控制与决策
Control and Decision

混合近邻和多簇合并的密度峰值聚类算法

吕莉, 赵妞, 肖人彬, 王新峰, 韩龙哲

引用本文:
吕莉, 赵妞, 肖人彬, 等. 混合近邻和多簇合并的密度峰值聚类算法[J]. 控制与决策, 2025, 40(7): 2194-2202.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2024.1232

混合近邻和多簇合并的密度峰值聚类算法

吕 莉[1,2,3†]，赵 妞[1,3]，肖人彬[4]，王新峰[1,2,3]，韩龙哲[1,2,3]

(1. 南昌工程学院 信息工程学院，南昌 330099；
2. 江西省水利大数据智能处理与预警技术工程研究中心，南昌 330099；
3. 南昌市智慧城市物联感知与协同计算重点实验室，南昌 330099；
4. 华中科技大学 人工智能与自动化学院，武汉 430074)

摘 要: 密度峰值聚类算法简单、高效, 可识别任意维度和形状类簇, 已在各领域得到广泛应用. 然而, 密度峰值聚类算法也存在一些问题, 如: 对截断距离参数敏感、难以发现低密度区域的类簇中心以及容易产生“多米诺效应”. 为此, 提出混合近邻和多簇合并的密度峰值聚类算法. 首先, 综合考虑样本的全局分布与局部结构, 引入自然近邻与k近邻重新定义局部密度, 消除对截断距离参数的敏感, 并提高低密度区域样本的局部密度以增加类簇中心的识别度; 其次, 将样本划分为多个微簇, 并利用簇间关联度进行合并, 减少距离类簇中心较远的样本的分配错误, 从而有效缓解分配错误连带效应. 使用人工数据与真实数据进行测试, 结果表明, 所提出算法的综合性能优于对比算法.

关键词: 聚类；自然近邻；k近邻；簇间关联度；密度峰值；局部密度

中图分类号: TP301.6 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2024.1232

引用格式: 吕莉, 赵妞, 肖人彬, 等. 混合近邻和多簇合并的密度峰值聚类算法 [J]. 控制与决策, 2025, 40(7): 2194-2202.

Density peak clustering algorithm with mixed nearest neighbors and multi-cluster merging

LV Li[1,2,3†]，*ZHAO Niu*[1,3]，*XIAO Ren-bin*[4]，*WANG Xin-feng*[1,2,3]，*HAN Long-zhe*[1,2,3]

(1. School of Information Engineering，Nanchang Institute of Technology，Nanchang 330099，China；2. Jiangxi Province Engineering Research Center for Intelligent Processing and Early Warning Technology of Water Conservancy Big Data，Nanchang 330099，China；3. Nanchang Key Laboratory of IoT Perception and Collaborative Computing for Smart City，Nanchang 330099，China；4. School of Artificial Intelligence and Automation，Huazhong University of Science and Technology，Wuhan 430074，China)

Abstract: The density peak clustering algorithm is simple and efficient, capable of identifying clusters of arbitrary dimensions and shapes, and has been widely applied in various fields. However, this algorithm also has some issues, such as sensitivity to the truncation distance parameter, difficulty in finding the cluster centers of low-density regions, and a tendency to produce the ‘domino effect’. To address these issues, this paper proposes a density peak clustering algorithm with mixed nearest neighbors and multi-cluster merging. First, by comprehensively considering the global distribution and local structure of the samples, natural nearest neighbors and k-nearest neighbors are introduced to redefine local density, eliminating sensitivity to the truncation distance parameter and augmenting the local density of samples in low-density regions to improve the identification of cluster centers. Then, the samples are divided into multiple micro-clusters, and inter-cluster association is utilized for merging to reduce the misallocation of samples that are far from the cluster centers, thereby alleviating the ripple effect of allocation errors. Finally, tests conducted on both synthetic and real datasets demonstrate that the proposed algorithm outperforms its comparative counterparts in overall performance.

Keywords: clustering； natural nearest neighbor； k-nearest neighbor； inter-cluster correlation degree； density peaks；local density

收稿日期: 2024-10-21；**录用日期:** 2025-01-29.
基金项目: 国家自然科学基金项目 (62066030).
责任编委: 黄敏.
†通信作者. E-mail: lvli623@163.com.

0　引　言

聚类分析简称聚类[1], 是一种重要的数据挖掘方法[2], 能够对数据进行分析寻找出其中的隐藏信息. 由于用途广泛, 聚类分析在数据挖掘和机器学习领域得到广泛的重视和应用[3]. 近年来, 国内外学者提出了多种聚类算法[4], 如基于层次、基于划分、基于图论、基于网格和基于密度的聚类算法[5-9]. 基于密度的聚类算法旨在识别由较低密度区域分隔出的稠密区域, 可在特征空间中识别任意形状的非球形类簇(如 DBSCAN 算法[10]), 但缺乏参数选择的理论基础.

随着研究的不断深入, Rodrigues 等[11] 提出了密度峰值聚类 (DPC) 算法. 该算法原理简单, 鲁棒性强, 可识别形状各异的类簇. 相较于 DBSCAN 等密度聚类算法, DPC 算法的参数更容易确定. 然而, DPC 算法也存在局限, 如: 低密度区域的类簇中心难以被发现, 非类簇中心点的分配易产生“多米诺效应”.

关于局部密度的改进, 陈梅等[12] 使用低密度分数将簇内每个区域的密度分布重构为单峰密度分布, 从而正确识别含多密度峰值的簇. Du 等[13] 引入模糊邻域并基于欧氏距离计算样本对局部密度的贡献, 在保持 DPC 算法高效性的前提下提升算法鲁棒性. 赵嘉等[14] 结合近邻与测地距离定义局部密度, 使密度峰值点与非密度峰值点的差异明显, 容易准确选择类簇中心. 谭鸿伟等[15] 定义局部密度时引入代表点思想, 通过代表点的代表值平衡密集簇与稀疏簇中代表点的局部密度, 从而缓解密度差对类簇中心选取的影响. 关于分配策略的改进, 陈蔚昌等[16] 引入共享近邻构造相似矩阵并基于此分配剩余样本, 有效避免了样本的误分配. 陈梅等[17] 提出两级分配策略, 对不同密度的数据点采用不同的分配方式, 减少了分配错误连带问题. 赵嘉等[18] 发挥k近邻的扩散作用分配剩余样本, 可更好地识别类簇结构, 从而提高了样本分配的准确率. 上述算法虽然提高了 DPC 的聚类性能, 但对于某些复杂数据集的聚类效果却不理想.

为此, 本文提出一种混合近邻和多簇合并的密度峰值聚类 (DPC-MNM) 算法. 该算法的创新点有: 1) 设计了一种新的局部密度定义方法, 使低密度区域样本的局部密度提高, 从而能更好地识别类簇中心. 2) 提出了一种新的簇间相似度计算方法, 使微簇之间的关联程度更加密切, 可有效指导微簇合并.

1　DPC 算法

DPC 算法通过分析样本的局部密度和相对距离来识别聚类中心. 该算法主要依赖于两个核心假设: 1) 聚类中心周围的局部密度较低; 2) 聚类中心与高密度样本间的距离较大. 基于此, DPC 算法为每个样本定义与计算以下两个关键属性.

1) 局部密度: 表示每个样本点周围邻域的密度, 可通过高斯核法或截断核法进行计算, 即

$$\rho_i = \sum \exp\left(-\frac{d_{ij}^2}{d_c^2}\right). \tag{1}$$

$$\rho_i = \sum_{j=1} \chi(d_{ij} - d_c),$$
$$\chi(x) = \begin{cases} 1, & x < 0; \\ 0, & x \geqslant 0. \end{cases} \tag{2}$$

其中: d_{ij}为样本间的欧氏距离; d_c为截断距离参数. 式 (1) 为高斯核法, 计算样本x_i与所有样本点高斯距离之和. 式 (2) 为截断核法, 是截断距离d_c邻域内样本的数量. 一般而言, 截断核适用于较大数据规模; 高斯核适用于较小的数据规模.

2) 相对距离: 样本x_i与其最近的高密度样本x_j之间的距离, 有

$$\delta_i = \begin{cases} \min\limits_{j:\rho_j > \rho_i}(d_{ij}), & \exists j \ \text{s.t.}\ \rho_j > \rho_i; \\ \max\limits_{j}(d_{ij}), & \text{othewise}. \end{cases} \tag{3}$$

DPC 算法选择决策图中ρ_i和δ_i均较大的样本作为类簇中心. 然而, 该方法通常依赖于人工观察和直观判断, 缺乏定量分析, 具有较大的主观性. 决策值较大的样本能够被自动识别为类簇中心, 通过定量计算每个样本的决策值[19], 避免了人为干预的偏差, 具有较高的客观性. 决策值

$$\gamma_i = \rho_i \cdot \delta_i. \tag{4}$$

DPC 算法的聚类过程为: 首先, 根据式 (1) ~ (3) 计算ρ_i和δ_i; 其次, 根据式 (4) 计算的决策值来选取类簇中心; 然后, 按密度降序逐一分配剩余样本到其最近的高密度样本所属类簇, 从而完成聚类.

2　混合近邻和多簇合并的密度峰值聚类算法

2.1　混合近邻的局部密度

DPC 的局部密度对参数敏感, 且难以发现低密度区域的类簇中心. 为准确计算局部密度, DPC-MNM 算法提出了混合近邻的局部密度. 结合自然近邻与k近邻定义局部密度, 在自然近邻的基础上, 结合k近邻调节不同分布样本的局部密度, 以增加类簇中心的辨识度.

定义 1 (k近邻 (KNN)[20])　将任一样本x_i与剩余样本x_j的距离升序排列, 取前k个样本构成KNN(x_i). d_{ij}是样本x_i与x_j之间的欧氏距离, KNN(x_i)的表达

式如下:

$$\mathrm{KNN}(x_i)=\{x_j\in D|d_{ij}\leqslant d_{ik}\}. \tag{5}$$

定义 2 (自然近邻[21]) 如果样本x_j属于$\mathrm{KNN}(x_i)$, 同时样本x_i也属于$\mathrm{KNN}(x_j)$, 即$x_j\in\mathrm{KNN}(x_i)$且$x_i\in\mathrm{KNN}(x_j)$, 则称样本x_i与x_j互为自然近邻点, 记作$\mathrm{NNN}(x_i,x_j)$. 其表达式为

$$\mathrm{NNN}(x_i,x_j)=\begin{cases}1,\ x_i\in\mathrm{KNN}(x_j)\ \text{and}\ x_j\in\mathrm{KNN}(x_i);\\0,\ \text{otherwise.}\end{cases} \tag{6}$$

定义 3(混合近邻的局部密度) 对于样本x_i, 其局部密度定义为

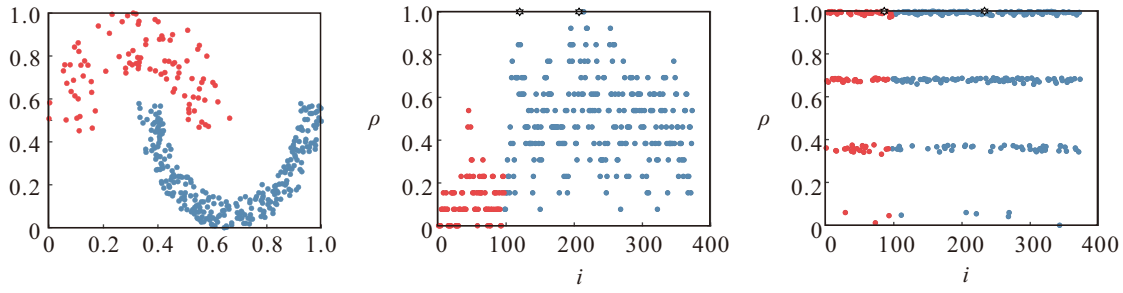
$$\rho_i=\frac{1}{N}\cdot\sum_{j=1}^{N}\exp(-d_{ij}^2)\ +\ |\mathrm{NNN}(i)|+\exp\Bigg(-\frac{\sum\limits_{j\in\mathrm{KNN}(i)}d_{ij}}{\sum\limits_{j\in\mathrm{KNN}(i)}\sum\limits_{v\in\mathrm{KNN}(j)}d_{vj}}\Bigg). \tag{7}$$

其中: d_{ij}是样本间的欧氏距离; N是样本规模; $\exp(-d)$是指数函数e^{-x}, 表示距离越近关系越密切; $|\mathrm{NNN}(i)|$是样本x_i的自然近邻点个数; $\sum\limits_{j\in\mathrm{KNN}(i)}d_{ij}$是样本的离群程度.

混合近邻的局部密度, 首先, 从全局考虑所有样本的相互影响, 以欧氏距离表示样本间的密切程度, 计算所有样本点对密度的贡献; 其次, 考虑样本所处的局部环境, 自然近邻点越多说明所处的环境越密集, 计算自然近邻点对密度的贡献; 最后, 在自然近邻的基础上, 结合k近邻调节不同分布样本的局部密度, 计算样本点与其k近邻点的相对密度对密度的贡献.

这样设计的局部密度, 不但从宏观角度考虑了全局样本的相互影响, 消除了截断距离参数的敏感性, 而且从微观角度考虑了样本所处的局部环境. 自然近邻更适合处理不规则分布的数据, k近邻对于均匀分布的数据更有效, 两者结合能够在更复杂的数据环境中提供更可靠的局部密度计算, 提升低密度区域样本的局部密度, 增加类簇中心的辨识度.

为验证混合近邻的局部密度有效性, 以 Jain 数据集为例进行实验, 其中“白色六角星”表示类簇中心, 用不同颜色区分不同类簇. Jain 数据集由上下两个半弧形类簇组成, 分布情况如图 1(a) 所示. 图 1(b) 和图 1(c) 分别是 DPC 算法和 DPC-MNM 算法计算的局部密度. 可以看出, 相较于图 1(b), 图 1(c) 中红色样本的局部密度整体得到增加, 且局部密度分布更有层次. 图 1(b) 中, 在蓝色样本中选取了两个类簇中心, 因为红色样本的局部密度大部分比较小, 而 DPC 的局部密度更倾向于在高密度区域寻找类簇中心, 不易识别红色样本的类簇中心; 图 1(c) 所示为改进的局部密度, 红色样本与蓝色样本中都有较大的局部密度, 所以能够在两个簇中各选取一个类簇中心.



(a) Jain 数据集的分布图 (b) DPC的局部密度 (c) 混合近邻的局部密度

图1 Jain 数据集上 DPC 和混合近邻的局部密度

2.2 微簇中心的选择

DPC 算法中, 相对距离的设计仅关注了第 2 个假设, 即类簇中心之间距离较大, 但忽略了第 1 个假设中类簇中心被低密度样本包围的情况. 借鉴文献[22] 思想, 本文实验采用调整距离来代替传统的相对距离, 有

$$\delta_i^{+}=\begin{cases}\min\limits_{j:\rho_j>\rho_i}(d_{ij}),\ \rho_i<\rho_{\max};\\\max\limits_{i\neq j}(\delta_j^{+}),\ \rho_i=\rho_{\max}.\end{cases} \tag{8}$$

$$\delta_i^{-}=\begin{cases}\min\limits_{j:\rho_j<\rho_i}(d_{ij}),\ \rho_i>\rho_{\max};\\\max\limits_{i\neq j}(\delta_j^{-}),\ \rho_i=\rho_{\min}.\end{cases} \tag{9}$$

$$\delta_i=\delta_i^{+}-\delta_i^{-}. \tag{10}$$

其中: δ_i^{+}是 DPC 算法中的相对距离, 该值越大, 说明样本x_i离高密度样本较远; δ_i^{-}是修正距离, 体现了类簇中心被低密度样本包围的情况; δ_i是调整距离, 该值为正时, 说明样本x_i既离高密度样本较远, 又离低密度样本较近, 更可能是正确的密度峰值. 因此, 将调整距离为正的样本作为潜在的类簇中心, 再通

过滤在类簇中心决策值是否大于所有样本决策值均值与方差的差, 衡量其是否能成为微簇中心.

初始微簇的构建过程: 首先, 根据式 (7) 计算局部密度; 其次, 通过基于该密度计算的调整距离选取潜在类簇中心, 并衡量其是否能成为微簇中心; 最后, 微簇中心确定后, 利用 DPC 算法的分配策略对非微簇中心进行分配, 形成初始微簇.

2.3　多簇合并策略

DPC 算法更倾向将某个样本分配给高密度区域, 一旦一个样本被错误分配, 容易导致后续大量样本被错误分配. 然而, 将样本分成多个微簇时, 使用 DPC 算法的分配策略可以准确分配[22]. 因此, 本文利用多簇合并思想优化分配策略, 能有效缓解分配错误连带效应. 具体合并过程为: 利用 DPC 算法选取类簇中心; 计算初始微簇之间的关联度, 并标记包含类簇中心的微簇; 从未标记微簇中, 选择与已标记的簇间关联度最大的微簇进行合并, 重复此过程, 直到所有簇间关联度为零.

定义 4 (共享近邻[21])　x为样本, D是数据集, $x_i,\ x_j \in D$. 称$\mathrm{KNN}(x_i)$与$\mathrm{KNN}(x_j)$的交集为x_i与x_j的共享近邻集合, 记作$\mathrm{SNN}(x_i, x_j)$. 其表达式为

$$\mathrm{SNN}(x_i, x_j) = \mathrm{KNN}(x_i) \bigcap \mathrm{KNN}(x_j). \tag{11}$$

定义 5 (簇间关联度)　微簇c_i与c_j间的关联度定义为

$$\rho_{c_i} = \frac{\sum\limits_{k\in c_i} \rho_k}{\mathrm{count}(c_i)}, \tag{12}$$

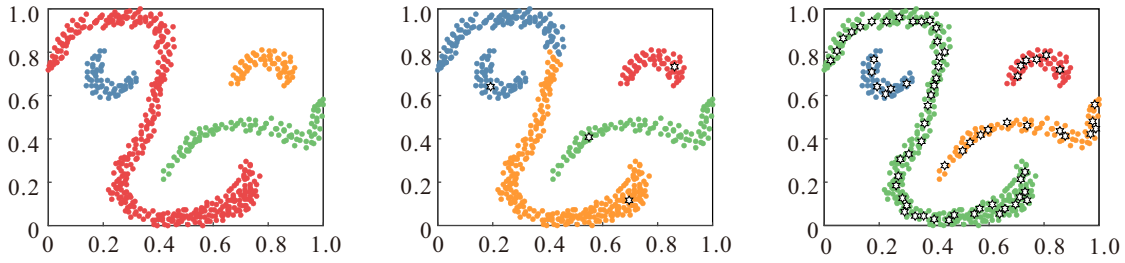
$$\mathrm{Attrativeness}(c_i, c_j) = \frac{\rho_{c_i} \cdot \rho_{c_j}}{d^2_{c_i\, c_j}}, \tag{13}$$

$$\mathrm{Correlations}(c_i, c_j) = |\mathrm{SNN}(c_i, c_j)| + \mathrm{Attrativeness}(c_i, c_j). \tag{14}$$

其中: ρ_{c_i}是微簇的密度; $\mathrm{count}(c_i)$是微簇c_i中样本的个数; $d_{c_i\, c_j}$是微簇中心间的距离; $|\mathrm{SNN}(c_i, c_j)|$是微簇间共享近邻的个数; $\mathrm{Attrativeness}(c_i, c_j)$是微簇$c_i$与$c_j$之间的吸引度.

$\mathrm{Correlations}(c_i, c_j)$表示微簇间的关联度, 是微簇间共享近邻数与吸引度之和, 反映了微簇间的密切程度. 微簇间共享近邻数表示每对微簇中样本间k近邻的重叠情况. 共享近邻存在意味着这两个簇的结构相似, 它们可能属于同一个更大的簇. 微簇间吸引度由微簇密度与微簇中心间的距离组成. 微簇密度用于衡量簇内样本的聚集程度, 密度高的簇一般较为紧凑. 微簇中心间的距离通常表示簇与簇之间的分离度, 距离越小, 意味着簇间存的相似性越大. 因此, 微簇密度越大, 微簇中心间距离越小, 微簇间吸引度越大, 两微簇关联性越强. 故而, 综合考虑微簇的局部结构、聚集性和分离度, 能够更准确衡量微簇之间的关联程度. 利用该簇间关联度指导微簇合并, 能有效缓解分配错误连锁反应.

为验证优化后分配策略的有效性, 保证局部密度和相对距离一致条件下, 仅改变分配策略进行对比实验, 具体以 Db 数据集为例进行说明. Db 数据集由 4 条弧形簇组成, 分布情况如图 2(a) 所示. 图 2(b) 和图 2 (c) 分别是利用 DPC 和 DPC-MNM 算法的分配策略得到的聚类结果. 图中“白色六角星”表示类簇中心或微簇中心. 可以发现, 图 2(b) 在准确选择类簇中心的前提下, 左上方本属于橙色簇的样本被错误地分配给了蓝色簇. 因为出现错误的样本离蓝色簇的类簇中心较近, 而 DPC 算法倾向将样本分配给高密度区域, 所以容易出现这类误分配问题. 而 DPC-MNM 算法的分配策略有效缓解了这种情况, 如图 2(c) 所示. 该算法充分考虑微簇的分布特性, 增强了同类簇内微簇之间的关联性, 可有效分配距簇中心较远的样本, 减弱了分配错误连锁效应.



(a) Db 数据集的分布图　(b) DPC 的分配策略　(c) DPC-MNM 的分配策略

图2　Db 数据集上不同分配策略的聚类结果

2.4　算法步骤

算法的步骤如下:

Input: 数据集X, 近邻参数k;

Output: 聚类结果.

step 1: 数据标准化.

step 2: 计算样本间欧氏距离, 根据式 (5) ∼ (7) 计

算样本局部密度.

step 3: 根据式 (8) ∼ (10) 计算样本相对距离, 并标记微簇中心.

step 4: 利用 DPC 的分配策略对非微簇中心点进行分配, 形成初始微簇;

step 5: 根据式 (4) 计算决策值, 并标记类簇中心.

step 6: 根据式 (11) ∼ (14) 计算簇间关联度, 并标记包含类簇中心的微簇.

step 7: 从未标记微簇中, 选择与已标记的簇间关联度最大的微簇, 进行合并.

step 8: 若所有簇间关联度非零, 则转至 step 7; 否则, 聚类结束.

3 实验结果与分析

3.1 实验设置

为验证 DPC-MNM 算法的聚类性能, 将其与 DPC[11]、DPC-FWSN[23]、DPC-CE[24]、FNDPC[13]、FKNN-DPC[25] 算法在人工和真实数据集上进行实验. 人工数据集和真实数据集的基本信息分别见表 1 和表 2. 人工数据集包括密度分布不均数据集和以弧线状或环状为主的流形数据集等. 真实数据集包含现实生活中真实存在的数据, 维度更高且类簇个数更多, 更能验证算法的性能.

表1 人工数据集

数据集	规模	维度	类簇个数
Compound	399	2	6
Cmc	1 002	2	3
Db	630	2	4
Circle	1 897	2	3
LineBlobs	266	2	3
Jain	373	2	2
Pathbased	300	2	3
Aggregation	788	2	7
Chainlink	1 000	3	2
Atom	800	3	2

表2 真实数据集

数据集	规模	维度	类簇个数
Wine	178	13	3
Yeast	1 484	8	10
Seeds	210	7	3
Wdbc	569	30	2
Thyroid	215	5	3
Ecoli	336	7	8
Waveform	5 000	21	3
Libras	360	90	15

为更好体现各算法性能, 本实验对各算法参数进行调优, 具体细节为: DPC-MNM、FKNN-DPC 和 DPC-FWSN 算法的近邻数 $k \in [2, 50]$, 步长为 1; FNDPC 算法的 $\varepsilon \in [0.01, 1]$, 步长为 0.01; DPC 算法的 $d_c \in [0.1\ \%, 5\ \%]$, 步长为 0.1 %; DPC-CE 算法没有输入参数, 不需要调优. 本文实验采用的实验环境为 Windows10 的 64 位操作系统, Intel(R) Core(TM)i5-7200U CPU @ 2.50 GHz 2.70 GHz, 8G 内存, Matlab R2021a.

本文采用 3 种常见评价指标评估聚类效果, 具体包括: 调整互信息 (AMI)[26]、调整兰德系数 (ARI)[26] 和 FM 指数 (FMI)[27]. AMI 和 ARI 分别评估聚类结果与真实类别标签间的相似性与一致性, FMI 评估聚类结果的精确度和召回率的平衡. 各指标越接近 1, 表示聚类效果越好.

3.2 人工数据集的实验结果与分析

本文先比较 6 种算法在人工数据集上的聚类效果. 表 3 呈现了各算法在表 1 所列数据集上聚类结果的评价指标值, 其中, 最优结果以加粗字体凸出; 最优参数用 “Arg-” 表示, 其值若为 “—” 代表该算法不需要进行调参. 由表 3 可知, DPC-MNM 算法在这 10 个数据集上评价指标整体高于其余算法. 其中, 评价指标达到最优值 1 的数据集有 Aggregation 等 8 个, 表明 DPC-MNM 能准确聚类这 8 个数据集. 而 DPC 和 FNDPC 算法仅能准确聚类 Chainlink 数据集. 其余算法能准确聚类的数据集分别为: 5 个、4 个、3 个. 综上, DPC-MNM 算法对人工数据集的聚类效果优于其他算法.

为更直观地体现 6 种算法对人工数据集的聚类效果, 图 3 列出了各算法在 Pathbased 数据集上的聚类结果. 其中, 不同簇以颜色进行区分, “白色六角星” 代表微簇中心或类簇中心. Pathbased 数据集既有流形簇, 又有密度存在差异的簇. 可以发现, DPC-MNM 和 FKNN-DPC 算法在 Pathbased 数据集上聚类效果较好, DPC 及其余改进算法聚类效果不理想. 因为 DPC-FWSN 算法未能找到密度较低的环形簇的类簇中心, 导致聚类效果较差; DPC-CE、FNDPC 和 DPC 算法虽能准确选取类簇中心, 但是就近分配剩余样本仅参考距离因素, 忽略了流形簇中样本的具体分布情况.

将 DPC-MNM 等算法的评价指标进行 Friedman 检验, 可更好地反映各算法的综合性能. 秩均值越高意味着算法的性能越好. 表 4 呈现了各算法在人工数据集上评价指标的秩均值. 可以发现, DPC-MNM 算法 AMI、ARI 和 FMI 的秩均值均是最高的, 说明其综合性能最优; DPC-FWSN 算法次之, FKNN-DPC

表3　DPC-MNM 等算法在人工数据集上的评价指标值

数据集	评价指标	算法					
		DPC-MNM	DPC-FWSN	DPC-CE	FNDPC	FKNN-DPC	DPC
Compound	AMI	**0.887 8**	0.866 4	0.808 2	0.851 6	0.846 7	0.775 4
	ARI	**0.895 5**	0.878	0.614 1	0.868 4	0.843	0.591
	FMI	**0.923 5**	0.911 5	0.706	0.904 6	0.888 4	0.687 6
	Arg-	23	16	—	0.38	7	3.8
Db	AMI	**1**	0.652 5	0.675 8	0.643 1	0.510 7	0.479 9
	ARI	**1**	0.494 2	0.558 8	0.441 2	0.271 8	0.363 3
	FMI	**1**	0.697	0.739 5	0.67	0.579 3	0.606 7
	Arg-	6	17	—	0.74	19	1
LineBlobs	AMI	**1**	**1**	**1**	0.779 4	**1**	0.837 5
	ARI	**1**	**1**	**1**	0.717 9	**1**	0.823 7
	FMI	**1**	**1**	**1**	0.814 8	**1**	0.884 2
	Arg-	6	21	—	0.11	7	4.2
Pathbased	AMI	**0.951 8**	0.587 8	0.446 2	0.575 1	0.930 5	0.533 5
	ARI	**0.969 2**	0.551 7	0.378 7	0.506 7	0.949 9	0.512 7
	FMI	**0.979 5**	0.747 2	0.628 5	0.706 5	0.966 5	0.732 2
	Arg-	7	7	—	0.01	9	0.1
Chainlink	AMI	**1**	**1**	**1**	**1**	**1**	**1**
	ARI	**1**	**1**	**1**	**1**	**1**	**1**
	FMI	**1**	**1**	**1**	**1**	**1**	**1**
	Arg-	2	4	—	0.04	7	0.5
Cmc	AMI	**1**	**1**	0.669 4	0.809 3	**1**	0.385 7
	ARI	**1**	**1**	0.736 2	0.842 1	**1**	0.266 1
	FMI	**1**	**1**	0.835 2	0.902 7	**1**	0.537 7
	Arg-	2	6	—	0.28	49	58
Circle	AMI	**1**	0.778	0.529	0.423 6	0.706 3	0.359 6
	ARI	**1**	0.761 5	0.255 5	0.273 2	0.613 9	0.301 5
	FMI	**1**	0.877 5	0.627 9	0.586 3	0.779	0.604 8
	Arg-	25	41	—	0.29	32	0.3
Jain	AMI	**1**	**1**	**1**	0.596 1	0.709 2	0.618 3
	ARI	**1**	**1**	**1**	0.725 7	0.822 4	0.714 6
	FMI	**1**	**1**	**1**	0.905 1	0.935 9	0.881 9
	Arg-	3	4	—	0.47	43	0.3
Aggregation	AMI	**1**	0.995 5	0.992 2	0.986 4	0.990 5	0.992 2
	ARI	**1**	0.997 8	0.995 6	0.991 3	0.994 9	0.995 6
	FMI	**1**	0.998 3	0.996 6	0.993 2	0.996	0.996 6
	Arg-	18	14	—	0.02	20	3.1
Atom	AMI	**1**	**1**	0.316 6	0.432 5	**1**	0.404 4
	ARI	**1**	**1**	0.256 9	0.409	**1**	0.371 4
	FMI	**1**	**1**	0.672 2	0.723 5	**1**	0.709 3
	Arg-	7	3	—	0.31	5	2.6

表4　DPC-MNM 等算法在人工数据集上的秩均值

算法	秩均值		
	AMI	ARI	FMI
DPC-MNM	**5.3**	**5.3**	**5.3**
DPC-FWSN	4.6	4.6	4.6
DPC-CE	3.05	2.85	3.05
FNDPC	2.45	2.45	2.35
FKNN-DPC	3.7	3.6	3.6
DPC	1.9	2.2	2.1

算法再次之.

综上, DPC-MNM、DPC-FWSN 和 FKNN-DPC 算法对人工数据集的聚类效果较好. 但仅 DPC-MNM 能够很好地处理流形数据集. 结合表 3 可知, 本文提出的 DPC-MNM 算法在多个人工数据集上均获得了最优的评价指标.

3.3　真实数据集的实验结果与分析

本小节将比较 6 种算法在真实数据集上的聚类

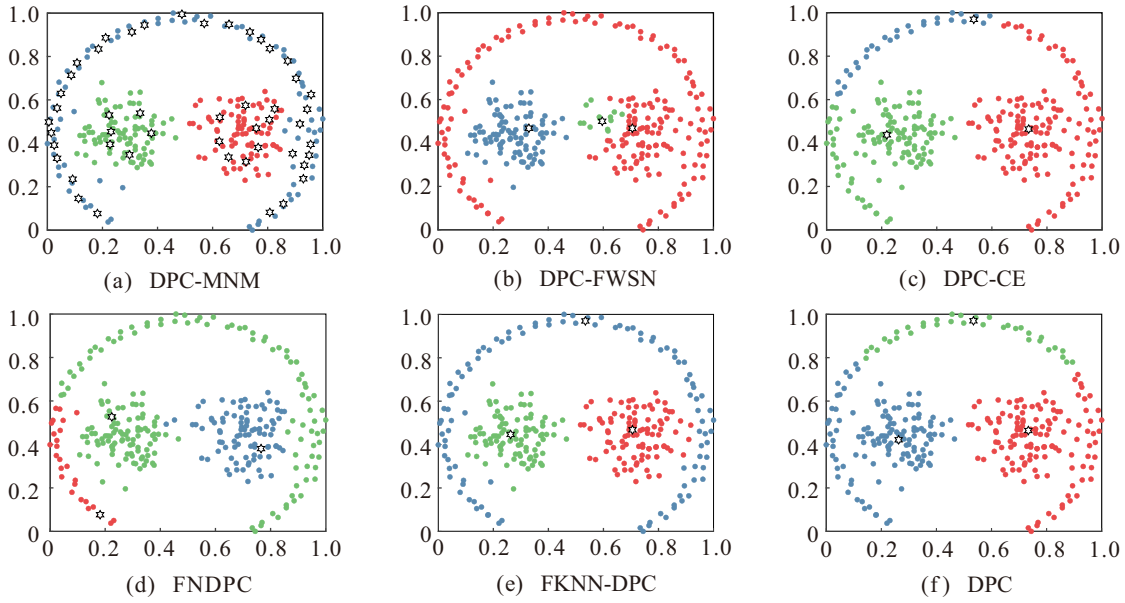


图3 Pathbased 数据集上的聚类结果

表5 DPC-MNM 等算法在真实数据集上的评价指标值

数据集	评价指标	算法					
		DPC-MNM	DPC-FWSN	DPC-CE	FNDPC	FKNN-DPC	DPC
Wine	AMI	0.7947	0.8598	0.5841	0.7898	0.8481	0.7695
	ARI	0.8309	0.8837	0.5362	0.8025	**0.8839**	0.7703
	FMI	0.8877	0.9227	0.6945	0.8686	**0.9229**	0.8474
	Arg-	13	32	—	0.26	8	2.4
Ecoil	AMI	**0.6804**	0.6637	0.6373	0.4833	0.5878	0.5179
	ARI	**0.7776**	0.7331	0.7398	0.5618	0.5894	0.4365
	FMI	**0.8415**	0.8059	0.819	0.7178	0.7027	0.5693
	Arg-	10	6	—	0.35	2	0.2
Yeast	AMI	**0.2933**	0.2725	0.1122	0.243	0.1368	0.1864
	ARI	**0.235**	0.1667	0.0915	0.1617	0.1347	0.1162
	FMI	0.4241	0.4772	0.4509	0.4606	**0.4833**	0.2748
	Arg-	47	2	—	0.04	6	0.4
Thyroid	AMI	**0.7078**	0.4456	0.2614	0.3961	0.341	0.2549
	ARI	**0.8142**	0.554	0.3452	0.5206	0.4022	0.3211
	FMI	**0.9174**	0.8314	0.7823	0.8194	0.7854	0.7138
	Arg-	27	4	—	0.23	16	3.4
Wdbc	AMI	**0.7198**	0.6697	0.3742	0.6076	0.6423	0.6375
	ARI	**0.8245**	0.7796	0.4355	0.7305	0.7613	0.7548
	FMI	**0.918**	0.8993	0.7743	0.8758	0.8894	0.8876
	Arg-	43	9	—	0.05	2	0.7
Seeds	AMI	0.7628	0.7366	0.7144	0.7136	**0.7757**	0.7298
	ARI	**0.8118**	0.7871	0.7448	0.7545	0.8024	0.767
	FMI	**0.874**	0.8574	0.8297	0.8361	0.8682	0.8444
	Arg-	5	7	—	0.07	9	0.7
Libras	AMI	**0.6062**	0.5342	0.557	0.5494	0.5554	0.5832
	ARI	**0.3906**	0.3287	0.3531	0.329	0.3549	0.3626
	FMI	**0.4506**	0.4038	0.4192	0.3869	0.4044	0.419
	Arg-	6	14	—	0.17	10	0.5
Waveform	AMI	0.3729	**0.4078**	0.3274	0.3293	0.3239	0.3261
	ARI	0.3682	**0.3905**	0.2836	0.283	0.2671	0.2698
	FMI	0.5893	**0.594**	0.5456	0.5442	0.5244	0.5292
	Arg-	42	324	—	0.34	2	0.1

效果. 表 5 是各算法在表 2 所列数据集上聚类结果的评价指标值. 各指标值越趋近 1, 说明聚类精度越高. 由表 5 可知, 相较于 DPC, DPC-MNM 算法在这 8 个数据集上聚类精度提升显著. 其中, 尤为显著的是 Ecoli 和 Thyroid 数据集. 在 Ecoli 数据集上, DPC-MNM 算法获得的评价指标较 DPC 算法分别提高了 0.16、0.34、0.27; 在 Thyroid 数据集上, DPC-MNM 获得的评价指标分别提高了 0.42、0.49、0.20, 是因为 DPC 算法的截断距离参数d_c对小规模数据敏感. 同时, 相较于其他改进算法, DPC-MNM 在 4 个数据集上聚类效果最优, 在 3 个数据集上聚类效果仅次于最优结果.

表 6 呈现了各算法在真实数据集上评价指标的秩均值. 由表 6 可知, DPC-MNM 算法各评价指标的秩均值均高于 5 种对比算法, 综合性能最优. 该算法设计的新局部密度可以提高低密度区域样本的局部密度, 所以更容易识别出准确的类簇中心, 而且优化的分配策略减少了离类簇中心较远的样本的误分配. 因此, DPC-MNM 算法在人工和真实数据集上均表现良好, 表明其鲁棒性较好.

表6　DPC-MNM 等算法在真实数据集上的秩均值

算法	秩均值		
	AMI	ARI	FMI
DPC-MNM	**5.5**	**5.63**	**5.13**
DPC-FWSN	4.63	4.38	4.5
DPC-CE	2.13	2.38	2.75
FNDPC	2.63	2.75	2.75
FKNN-DPC	3.38	3.5	3.75
DPC	2.75	2.38	2.13

4　结　论

针对 DPC 算法难以发现低密度区域的密度峰值且易出现分配错误连带效应问题, 本文提出了混合近邻和多簇合并的密度峰值聚类算法. 该算法通过新的局部密度增大低密度区域样本的局部密度, 以提高该区域类簇中心的辨识率; 利用簇间关联度合并微簇, 减少离类簇中心较远的样本的误分配, 以缓解分配错误连带效应. 最后, 通过对比实验, 验证了 DPC-MNM 算法可以提高低密度区域类簇中心的辨识度, 缓解分配错误连带效应, 而且聚类效果整体较好. 此外, 接下来的研究将着重考虑: 1) 提升算法的运行效率. DPC-MNM 算法虽然提高了聚类精度, 但处理大规模高维数据[28-29] 时, 聚类效率较低. 因此, 下一步将在确保准确性的前提下, 探索如何提升算法的运行效率. 2) 探索算法的应用场景. 尝试将 DPC-MNM 算法运用在异常用电检测中, 以识别和预防潜在的电力故障.

参考文献 (References)

[1] Mining W I D. Data mining: Concepts and techniques[J]. Morgan Kaufinann, 2006, 10(559~569): 4.

[2] 李松, 吴润秀, 康平, 等. 基于自适应剪辑与概率参数的 Tri-Training 算法[J]. 江西师范大学学报: 自然科学版, 2023, 47(5): 490-496.
(Li S, Wu R X, Kang P, et al. The ADP-Tri-Training: Tri-Training with adaptive editing and probability parameters [J]. Journal of Jiangxi Normal University: Natural Sciences Edition, 2023, 47(5): 490-496.)

[3] Aggarwal C C, Reddy C K. Data clustering: Algorithms and applications[M]. Boca Raton: CRC Press, 2013: 111-124.

[4] 陈梅, 柳博雅, 王钰, 等. 基于时间序列形态的模糊聚类算法[J]. 控制与决策, 2025, 40(4): 1116-1126.
(Chen M, Liu B Y, Wang Y, et al. Fuzzy clustering algorithm based on time series morphology[J]. Control and Decision, 2025, 40(4): 1116-1126.)

[5] Chu Z Y, Wang W F, Li B Z, et al. An operation health status monitoring algorithm of special transformers based on BIRCH and Gaussian cloud methods[J]. Energy Reports, 2021, 7: 253-260.

[6] Tavallali P, Tavallali P, Singhal M. K-means tree: An optimal clustering tree for unsupervised learning[J]. The Journal of Supercomputing, 2021, 77(5): 5239-5266.

[7] Von Luxburg U. A tutorial on spectral clustering[J]. Statistics and Computing, 2007, 17(4): 395-416.

[8] Bureva V, Popov S, et al. Generalized net model of cluster analysis using CLIQUE: Clustering in quest[J]. International Journal Bioautomation, 2019, 23(2): 131-138.

[9] Zhu Q D, Tang X M, Elahi A. Application of the novel harmony search optimization algorithm for DBSCAN clustering[J]. Expert Systems with Applications, 2021, 178: 115054.

[10] Ester M, Kriegel H P, Sander J, et al. 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. Portland, 1996: 226-231.

[11] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.

[12] 陈梅, 尤远毓秀, 魏礼磊, 等. 基于低密度分数的密度峰值聚类算法[J]. 控制与决策, 2025, 40(5): 1599-1609.
(Chen M, You Y Y X, Wei L L, et al. A density peaks clustering algorithm based on low density score[J]. Control and Decision, 2025, 40(5): 1599-1609.)

[13] Du M J, Ding S F, Xue Y. A robust density peaks clustering algorithm using fuzzy neighborhood[J]. International Journal of Machine Learning and Cybernetics, 2018, 9(7): 1131-1140.

[14] 赵嘉, 王刚, 吕莉, 等. 面向流形数据的测地距离与余弦互逆近邻密度峰值聚类算法[J]. 电子学报, 2022,

50(11): 2730-2737.
(Zhao J, Wang G, Lü L, et al. Density peaks clustering algorithm based on geodesic distance and cosine mutual reverse nearest neighbors for manifold datasets[J]. Acta Electronica Sinica, 2022, 50(11): 2730-2737.)

[15] 谭鸿伟, 吕莉, 郝筱萱, 等. 基于近邻与代表点的密度峰值聚类算法[J]. 控制工程, DOI: 10.14107/j.cnki.kzgc.20240081.
(Tan H W, Lv L, Hao X X, et al. Density peaks clustering algorithm based on nearest neighbors and representative points[J]. Control Engineering of China, DOI: 10.14107/j.cnki.kzgc.20240081.)

[16] 陈蔚昌, 赵嘉, 肖人彬, 等. 面向密度分布不均数据的近邻优化密度峰值聚类算法[J]. 控制与决策, 2024, 39(3): 919-928.
(Chen W C, Zhao J, Xiao R B, et al. Density peaks clustering algorithm with nearest neighbor optimization for data with uneven density distribution[J]. Control and Decision, 2024, 39(3): 919-928.)

[17] 陈梅, 魏礼磊, 尤远毓秀, 等. 基于k近邻图的密度峰值聚类算法[J]. 控制与决策, DOI: 10.13195/j.kzyjc.2024.1152.
(Chen M, Wei L L, You Y Y X, et al. Density peaks clustering algorithm based on k-nearest neighbor graph[J]. Control and Decision, DOI: 10.13195/j.kzyjc.2024.1152.)

[18] 赵嘉, 陈蔚昌, 肖人彬, 等. 面向流形数据的共享近邻和二阶K近邻密度峰值聚类算法[J]. 控制理论与应用, DOI: 10.7641/CTA.2024.30570.
(Zhao J, Chen W C, Xiao R B, et al. Clustering algorithm of shared nearest neighbor and second-order K nearest neighbor density peak for manifold data[J]. Control Theory & Applications, DOI: 10.7641/CTA.2024.30570.)

[19] 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.
(Zhao J, Yao Z F, Lyu L, et al. Density peaks clustering based on mutual neighbor degree[J]. Control and Decision, 2021, 36(3): 543-552.)

[20] 赵嘉, 陈磊, 吴润秀, 等. K近邻和加权相似性的密度峰值聚类算法[J]. 控制理论与应用, 2022, 39(12): 2349-2357.
(Zhao J, Chen L, Wu R X, et al. Density peaks clustering algorithm with K-nearest neighbors and weighted similarity[J]. Control Theory & Applications, 2022, 39(12): 2349-2357.)

[21] 吕莉, 朱梅子, 康平, 等. 面向密度分布不均数据的混合近邻密度峰值聚类算法[J]. 控制理论与应用, 2024, 41(10): 1821-1830.
(Lv L, Zhu M Z, Kang P, et al. Multiplex neighbor density peaks clustering for uneven density data sets[J]. Control Theory & Applications, 2024, 41(10): 1821-1830.)

[22] 郭佳, 韩李涛, 孙宪龙, 等. 自动确定聚类中心的比较密度峰值聚类算法[J]. 计算机应用, 2021, 41(3): 738-744.
(Guo J, Han L T, Sun X L, et al. Comparative density peaks clustering algorithm with automatic determination of clustering center[J]. Journal of Computer Applications, 2021, 41(3): 738-744.)

[23] Zhao J, Wang G, Pan J S, et al. Density peaks clustering algorithm based on fuzzy and weighted shared neighbor for uneven density datasets[J]. Pattern Recognition, 2023, 139: 109406.

[24] Guo W J, Wang W H, Zhao S P, et al. Density peak clustering with connectivity estimation[J]. Knowledge-Based Systems, 2022, 243: 108501.

[25] Xie J Y, Gao H C, Xie W X, 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.

[26] Vinh N X, 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: 2837-2854.

[27] 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.

[28] 张曦, 李璠, 付雪峰, 等. 随机学习萤火虫算法优化的模糊软子空间聚类算法[J]. 江西师范大学学报: 自然科学版, 2021, 45(2): 137-144.
(Zhang X, Li F, Fu X F, et al. The fuzzy soft subspace clustering algorithm optimized by random learning firefly algorithm[J]. Journal of Jiangxi Normal University: Natural Science Edition, 2021, 45(2): 137-144.)

[29] 马福民, 宫婷, 杨帆, 等. 基于Zipf分布的网格密度峰值聚类算法[J]. 控制与决策, 2024, 39(2): 577-587.
(Ma F M, Gong T, Yang F, et al. Grid density peak clustering algorithm based on Zipf distribution[J]. Control and Decision, 2024, 39(2): 577-587.)

作者简介

吕莉 (1982−), 女, 教授, 博士, 硕士生导师 , 主要研究方向为智能计算与计算智能、目标跟踪与检测, E-mail: lvli623@163.com;

赵妞 (1998−), 女, 硕士生, 主要研究方向为数据挖掘, E-mail: zhaoniu23@163.com;

肖人彬 (1965−), 男, 教授, 博士, 博士生导师 , 主要研究方向为复杂系统建模与分析、群集智能, E-mail: rbxiao@163.com;

王新峰 (1986−), 男, 讲师, 博士, 主要研究方向为人工智能、生物信息处理, E-mail: wangxf59@mail3.sysu.edu.cn;

韩龙哲 (1976−), 男, 教授, 博士, 主要研究方向为下一代网络体系结构、人工智能算法, E-mail: longzhehan@gmail.com.