

# 基于多密度峰值的 CFSFDP 算法改进

孙 绵 侯再恩\* 韩肖赞  
(陕西科技大学文理学院 西安 710021)

**摘 要** CFSFDP(Clustering by Fast Search and Find of Density Peaks)算法在单个簇中存在多个密度峰值时,使用决策图难以确定聚类中心数量,导致聚类效果不佳的情况。对此提出将所有密度大于当前位置的数据点以及与当前位置的最小距离各归为一个集合,并对高斯核求得的局部密度排序。当存在多个密度峰值时,只选择第一个点作为聚类中心,同时利用归一化的 $\gamma$ 值分布图确定聚类中心数。人工数据集和 UCI 数据集的数值模拟实验表明,改进 CFSFDP 算法在调整兰德系数、同质性、完整性、V-measure 和标准互信息评分等各指标值均优于 CFSFDP 算法、DBSCAN 算法和 k-means 算法。该算法弥补了 CFSFDP 算法对多密度峰值不能很好聚类的缺陷,适用于对较低维度的任意形数据集的聚类。

**关键词** 聚类 密度峰值 CFSFDP 归一化 集合

**中图分类号** TP3 **文献标识码** A **DOI**:10.3969/j.issn.1000-386x.2019.08.040

## IMPROVEMENT OF CFSFDP BASED ON MULTI-DENSITY PEAKS

Sun Mian Hou Zaien\* Han Xiaoyun  
(School of Arts and Sciences, Shaanxi University of Science and Technology, Xi'an 710021, Shaanxi, China)

**Abstract** When there are multiple density peaks in a single cluster of the CFSFDP algorithm, it is difficult to determine the number of cluster centers using the decision graph, resulting in a poor clustering effect. This paper proposed to classify all the data points whose density was greater than the current position and grouped the minimum distance from the current position into one set, and sorted the local density obtained by the Gaussian kernel. Thus, when there were multiple density peaks, only the first point was selected as the cluster center, and the number of cluster centers was determined using the normalized  $\gamma$  value distribution map. Finally, numerical simulation experiments of artificial data sets and UCI data sets show that the improved CFSFDP algorithm is superior to CFSFDP algorithm, DBSCAN algorithm and k-means algorithm in adjusting rand coefficient, homogeneity, completeness, V-measure, standard mutual information score and other index values. The improved algorithm makes up of the shortcomings of the CFSFDP algorithm that cannot cluster well for multi-density peaks, and it is suitable for clustering of arbitrary data sets with lower dimensions.

**Keywords** Clustering Density peaks CFSFDP Normalization Set

## 0 引 言

基于密度的聚类算法是遵循数据点的密集程度展开聚类,主要针对任意形状的数据集。经典的基于密度的聚类算法有 OPTICS 算法、DBSCAN 算法等,其中,OPTICS 算法存在扩展性低、算法复杂度高的缺陷;DB-

SCAN 算法对输入参数敏感、聚类收敛时间较长等。

针对这些不足,CFSFDP 算法<sup>[1]</sup>作为一种新的密度算法被提出,其对任意形状的数据集尤其是非球形数据集能够有效识别,并且算法仅需设定一个参数,算法结果对参数选取也不敏感<sup>[2]</sup>。同时因为其算法原理简单易实现、聚类效果优而备受关注。经过深入研究,尽管该算法有许多优点,但是也存在一些缺陷有待完

善。文献[3-4]采用簇中心点自动选择策略,避免通过决策图主观确定聚类中心个数带来的误差;文献[5]将模糊的概念用到 CFSFDP 算法来实现聚类中心的自适应选取;文献[6]利用局部反差方法解决了 CFSFDP 算法难以发现稀疏类簇的问题;文献[7-8]采用熵实现了截断距离的自适应,并取得了良好的聚类效果;文献[9-12]提出了一种基于网格的 CFSFDP 算法,该算法利用网格划分来计算局部密度并降低算法的时间复杂度;文献[13]将密度比例引入到 CFSFDP 算法,解决了对类簇间密度较小数据集不能很好聚类的问题,提高了聚类准确率;文献[14]为了克服无法准确聚类多密度峰值的缺陷,提出了一种基于近邻距离曲线和类合并优化的 CFSFDP 算法。

考虑到单个簇中存在多密度峰值时易造成簇的分裂,以及利用决策图无法正确选取存在多个聚类中心个数的问题,本文提出把所有密度大于当前位置的数据点以及与当前位置的最小距离分别做成一个集合,对局部密度进行排序。在存在多个密度峰值时,仅选择第一个点作为聚类中心,并且利用归一化的  $\gamma$  函数数值分布做决策图辅助图选取聚类中心数量。模拟实验结果表明,本文提出的改进 CFSFDP 算法与 CFSFDP 相比,聚类效果更好,更适合于对较低维度的不规则形数据集进行聚类。

## 1 CFSFDP 算法

CFSFDP 算法是一种简洁优美的聚类算法,可识别任意形状类簇,并且仅有一个参数。核心思想是,在聚类中心的选择中,聚类中心本身具有相对高的密度,而密度较大的其他数据点之间的距离相对较大。

算法的基本模型是:

- 1) 计算各数据点的密度以及高局部密度点距离;
- 2) 根据  $\gamma$  函数选取聚类中心;
- 3) 对非聚类中心点进行聚类,确定每个数据点的最终归类。

考虑待聚类的数据集  $S = \{x_i\}_{i=1}^N \subset R^m, m = 1, 2, \dots, n$ , 对于  $S$  中的任何数据点  $x_i$ , 为其定义局部密度  $\rho_i$  和与高局部密度点之间的最小距离  $\delta_i$  两个量。

1) 局部密度  $\rho$ 。计算局部密度有截断核和高斯核两种方式,前者定义如下:

$$\rho_i = \sum_{\substack{i, j \in I_s \\ i \neq j}} \chi(d_{ij} - d_c) \quad (1)$$

式中:

$$\chi(x) = \begin{cases} 1 & x < 0 \\ 0 & x \geq 0 \end{cases}$$

高斯核定义为:

$$\rho_i = \sum_{\substack{i, j \in I_s \\ i \neq j}} e^{-\left(\frac{d_{ij}}{d_c}\right)^2} \quad (2)$$

式中:  $I_s = \{1, 2, \dots, N\}$  为数据集  $S$  对应的指标集,  $d_{ij} = \text{dist}(x_i, x_j)$  表示数据点  $x_i$  和  $x_j$  之间的距离, 参数  $d_c > 0$ , 需提前设定, 通常取所有数据点两两之间的距离经过升序排列后的前 1% ~ 2%。

截断核表示离散值, 高斯核表示连续值。考虑到截断核对于不同数据点易出现具有同一局部密度值的情况, 本文采用高斯核计算方式。

2) 高局部密度点距离  $\delta$ 。设  $\{q_i\}_{i=1}^N$  表示  $\{\rho_i\}_{i=1}^N$  降序排列的下标, 它满足:

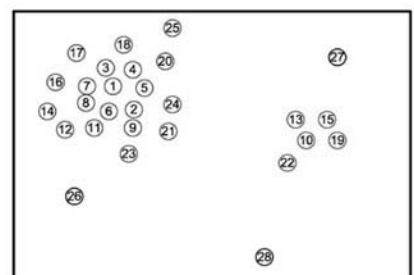
$$\rho_{q_1} \geq \rho_{q_2} \geq \dots \geq \rho_{q_N}$$

则考虑  $\delta_i$  定义如下:

$$\delta_{q_i} = \begin{cases} \min_{q_j, j < i} \{d_{q_i q_j}\} & i \geq 2 \\ \max_{j \geq 2} \{\delta_{q_j}\} & i = 1 \end{cases} \quad (3)$$

由定义可知当数据点  $x_{q_i}$  具有最大局部密度时,  $\delta_{q_i}$  取最大值, 足以保证  $x_{q_i}$  被选为聚类中心。

3) 决策图。文献[2]中对 28 个数据点的原始数据图分布以及关于  $(\rho, \delta)$  做出的决策图如图 1 所示, 决策图对聚类中心的选取起着关键作用。对照图 1(a) 和 (b) 发现, 1 号和 10 号数据点同时有着较大的  $\rho$  和  $\delta$  值, 并且恰好是原始数据集中的聚类中心; 而对于 26、27、28 号数据, 由于其  $\delta$  值很大,  $\rho$  值很小, 在原始数据集中充当离群点。上述聚类中心的选取方案中定性因素太多, 即使是同样的决策图, 不同的人得到的结果也不同, 尤其是对有着多个聚类中心并且个数难以确定的数据, 利用决策图无法正确选出聚类中心。文献[2]中提到综合考虑  $\rho$  和  $\delta$  值, 建立  $\gamma$  函数, 依据  $\gamma_i = \rho_i \delta_i$  的值选取聚类中心, 值越大, 越有可能是聚类中心。并将所有  $\gamma$  值进行降序排列, 取前若干个数据点作为聚类中心即可, 但是若干个并非自动选取, 也是带有一定的主观性。同时, 对于较大的  $\gamma$  值存在  $\rho$  值很大  $\delta$  值很小或者  $\delta$  值很大  $\rho$  值很小的情况, 易误将离群点选作聚类中心。针对这一缺陷, 本文将做出一定改进, 提高聚类中心选取的准确率。



(a)

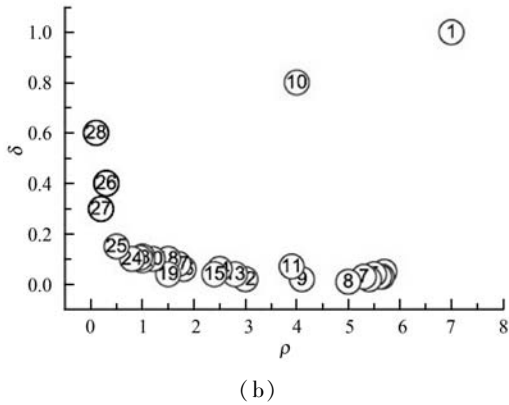


图 1 关于决策图的实例示意图

## 2 CFSFDP 算法的改进

### 2.1 距离 $\delta$ 的确定

对于单个簇中存在多个密度大于自身且距离自身的最小距离  $\delta$  相同的数据点,这些点均易被选为聚类中心,从而造成簇的分裂,以及聚类中心的多选。

针对这一问题,对式(2)所得数据点的密度进行降序排列,形成先后顺序,并且对点密度大于当前点的所有数据点以及与当前点的所有最小距离分别做成一个集合,保证在存在多个密度大于自己且距离自己最近的点中只选择排在前面的第一个点作为聚类中心,避免簇的分裂。

### 2.2 聚类中心的选取

针对 CFSFDP 算法易将数据集中密度大而距离小或者距离大而密度小的数据点选为聚类中心,从而造成聚类中心的错选,文献[8]提出利用信息熵实现截断距离  $d_c$  的自适应,并将得到的最优值设为潜在在聚类中心的距离阈值,避免将密度大而距离小的数据错选为聚类中心。

本文分析了局部密度和最小距离的数值,可知它们是两个不同数量级的。为了避免不同数量级数字之间相互影响,防止大数吃小数等情况,对其进行归一化处理,原始数据经过处理后,各指标处于同一数量级的,使得聚类中心点到非聚类中心点的  $\gamma$  值变化趋势更加明显,尤其是非聚类中心点的  $\gamma$  值变化趋势更加平滑,从而实现聚类中心的正确选取。

文献[2]中提到聚类中心的选取依靠  $\gamma$  函数,综合考虑  $\rho$  和  $\delta$  值,定义如下:

$$\gamma_i = \rho_i \delta_i \quad i \in I_s$$

本文选用 max-min 标准化方法,对  $\rho$  和  $\delta$  原始数据进行线性变换,则  $\gamma$  函数重新定义如下:

$$\gamma_i = \rho'_i \delta'_i = \frac{(\rho_i - \rho_{\min})(\delta_i - \delta_{\min})}{(\rho_{\max} - \rho_{\min})(\delta_{\max} - \delta_{\min})} \quad i \in I_s \quad (4)$$

对有着多个聚类中心并且个数难以确定的数据集,利用决策图无法正确选取出聚类中心。利用上式求出各数据点的  $\gamma$  值,并对各数据点  $\gamma$  值的分布做决策图辅助图,图中从聚类中心到非聚类中心时会有一个明显的跳跃,根据这个趋势可实现聚类中心个数的准确选取。

综上所述,改进 CFSFDP 算法以代价函数取得最小值作为聚类准则:

$$f(\delta_{q_i}) = \sum_{i=1}^N \min \delta_{q_i} = \sum_{i=2}^N \sum_{j=1}^{i-1} \text{mindist}(x_{q_i}, x_{q_j})$$

s. t.  $q_j > q_i, j < i$

式中:  $\{q_i\}_{i=1}^N$  表示密度  $\{\rho_i\}_{i=1}^N$  降序排列的下标,  $\{x_{q_i}\}_{i=1}^N \subset S = \{x_1, x_2, \dots, x_N\}$  表示赋予密度下标的各数据点,  $\{x_{q_1}, x_{q_2}, \dots, x_{q_{(i-1)}}\} \subset \{m_k\}_{k=1}^{n_c}, n_c \leq N$  代表了聚类中心个数。

### 2.3 改进算法实现

算法的具体实现步骤如下:

输入:数据集  $S = \{x_1, x_2, \dots, x_N\}$ , 截断距离参数  $d_c$ 。

输出:数据集  $S$  各样本点的聚类结果

步骤:

- 1) 计算数据集  $S$  中任意两点间的距离矩阵;
- 2) 根据  $t(t=2)$  选择合适的  $d_c$ ;
- 3) 按照式(2)计算各样本点的局部密度  $\rho$ ;
- 4) 根据式(3)计算各样本点的距离  $\delta$ ;
- 5) 把大于当前位置的多个密度值和最小距离各归为一个集合,只选择第一个点作为聚类中心;
- 6) 按照式(4)计算各样本点的  $\gamma$  值;
- 7) 利用各点降序排列的  $\gamma$  值分布图做决策图辅助图选取聚类中心数量;
- 8) 对非聚类中心点完成聚类,确定各点最终的归属。

本文提出的改进 CFSFDP 算法在保持原有算法寻找聚类中心的思想,对  $\gamma$  函数进行了归一化处理,利用各数据点  $\gamma$  值的分布图做决策图辅助图,实现聚类中心个数的选取。并且为了防止簇的分裂,对于同一簇中存在多个密度峰值的簇,只选择第一个高局部密度点计算局部高密度最小距离,避免簇的分裂,也提高了聚类中心选取的准确率。

## 3 数值模拟

为了检验改进算法的聚类效果,本文利用人工数据集和 UCI 数据集进行数值模拟实验,分别对 DBSCAN 算法、k-means 算法、CFSFDP 算法和改进的 CFS-

FDP 算法的实验结果以聚类效果图和各聚类算法的调整兰德系数、互信息评分、同质性、完整性和 V-measure 等评价指标结果对比呈现改进性能。

实验软件和硬件环境是: CPU: Inter (R) Core (TM) i5 - 3470@ 3.20 GHz; 操作系统: Windows10 64 位, 内存: 4 GB; 编程软件: Python 3。

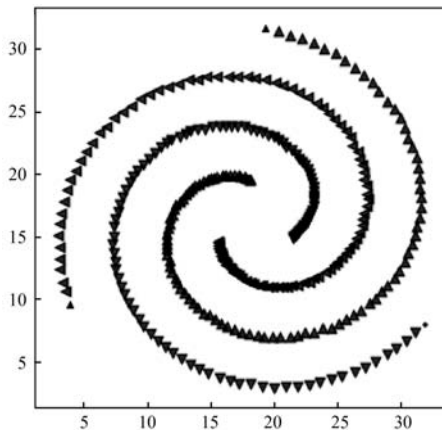
### 3.1 聚类效果图

这部分实验采用数据集 Spiral 和 Aggregation, 具体属性如表 1 所示。

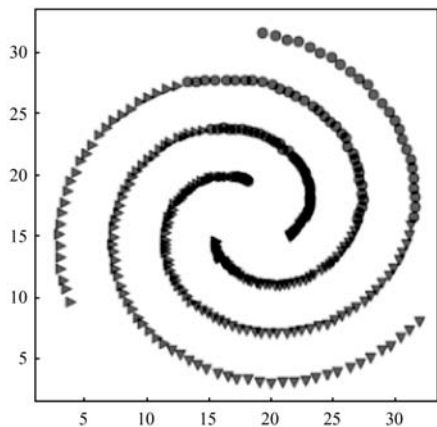
表 1 实验各数据集属性表

数据集	样本点数	类别	维数 m
Spiral	312	3	2
Aggregation	788	7	2

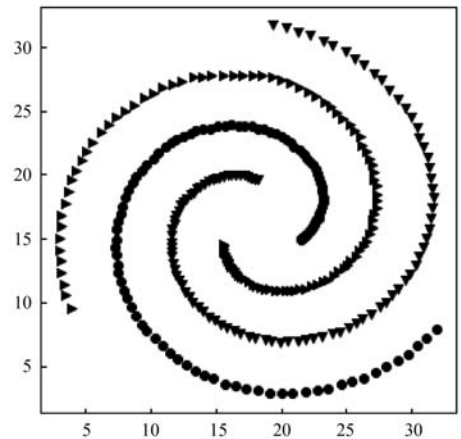
Spiral 数据集分布较为特殊, 呈环状分布, 因此其对算法的参数设置十分敏感, 图 2 是各算法对 Spiral 数据集的聚类效果图。由图 2(a) 可知 DBSCAN 算法因为对参数选取敏感, 故其在聚类过程中产生了少量噪声点; 图 2(b) 中 K-means 算法出现了混乱的聚类结果; 图 2(c) 和 2(d) 所示的 CFSFDP 算法和改进的 CFSFDP 算法因为仅有一个参数, 并且算法对参数的选取有一定鲁棒性, 因此都产生了较好的聚类效果。



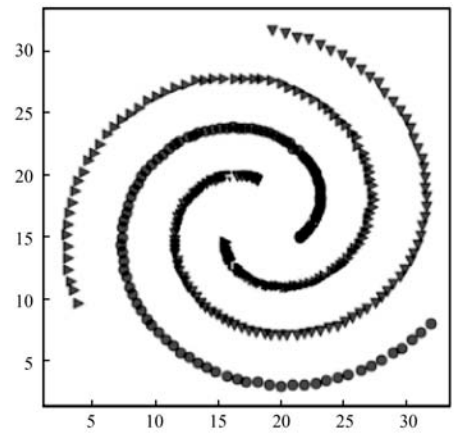
(a)



(b)



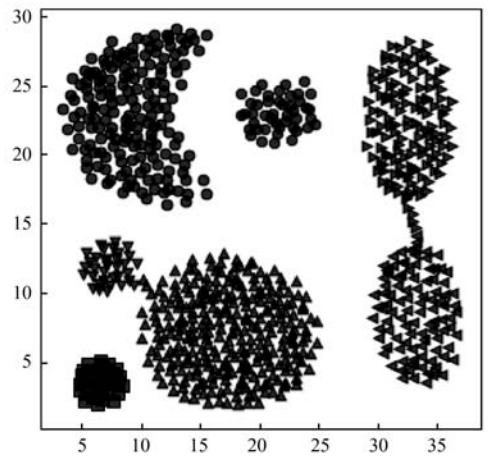
(c)



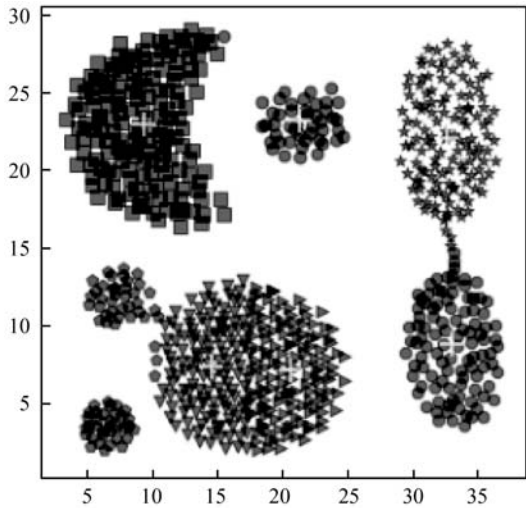
(d)

图 2 Spiral 数据集各算法聚类效果图

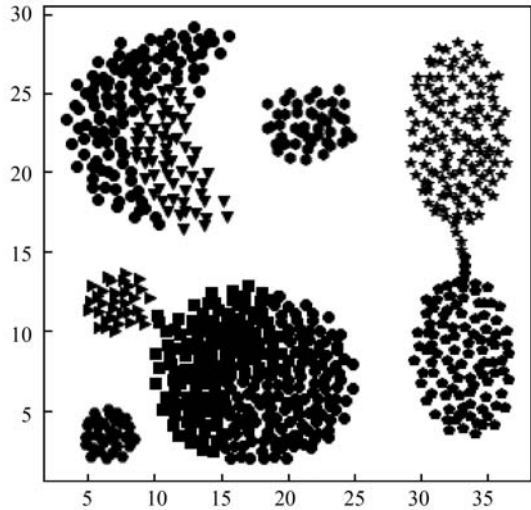
Aggregation 数据集有几个类簇距离较近, 易产生簇的混乱。图 3(b) 和 (c) 分别为 K-means 算法和 CFSFDP 算法的聚类效果图, 聚类效果不佳, CFSFDP 算法由于利用决策图主观性地对聚类中心个数进行选择, 导致聚类中心个数的错选, 使得聚类结果出错; 图 3(a) 和 (d) 所示, DBSCAN 算法和改进的 CFSFDP 算法对数据集的聚类效果良好, 尤其是改进的 CFSFDP 算法利用重新定义的  $\gamma$  函数值做决策图辅助图, 实现了聚类中心的正确选取。



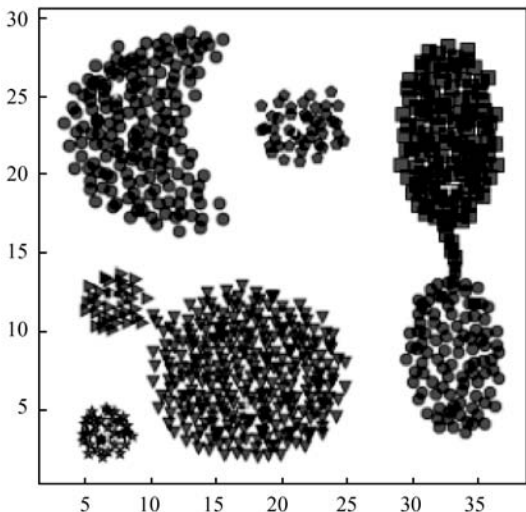
(a)



(b)



(c)



(d)

图 3 Aggregation 数据集各算法聚类效果图

### 3.2 各聚类算法评价指标对比

实验采用人工数据集 R15 和 UCI 数据集 Iris 和 waveform 进行测试,数据的具体属性如表 2 所示。利

用这些数据集主要对各算法的调整兰德系数、同质性、完整性、V-measure 和标准互信息评分等评价指标的结果进行对比。

表 2 实验各数据集属性表

数据集	样本点数	类别	维数 m
Iris	150	3	4
R15	600	15	2
waveform	5 000	3	21

设  $U$  为样本实际类别分配情况,  $V$  为样本聚类后的标签预测情况。各评价指标<sup>[15]</sup>定义如下:

1) 调整兰德系数:

$$ARI = \frac{RI - E|RI|}{\max(RI) - E|RI|}$$

2) 同质性:

$$Homogeneity = H = 1 - \frac{H(C|K)}{H(C)}$$

3) 完整性:

$$Completeness = C = 1 - \frac{H(K|C)}{H(K)}$$

4) V\_measure:

$$V\text{-measure} = 2 \times \frac{H \times C}{H + C}$$

5) 标准互信息评分:

$$NMI(U, V) = \frac{MI(U, V)}{\sqrt{H(U)H(V)}}$$

上述公式中,  $ARI$  测量两个数据分布之间的一致程度,它具有高度的区分性,并且值范围是  $[-1, 1]$ 。  $V\text{-measure}$  是  $H$  和  $C$  的调和平均,取值范围为  $[0, 1]$ ,常用来评估聚类模型的性能好坏。  $NMI$  是  $MI$  的标准化,用来衡量实际类别和预测类别的相似性,取值范围为  $[0, 1]$ ,上述范围越接近于 1 说明聚类结果越好。实验结果如表 3 至表 5 所示。

表 3 各算法在 Iris 数据集的各指标值对比

数据集	评价指标	DBSCAN	K-means	CFSFDP	本文算法
Iris	ARI	0.249 29	0.425 13	0.759 20	0.960 07
	Homogeneity	0.336 72	0.661 28	0.815 65	0.940 50
	Completeness	0.361 73	0.537 41	0.795 98	0.940 39
	V-measure	0.348 77	0.592 94	0.805 69	0.940 44
	NMI	0.349 00	0.596 13	0.805 75	0.940 44

表 4 各算法对 R15 数据集的各指标值对比

数据集	评价指标	DBSCAN	K-means	CFSFDP	本文算法
R15	ARI	0.930 23	0.716 41	0.949 34	0.992 78
	Homogeneity	0.917 71	0.936 51	0.959 70	0.994 24
	Completeness	0.961 63	0.892 68	0.959 45	0.994 21
	V-measure	0.939 15	0.894 25	0.959 58	0.994 23
	NMI	0.939 41	0.895 17	0.959 58	0.994 23

表 5 各算法对 waveform 数据集的各指标值对比

数据集	评价指标	DBSCAN	K-means	CFSFDP	本文算法
waveform	ARI	0.000 47	0.254 11	0.242 16	0.290 48
	Homogeneity	0.002 96	0.366 19	0.348 09	0.346 01
	Completeness	0.001 13	0.364 12	0.210 91	0.330 40
	V-measure	0.001 64	0.365 15	0.262 66	0.338 02
	NMI	0.001 83	0.365 15	0.270 95	0.338 11

Iris 数据集是聚类分析中常用的数据集,由表 3 可知,本文改进的 CFSFDP 算法的各项评价指标都优于其他算法,并且 ARI 的指标值最好,高于 0.95, DBSCAN 算法的聚类效果最差,这是因为此数据集中存在样本分布稀疏的簇,使得 DBSCAN 算法将这些稀疏点作为噪声点处理,导致聚类精度较差。R15 数据集中间的 8 个类簇距离很近,易造成簇混乱的情况,从表 4 的结果来看,改进的 CFSFDP 算法取得了较好的聚类结果,各项评价指标都很优,K-means 算法中的  $k$  是需要提前指定的, $k$  的随机选取易使得算法陷入局部最优,导致聚类结果不理想,造成各项评价指标最低。waveform 数据集共 21 维,有 5 000 个样本点,是一个有着较高维度的大数据集,从表 5 的实验结果来看,改进的 CFSFDP 算法 ARI 指标值虽然是最好的,但是其他指标的值还是略低于 K-means 算法,但是远好于 DBSCAN 算法。

### 3.3 各算法时间复杂度分析

观察到改进 CFSFDP 算法对较高维度的大数据集聚类效果不理想,因此比较和分析各算法的时间复杂度,对上述实验各算法运行时间进行对比分析,结果如表 6 所示。

表 6 各算法运行时间对比分析

实验数据集	DBSCAN	K-means	CFSFDP	本文算法
Spiral	3.629 19	5.448 66	3.345 34	6.194 78
Aggregation	4.510 93	7.482 24	10.324 94	11.008 26
R15	4.393 46	4.532 68	10.861 27	6.284 76
Iris	6.323 00	1.434 77	1.109 95	4.472 14
waveform	230.639 91	25.708 70	257.193 84	185.256 95

从表 6 可以看出,改进 CFSFDP 算法虽然产生了较好的聚类效果,但是其算法运行时间略长于其他算法。这是由于其在具体计算中,需要对数据集中任意两点间的欧式距离进行计算,尤其是对维度较高的大数据集进行计算时,需要完成大量欧式距离的计算,还要完成数据集样本点局部密度和最小距离的计算,因此算法的时间复杂度至少为  $O(N^2)$ 。CFSFDP 算法的时间复杂度是  $O(N^2)$ ,K-means 算法的时间复杂度为  $O(Nkt)$ ,DBSCAN 算法的时间复杂度与数据集的维度有关,上界为  $O(N^2)$ 。

综上所述,改进 CFSFDP 算法的聚类性能整体上优于基本 CFSFDP 算法、K-means 算法和 DBSCAN 算法,该算法非常适合于对较低维度数据集的聚类处理。

## 4 结 语

本文针对 CFSFDP 算法对单个簇中存在多密度峰值时聚类效果不理想的情况,提出将所有密度大于当前位置的数据点以及与当前位置的较近距离各做成一个集合,对局部密度进行排序,在存在多个密度峰值时只选择第一个点作为聚类中心,并且利用各数据点归一化的  $\gamma$  值分布图确定聚类中心数。通过数值分析发现,改进的 CFSFDP 算法适用于对任意形状特别是存在多密度峰值的较低维数据集的聚类。但是其算法运行时间还是略长,如何提升算法运行效率,尤其是对较高维度数据集的聚类将是下一步的研究重点。

## 参 考 文 献

- [1] Rodriguez A, Laio A. Machine learning. Clustering by fast search and find of density peaks [J]. Science, 2014, 344 (6191):1492.
- [2] Rodriguez A, Laio A. Supplementary Materials for Clustering by fast search and find of density peaks[OL]. www.sciencemag.org/content/344/6191/1492/suppl/DC1.
- [3] Ma C L, Ma T, Shan H. A new important-place identification method[C]// IEEE International Conference on Computer and Communications. 2016:151 - 155.
- [4] 马春来,单洪,马涛.一种基于簇中心点自动选择策略的密度峰值聚类算法[J].计算机科学,2016,43(7):255 - 258,280.
- [5] Mehmood R, Bie R, Dawood H, et al. Fuzzy Clustering by Fast Search and Find of Density Peaks[C]// International Conference on Identification, Information, and Knowledge in the Internet of Things. IEEE, 2016:258 - 261.

据中心的虚拟机合并,算法可以建立虚拟机迁移方案,通过利用不充分的主机上的虚拟机迁移和合并降低物理主机的请求量。在此过程中,目标函数设置为最大化主机释放量与最小化虚拟机迁移量。以大量的实验对算法进行了验证,并与两种已有的蚂蚁优化算法作对比,所提算法在多种实验场景下的性能均是较优的。进一步研究算法实际的能效,本文仅从优化主机使用数量与虚拟机迁移次数上着手优化虚拟机合并,但虚拟机的具体部署结果也会对能效产生影响,此时需要计算具体的部署时的能效问题,然后再设计能效更高的部署算法。

## 参 考 文 献

- [1] Dayarathna M, Wen Y, Fan R. Data Center Energy Consumption Modeling: A Survey [J]. IEEE Communications Surveys & Tutorials, 2017, 18(1):732-794.
- [2] Beloglazov A, Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers [J]. Concurrency & Computation Practice & Experience, 2012, 24(13):1397-1420.
- [3] Farahnakian F, Ashraf A, Pahikkala T, et al. Using Ant Colony System to Consolidate VMs for Green Cloud Computing [J]. IEEE Transactions on Services Computing, 2015, 8(2):187-198.
- [4] Corradi A, Fanelli M, Foschini L. VM consolidation: A real case based on OpenStack Cloud [J]. Future Generation Computer Systems, 2014, 32(1):118-127.
- [5] Hwang I, Pedram M. Hierarchical Virtual Machine Consolidation in a Cloud Computing System [C]// IEEE 6th International Conference on Cloud Computing. IEEE, 2013:196-203.
- [6] Ahmad R W, Gani A, Hamid S H A, et al. A survey on virtual machine migration and server consolidation frameworks for cloud data centers [J]. Journal of Network & Computer Applications, 2015, 52(C):11-25.
- [7] Pires F L, Baran B. A Virtual Machine Placement Taxonomy [C]// Ieee/acm International Symposium on Cluster, Cloud and Grid Computing. IEEE, 2015:159-168.
- [8] Farahnakian F, Ashraf A, Liljeberg P, et al. Energy-Aware Dynamic VM Consolidation in Cloud Data Centers Using Ant Colony System [C]// IEEE International Conference on Cloud Computing. IEEE Computer Society, 2014:104-111.
- [9] Feller E, Morin C, Esnault A. A Case for Fully Decentralized Dynamic VM Consolidation in Clouds [C]// IEEE, International Conference on Cloud Computing Technology and Science. IEEE, 2013:26-33.
- [10] Ferdous M H, Murshed M, Calheiros R N, et al. Virtual Machine Consolidation in Cloud Data Centers Using ACO Metaheuristic [C]// 20th European International Conference on Parallel Processing (EuroPar 2014), 2014.
- [11] Hu X B, Wang M, Paolo E D. Calculating Complete and Exact Pareto Front for Multiobjective Optimization: A New Deterministic Approach for Discrete Problems [J]. IEEE Transactions on Cybernetics, 2013, 43(3):1088-1101.
- [12] Ashraf A, Porres I. Using Ant Colony System to Consolidate Multiple Web Applications in a Cloud Environment [C]// Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. IEEE, 2014:482-489.
- [13] Gao Y, Guan H, Qi Z, et al. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing [J]. Journal of Computer & System Sciences, 2013, 79(8):1230-1242.
- ~~~~~
- (上接第 240 页)
- [6] Chen B, Kai M T, Washio T, et al. Local contrast as an effective means to robust clustering against varying densities [J]. Machine Learning, 2018(2):1-25.
- [7] 凌文燕,刘冲.一种改进的搜索密度峰值的聚类算法 [J]. 智能系统学报, 2017, 12(2):229-236.
- [8] 杨震,王红军,周宇.一种截断距离和聚类中心自适应的聚类算法 [J]. 数据分析与知识发现, 2018, 2(3):43-52.
- [9] 孙昊,张明新,戴娇,等.基于网格的快速搜寻密度峰值的聚类算法优化研究 [J]. 计算机工程与科学, 2017, 39(5):964-970.
- [10] 王飞,王国胤,李智星,等.一种基于网格的密度峰值聚类算法 [J]. 小型微型计算机系统, 2017, 38(5):1034-1038.
- [11] 王飞. 基于网格的密度峰值聚类算法研究 [D]. 重庆:重庆邮电大学, 2017.
- [12] 杨洁,王国胤,王飞. 基于密度峰值的网格聚类算法 [J]. 计算机应用, 2017, 37(11):3080-3084.
- [13] 高诗莹,周晓锋,李帅. 基于密度比例的密度峰值聚类算法 [J]. 计算机工程与应用, 2017, 53(16):10-17.
- [14] 蒋礼青,张明新,郑金龙,等. 快速搜索与发现密度峰值聚类算法的优化研究 [J]. 计算机应用研究, 2016, 33(11):3251-3254.
- [15] 石胜飞. 大数据分析挖掘 [M]. 北京:人民邮电出版社, 2018:166-228.