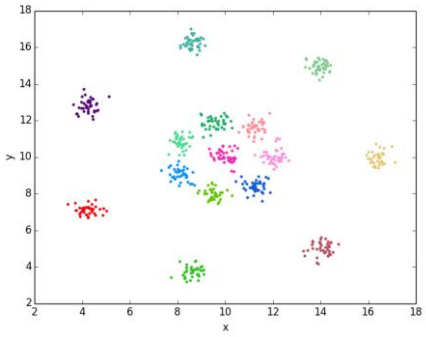
(e) CFSFDP algorithm results (D31)



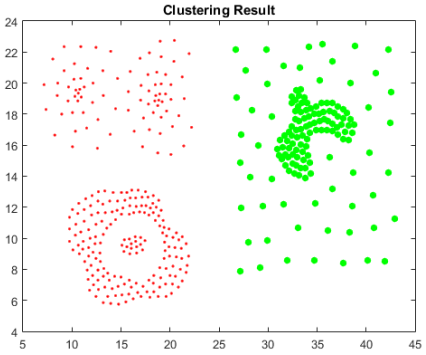
(f) PT-CFSFDP algorithm results (D31)



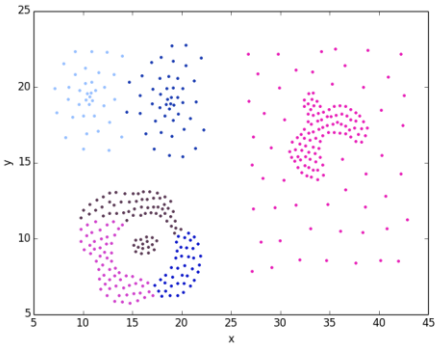
(g) CFSFDP algorithm results (R15)



(h) PT-CFSFDP algorithm results (R15)

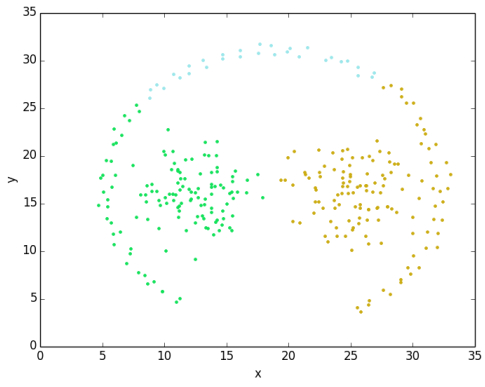


(i) CFSFDP algorithm results(Compound)



(j) PT-CFSFDP algorithm results (R15)

(k) CFSFDP algorithm results(Pathbased)



(l) PT-CFSFDP algorithm results(Pathbased)

Fig. 5. Two algorithms clustering results for four data sets.

Table 4 indicates dataset clustering center selection accuracy using CFSFDP algorithm and the PT-CFSFDP proposed in this paper for six different dataset.

Table 4. Clustering Center Selection Accuracy

	Aggregation	Flame	D31	R15	Compound	Pathbased
category	7	2	31	15	6	3
CFSFDP	7	1	31	11	2	2
PT-CFSFDP	7	2	31	15	6	3
accuracy	100%	50%	100%	73%	33%	67%
	100%	100%	100%	100%	100%	100%

From Table we observe the PT-CFSFDP algorithm proposed in this paper can accurately obtain the clustering centers for all six data sets. The CFSFDP algorithm has good effect on the dataset Aggregation and D31, but the dataset Flame, R15, Compound and Pathbased do not work well.

4.2.2. Clustering accuracy and F-measure

In this paper, we used Precision and F-measure as the evaluation index of clustering results and the datasets given in Table 3 to test. The class i to which the data belongs can be regarded as the item to be queried in the set S_i .The cluster D_k generated by the algorithm can be regarded as the item retrieved in the set S_k . S_{ik} is the number of class i in cluster D_k . The accuracy and recall of class i and cluster D_k are respectively:

$$precision(i, D_k) = \frac{S_{ik}}{S_k} \tag{7}$$

$$recall(i, D_k) = \frac{S_{ik}}{S_i} \tag{8}$$

$$F - measure(i, D) = \frac{(\beta^2 + 1) precision(i, D_k) * recall(i, D_k)}{\beta^2 precision(i, D_k) + recall(i, D_k)} \tag{9}$$

The experiment that measured the accuracy and F-measure of CFSFDP algorithm and PT-CFSFDP algorithm. The results are shown in Figures 6 and 7.

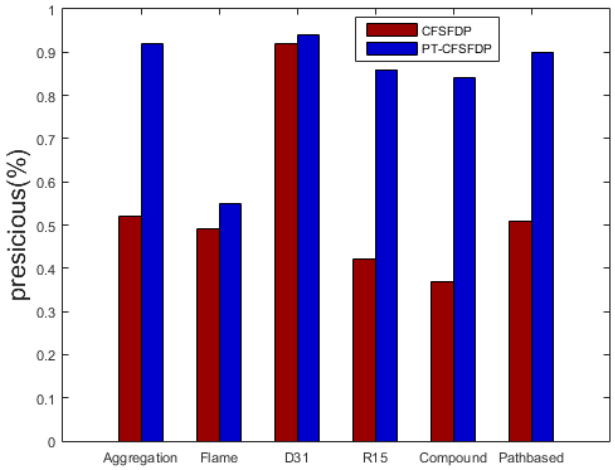


Fig. 6. Presicious.

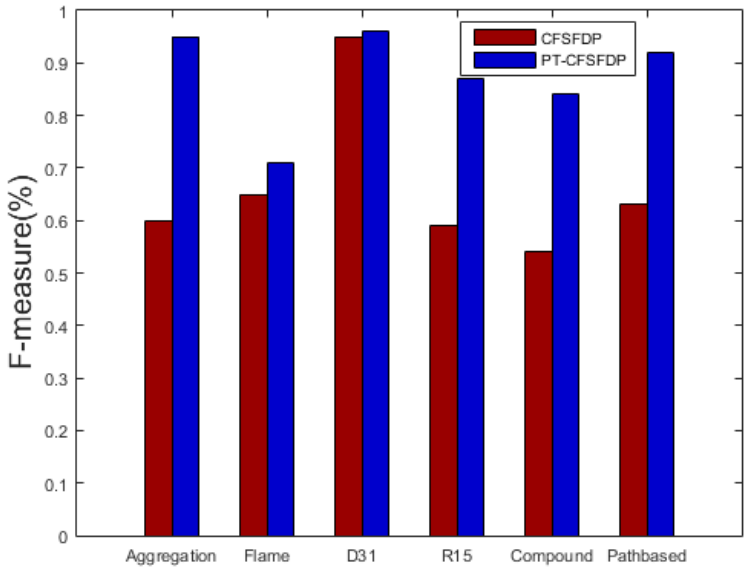


Fig. 7. F-measure.

As can be seen from Figure 6, the accuracy of the PT-CFSFDP algorithm is greatly improved compared to the CFSFDP algorithm for the dataset Aggregation, R15, Compound and Pathbased, and the accuracy of the dataset Flame and D31 is slightly improved. The accuracy is up to 94%. We observe from Figure 7 that the same as the accuracy, the F-measure of CFSFDP algorithm is greatly improved for the dataset Aggregation , R15, Compound and Pathbased, and the improvement effect is not obvious for the other two data sets.

4.3. Distance Matrix Parallel Optimization

Ji C [11]optimized the CFSFDP algorithm from the space complexity, reducing the space complexity of the algorithm from $\Theta(n(n-1)/2)$ to $\Theta(Kn)$, the optimized algorithm can process large dataset clustering. More importantly, the parallel algorithm is load balanced and has high expansibility. This paper optimizes and improves the CFSFDP algorithm from the perspective of time complexity to make up for the lack of research by Ji C et al.

When calculating the distance matrix, multi-threading optimization with OpenMP is performed using the standard dataset data.dat (sample size 3200), and the experimental results are shown in Table 3, with the speedup ratio shown in Fig. 6.

Table 3. OpenMP-based Time Testing

Threads	Time
1	0.0353
2	0.0189
4	0.0109
8	0.0088
12	0.0083
16	0.0238

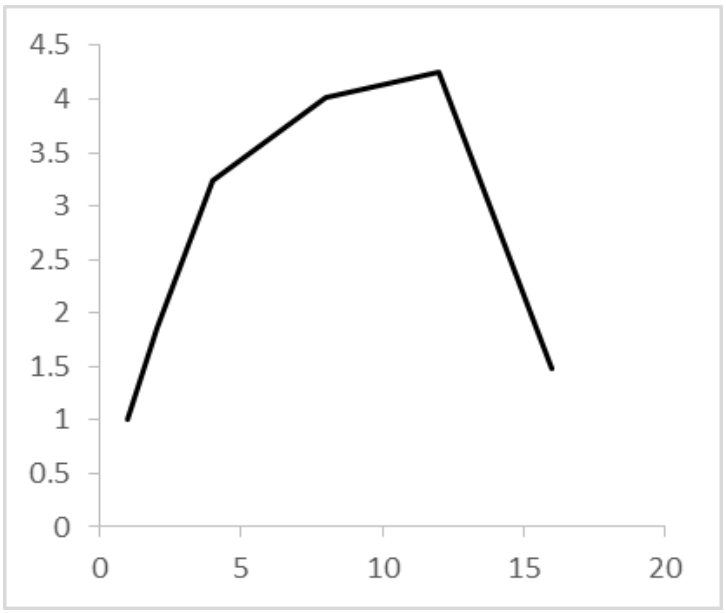


Fig. 6. Speedup ratio.

The experiment uses time as a standard to measure the speedup. From the experimental results, it can be seen that when using OpenMP for parallel optimization, as the number of threads increases, the speedup ratio gradually increases and the running time of the program becomes shorter and shorter. When the threshold is crossed, the communication overhead between the threads starts to become larger , then the efficiency of the program will be reduced. Of course, the number of experimental samples used is limited. We can see that when the number of threads is 12, the optimal speedup is obtained.

5. Conclusion

The proposed algorithm in this paper is based on the lack of CFSFDP algorithm for optimization and improvement. Aiming at the disadvantage of artificially choosing the clustering center of CFSFDP algorithm, this paper set the threshold value for ρ and δ .When the parameter of the point is greater than the threshold, it will be selected the clustering center. The density calculation process has nonlinear time complexity $o(n^2)$. It will cost a large amount of time to compute the distance matrix for big dataset clustering. To solve this problem, in this paper, OpenMP is used to parallelize the calculation of distance matrix. After testing, we found that the maximum speedup up to 4.25 for dataset data.dat. As the dataset increases, the acceleration effect of the algorithm will be more obvious. For six different datasets, the experimental results show that we proposed algorithm has better clustering performance than the CFSFDP algorithm.

Acknowledgment

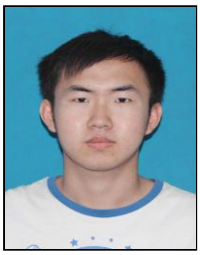
This work was sponsored by National Natural Science Foundation of China (grant number 61300029, 61672168) and Guangzhou Major Science and Technology Projects (201604010096).

References

[1] MacQueen, J. (1967). *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*.

[2] Park, H. S., & Jun, C H. (2009). A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, *36(2)*, 3336-3341.

[3] Jain, A. K. (2010). *Pattern Recognit. Lett*, *31*, 651–666.
[4] Gracia, C., & Binefa X. (2011). On hierarchical clustering for speech phonetic segmentation. *Proceedings of the Signal Processing Conference*.
[5] Umam, K., Bustamam, A., & Lestari D. (2017). Application of hybrid clustering using parallel k-means algorithm and DIANA algorithm. *Proceedings of the Symposium on Biomathematics*.
[6] Douglas, R. (2009). Gaussian mixture models. *Encyclopedia of Biometrics*, 659–663.
[7] Ester, M., Kriegel, H.-P., Sander, J., Xu, X. (1996). P*roceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*.
[8] Alex, R., & Alessandr, L. (2014). Clustering by fast search and find of density peaks. *Science*.
[9] Comaniciu, D., & Meer, P. M. (2002). Shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*.
[10] Chandra, R. (2001). *Parallel Programming in OpenMP*.
[11] Ji, C., Lei, Y. (2017). Parallel clustering by fast search and find of density peaks. *Proceedings of the International Conference on Audio, Language and Image Processing*.



Anbo Qiu was born in Hubei province, China, in 1993. he graduated in software engineering from Wuhan Qingchuan College in 2016. He is a master's student at the Computer Institute of Guangdong University of Technology from September 2016 until the present time. His current scientific research primarily focuses on data mining and high performance computing.

Zhuowei Wang received PhD degree in Wuhan University. She is now associate professor of institute of computer in Guangdong University of Technology. Her research interests focus on high performance computing, low power optimization, distributed systems and etc.