

# 控制与决策

Control and Decision

## 基于相互邻近度的密度峰值聚类算法

赵嘉, 姚占峰, 吕莉, 樊棠怀

引用本文:

赵嘉, 姚占峰, 吕莉, 等. 基于相互邻近度的密度峰值聚类算法[J]. *控制与决策*, 2021, 36(3): 543–552.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0795>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 基于复杂昂贵仿真的体系效能多目标优化

Complex and expensive simulation based multi-objective optimization to system-of-system effectiveness

*控制与决策*. 2021, 36(3): 589–598 <https://doi.org/10.13195/j.kzyjc.2019.0844>

### 基于KPCA和G-G聚类的多元时间序列模糊分段

Fuzzy segmentation of multivariate time series with KPCA and G-G clustering

*控制与决策*. 2021, 36(1): 115–124 <https://doi.org/10.13195/j.kzyjc.2019.0849>

### 基于稀疏度阶数优化的杂波密度估计算法

A clutter density estimation algorithm by optimized sparsity order

*控制与决策*. 2020, 35(12): 2923–2930 <https://doi.org/10.13195/j.kzyjc.2019.0429>

### 基于社交网络的双知识表达分类方法

Double knowledge representations based classification method from perspective of social networks

*控制与决策*. 2020, 35(11): 2653–2664 <https://doi.org/10.13195/j.kzyjc.2019.0141>

### 基于搜索空间划分与Canopy K-means聚类的种群初始化方法

Population initialization based on search space partition and Canopy K-means clustering

*控制与决策*. 2020, 35(11): 2767–2772 <https://doi.org/10.13195/j.kzyjc.2019.0358>

# 基于相互邻近度的密度峰值聚类算法

赵嘉<sup>1,2,3†</sup>, 姚占峰<sup>1</sup>, 吕莉<sup>1,2</sup>, 樊棠怀<sup>1,2,3</sup>

(1. 南昌工程学院 信息工程学院, 南昌 330099; 2. 江西省水信息协同感知与智能处理重点实验室, 南昌 330099; 3. 鄱阳湖流域水工程安全与资源高效利用国家地方联合工程实验室, 南昌 330099)

**摘要:** 密度峰值聚类算法对密集程度不一数据的聚类效果不佳, 样本分配过程易产生连带错误. 为此, 提出一种基于相互邻近度的密度峰值聚类算法. 所提算法引入  $k$  近邻思想计算局部密度, 以此保证密度的相对性. 定义综合数据全局和局部特征的样本相互邻近度的度量准则, 据此准则, 提出一种新的样本分配策略. 新的分配策略采用  $k$  近邻思想寻找密度峰值, 将密度峰值的  $k$  个近邻点分配给其对应类簇, 对所有已分配数据点寻找相互邻近度最高的未分配数据点, 将未分配数据点分配给已分配数据点所在类簇. 在合成和 UCI 数据集上, 将所提算法与 DPC、DBSCAN、OPTICS、AP、 $K$ -Means 及 DPC 的改进算法进行比较, 实验结果表明, 所提出的算法性能最优.

**关键词:** 密度峰值聚类; 相互邻近度; 分配策略;  $k$  近邻

中图分类号: TP301.6

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0795

开放科学(资源服务)标识码(OSID):



引用格式: 赵嘉, 姚占峰, 吕莉, 等. 基于相互邻近度的密度峰值聚类算法[J]. 控制与决策, 2021, 36(3): 543-552.

## Density peaks clustering based on mutual neighbor degree

ZHAO Jia<sup>1,2,3†</sup>, YAO Zhan-feng<sup>1</sup>, LYU Li<sup>1,2</sup>, FAN Tang-huai<sup>1,2,3</sup>

(1. School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China; 2. Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang 330099, China; 3. National-Local Engineering Laboratory of Water Engineering Safety and Effective Utilization of Resources in Poyang Lake Area, Nanchang 330099, China)

**Abstract:** Density peaks clustering has poor clustering effect on data with different densities, and the problem of the ripple may occur in the sample allocation process. To solve the problems, a new clustering method, density peaks clustering based on mutual neighbor degree, is proposed. Firstly, the method introduces  $k$  nearest neighbors to calculate local density, so as to ensure the relativity of the density. Then, it defines the metrics of mutual neighbour degree with global and local features of data. We propose a new allocation strategy according to the metrics. The strategy adopts  $k$  nearest neighbors to find the density peaks, assigns the  $k$  nearest neighbors to their corresponding clusters, searches the unallocated data points with the highest mutual neighbor degree for all allocated data points, and then assigns the unallocated points to the cluster where the allocated data points are located. The experiments on synthetic and UCI datasets show that the proposed method outperforms DPC, DBSCAN, OPTICS, AP,  $K$ -Means and other DPC variant algorithms.

**Keywords:** density peaks clustering; mutual neighbor degree; allocation strategy;  $k$  nearest neighbors

## 0 引言

聚类是一类重要的数据挖掘技术,它依据对象的相似性度量准则将给定的数据集划分为具有共同特征的类簇,相似度高的对象分配在一组,而相似度较低的对象隶属于不同的分组. 聚类在数据分析、数据挖掘、模式识别和机器学习等领域有着广泛应用.

由于聚类的重要作用,学者们提出了众多的聚类算法. 最常用的基于划分的聚类方法是  $K$ -means<sup>[1]</sup>, 它通过迭代寻优类簇中心. 这种方法实现简单,多数情况下都能获得不错的聚类效果,但该方法要求用户指定类簇数,并且很难发现非凸形簇. BIRCH (balanced iterative reducing and clustering using

收稿日期: 2019-06-05; 修回日期: 2019-08-13.

基金项目: 国家自然科学基金项目(51669014, 61663029, 62069014, 62066030); 江西省杰出青年基金项目(2018ACB21029); 江西省自然科学基金项目(20192BAB207031).

责任编辑: 阳春华.

†通讯作者. E-mail: zhaojia925@163.com.

hierarchies)<sup>[2]</sup>是一种基于层次的聚类算法,该算法只需扫描一遍数据集即可完成聚类,且能够发现离群点.但该算法对具有高维特征的数据集或数据集的分布为非凸时,聚类效果不理想. DBSCAN (density-based algorithm for discovering clusters in large spatial databases with noise)<sup>[3]</sup>是一种典型的基于密度的聚类算法,它可以发现任意形状类簇.但该算法需要设置邻域半径 $\varepsilon$ 和核心点的最少近邻样本数 $\text{Minpts}$ ,且不同的参数设置对聚类结果影响显著.近邻传播算法 (affinity propagation, AP)<sup>[4]</sup>通过迭代不断更新每一个点的吸引度和归属感,从而找到类簇中心.该算法无需指定类簇数,但算法的时间复杂度较高,对海量数据聚类时,需消耗较长时间.基于网格的STING (statistical information grid)算法<sup>[5]</sup>将数据空间划分成有限个单元,所有的处理都以单元为对象.该算法聚类速度快,但聚类精度不高.

Rodriguez等<sup>[6]</sup>于2014年提出快速搜索和寻找密度峰值的聚类 (clustering by fast search and find of density peaks),简称密度峰值聚类 (density peaks clustering, DPC)算法. DPC算法基于如下假设:1) 类簇中心被类簇中其他密度较低的数据点包围;2) 类簇中心间的距离相对较远. DPC算法的优点主要包括:不需要事先指定类簇数;能够发现非球形类簇;只有一个参数需要预先取值.

DPC算法定义的局部密度未考虑数据内部的结构差异,当类簇间的数据密集程度差异较大时, DPC算法并不能获得较好的聚类效果;另外, DPC算法的样本分配策略存在分配连带错误.为此,本文提出一种基于相互邻近度的密度峰值聚类 (density peaks clustering based on mutual neighborhood degrees, DPC-MND)算法. DPC-MND算法采用 $k$ 近邻思想计算样本局部密度和寻找密度峰值;将密度峰值的 $k$ 个近邻点分配给其对应类簇;基于样本的相互邻近度,对所有已分配数据点寻找相互邻近度最高的未分配数据点,将未分配数据点分配给已分配数据点所在类簇,直至完成聚类过程.仿真实验结果表明, DPC-MND算法能够有效解决密集程度不一数据的聚类效果不佳和样本分配过程出现的分配连带错误问题.

## 1 DPC算法

DPC算法是一种基于密度的聚类方法,它既能找到各类簇的类簇中心并有效聚类,又能够发现离群点. DPC算法包括两个重要概念:1) 数据点 $i$ 的局部

密度 $\rho_i$ ;2) 数据点 $i$ 的相对距离 $\delta_i$ .

对于给定数据集 $X_{N \times M} = [x_1, x_2, \dots, x_N]^T$ ,其中 $x_i = [x_{i1}, x_{i2}, \dots, x_{iM}]$ ,  $N$ 为样本个数,  $M$ 为样本维数.对于样本点 $i$ 的局部密度, DPC算法给出了两种度量方式:截断核和高斯核.截断核度量方式为

$$\rho_i = \sum_{i \neq j} \chi(d_{ij} - d_c).$$

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

高斯核度量方式为

$$\rho_i = \sum_{i \neq j} \exp \left[ - \left( \frac{d_{ij}}{d_c} \right)^2 \right]. \quad (2)$$

其中: $d_{ij}$ 为数据点 $i$ 与数据点 $j$ 的欧氏距离, $d_c$ 为数据点的邻域截断距离.

截断核定义的局部密度指样本的截断距离范围内存在样本点的总个数;高斯核定义的局部密度指所有样本点到该点的高斯距离之和.文献[6]指出,聚类较大规模数据集,截断核方式效果较好,而聚类小规模数据集,高斯核的效果更为明显.

相对距离 $\delta_i$ 指样本点 $i$ 与密度比它高且距离它最近样本的距离.在计算样本点 $i$ 前需要对每个数据点的局部密度进行排序.对于密度最高的样本,相对距离定义为

$$\delta_i = \max_{i \neq j} (d_{ij}); \quad (3)$$

对于其余数据点,相对距离定义为

$$\delta_i = \min_{j: \rho_j > \rho_i} (d_{ij}). \quad (4)$$

由于密度最高的样本不存在比其密度更高的点, DPC认为该点必为密度峰值(类簇中心),人为设定其相对距离为最大值.剩余的密度峰值需要同时满足两个条件:局部密度 $\rho$ 较高,相对距离 $\delta$ 较大.为此,文献[6]通过决策值 $\gamma$ 寻找这类密度峰值,下式给出了 $\gamma_i$ 的定义:

$$\gamma_i = \rho_i \cdot \delta_i. \quad (5)$$

找到密度峰值后, DPC将剩余数据点分配给密度比它高的最近数据点所在类簇,形成多个从密度峰值出发的树状结构,每一个树状结构代表一个类簇.

## 2 DPC算法缺陷分析

与其他基于密度的聚类方法类似, DPC算法也存在缺陷.

1) DPC算法定义的样本局部密度未考虑数据内

部的结构差异,当类簇间的数据密度程度差异较大时,DPC并不能获得较好的聚类效果.理想情况下,每个类簇都存在一个密度峰值,并且这些密度峰值具有密度高和相对距离远两个特征.DPC算法通过局部密度与相对距离的乘积选择密度峰值.当数据集中各类簇的密集程度差异较大时,相对距离对密度峰值的影响将会变小,局部密度对密度峰值的选择将起着决定性的作用.这种情况下,很容易在一个类簇中找到多个密度峰值或寻找到的密度峰值不正确,这是人们不想看到的.

如图1所示,使用DPC定义的两种局部密度定义方式均无法找到Jain数据集正确的密度峰值(图中“六角星”代表密度峰值).

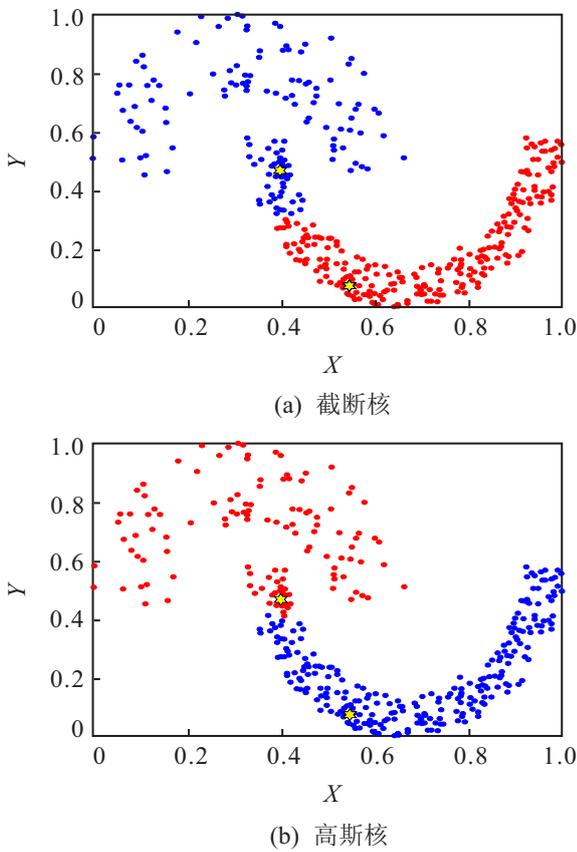


图1 使用高斯核和核截断核对Jain数据集的聚类结果

2) DPC算法的样本分配策略存在分配连带错误.DPC算法的分配策略将所有非密度峰值分配给密度比它高且距离它最近的数据点.这种分配策略容错性差,一旦某一个数据点被错误分配,后续分配将会进一步放大该错误,致使聚类效果变差.如图2所示,DPC算法成功地找到了正确的密度峰值,但由于该数据集环形的类簇较稀疏且一部分数据点与中心的两个类簇距离较近,环形稀疏类簇中的大量数据点被错误地分配给了中心的两个类簇.

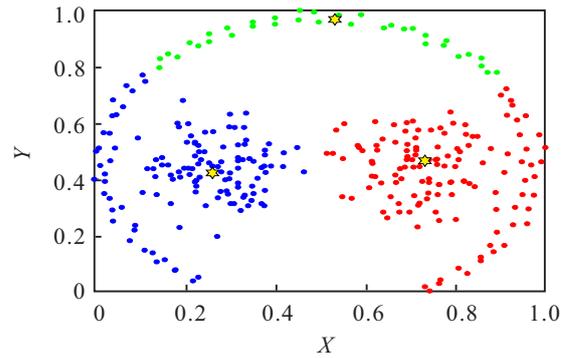


图2  $d_c = 3.8$ 时,DPC对Pathbased数据集的聚类结果

### 3 DPC-MND算法

#### 3.1 局部密度定义

DPC算法的局部密度定义方式对密度峰值的选取有重要影响,一旦密度峰值选择错误,后续操作都将变得毫无意义.通过分析发现,当样本类簇密集程度相差较大时,全局范围内密度较高的点可能均存在于密集类簇中,这将难以发现正确的密度峰值;另外,每个类簇中的密度峰值,它的密度是所在类簇中密度较高的样本,即局部范围内密度较高的样本.因此,样本与其近邻点的相对密度可更加准确地反映该样本是否能成为类簇中心.为了被选为密度峰值,只需要比该类簇中其他样本拥有更高的密度.所以需要计算的是样本局部范围内的相对密度,而不是全局范围内的相对密度.为此,定义样本的局部密度如下.

对于每个数据点 $x_i$ ,其局部密度 $\rho_i$ 定义为

$$\rho_i = \frac{\sum_{j=\text{knn}(i)} \sum_{v=\text{knn}(j)} d_{vj}^2}{2 \cdot k \cdot \sum_{j=\text{knn}(i)} d_{ij}^2} \quad (6)$$

其中: $d_{ij}$ 为数据点 $i$ 与数据点 $j$ 的欧氏距离, $k$ 为近邻点个数, $\text{knn}(i)$ 为数据点 $i$ 的 $k$ 个近邻点的集合, $2 \cdot k$ 为 $\rho_i$ 的归一化参数.该局部密度分为两部分, $\sum_{j=\text{knn}(i)} d_{ij}^2$ 代表 $i$ 点的离群程度, $i$ 点的离群程度越大,该点的局部越稀疏; $\sum_{j=\text{knn}(i)} \sum_{v=\text{knn}(j)} d_{vj}^2$ 代表 $i$ 点局部的疏密程度,该值越大,该局部越稀疏.使用上述公式计算局部密度的优势在于它计算的局部密度为样本局部范围内的相对密度,它将稀疏类簇中样本的局部密度放大,而密集类簇中样本的局部密度缩小,降低了类簇间数据点密集程度不对聚类结果的影响.

#### 3.2 分配策略

一般情况下,类簇边缘点较类簇核心点会更加稀疏,但在样本分布不均匀的数据集上,可能出现类簇

边缘点的某些样本间相距较近,而核心点的某些样本间相距较远的情形.此时依据式(6)计算样本的局部密度,两者分母部分的数值差异较大,而分子部分的数值差异较小,导致类簇边缘点中距离较近的样本局部密度较大,而类簇核心点中距离较远的样本局部密度较小.因此,存在一个类簇中类簇边缘点的局部密度比类簇核心点的局部密度更大的情形,若此时使用DPC算法的分配策略势必会放大DPC的连带错误.为解决此问题,本文定义了一种新的数据点间邻近程度的度量准则,并据此度量准则,提出一种新的样本分配策略.

**定义1(邻近度)** 用数据点间的距离指标定义其邻近度,计算公式如下所示:

$$\omega_{ij} = \begin{cases} e^{-\frac{d_{ij}^2}{\sigma^2}}, & j \in \text{knn}(i); \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

其中

$$\sigma^2 = \frac{1}{N \cdot k} \sum_{i=1}^N \sum_{j \in \text{knn}(i)} d_{ij}^2.$$

$\omega_{ij}$ 为数据点*i*和数据点*j*的邻近度,两数据点距离越远,相似程度越低,邻近度也越小;数据点间的邻近度仅与其*k*个近邻点有关,将样本与距离它较远的样本间的邻近关系完全剔除,减少了较远且无关数据的干扰.由式(7)可知,无论在何种密集程度的类簇中,距离相等的两数据点的邻近度是一样的,该邻近度体现了两数据点在全体数据集中的邻近程度.

本文期望的是,在密集程度不同的类簇中,距离相等的两数据点的邻近程度应是不同的.如在距离相等的情况下,密集类簇中两数据点的邻近程度应该比稀疏类簇中两数据点的低.式(7)定义的邻近度仅考虑了数据的全局信息,而未考虑数据点的局部信息,从而未达到效果.为此,定义数据点的相对邻近度如下.

**定义2(相对邻近度)** 在计算数据点的相对邻近度时,引入两数据点的局部信息,计算公式如下所示:

$$\text{deg}_{i \rightarrow j} = \frac{1}{k+1} \sum_{v \in [\text{knn}(i), i]} \omega_{vj}, \quad i \neq j. \quad (8)$$

**定义3(相互邻近度)** 在数据点相对邻近度的基础上,定义两数据点的相互邻近度如下:

$$A_{i,j} = \text{deg}_{i \rightarrow j} \cdot \text{deg}_{j \rightarrow i}. \quad (9)$$

其中: $\text{deg}_{i \rightarrow j}$ 表示数据点*i*到数据点*j*的相对邻近度,

$\text{deg}_{j \rightarrow i}$ 表示数据点*j*到数据点*i*的相对邻近度, $A_{i,j}$ 表示两个数据点间的相互邻近度.

两数据点的相互邻近度越高,说明两数据点的接近程度越高,联系越紧密.通过分析式(9)可知,两数据点间的相互邻近度与下述因素有关:1) 两数据点间的距离;2) 与全体数据有关,体现了数据集的密集或稀疏程度;3) 与两数据点近邻的局部数据有关,体现了数据点的局部紧密或稀疏程度.满足上述特征计算出来的相互邻近度更能体现数据间的密集或稀疏关系.

新的样本分配策略执行过程包括:首先,采用*k*近邻思想寻找密度峰值,将密度峰值的*k*个近邻点分配给其对应的类簇;其次,计算数据间的相互邻近度,对所有已分配的数据点寻找相互邻近度最高的未分配数据点,将该点分配给已分配数据点所在类簇,循环该操作直至所有数据点分配完毕或所有未分配数据点与已分配数据点的相互邻近度为0.当还有数据点未分配时,采用DPC算法的分配策略对剩余点进行分配.

### 3.3 算法步骤

输入:数据集data,样本邻近数*k*;

输出:聚类结果*C*.

step 1:数据归一化;

step 2:计算数据点间欧氏距离,根据式(6)、(3)和(4)分别计算数据点的 $\rho$ 和 $\delta$ 值;

step 3:根据式(5)计算数据点的决策值 $\gamma$ ,选出密度峰值集合*C<sub>n</sub>*;

step 4:根据式(7)~(9)计算相互邻近度矩阵;

step 5:将密度峰值的*k*个近邻点分配给其对应的类簇;

step 6:对所有已分配数据点寻找相互邻近度最高的未分配数据点,将该点分配给已分配数据点所在类簇;

step 7:若最高相互邻近度不为0,则转至step 6,否则转至step 8;

step 8:若存在尚未分配的数据点,则将这些数据点分配给距离最近且密度比它高的已分配数据点,否则聚类结束.

## 4 实验结果与分析

实验使用经典合成数据集和真实数据集对DPC-MND算法进行测试和评价.将DPC-MND算法与FKNN-DPC<sup>[7]</sup>、DPC<sup>[6]</sup>、DBSCAN<sup>[3]</sup>、OPTICS<sup>[8]</sup>、

AP<sup>[4]</sup>和  $K$ -Means<sup>[1]</sup>及 DPC的改进算法进行比较. 其中: DBSCAN和  $K$ -means 算法参照原文献使用 Matlab 2016a 编程实现, AP在 Python的 Sklearn库中实现; OPTICS在 Python的 PyClustering库中实现; 对于 DPC算法, 基于作者提供的源代码, 但本文的数据集不包含噪声, 因此删除“Halo”部分. 对于 FKNN-DPC算法, 由于无法从原作者处获得源代码, 参照原文献实现该过程. 实验环境均为 Win 10 64bit 操作系统, Matlab 2016a 软件, 4 GB 内存, Intel(R) Core(TM) i5-4200M CPU@2.50 GHz 处理器.

#### 4.1 评价指标

使用 3 个独立于标签绝对值的评价指标评估聚类效果, 分别是调整互信息 (adjusted mutual information, AMI)<sup>[9]</sup>、调整兰德系数 (Adjusted Rand Index, ARI)<sup>[9]</sup>和 Fowlkes-Mallows 指数 (fowlkes-mallows index, FMI)<sup>[10]</sup>. 这 3 个指标的上限均为 1, 该值越接近 1, 表示聚类效果越好.

调整互信息 (AMI) 用来衡量两组数据的吻合程度. 假设  $U$  和  $V$  为  $N$  个数据的两种标签 ( $U$  为实际类别,  $V$  为聚类结果), 则这两种标签的熵为

$$H(U) = \sum_{i=1}^{|U|} P(i) \log(P(i)), \quad (10)$$

$$H(V) = \sum_{j=1}^{|V|} P'(j) \log(P'(j)). \quad (11)$$

其中

$$P(i) = |U_i|/N,$$

$$P'(j) = |V_j|/N.$$

$U$  和  $V$  之间互信息为

$$MI(U, V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} P(i, j) \log \left( \frac{P(i, j)}{P(i)P'(j)} \right). \quad (12)$$

调整互信息 AMI 定义为

$$AMI = \frac{MI(U, V)}{\sqrt{H(U)H(V)}}. \quad (13)$$

调整兰德系数 ARI 定义为

$$ARI = \frac{2(A \cdot D - B \cdot C)}{(B + D) \cdot (A + C) + (C + D) \cdot (A + B)}. \quad (14)$$

Fowlkes-Mallows 指数为

$$FMI = \frac{A}{\sqrt{(A + B) \cdot (A + C)}}. \quad (15)$$

其中:  $A$  表示在  $U$  中为同一类、在  $V$  中也为同一类的

数据点对数,  $B$  表示在  $U$  中为同一类、在  $V$  中不为同一类的数据点对数,  $C$  表示在  $U$  中不为同一类、在  $V$  中为同一类的数据点对数,  $D$  表示在  $U$  中不为同一类、在  $V$  中也不为同一类的数据点对数.

#### 4.2 数据预处理

本实验选用的合成数据集和来源于 UCI 机器学习数据库的真实数据集是广泛应用于测试聚类算法有效性且非常流行的数据集. 这些数据集的样本规模、特征数、类簇数存在较大差异. 表 1 和表 2 详细描述了本文实验所用的数据集.

表 1 合成数据集

数据集	数据来源	样本规模	属性个数	类簇个数
Aggregation	文献 [11]	788	2	7
Flame	文献 [12]	240	2	2
Jain	文献 [13]	373	2	2
Pathbased	文献 [14]	300	2	3
R15	文献 [15]	600	2	15
Spiral	文献 [14]	312	2	3
D31	文献 [15]	3 100	2	31
DIM512	文献 [16]	1 024	512	16
S2	文献 [17]	5 000	2	15

表 2 UCI 数据集

数据集	数据来源	样本规模	属性个数	类簇个数
Iris	文献 [18]	150	4	3
Wine	文献 [18]	178	13	3
WDBC	文献 [19]	569	30	2
Seeds	文献 [20]	210	7	3
Segmentation	文献 [18]	2 310	19	7
Libras	文献 [21]	360	90	15
Ionosphere	文献 [22]	351	34	2
Waveform	文献 [23]	5 000	21	3
Ecoli	文献 [18]	336	8	8
Dermatology	文献 [18]	366	33	6
Parkinsons	文献 [18]	197	23	2
Balance Scale	文献 [18]	625	4	3

在进行聚类实验之前, 需要进行数据预处理. 数据预处理包括填补缺失值和对数据进行归一化. 使用同一维的所有有效数据的平均值代替缺失值. 归一化则采用下式所示的最大最小化方法:

$$x_{ij}^* = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}. \quad (16)$$

其中:  $x_{ij}$  为第  $i$  个样本的第  $j$  个属性的取值,  $x_j$  为所有数据的第  $j$  个属性的集合. 对数据归一化有利于消除不同维度对实验结果的影响.

### 4.3 算法参数选择

为更加客观地测试各算法的聚类性能,对每种算法都进行了参数调优. DPC-MND 及 FKNN-DPC 算法需要设定样本近邻数  $k$ , 在  $1 \sim 50$  之间选出一个最优参数作为样本近邻数  $k$ . DPC 算法存在两种局部密度定义方式, 同时对这两种局部密度定义方式进行实验, 对每个数据集选出最适合的局部密度定义方式. 另外, DPC 算法需要设定截断距离  $d_c$ , 通过实验发现, 文献 [6] 给出的经验法则选择  $d_c$  并不具有普适性, 所以调整这个百分比, 选出最好的参数进行聚类. DBSCAN 和 OPTICS 算法需要设定两个参数: 邻域半径  $\varepsilon$  和邻域内包含的最少样本数  $Minpts$ . 邻域半径  $\varepsilon$  以 0.01 为步长, 在  $0.01 \sim 1$  之间选取; 邻域内包含的最少样本数  $Minpts$ , 在  $1 \sim 50$  之间选取. 由于 AP 算法没有通用的规则来选择参数, 将参数搜索上限设置为最大相似度的几倍, 逐渐缩小搜索范围<sup>[24]</sup>.  $K$ -means 算

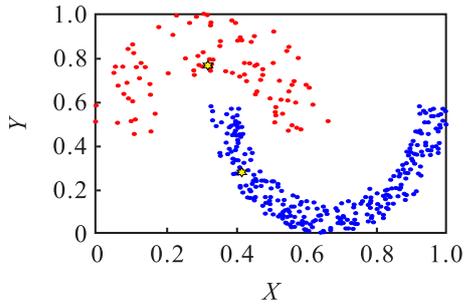
法需要给定类簇个数, 由于  $K$ -means 算法选取的初始类簇中心不同聚类结果也不同, 对每个数据集进行 50 次重复实验, 取其中最好的结果. 虽然 DPC-MND、FKNN-DPC 及 DPC 算法都有确定类簇个数的机制, 但是该机制并不能对所有数据集都正确确定类簇个数, 因此人为指定生成的类簇个数.

### 4.4 合成数据集实验结果分析

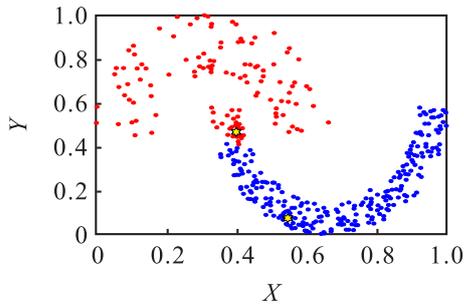
表 3 显示了 7 种算法对表 1 中所有合成数据集的聚类结果 (3 种评价指标 AMI、ARI、FMI 的值) 及各算法的最优参数 Arg-, 表中加粗、加黑的值表示较好的实验结果. 受论文篇幅所限, 图 3 和图 4 仅展示了 DPC-MND、DPC、DBSCAN 和  $K$ -means 算法对 Jain 和 Pathbased 数据集的聚类结果, 图中不同颜色的点代表不同的类簇. 除 DBSCAN 算法以外, 其他算法的聚类中心用“六角星”表示, DBSCAN 算法中的“叉”表示该算法确定的噪声点.

表 3 7 种聚类算法在 9 个合成数据集上的聚类性能

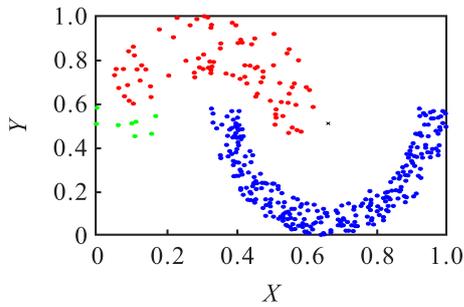
聚类算法	Aggregation				Flame				Jain			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
DPC-MND	0.9955	0.9978	0.9983	28	0.9318	0.9666	0.9846	43	<b>1</b>	<b>1</b>	<b>1</b>	5
FKNN-DPC	0.9775	0.9855	0.9886	20	<b>1</b>	<b>1</b>	<b>1</b>	6	0.0562	0.1318	0.6430	10
DPC	<b>1</b>	<b>1</b>	<b>1</b>	3.4	<b>1</b>	<b>1</b>	<b>1</b>	2.8	0.6183	0.7146	0.8819	0.9
DBSCAN	0.9529	0.9779	0.9827	0.04/6	0.8234	0.9388	0.9712	0.09/8	0.8650	0.9758	0.9906	0.08/2
OPTICS	0.9221	0.9753	0.9807	0.06/10	0.6898	0.8968	0.9508	0.10/8	0.8542	0.9756	0.9905	0.08/1
AP	0.7873	0.7658	0.8150	-1.21	0.4987	0.5403	0.7498	-6.36	0.6582	0.7952	0.9212	-1.77
$K$ -means	0.7935	0.7300	0.7884	7	0.3863	0.4534	0.7364	2	0.4916	0.5767	0.8200	2
聚类算法	Pathbased				R15				Spiral			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
DPC-MND	<b>0.9524</b>	<b>0.9696</b>	<b>0.9797</b>	10	<b>0.9938</b>	<b>0.9928</b>	<b>0.9933</b>	11	<b>1</b>	<b>1</b>	<b>1</b>	1
FKNN-DPC	0.8344	0.8744	0.9165	9	0.9907	0.9892	0.9899	27	<b>1</b>	<b>1</b>	<b>1</b>	5
DPC	0.5212	0.4717	0.6664	3.8	<b>0.9938</b>	<b>0.9928</b>	<b>0.9933</b>	0.6	<b>1</b>	<b>1</b>	<b>1</b>	1.8
DBSCAN	0.8710	0.9011	0.9340	0.08/10	0.9825	0.9819	0.9831	0.04/12	<b>1</b>	<b>1</b>	<b>1</b>	0.04/2
OPTICS	0.4364	0.6364	0.7517	0.06/4	0.9734	0.9785	0.9799	0.04/11	<b>1</b>	<b>1</b>	<b>1</b>	0.04/1
AP	0.5199	0.4775	0.6577	-4.1	0.9907	0.9891	0.9898	-0.17	0.2932	0.1569	0.3409	-0.19
$K$ -means	0.5098	0.4613	0.6617	3	<b>0.9938</b>	<b>0.9928</b>	<b>0.9933</b>	15	-0.0055	-0.006	0.3274	3
聚类算法	D31				DIM512				S2			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
DPC-MND	<b>0.9639</b>	<b>0.9503</b>	<b>0.9519</b>	44	<b>1</b>	<b>1</b>	<b>1</b>	3	<b>0.9461</b>	<b>0.9379</b>	<b>0.9420</b>	38
FKNN-DPC	0.9522	0.9275	0.9298	28	<b>1</b>	<b>1</b>	<b>1</b>	20	0.9180	0.8889	0.8963	22
DPC	0.9554	0.9365	0.9385	0.6	<b>1</b>	<b>1</b>	<b>1</b>	0.6	0.9437	0.9352	0.9395	1.5
DBSCAN	0.8895	0.8078	0.8186	0.04/48	<b>1</b>	<b>1</b>	<b>1</b>	0.36/2	0.8511	0.7485	0.7744	0.04/30
OPTICS	0.8211	0.8673	0.8763	0.03/23	0.9029	0.9432	0.9478	0.19/1	0.6723	0.7713	0.7891	0.03/27
AP	0.8367	0.7425	0.7665	0.23	<b>1</b>	<b>1</b>	<b>1</b>	-1	0.4616	0.5704	0.6080	-3.06
$K$ -means	0.9593	0.9453	0.9470	31	<b>1</b>	<b>1</b>	<b>1</b>	16	<b>0.9461</b>	<b>0.9379</b>	<b>0.9420</b>	15



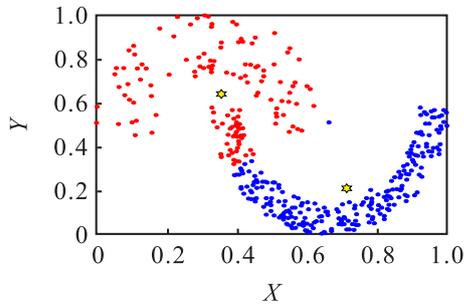
(a) DPC-MND 算法



(b) DPC 算法



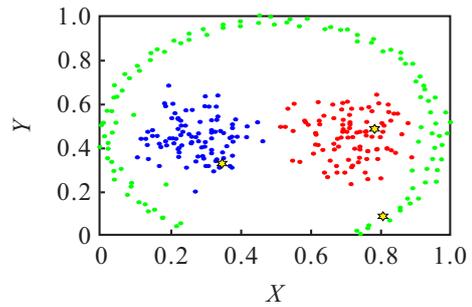
(c) DBSCAN 算法



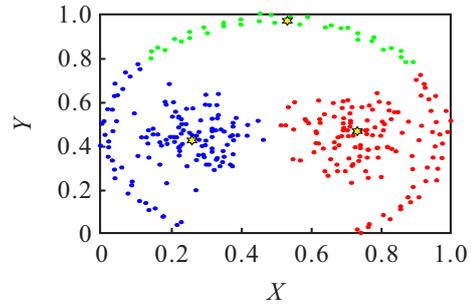
(d) K-mean 算法

图3 4种算法对Jain数据集的聚类结果

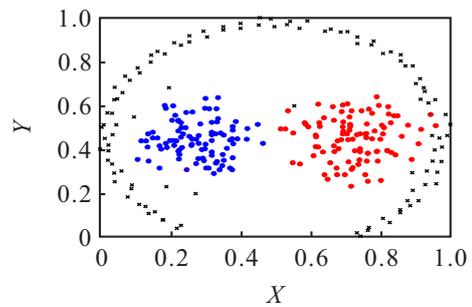
Jain数据集由两个月牙形的类簇组成,这两个类簇相互交错,并且密集程度相差较大.对于由稀疏类簇与密集类簇组成并且存在相互交叠的数据集,从图3可以看出,DPC-MND算法能同时发现正确的密度峰值及其类簇分布,而DPC、DBSCAN和K-means算法均不能对Jain数据集进行准确聚类. DPC算法没有找到正确的密度峰值,使得密集类簇中的部分数据点错误分配.由于存在稀疏类簇,DBSCAN算法错误地将稀疏类簇分为两个类簇,并且将一个较远的边缘点标记为了噪声点.由于该数据集并非球状数据集,K-means算法未能成功聚类.表3的数



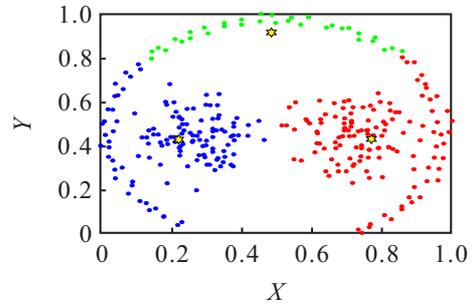
(a) DPC-MND 算法



(b) DPC 算法



(c) DBSCAN 算法



(d) K-mean 算法

图4 4种算法对Pathbased数据集的聚类结果

据表明,仅有DPC-MND算法能对该数据集进行准确聚类,DBSCAN和OPTICS算法能获得较好的聚类结果,其余算法不能对该数据集进行聚类.

Pathbased数据集由3个类簇组成,并且中间的两个密集类簇与四周的环形稀疏类簇距离较近.从图4可以看出,DPC-MND不仅能成功寻找到正确的密度峰值,并且还能准确地将数据点分配给对应的类簇. DPC算法虽然能够成功找出密度峰值,但是在分配剩余数据点时,由于四周稀疏环形类簇的局部密度较低,并且部分数据点与中间密集类簇距离较近,所以环形类簇中离密集类簇较近的点较易被分配给密集

类簇,一旦某个数据点被分配错误,DPC算法分配策略的连带错误效应将被激活,使得后续的数据点被错误分配.由于四周的环形类簇相较于中间的两个类簇较稀疏,DBSCAN算法直接将四周的稀疏数据定义为噪声点.由于 $K$ -means算法仅能处理球状数据集,其不能成功聚类Pathbased数据集.由表3可以看

出,对于该数据集,仅DPC-MND能进行较好的聚类,其余算法的聚类结果均不能令人满意.

#### 4.5 UCI数据集实验结果分析

表4是DPC-MND、KNN-DPC、DPC、DBSCAN、OPTICS、AP和 $K$ -means算法在12个UCI数据集上的聚类结果,加粗、加重的值表示较优的实验结果.

表4 7种聚类算法在12个UCI数据集上的性能

聚类算法	Iris				Wine				WDBC			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
DPC-MND	0.878 6	0.903 7	0.935 4	17	<b>0.888 6</b>	<b>0.914 9</b>	<b>0.943 5</b>	37	<b>0.685 3</b>	<b>0.785 7</b>	<b>0.902 8</b>	10
FKNN-DPC	<b>0.883 1</b>	<b>0.903 8</b>	<b>0.935 5</b>	22	0.803 8	0.799	0.866 7	6	0.356 0	0.400 9	0.765 8	9
DPC	0.860 6	0.885 7	0.923 3	0.2	0.706 5	0.672 4	0.783 5	2	0.000 7	-0.002 8	0.725 7	1.3
DBSCAN	0.569 2	0.612 0	0.729 1	0.12/5	0.548 4	0.529 2	0.712 1	0.50/21	0.358 1	0.478 6	0.757	0.46/38
OPTICS	0.451 3	0.688 6	0.786 8	0.15/5	0.369 8	0.411 9	0.629 6	0.59/7	0.085 6	0.430 5	0.676 7	0.51/65
AP	0.547 9	0.570 1	0.709 9	-0.57	0.333	0.317	0.612 6	-2.02	0.593 6	0.132 2	0.787 9	2.62
$K$ -means	0.733 1	0.716 3	0.811 2	3	0.847 3	0.868 5	0.912 6	3	0.611	0.730 2	0.877	2
聚类算法	Seeds				Segmentation				Libras			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
DPC-MND	<b>0.756 6</b>	<b>0.801 1</b>	<b>0.867 0</b>	7	0.691 3	<b>0.603 5</b>	0.660 4	40	<b>0.564 6</b>	<b>0.373 6</b>	<b>0.432 8</b>	16
FKNN-DPC	0.697 1	0.742 2	0.827 6	9	0.583 0	0.436 7	0.558 1	27	0.475 4	0.318 4	0.397 6	11
DPC	0.729 9	0.767 0	0.844 4	0.7	<b>0.692 7</b>	0.600 4	<b>0.673 0</b>	1.5	0.535 8	0.319 3	0.371 7	0.3
DBSCAN	0.530 2	0.529 1	0.671 1	0.24/16	0.496 5	0.454 3	0.527 7	0.15/2	0.418 3	0.196 5	0.257 0	0.90/2
OPTICS	0.380 2	0.419 0	0.635	0.81/5	0.431 2	0.46	0.536 1	0.15/1	0.137 7	0.082 8	0.212 6	0.59/1
AP	0.446 5	0.393 6	0.693 3	-2.07	0.208 9	0.344 5	0.340 9	1.8	0.148 7	0.205 6	0.197 1	4.31
$K$ -means	0.670 5	0.704 9	0.802 6	3	0.610 2	0.504 9	0.575 8	7	0.523 2	0.309 4	0.361 2	15
聚类算法	Ionosphere				Waveform				Ecoli			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
DPC-MND	0.402 1	0.527 2	0.797 7	44	<b>0.364 1</b>	<b>0.349 8</b>	<b>0.587 6</b>	14	<b>0.630 5</b>	<b>0.728 3</b>	<b>0.802</b>	39
FKNN-DPC	0.131 4	0.132 1	0.584 1	26	0.077 4	0.008 6	0.505	6	0.475 5	0.553 5	0.691 9	9
DPC	0.150 4	0.235 7	0.649 1	0.5	0.326 1	0.269 8	0.529 2	0.1	0.497 8	0.446 5	0.577 5	0.4
DBSCAN	<b>0.594 7</b>	<b>0.722 6</b>	<b>0.874</b>	0.78/9	0.085 6	0.009 7	0.481 3	0.38/5	0.451 6	0.525 5	0.662 3	0.20/22
OPTICS	0.097	0.338 3	0.608 5	0.58/1	0.028 6	0.091 8	0.266 1	0.47/48	0.426	0.664 2	0.751 5	0.23/29
AP	0.136 7	0.077 3	0.513 7	1.92	0.289 1	0.301 4	0.517 8	-2.2	0.533 9	0.490 7	0.613 4	-0.86
$K$ -means	0.129 4	0.177 6	0.605 3	2	0.363	0.253 6	0.503 7	3	0.505 1	0.419	0.554 2	8
聚类算法	Dermatology				Parkinsons				Balance Scale			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
DPC-MND	0.838 9	0.786 9	0.834 1	11	<b>0.272 8</b>	<b>0.391</b>	<b>0.832 2</b>	33	<b>0.142 5</b>	<b>0.198 5</b>	0.514 1	35
FKNN-DPC	0.835 5	<b>0.812 7</b>	<b>0.850 4</b>	35	0.072 8	0.160 1	0.658 2	7	0.035 1	0.023 6	<b>0.554 8</b>	9
DPC	0.784	0.776	0.822 1	1.6	0.247 8	0.125 6	0.618 7	1.2	0.115 4	0.139 4	0.502 4	1.1
DBSCAN	0.572 1	0.416 5	0.539 5	0.99/3	0.007 1	0.025 2	0.577 5	0.50/17	0.090 2	0.139 4	0.151	0.03/1
OPTICS	0.293 4	0.343	0.456 3	0.99/1	0.036 8	0.098 6	0.504 9	0.45/9	0.063 3	0.106 2	0.116 5	0.03/2
AP	0.689 8	0.593 5	0.676 6	-0.84	0.109 8	0.034 3	0.224 6	0.23	0.090 2	0.142	0.155 3	0.97
$K$ -means	<b>0.874 8</b>	0.742 6	0.794 7	6	0.212 9	0.052	0.595 7	2	0.013 2	0.001 5	0.044	3

表4的实验结果表明,处理 Iris 数据集时,DPC-MND算法要略逊于FKNN-DPC算法,但是优于剩余算法.处理 Dermatology 数据集时,DPC-MND算法聚类效果仅比K-means算法差.处理 Segmentation 数据集时,DPC-MND算法聚类效果优于除DPC算法外的其余算法.处理 Ionosphere 数据集时,DPC-MND算法要略逊于DBSCAN算法,但要好于其余算法.对于剩余的8个数据集,DPC-MND算法都要优于与之比较的算法.

#### 4.6 与其他DPC改进算法的实验结果分析

为进一步验证本文算法的性能,将DPC-MND算法与近年提出的各种DPC的改进算法进行比较.实验数据来自各参考文献,由于各文献所列的对比数据集及评价指标各不相同,选择出现频率最高的5个数据集作为实验数据集,选择出现频率最高的ARI作为算法评价指标,见表5,其中“-”表示原文献中并未提供该数据,加粗、加重的值表示较优的实验结果.

表5 9种改进算法在5种数据集上的ARI指标值

聚类算法	Iris	Wine	WDBC	Seeds	Segmentation
DPC-MND	0.904	<b>0.915</b>	<b>0.786</b>	<b>0.801</b>	<b>0.604</b>
ALPC <sup>[25]</sup>	0.561	0.338	0.011	0.462	-
ADPC-DNAGA <sup>[26]</sup>	<b>0.922</b>	0.852	<b>0.786</b>	0.790	-
DPC-GEV <sup>[27]</sup>	0.834	0.302	-	0.713	0.296
DPC-CI <sup>[27]</sup>	0.834	0.285	-	0.713	0.296
ADPC-KNN <sup>[28]</sup>	0.760	0.370	-	0.770	0.250
FN-DP <sup>[29]</sup>	0.904	0.803	0.528	-	0.277
A-IVDPC <sup>[30]</sup>	0.851	0.699	0.743	0.778	-
IVDPC <sup>[30]</sup>	0.851	0.594	0.615	0.778	-

表5的数据表明,DPC-MND算法在各数据集上均获得了较好的聚类结果,仅在Iris数据集上略逊于ADPC-DNAGA算法,WDBC数据集上的聚类效果与ADPC-DNAGA算法相同.处理剩余数据集时,本文算法的聚类效果均优于与之比较的DPC改进算法.

## 5 结论

针对DPC算法存在的不足,本文提出了一种基于相互邻近度的密度峰值聚类算法.该算法引入k近邻思想,计算样本局部密度和寻找密度峰值,采用相互邻近度衡量样本邻近程度,对所有已分配数据点寻找相互邻近度最高的未分配数据点,将未分配数据点分配给已分配数据点所在类簇.DPC-MND算法有效解决了DPC算法样本分配策略中存在的连带错误效应以及数据密集程度差异较大时聚类效果不佳的

问题.通过对多个合成和UCI数据集实验发现,DPC-MND算法可以快速搜索和发现任意大小、任意形状数据集的密度峰值,且算法的聚类性能优异.本文使用k近邻思想,由于k的取值为整数,较 $d_c$ 更容易确定取值.但是,参数k仍然需要人为确定,参数k的自适应确定将是下一步的努力方向.

#### 参考文献(References)

- [1] Macqueen J. Some methods for classification and analysis of multi variate observations[C]. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Palo Alto: University of California Press, 1967: 281-297.
- [2] Zhang T, Ramakrishnan R, Livny M. BIRCH: A new data clustering algorithm and its applications[J]. Data Mining and Knowledge Discovery, 1997, 1(2): 141-182.
- [3] Ester M. A density-based algorithm for discovering clusters in large spatial databases with noise[C]. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. Palo Alto: AAAI Press, 1996: 226-231.
- [4] Frey B J, Dueck D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814): 972-976.
- [5] Wei W, Yang J, Muntz R R. STING: A statistical onformation grid approach to spatial data mining[C]. Proceedings of the 23rd International Conference on Very Large Data Bases. San Francisco: Morgan Kaufmann Publishers Inc., 1997: 186-195.
- [6] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.
- [7] Xie J, Gao H, Xie W, et al. Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors[J]. Information Sciences, 2016, 354: 19-40.
- [8] Ankerst M, Breunig M M, Kriegel H P, et al. Optics: Ordering points to identify the clustering structure[C]. Proceedings of the ACM Sigmod Record. New York: ACM, 1999: 49-60.
- [9] Vinh N, Epps J, Bailey J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance[J]. Journal of Machine Learning Research, 2010, 11(1): 2837-2854.
- [10] Fowlkes E B, Mallows C L. A method for comparing two hierarchical clusterings[J]. Journal of the American Statistical Association, 1983, 78(383): 553-569.
- [11] Gionis A, Mannila H, Tsaparas P. Clustering aggregation[J]. ACM Transactions on Knowledge Discovery from Data, 2007, 1(1): 1-30.

- [12] Fu L, Medico E. FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data[J]. BMC Bioinformatics, 2007, 8(1): 3.
- [13] Jain A K, Law M H. Data clustering: A user's dilemma[C]. Proceedings of the 1st International Conference on Pattern Recognition and Machine Intelligence. Heidelberg: Springer, 2005: 1-10.
- [14] Chang H, Yeung D Y. Robust path-based spectral clustering[J]. Pattern Recognition, 2008, 41(1): 191-203.
- [15] Veenman C J, Reinders M J T, Backer E. A maximum variance cluster algorithm[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(9): 1273-1280.
- [16] Franti P, Virtajoki O, Hautamaki V. Fast agglomerative clustering using a  $k$ -nearest neighbor graph[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(11): 1875-1881.
- [17] Frnti P, Virtajoki O. Iterative shrinking method for clustering problems[J]. Pattern Recognition, 2006, 39(5): 761-775.
- [18] Blake C L, Merz C J. UCI repository of machine learning database[EB/OL]. [2016-12-28](2019-02-03). <http://archive.ics.uci.edu/ml/index.html>.
- [19] Street W N, Wolberg W H, Mangasarian O L. Nuclear feature extraction for breast tumor diagnosis[C]. Proceedings of the IS and T/SPIE International Symposium on Electronic Imaging: Science and Technology. San Jose: SPIE, 1993: 861-870.
- [20] Charytanowicz M, Niewczas J, Kulczycki P, et al. Complete gradient clustering algorithm for features analysis of  $x$ -ray images[J]. Advances in Intelligent and Soft Computing, 2010, 69: 15-24.
- [21] Dias D B, Madeo R C B, Rocha T, et al. Hand movement recognition for brazilian sign language: A study using distance-based neural networks[C]. Proceedings of the International Joint on Neural Networks. Atlanta: IEEE, 2009: 697-704.
- [22] Sigillito V G, Wing S P, Hutton L V, et al. Classification of radar returns from the ionosphere using neural networks[J]. Johns Hopkins APL Technical Digest, 1989, 10(3): 262-266.
- [23] Breiman L, Friedman J, Stone C J, et al. Classification and regression trees[Z]. Boca Raton: CRC Press, 1984.
- [24] Liu R, Wang H, Yu X. Shared-nearest-neighbor-based clustering by fast search and find of density peaks[J]. Information Sciences, 2018, 450: 200-226.
- [25] 邱保志, 程栾. 基于拉普拉斯中心性和密度峰值的无参数聚类算法[J]. 计算机应用, 2018, 38(9): 2511-2514.  
(Qiu B Z, Cheng L. Parameter-free clustering algorithm based on Laplace centrality and density peaks[J]. Journal of Computer Applications, 2018, 38(9): 2511-2514.)
- [26] Zang W, Ren L, Zhang W, et al. Automatic density peaks clustering using DNA genetic algorithm optimized data field and gaussian process[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2017, 31(8): 1750023.
- [27] Ding J, He X, Yuan J, et al. Automatic clustering based on density peak detection using generalized extreme value distribution[J]. Soft Computing, 2018, 22: 2777-2796.
- [28] Yaohui L, Zhengming M, Fang Y. Adaptive density peak clustering based on  $K$ -nearest neighbors with aggregating strategy[J]. Knowledge-Based Systems, 2017, 133: 208-220.
- [29] Du M, Ding S, Xue Y. A robust density peaks clustering algorithm using fuzzy neighborhood[J]. International Journal of Machine Learning and Cybernetics, 2017, 12: 1-10.
- [30] Zhou Z, Gangquan S, Yanbin Z, et al. Robust clustering by identifying the veins clusters based on kernel density estimation[J]. Knowledge-Based Systems, 2018, 159: 309-320.

### 作者简介

赵嘉(1981—), 男, 教授, 博士, 从事机器学习与数据挖掘、计算智能等研究, E-mail: zhaojia925@163.com;

姚占峰(1995—), 男, 硕士生, 从事机器学习与数据挖掘的研究, E-mail: yao1020242939@163.com;

吕莉(1982—), 女, 副教授, 博士, 从事数据挖掘、视频目标跟踪等研究, E-mail: lvli@nit.edu.cn;

樊棠怀(1962—), 男, 教授, 博士, 从事传感器信息获取与处理等研究, E-mail: fantanghuai@163.com.

(责任编辑: 齐 霖)