

文章编号: 1008-1542(2026)01-0049-11

# 基于最小生成树与统计特征的层次聚类算法

刘子康<sup>1</sup>, 周长杰<sup>1</sup>, 姚 卫<sup>2</sup>

(1. 河北科技大学理学院, 河北石家庄 050018;  
2. 南京信息工程大学数学与统计学院, 江苏南京 210044)

**摘要:** 针对 Chameleon 算法在参数敏感性、噪声鲁棒性及计算效率上的不足, 提出一种基于最小生成树与统计特征的层次聚类算法 (statistical-MST integrated hierarchical clustering algorithm, SHCA)。采用最小生成树构建稀疏图, 消除人工参数干预, 利用最小生成树的全局最优性避免跨簇伪连接; 设计动态统计合并策略, 结合局部距离阈值过滤噪声, 并通过簇间连通性检验, 迭代合并子簇, 确保簇内紧密性与簇间分离性; 在 20 个人工数据集与 10 个真实数据集上进行对比实验。结果表明: SHCA 的聚类性能优于对比算法; 针对部分数据集表现下降的情况, 分析发现流形重叠是主要影响因素。SHCA 有效提升了聚类精度与结果稳定性, 为后续大规模、复杂流形数据的聚类研究提供了参考。

**关键词:** 人工智能理论; 聚类; 层次聚类算法; 最小生成树; 动态统计合并策略

**中图分类号:** TP181 **文献标识码:** A **DOI:** 10.7535/hbkd.2026yx01006

## Hierarchical clustering algorithm based on minimum spanning tree and statistical features

LIU Zikang<sup>1</sup>, ZHOU Changjie<sup>1</sup>, YAO Wei<sup>2</sup>

(1. School of Sciences, Hebei University of Science and Technology, Shijiazhuang, Hebei 050018, China;  
2. School of Mathematics and Statistics, Nanjing University of Information Science and Technology, Nanjing, Jiangsu 210044, China)

**Abstract:** To address the limitations of the Chameleon algorithm in terms of parameter sensitivity, noise robustness, and computational efficiency, this study proposed a statistical-MST integrated hierarchical clustering algorithm (SHCA) based on the minimum spanning tree and statistical features. The minimum spanning tree was used to construct a sparse graph, eliminating manual parameter intervention, and the global optimality of the minimum spanning tree was used to avoid false cross cluster connections. The dynamic statistical merging strategy was designed to filter the noise combined with the local distance threshold, and the sub clusters were merged iteratively through the inter cluster connectivity test to ensure the intra cluster compactness and inter cluster separation. Experiment on 20 synthetic datasets and 10 real-world datasets was

收稿日期: 2025-03-03; 修回日期: 2025-09-01; 责任编辑: 冯民

基金项目: 国家自然科学基金 (12371462)

第一作者简介: 刘子康 (1999—), 男, 河北保定人, 硕士研究生, 主要从事机器学习算法方面的研究。

通信作者: 周长杰, 副教授。E-mail: zhch.jie@163.com

刘子康, 周长杰, 姚卫. 基于最小生成树与统计特征的层次聚类算法[J]. 河北科技大学学报, 2026, 47(1): 49-59.

LIU Zikang, ZHOU Changjie, YAO Wei. Hierarchical clustering algorithm based on minimum spanning tree and statistical features[J]. Journal of Hebei University of Science and Technology, 2026, 47(1): 49-59.

conducted. The result shows that the proposed SHCA algorithm outperforms existing methods in clustering performance; In cases where performance degradation is observed on certain datasets, the analysis reveals that manifold overlap is the primary contributing factor. Overall, SHCA significantly enhances clustering accuracy and result stability, providing some reference for subsequent research on clustering of large-scale and complex manifold data.

**Keywords:** artificial intelligence theory; clustering; hierarchical clustering algorithm; minimum spanning tree; dynamic statistical merging strategy

一般而言,聚类是指将无标签数据集划分为若干类别,使得类内数据相似度高、类间数据相似度低的过程,是一种无监督的机器学习分类方法<sup>[1-2]</sup>。聚类算法大致包括 5 类:基于划分的(如  $k$ -means<sup>[3]</sup>)、基于密度的(如 DBSCAN<sup>[4]</sup>、CTW-DPC<sup>[5]</sup>)、基于网格的(如 STING<sup>[6]</sup>)、基于模型的(如高斯混合模型<sup>[7]</sup>)和基于层次的聚类算法,其中层次聚类算法(hierarchical clustering algorithm, HCA)广泛应用于发现文本间嵌套结构<sup>[8]</sup>、图像中物体的空间层次<sup>[9]</sup>、社交网络中社区演化的动态关联<sup>[10]</sup>以及生物学中物种进化树的构建<sup>[11]</sup>等领域。

Chameleon 算法是一个 HCA<sup>[12]</sup>,其包含自适应相似性度量与两阶段合并策略,在处理非球形簇和不均匀分布方面取得了一定进展。但其仍存在以下主要问题:一是在构建稀疏图时对  $k$  值敏感,当  $k$  取值偏小或偏大时容易导致簇断裂或误连,且在高噪声场景下,由于构建  $k$  近邻图对噪声敏感,会引入错误合并,而合并之后难以取消,易导致后续聚类结果的不合理性;二是构建和更新  $k$  近邻图在大规模或高维数据上计算复杂度剧增;三是相似性度量问题,Chameleon 算法将相对互连性定义为边权重之和,将相对紧密性定义为切割边的平均权重,当簇内密度发生变化时,相对互连性和相对紧密性无法准确反映数据的分布特征,进而导致无法获得良好的聚类效果。近年来,为了解决构建稀疏图问题,大部分通过采用其他算法构建稀疏图进行改进<sup>[13-17]</sup>,对于相似性度量问题,一般采用引入参数描述密度变化<sup>[18-20]</sup>,与此同时,基于最小生成树(minimum spanning tree, MST)的聚类方法<sup>[21]</sup>因其全局最优性和无参数特性受到关注,但尚缺乏结合统计特征进行样本分配的有效策略。

针对上述挑战,本文提出一种基于 MST 与统计特征的层次聚类算法(statistical-MST integrated hierarchical clustering algorithm, SHCA)。在算法设计上,首先利用 Prim 算法构建 MST,代替  $k$  近邻图以消除参数  $k$  的主观影响;随后基于节点度数、子树大小和局部密度等统计特征,设计动态合并策略,对剩余样本进行合理分配,同时抑制噪声干扰。

## 1 相关工作

### 1.1 Chameleon 算法

Chameleon 算法通过数据建模和层次化聚类 2 个阶段识别复杂簇结构。在数据建模阶段,每个数据点转化为图中的节点,边的权重定义为

$$\begin{cases} w(u, v) = \frac{1}{d(u, v)}, \\ d(u, v) = \| \mathbf{x}_u - \mathbf{x}_v \|_2, \end{cases} \quad (1)$$

式中  $\mathbf{x}_u$  和  $\mathbf{x}_v$  是节点  $u$  和  $v$  的特征向量。为了简化图结构并减少噪声,仅保留每个节点的  $k$  个最近邻边,形成  $k$  近邻图。

层次聚类首先将图分割为多个子簇  $\{C_1, C_2, \dots, C_m\}$ ,目标是最小化子簇间的边割,并保证每个子簇至少有  $t$  个节点。然后,基于相对互连性 RI 和相对紧密度 RC 评估子簇相似性。RI 定义为跨簇边割与簇内连接均值的比值,见式(2)。

$$RI(C_i, C_j) = \frac{2 | EC(C_i, C_j) |}{| EC(C_i) | + | EC(C_j) |}, \quad (2)$$

式中:EC 是连接 2 个簇的边的权重之和;EC( $C_i$ ) 指将  $C_i$  大致等分为 2 部分时所切割边的权重之和。RC 衡量的是跨簇边平均权重的归一化形式,见式(3)。

$$RC(C_i, C_j) = \frac{\bar{SEC}(C_i, C_j)}{\frac{| C_i | \bar{SEC}(C_i) + | C_j | \bar{SEC}(C_j)}{| C_i | + | C_j |}}, \quad (3)$$

式中:  $\overline{\text{SEC}}(C_i)$  指将  $C_i$  大致等分为 2 部分时所切割边的平均权重;  $\overline{\text{SEC}}(C_i, C_j)$  是连接  $C_i, C_j$  的边的平均权重;  $|C_i|$  表示  $C_i$  中数据点数量。

综合 RI 和 RC 的得分函数为

$$\text{Score}(C_i, C_j) = \text{RI}(C_i, C_j) \times \text{RC}(C_i, C_j)^\alpha, \quad (4)$$

式中  $\alpha$  是用户指定的参数。得分函数用于指导子簇合并过程, 优先合并得分最高的簇对, 直至达到预定的目标簇数。

## 1.2 MST 算法

为了在层次聚类中实现一种局部自适应的互连机制, 即一种偏向最近邻的机制, 采用 MST 这一强有力的工具。Prim 算法生成的 MST 是一个连通无环图, 它包含图中所有顶点, 并且边权之和最小。在处理复杂数据集时, 其优势显著<sup>[22]</sup>。

从构建角度看, MST 的构建遵循全局最优连通准则, 从数据集中挑选总权重最小的边集, 确保所有节点连通且无环路。对于凝聚型层次聚类, 一旦在合并过程中出错则合并簇难以取消。MST 与  $k$  近邻算法相比, 无需预设邻域数  $k$ , 有效避免了因参数选择不当而引入的伪连接, 能更精准地反映数据的内在拓扑结构。在处理不同密度分布的数据集时, MST 通过最小化全局边权, 利用长边隔离机制, 将跨密度边排除在外, 从而保证了簇内部的紧凑性, 而  $k$  近邻算法可能因局部连接在密度边界处产生跨簇短边, 导致聚类错误。并且, 当簇间距离大于簇内最大距离时, MST 能天然保持簇的连通性, 为后续聚类的动态合并提供了可靠的初始分割基础。

## 2 基于 MST 与统计特征的层次聚类

通过改进 Chameleon 算法中的稀疏图构造方式和相似性度量策略, 提出了基于 MST 及统计特征的 SHCA。对于 SHCA 构造的稀疏图, 给出过滤策略形成初始连通分量, 通过合并策略反复迭代得到初始聚类结果, 经过离群点分配后, 根据用户输入聚类个数, 得到最终聚类结果。聚类流程如图 1 所示。

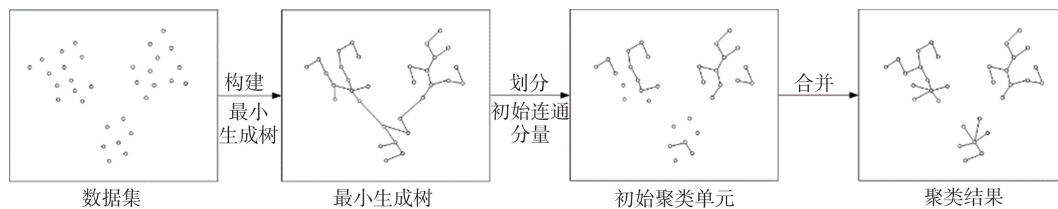


图 1 聚类流程示意图

Fig. 1 Clustering process diagram

### 2.1 基于 MST 的稀疏图构造

为了消除算法参数敏感性以及使算法能够处理非均匀密度分布的数据集, 提出使用 MST 代替  $k$  近邻算法构造稀疏图。这一改进的核心在于利用 MST 的全局最优性和参数无关性, 结合数据集的变异系数计算阈值, 实现更加鲁棒和高效的连通分量划分。MST 的构建遵循全局最优连通准则, 即从数据集中选择总权重最小的边集, 确保所有节点连通且无环路。

**定理 1**(MST 连通性保持定理) 若数据集  $D$  可划分为若干子簇  $\{C_1, C_2, \dots, C_m\}$ , 且满足:

1) 任意子簇  $C_i$  内部的最大分割距离  $\text{diam}(C_i)$  小于其与最近邻子簇  $C_j$  间的最小距离, 即

$$\max_{C_i} (\text{diam}(C_i)) < \min_{x \in C_i, y \in C_j} \| \mathbf{x} - \mathbf{y} \| . \quad (5)$$

2) 子簇间距离函数满足三角不等式。则其 MST 中不存在跨子簇连接边, 即 MST 天然保持簇内连通性。

**证明** 假设跨簇边  $e$  在  $C_i$  中, 此时,  $C_i$  与  $C_j$  存在跨簇边  $e' = \min_{x \in C_i, y \in C_j} \| \mathbf{x} - \mathbf{y} \|$ , 根据条件 1) 应有:

$$e = \max_e (\max_{e \in C_i} (\text{diam}(C_i)), \max_{e \in C_j} (\text{diam}(C_j))) < e' = \min_{x \in C_i, y \in C_j} \| \mathbf{x} - \mathbf{y} \| , \quad (6)$$

即从 2 个子簇中选出最大的边  $e$  应该是小于跨簇边  $e'$  的, 然而  $e > e'$ , 如图 2 b) 所示, 上述假设与条件 1) 矛盾, 因此子簇中不存在跨簇边  $e$ 。

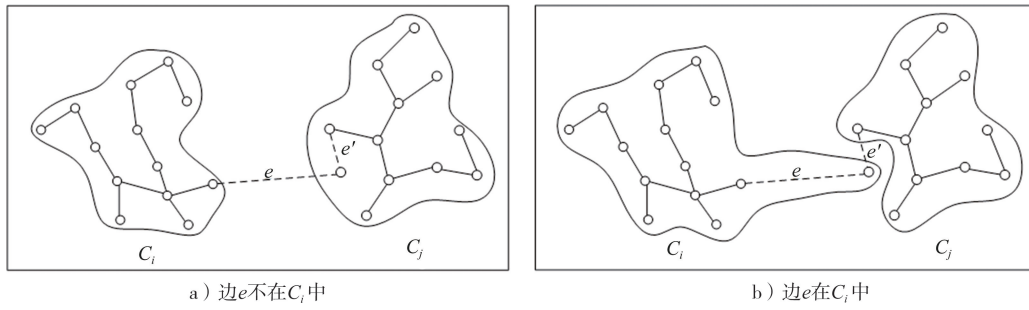


图2 定理1证明示意图

Fig. 2 Illustration of the proof of Theorem 1

此外,Prim 算法生成 MST 的边长是潜在的数据概率密度函数的一维展开表示,表现出波峰和波谷,如图 3 所示,其中波峰部分表示簇间连接边(第 10、21 条边)或数据集噪声边(第 20 条边)。

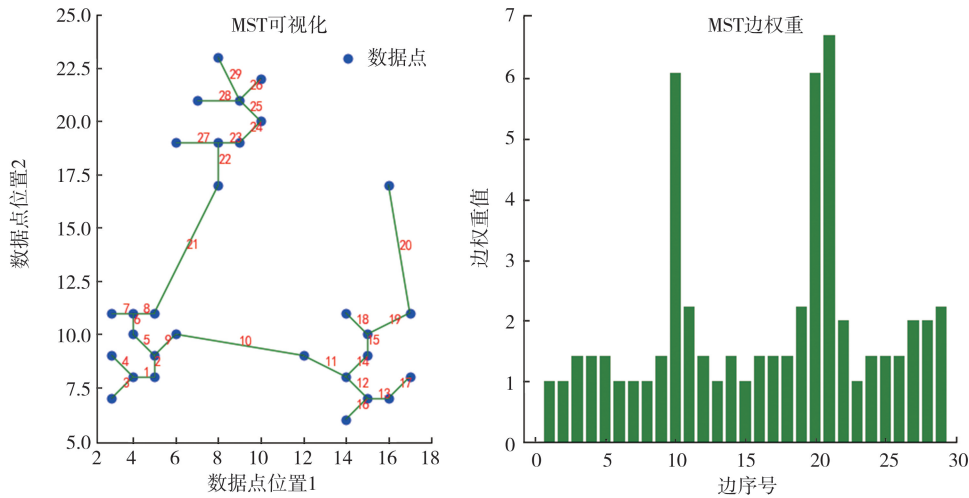


图3 MST 拓扑结构与边权重特性说明

Fig. 3 Description of MST topology and edge weight characteristics

**定理 2**(MST 噪声末端性定理) 对于含噪声数据集  $D = D_c \cup D_n$ , 其中核心数据点  $D_c$  位于紧致区域, 其最近邻距离方差有界, 噪声点  $D_n$  分散于低密度区域, 则其 MST 满足:

1) 噪声点多为叶子节点 若噪声点  $v \in D_n$  满足:

$$\min_{u \in D_c} \|v - u\| > \max_{(p,q) \in \text{MST}(D_c)} \|p - q\| \text{ 且 } \min_{v' \in D_n, v' \neq v} \|v - v'\| > \min_{u \in D_c} \|v - u\|, \quad (7)$$

则  $v$  在 MST 中度数为 1, 是叶子节点, 且通过一条长边与核心簇相连。

2) 噪声边可分离 设  $w_{\max} = \max_{(p,q) \in \text{MST}(D_c)} \|p - q\|$  为核心点集的最大边权重, 则所有连接噪声点与核心点的边权重均满足:

$$\|v - u\| > w_{\max} \quad (\forall v \in D_n, u \in D_c). \quad (8)$$

**证明**

1) 噪声点为叶子节点的证明

反证法: 假设满足条件的噪声点  $v \in D_n$  在 MST 中度数不为 1, 即至少存在 2 条边  $(v, a)$  和  $(v, b)$ , 其中  $a, b \in D_c$ 。

若  $a, b \in D_c$ : 由核心点集的紧致性,  $\text{MST}(D_c)$  中存在连接  $a$  和  $b$  的路径, 其最大边权重为  $w_{\max}$ 。根据条件 1,  $\|v - a\| > w_{\max}$  且  $\|v - b\| > w_{\max}$ , 故  $\|a - b\| \leq w_{\max} < \min(\|v - a\|, \|v - b\|)$ 。此时, 若在 MST 中移除边  $(v, a)$  和  $(v, b)$ , 添加边  $(a, b)$ , 由于  $\|a - b\| < \|v - a\| + \|v - b\|$ , 生成树总权重减少, 与 MST 定义矛盾。

若  $a \in D_c$  且  $b \in D_n$ : 根据条件 1),  $\|v - b\| > \min_{u \in D_c} \|v - u\| = \|v - a\|$ , 且  $\|v - a\| > w_{\max}$ 。若在 MST 中移除边  $(v, b)$ , 添加边  $(b, a)$ , 此时因  $\|b - a\| \leq \|b - v\| + \|v - a\|$ , 且由条件 1) 可知  $\|b - a\| > w_{\max}$ , 此时  $\|b - a\| < \|v - b\|$ , 生成树总权重减少, 与 MST 定义矛盾。

若  $a, b \in D_n$ : 同理, 噪声点间距离  $\|v-a\| > \|v-u\|$  且  $\|v-b\| > \|v-u\| (\forall u \in D_c)$ , 移除 2 条噪声边并替换为  $(a, b)$  会导致总权重减少, 矛盾。

综上, 噪声点  $v$  不可能存在 2 条及以上边, 故度数必为 1, 是叶子节点, 且仅连接核心簇。

## 2) 噪声边可分离的证明

反证法: 假设存在噪声点  $v \in D_n$  和核心点  $u \in D_c$ , 使得边  $(v, u) \in \text{MST}$  且  $\|v-u\| \leq w_{\max}$ 。MST( $D_c$ ) 中存在最大边  $e = (p, q)$ , 满足  $\|p-q\| = w_{\max}$ 。由于  $v \in \text{MST}$ , 已经有 1 条边连接  $v$  与其他点, 又将边  $(v, u)$  加入 MST( $D_c$ ) 会形成包含  $e$  和  $(v, u)$  的环。由于  $\|v-u\| \leq \|p-q\|$ , 移除边  $e$  并保留  $(v, u)$  后, 新生成树总权重更小, 与 MST( $D_c$ ) 是核心点集的最小生成树矛盾。因此, 所有连接噪声点与核心点的边权重必大于  $w_{\max}$ 。

## 2.2 基于最近邻距离的过滤策略

采用局部距离阈值  $\delta$  作为边的过滤条件, 使用  $\delta$  将数据集的 MST 划分为若干个连通分量。每个连通分量由一组紧密相连的节点组成, 代表潜在的簇。当 2 个节点之间的边权重超过  $\delta$  时, 它们不再属于同一个连通分量。聚类过滤算法如表 1 所示。

表 1 聚类过滤算法

Tab.1 Clustering filtering algorithm

算法 1: 聚类过滤算法流程	
输入:	数据集 $D = \{p_1, p_2, \dots, p_n\}$ ;
输出:	连通分量集合 $C = \{C_1, C_2, \dots, C_m\}$ , 离群点集合 $O \subseteq D$ ;
1:	计算点对间距离, 得到距离矩阵 $M$ , 根据式(9)计算平均最近邻距离 $\mu_{NN}$ , 根据式(11)计算局部距离阈值 $\delta$ , 基于 $M$ 构建最小生成树 $T$ ;
2:	for each $e \in T$ do // $e$ 表示 $T$ 的边
3:	if $w(e) > \delta$ then // $w(e)$ 表示边 $e$ 的权重
4:	去掉边 $e$ ;
5:	end if
6:	end for
7:	初始化 $O = \emptyset$ ;
8:	for each $v \in T$ do // $v$ 表示 $T$ 的顶点
9:	if $\text{deg}(v) = 0$ then // $\text{deg}$ 表示节点度数
10:	将 $v$ 添加到 $O$ ;
11:	end if
12:	end for

**定义 1(平均最近邻距离)** 对于每个节点  $v_i$ , 设  $d_i$  表示  $v_i$  到其最近邻的距离(即边权重), 则节点平均最近邻距离  $\mu_{NN}$  定义为

$$\mu_{NN} = \frac{1}{n} \sum_{i=1}^n d_i. \quad (9)$$

平均最近邻距离反映了数据集的局部密度分布特性。具体而言, 较低的值表明数据在局部区域内分布紧凑, 较高的值则说明数据分布稀疏或存在噪声干扰。结合定理 2, 噪声点的最近邻距离通常显著大于核心簇内边的权重, 因此  $\mu_{NN}$  可作为区分核心区域与噪声的参考基准。

**定义 2(变异系数)** 对于均值为  $\mu$ 、标准差为  $\sigma$  的数据集  $D$ , 其变异系数  $c_v$  定义为

$$c_v = \frac{\sigma}{\mu}. \quad (10)$$

变异系数是数据集  $D$  离散程度的相对度量。

**定义 3(局部距离阈值)** 局部距离阈值  $\delta$  定义为

$$\delta = (1 + c_v) \mu_{NN}. \quad (11)$$

$\delta$  可以在保留核心簇完整性的同时, 有效过滤噪声边, 保证聚类结果的紧密性。例如, 当  $c_v = 1.5$  时, 所有边权重超过  $2.5\mu_{NN}$  的 MST 边将被切断, 形成连通分量集合。

## 2.3 基于统计特征的合并策略

在初始子簇生成后, 需基于统计特征迭代合并相似子簇。合并策略的核心思想是: 若 2 个子簇合并后满足簇间连通性检验, 则认为它们应归属同一簇。为了判断 2 个子簇  $C_i$  和  $C_j$  是否应该合并, 本文提出以下连

通性检验准则:

若连接  $C_i$  和  $C_j$  的边  $e_{ij}$  的权重  $w_{ij}$  满足

$$w_{ij} \leq \mu_{i \cup j} + \alpha \cdot \sigma_{i \cup j}, \quad (12)$$

则判定  $C_i$  和  $C_j$  具有连通性,应合并为一个簇。其中: $C_{i \cup j}$  表示合并后的候选簇; $\mu_{i \cup j}$  和  $\sigma_{i \cup j}$  分别表示合并后的候选簇  $C_{i \cup j}$  的边权重均值及标准差; $\alpha$  为合并系数(通常取  $\alpha \in [2.0, 3.0]$ ),用于控制合并的宽松程度。

由定理 2 可知,若跨簇边权重未显著偏离合并后的分布(即未超出  $\mu + \alpha\sigma$ ),则认为 2 个子簇本质属于同一分布,应当合并。此方法可处理不同密度的簇,避免因密度差异导致的误分割。合并策略算法如表 2 所示。

表 2 聚类合并算法

Tab. 2 Clustering merging algorithm

算法 2: 聚类合并算法流程	
输入:	连通分量集合 $C = \{C_1, C_2, \dots, C_m\}$ , 合并系数 $\alpha$ ;
输出:	合并结果 $C_{\text{merge}}$ ;
1:	初始化簇集合 $C_{\text{merge}} = \{C_1, C_2, \dots, C_m\}$ ; //每个连通分量视为独立簇
2:	初始化队列 $Q$ ;
3:	for each $(C_i, C_j) \in C_{\text{merge}}$ do
4:	计算最小连接边权重 $w_{ij}$ , 并将 $(C_i, C_j, w_{ij})$ 按 $w_{ij}$ 升序插入队列 $Q$ ;
5:	end for
6:	While $Q \neq \emptyset$ do
7:	取出 $Q$ 中权重最小元素 $(C_i, C_j, w_{\min})$ ;
8:	模拟合并簇 $C_i$ 和 $C_j$ , 计算合并后的边权重均值 $\mu_{i \cup j}$ 和标准差 $\sigma_{i \cup j}$ ;
9:	if $w_{ij} \leq \mu_{i \cup j} + \alpha \cdot \sigma_{i \cup j}$ then
10:	执行合并操作, 更新簇集合 $C_{\text{merge}}$ ;
11:	重新计算新簇与相邻簇的连接边信息, 更新队列 $Q$ ;
12:	end if
13:	end While
14:	输出 $C_{\text{merge}}$ 作为合并结果。

## 2.4 根据指定类数进行聚类

在层次聚类框架中,经过基于统计特征的合并策略后,算法已生成一组聚类结果  $C_{\text{merge}}$ 。然而,实际应用中用户通常期望指定目标簇数  $k$  以获得特定粒度的聚类结果。为此,本文提出一种自适应调整策略,通过迭代合并最相似的簇对,逐步缩减簇数量至用户指定的目标值。

具体而言,算法在基于 2.3 节的合并策略生成初始聚类集合  $C_{\text{merge}}$  后的工作是:若此时簇数小于  $k$ ,则提示用户最优聚类结果小于指定  $k$  值;若等于  $k$ ,则直接输出结果;若大于  $k$ ,则需进一步合并至  $k$  个簇。合并过程遵循以下准则:优先选择簇间相似性最高的簇对,即跨簇边权重最小的簇对。此阶段的关键在于平衡全局相似性度量与局部密度分布特性,确保合并后的簇既符合用户需求,又保持内部结构的紧密性。

## 2.5 离群点分配策略

离群点的分配需在保证原有簇结构稳定的前提下进行。根据定理 2,离群点仅通过一条长边与核心簇相连,其分配应避免扰动核心簇的拓扑完整性。为此,本文提出一种保守分配策略:

1) 计算离群点到各簇的距离

对于每个离群点  $p$ ,计算其到各簇  $C_i$  的距离。这里的距离是离群点  $p$  到簇  $C_i$  中最近点的距离:

$$d(p, C_i) = \min_{q \in C_i} d(p, q), \quad (13)$$

式中  $d(p, q)$  表示点  $p, q$  之间的距离。

2) 选择最近的簇进行分配

找到离群点  $p$  到所有簇的最小距离:

$$d_{\min}(p) = \min_{C_i} d(p, C_i), \quad (14)$$

将离群点  $p$  分配到距离最近的簇  $C_{\text{final}}$  中:

$$C_{\text{final}} = \arg \min_{C_i} d(p, C_i), \quad (15)$$

式中  $\arg \min$  返回最小值的序号。

### 3) 分配后不更新簇结构

离群点  $p$  分配到簇  $C_{\text{final}}$  后,不更新簇  $C_{\text{final}}$  的结构,即不重新调整簇内其他点的分配。这样可以保持原有聚类的紧密性,防止噪声传播。

## 2.6 SHCA 实现

综合算法的过滤、合并、根据指定类数合并、离群点分配过程,SHCA 流程见表 3。

表 3 SHCA 流程

Tab. 3 SHCA flow

算法 3:SHCA 流程
输入:数据集 $D = \{p_1, p_2, \dots, p_n\}$ , 合并系数 $\alpha$ , 指定聚类数 $k$ ;
输出:最终聚类结果 $C_{\text{final}}$ ;
1: 使用算法 1 得到连通分量集合 $C$ 和离群点集合 $O$ ;
2: 使用算法 2 得到聚类结果 $C_{\text{merge}}$ ;
3: While $ C_{\text{merge}}  > k$ do
4: 初始化队列 $Q$ ;
5: for each $(C_i, C_j) \in C_{\text{merge}}$ do
6: 找到最小连接边权重 $w_{ij}$ , 并将 $(C_i, C_j, w_{ij})$ 按 $w_{ij}$ 升序插入队列 $Q$ ;
7: end for
7: 取出 $Q$ 中权重最小元素 $(C_i, C_j, w_{\min})$ , 合并簇 $C_i$ 和 $C_j$ ;
8: 重新计算新簇与相邻簇的连接边信息, 更新队列 $Q$ ;
9: end While
10: for each $o \in O$ do
10: 根据式(13)计算离群点到各簇的最近距离;
11: 根据式(15)分配离群点;
12: end for
13: 输出最终聚类结果 $C_{\text{final}}$ 。

## 2.7 算法复杂度分析

SHCA 算法的时间复杂度主要取决于构建 MST、过滤策略、合并策略、指定簇数合并,若  $n$  为数据点个数,  $m$  为初始聚类单元个数,那么时间复杂度分别为  $O(n \log n)$ 、 $O(n)$  和  $O(m^2 \log m)$ , 合并至  $k$  簇的时间复杂度为  $O((m-k) \log m)$ , 因此算法的时间复杂度为  $O(n \log n)$ 。

## 3 实验结果与分析

为了检验本文所提算法的准确性,在 30 个数据集上与 Chameleon 算法<sup>[23]</sup>、CURE 算法<sup>[24]</sup>、BIRCH 算法<sup>[23]</sup> 和基于粒球的 GBSC 算法<sup>[25]</sup> 进行对比实验。Chameleon 算法是一种基于图的层次聚类算法,核心流程包括构建  $k$  近邻图、图划分、簇相似度计算、迭代合并簇 4 个阶段,对于包含  $n$  个样本点、每个样本点维度为  $d$  的数据集,时间复杂度分别为  $O(n^2 d)$ 、 $O(kn + n \log n)$ 、 $O(c^2)$ 、 $O(c \log c)$ , 其中  $c$  为被合并的小簇数量,因此当小簇较多时复杂度较高。CURE 算法是一种基于密度和层次的聚类算法,通过选取代表性点来捕捉簇的形状,适用于非球形簇和不同大小的簇。其时间复杂度主要由初始化、簇距离计算、合并操作及代表点更新等步骤决定,对于包含  $n$  个样本点的数据集,时间复杂度分别为  $O(n)$ 、 $O(c^2)$ 、 $\sum_{c=k+1}^n O(c^2)$  和  $O(n)$ , 因此算

法时间复杂度为  $\sum_{c=k+1}^n O(c^2) \approx O(n^3)$ , 其中  $c$  为当前簇数量。BIRCH 算法是一种面向大规模数据集的高效聚类算法,其优势在于通过聚类特征和 CF 树实现数据压缩,从而降低时间和空间开销,其时间复杂度整体为  $O(n)$ 。GBSC 算法是基于粒球的谱聚类算法,核心思想是通过粒球对数据进行多粒度表示,大幅降低传统谱聚类的计算复杂度,核心流程包括粒球生成、相似矩阵构建、谱聚类和标签分配,对于包含  $n$  个样本点、每个样本点维度为  $d$  的数据集,通过构造  $g$  个粒球的聚类过程,时间复杂度分别为  $O(n^2 d)$ 、 $O(g^2 d)$ 、 $O(g^3)$ 、 $O(ng)$ , 若粒球数远小于样本点个数,则算法整体时间复杂度可降低为  $O(n^2 d)$ 。

30 个数据集包括 20 个二维的人工数据集(见表 4)和 10 个来源于 UCI 的真实世界数据集(见表 5),

表4 人工数据集

Tab. 4 Synthetic datasets

数据集	样本数	特征数	类别数	数据集	样本数	特征数	类别数
3MC	300	2	3	Lsun	400	2	3
3-spiral	312	2	3	Shapes	1 000	2	4
Atom	800	2	3	Sizes5	1 000	2	4
Cassini	1 000	2	3	Smile1	1 000	2	4
Complex8	2 551	2	8	Spherical_6_2	300	2	6
Complex9	3 031	2	9	Target	770	2	6
Compound	399	2	6	Triangle2	1 000	2	4
Cure-t0-2000n-2D	2 000	2	3	Wingnut	1 016	2	2
Dartboard2	1 000	2	4	Zelnik3	266	2	3
Donut3	999	2	3	Zelnik5	512	2	4

其中包含有 10 992 个样本的 Pendigits 数据集以及有 1 558 个特征的 Internet-ad 数据集。由于进行对比的算法求得最优解时的参数不尽相同,因此本文采取网格搜索方式确定各个算法的最优参数。

### 3.1 人工数据集上的实验结果

在 20 个人工数据集上对聚类算法性能进行评估,实验结果如表 6 所示(加粗数值表示对应数据集上各对比算法中的最优结果)。实验采用 2 个指标分别从不同维度评估聚类质量:FMI 通过结合精度和召回率来量化聚类结果与真实标

签的匹配程度,其值域为 $[0, 1]$ ,数值越大表示聚类效果越优;AMI 是一种基于信息论的评估方法,用于度量聚类结果与真实标签之间的相关性,其取值范围为 $[-1, 1]$ ,数值越大表明两者一致性越高。由表 6 可以看出,本文所提算法在 Complex8、Complex9、Compound 等数据集上表现都有所提升。对于 3-spiral 和 Atom 数据集,SHCA 因 MST 的全局连通性优势,实现了 100% 的聚类精度(AMI=1.0),而 Chameleon 算法因  $k$  近邻算法构建稀疏图的局部连接碎片化问题,正确率较低。此外,SHCA 在密度差异显著的 Sizes5 数据集中,有效隔离了低密度噪声区域,FMI 为 0.991 5,优于其他对比算法。图 4 的可视化结果进一步验证了 SHCA 对结构非凸数据集的适应性。

表6 人工数据集实验结果

Tab. 6 Result on synthetic datasets

数据集	AMI					FMI				
	SHCA	Chameleon	CURE	BIRCH	GBSC	SHCA	Chameleon	CURE	BIRCH	GBSC
3MC	<b>1.000 0</b>	0.754 2	<b>1.000 0</b>	0.491 2	0.961 9	<b>1.000 0</b>	0.812 9	<b>1.000 0</b>	0.577 4	0.984 7
3-spiral	<b>1.000 0</b>	0.410 7	0.046 6	0.026 7	0.002 3	<b>1.000 0</b>	0.557 3	0.387 0	0.350 2	0.339 8
Atom	<b>1.000 0</b>	<b>1.000 0</b>	0.306 0	0.288 9	0.000 0	<b>1.000 0</b>	<b>1.000 0</b>	0.658 1	0.653 9	0.706 7
Cassini	<b>1.000 0</b>	<b>1.000 0</b>	0.738 4	<b>1.000 0</b>	0.670 7	<b>1.000 0</b>	<b>1.000 0</b>	0.779 0	<b>1.000 0</b>	0.703 0
Complex8	<b>0.944 7</b>	0.742 1	0.701 2	0.580 1	0.715 2	<b>0.927 1</b>	0.626 1	0.611 6	0.467 6	0.647 8
Complex9	<b>0.968 6</b>	0.783 6	0.671 2	0.706 0	0.746 0	<b>0.946 3</b>	0.589 3	0.472 3	0.541 8	0.585 0
Compound	<b>0.804 1</b>	0.742 8	0.736 9	0.720 2	0.419 3	<b>0.830 4</b>	0.614 6	0.673 7	0.637 6	0.512 7
Cure-t0-2000n-2D	<b>1.000 0</b>	0.590 8	0.626 1	0.498 6	<b>1.000 0</b>	<b>1.000 0</b>	0.698 5	0.773 5	0.669 0	<b>1.000 0</b>
Dartboard2	<b>1.000 0</b>	0.577 6	0.532 8	0.463 5	0.000 0	<b>1.000 0</b>	0.632 5	0.586 1	0.539 2	0.499 2
Donut3	<b>1.000 0</b>	<b>1.000 0</b>	0.291 6	0.000 0	0.982 6	<b>1.000 0</b>	<b>1.000 0</b>	0.547 6	0.576 8	0.994 0
Lsun	<b>1.000 0</b>	<b>1.000 0</b>	<b>1.000 0</b>	0.722 7	<b>1.000 0</b>	<b>1.000 0</b>	<b>1.000 0</b>	<b>1.000 0</b>	0.719 2	<b>1.000 0</b>
Shapes	<b>1.000 0</b>	0.842 6	<b>1.000 0</b>	0.974 3	<b>1.000 0</b>	<b>1.000 0</b>	0.816 9	<b>1.000 0</b>	0.984 2	<b>1.000 0</b>
Sizes5	0.952 3	0.546 8	0.634 3	0.898 0	0.952 5	0.991 5	0.655 4	0.755 4	0.785 1	0.992 6
Smile1	<b>1.000 0</b>	<b>1.000 0</b>	0.590 5	0.766 0	0.995 2	<b>1.000 0</b>	<b>1.000 0</b>	0.612 4	0.785 0	0.997 9
Spherical_6_2	<b>1.000 0</b>	0.648 2	<b>1.000 0</b>	0.613 9	0.842 6	<b>1.000 0</b>	0.522 8	<b>1.000 0</b>	0.404 6	0.851 3
Target	<b>1.000 0</b>	0.564 4	0.392 8	0.583 5	0.720 2	<b>1.000 0</b>	0.614 4	0.667 0	0.766 9	0.833 2
Triangle2	<b>0.981 5</b>	0.977 7	0.854 8	0.859 9	0.972 8	<b>0.991 3</b>	0.989 7	0.896 6	0.869 7	0.990 7
Wingnut	<b>1.000 0</b>	<b>1.000 0</b>	<b>1.000 0</b>	0.602 6	0.553 4	<b>1.000 0</b>	<b>1.000 0</b>	<b>1.000 0</b>	0.853 1	0.793 9
Zelnik3	<b>1.000 0</b>	0.686 8	0.519 9	0.666 5	<b>1.000 0</b>	<b>1.000 0</b>	0.690 6	0.609 7	0.671 0	<b>1.000 0</b>
Zelnik5	<b>1.000 0</b>	0.592 8	0.668 3	0.589 5	<b>1.000 0</b>	<b>1.000 0</b>	0.562 2	0.649 6	0.560 2	<b>1.000 0</b>

表5 真实世界数据集

Tab. 5 Real-world datasets

数据集	样本数	特征数	类别数
Breast-cancer-wisconsin	699	9	2
Dermatology	366	34	6
Haberman	306	3	2
Internet-ad	3 279	1 558	2
Iris	150	4	3
Newthyroid	215	5	3
Pendigits	10 992	16	10
WDBC	569	30	2
Wine	178	13	3
Zoo	101	16	7

图 4 中每一行对应一个数据集,每一列对应一个算法。可以看出,本文所提算法在条带状、螺旋状、密度差异较大等的数据集上表现均优于对比算法。

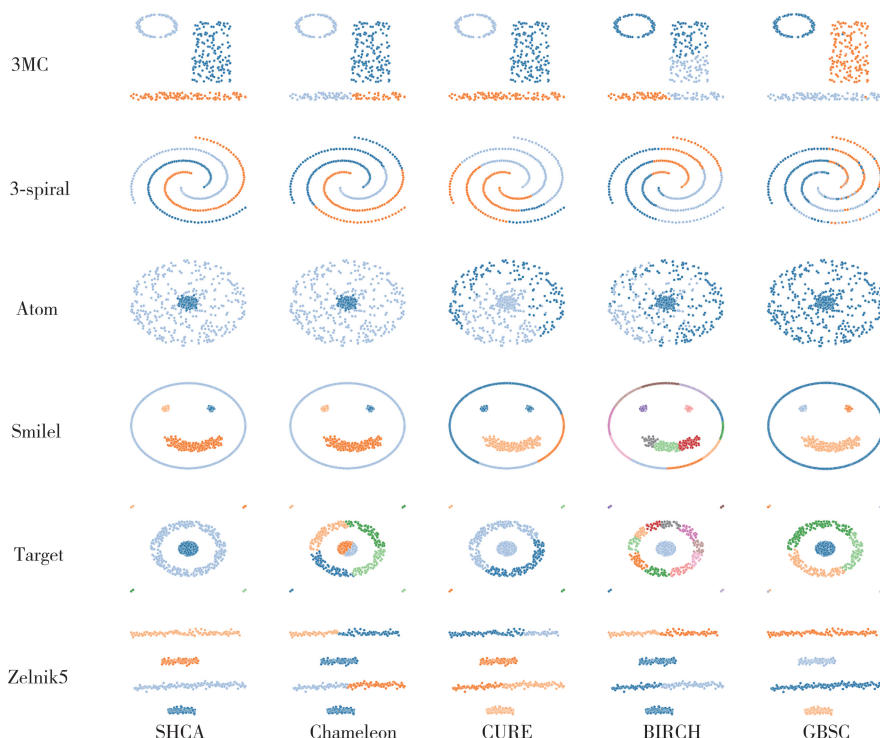


图 4 聚类结果对比图

Fig. 4 Comparison of clustering results

### 3.2 真实世界数据集上的实验结果

表 7 给出了各个算法在 10 个真实世界数据集上的对比结果,其中加粗内容为最优结果,括号中的值表示每行指标值的排名。SHCA 的数值相较于对比算法在大部分数据集上都有所提升。例如,在 Iris 数据集中,

表 7 真实世界数据集上的实验结果

Tab. 7 Results on real-world datasets

数据集	评价指标	SHCA	Chameleon	CURE	BIRCH	GBSC
Breast-cancer-wisconsin	AMI	0.669 8(3)	<b>0.811 4(1)</b>	0.524 1(4)	0.518 2(5)	0.731 6(2)
	FMI	0.897 3(3)	<b>0.949 9(1)</b>	0.840 3(4)	0.829 7(5)	0.923 9(2)
Dermatology	AMI	<b>0.656 7(1)</b>	0.342 2(3)	0.259 4(4)	0.099 0(5)	0.600 1(2)
	FMI	<b>0.642 4(1)</b>	0.352 8(3)	0.306 3(4)	0.230 7(5)	0.536 4(2)
Haberman	AMI	<b>0.069 3(1)</b>	0.002 0(3)	0.011 3(2)	0.000 2(4)	-0.001 7(5)
	FMI	<b>0.761 6(1)</b>	0.566 8(4)	0.758 6(2)	0.634 6(3)	0.550 2(5)
Internet-ad	AMI	<b>0.343 3(1)</b>	0.041 6(4)	0.322 1(3)	0.334 4(2)	0.005 9(5)
	FMI	<b>0.888 4(1)</b>	0.856 5(4)	0.882 0(3)	0.885 6(2)	0.854 2(5)
Iris	AMI	<b>0.828 7(1)</b>	0.803 2(2)	0.709 3(4)	0.701 2(5)	0.710 1(3)
	FMI	<b>0.899 9(1)</b>	0.840 7(2)	0.759 3(4)	0.751 5(5)	0.771 2(3)
Newthyroid	AMI	0.245 3(4)	0.282 6(3)	0.119 5(5)	0.423 9(2)	<b>0.446 2(1)</b>
	FMI	0.759 9(3)	0.607 5(5)	0.742 2(4)	0.798 6(2)	<b>0.809 5(1)</b>
Pendigits	AMI	0.542 8(2)	<b>0.581 9(1)</b>	0.463 9(5)	0.526 3(3)	0.496 6(4)
	FMI	0.456 0(2)	<b>0.441 0(1)</b>	0.382 8(5)	0.407 1(3)	0.394 4(4)
WDBC	AMI	<b>0.625 0(1)</b>	0.413 5(2)	0.040 2(5)	0.318 0(3)	0.077 9(4)
	FMI	<b>0.8732(1)</b>	0.717 0(2)	0.725 0(4)	0.739 2(3)	0.688 9(5)
Wine	AMI	0.5697(2)	0.413 8(3)	0.389 4(5)	0.409 9(4)	0.820 1(1)
	FMI	0.700 1(2)	0.576 2(5)	0.640 9(3)	0.582 1(4)	<b>0.900 4(1)</b>
Zoo	AMI	<b>0.757 7(1)</b>	0.665 7(4)	0.747 6(2)	0.706 8(3)	0.551 9(5)
	FMI	<b>0.718 1(1)</b>	0.592 9(4)	0.7180(2)	0.703 4(3)	0.507 7(5)
AMI 平均排名		1.7	2.6	3.9	3.6	3.2
FMI 平均排名		1.6	3.1	3.5	3.5	3.3

SHCA 的 AMI 和 FMI 分别为 0.828 7 和 0.899 9, 排名第一, 优于 Chameleon 算法的 0.803 2 和 0.840 7, 主要归因于其动态统计合并策略对类别重叠的鲁棒性。虽然在 Breast-cancer-wisconsin、Newthyroid、Pendigits 等数据集上的表现不如对比算法, 但总体表现稳定, 且排名靠前, 与最优评价指标结果相差不大。通过评价指标平均排名可以看出, SHCA 的 FMI 平均排名为 1.6, AMI 平均排名为 1.7, 优于其他算法。

然而, 对于真实世界数据集, 不同的类别会有所重叠, CBO 值<sup>[26]</sup>可以衡量数据集的重叠程度:

$$\text{CBO} = \frac{\sum_{(i,j) \in \text{CL}} s(x_i, x_j) + \sum_{\substack{(i,j) \in \text{CL} \\ (k,l) \in \text{ML}}} \min\{s(x_i, x_k)s(x_j, x_l), s(x_i, x_l)s(x_j, x_k)\}}{\sum_{\substack{(a,b) \in \text{ML} \cup \text{CL} \\ (c,d) \in \text{ML} \cup \text{CL}}} \min\{s(x_a, x_c)s(x_b, x_d), s(x_a, x_d)s(x_b, x_c)\}}, \quad (16)$$

式中: ML 表示应同簇的样本对; CL 表示应异簇的样本对;  $s(x_i, x_j)$  表示相对相似度。

由图 5 可知, Haberman、Internet-ad、Wine 等数据集的 CBO 值较高, 因此易出现簇边界错分、簇内纯度低、对超参数敏感等问题, 导致聚类结果差, 尤其对于标签型数据集, 重叠部分的流形连通且密度较大, 因此会造成误分类问题。

#### 4 结 语

通过引入 MST 替代  $k$  近邻算法, SHCA 消除了人工参数干预, 提升了算法对非均匀密度数据分布的适应性。

MST 的全局最优性和稀疏性不仅确保了初始分割阶段的簇内连通性, 还显著降低了计算复杂度。通过引入局部距离阈值作为过滤机制, 结合动态统计合并策略, 实现了跨簇相似性的无监督评估, 能够有效区分噪声与核心簇。此外, 离群点的保守分配策略进一步保障了核心簇结构的稳定性。所提算法为处理复杂流形数据提供了有效方案。

然而, 在某些真实世界数据集中, 由于类别之间的重叠导致了 SHCA 部分误分类问题, 未来拟研究如何处理数据集中的流形连通性和密度变化问题, 以提升 SHCA 在真实复杂场景中的表现。

#### 参考文献/References:

- [1] 章永来, 周耀鉴. 聚类算法综述[J]. 计算机应用, 2019, 39(7): 1869-1882.  
ZHANG Yonglai, ZHOU Yaojian. Review of clustering algorithms[J]. Journal of Computer Applications, 2019, 39(7): 1869-1882.
- [2] EZUGWU A E, IKOTUN A M, OYELADE O O, et al. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects[J]. Engineering Applications of Artificial Intelligence, 2022. DOI: 10.1016/j.engappai.2022.104743.
- [3] MACQUEEN J B. Some methods for classification and analysis of multivariate observations[C]//Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. California: University of California Press, 1967: 281-297.
- [4] 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 Second International Conference on Knowledge Discovery and Data Mining. Portland: AAAI Press, 1996: 226-231.
- [5] 王江元, 蔡明杰, 祁明月. 基于连通性的三支密度峰值聚类算法[J]. 河北科技大学学报, 2023, 44(6): 632-640.  
WANG Jiangyuan, CAI Mingjie, QI Mingyue. A three-way density peaks clustering algorithm based on connectivity[J]. Journal of Hebei University of Science and Technology, 2023, 44(6): 632-640.
- [6] WANG Wei, YANG Jiong, MUNTZ R R. Sting: A statistical information 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.
- [7] DEMPSTER A P, LAIRD N M, RUBIN D B. Maximum likelihood from incomplete data via the EM algorithm[J]. Journal of the Royal Statistical Society: Series B (Methodological), 1977, 39(1): 1-22.
- [8] YANG Ben, XUE Zhiyuan, WU Jinghan, et al. Anchor-graph regularized orthogonal concept factorization for document clustering[J]. Neurocomputing, 2024. DOI: 10.1016/j.neucom.2023.127173.
- [9] MITTAL H, PANDEY A C, SARASWAT M, et al. A comprehensive survey of image segmentation: Clustering methods, performance

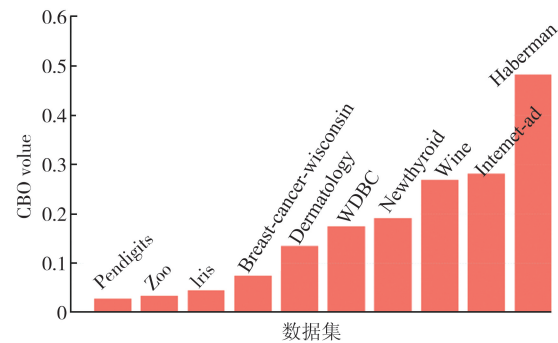


图 5 数据集重叠指标 CBO 值图

Fig. 5 Dataset overlap metrics: CBO value graph

- parameters, and benchmark datasets[J]. *Multimedia Tools and Applications*, 2022, 81(24): 35001-35026.
- [10] LIU Peide, ZHANG Kuo, WANG Peng, et al. A clustering-and maximum consensus-based model for social network large-scale group decision making with linguistic distribution[J]. *Information Sciences*, 2022, 602: 269-297.
- [11] KARIM M R, BEYAN O, ZAPPA A, et al. Deep learning-based clustering approaches for bioinformatics[J]. *Briefings in Bioinformatics*, 2021, 22(1): 393-415.
- [12] KARYPIS G, HAN E H, KUMAR V. Chameleon: Hierarchical clustering using dynamic modeling[J]. *Computer*, 1999, 32(8): 68-75.
- [13] CAO Xiaoxiao, SU Tianyun, WANG Pengyu, et al. An optimized chameleon algorithm based on local features[C]//*Proceedings of the 2018 10th International Conference on Machine Learning and Computing*. New York: Association for Computing Machinery, 2018: 184-192.
- [14] ZHANG Yuru, DING Shifei, WANG Lijuan, et al. Chameleon algorithm based on mutual  $k$ -nearest neighbors[J]. *Applied Intelligence*, 2021, 51(4): 2031-2044.
- [15] ZHANG Yuru, DING Shifei, WANG Yanru, et al. Chameleon algorithm based on improved natural neighbor graph generating sub-clusters [J]. *Applied Intelligence*, 2021, 51(11): 8399-8415.
- [16] 张添翼, 闫飞. 基于改进变色龙算法的交通控制子区划分方法[J]. *计算机工程与设计*, 2025, 46(1): 15-22.  
ZHANG Tianyi, YAN Fei. Traffic control sub-area division method based on improved chameleon algorithm[J]. *Computer Engineering and Design*, 2025, 46(1): 15-22.
- [17] SINGH P, AHUJA K. Chameleon2++: An efficient chameleon 2 clustering with approximate nearest neighbors[EB/OL]. (2025-01-05) [2025-06-25]. <https://arxiv.org/abs/2501.02612>.
- [18] BARTON T, BRUNA T, KORDIK P. Chameleon 2: An improved graph-based clustering algorithm[J]. *ACM Transactions on Knowledge Discovery From Data*, 2019, 13(1): 1-27.
- [19] DONG Yuxin, WANG Ye, JIANG Kai. Improvement of partitioning and merging phase in chameleon clustering algorithm[C]//*2018 3rd International Conference on Computer and Communication Systems (ICCCS)*. Nagoya: IEEE, 2018: 29-32.
- [20] GUO Dongwei, ZHAO Jingjing, LIU Jici. Research and application of improved chameleon algorithm based on condensed hierarchical clustering method[C]//*Proceedings of the 2019 8th International Conference on Networks, Communication and Computing*. New York: Association for Computing Machinery, 2019: 14-18.
- [21] 林钰莹, 侯新民. 基于局部密度峰和标签传播的最小生成树聚类[J]. *计算机系统应用*, 2024, 33(8): 18-29.  
LIN Yuying, HOU Xinmin. Minimum spanning tree clustering based on local density peak and label propagation[J]. *Computer Systems & Applications*, 2024, 33(8): 18-29.
- [22] MA Yan, LIN Hongren, WANG Yan, et al. A multi-stage hierarchical clustering algorithm based on centroid of tree and cut edge constraint[J]. *Information Sciences*, 2021, 557: 194-219.
- [23] ZHANG Tian, RAMAKRISHNAN R, LIVNY M. BIRCH: An efficient data clustering method for very large databases[J]. *ACM SIGMOD Record*, 1996, 25(2): 103-114.
- [24] GUHA S, RASTOGI R, SHIM K. CURE: An efficient clustering algorithm for large databases[J]. *Information Systems*, 1998, 27(2): 73-84.
- [25] XIE Jiang, KONG Weiyu, XIA Shuyin, et al. An efficient spectral clustering algorithm based on granular-ball[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2023, 35(9): 9743-9753.
- [26] ADAM A, BLOCKEEL H. Constraint-based measure for estimating overlap in clustering[C]//*Proceedings of the Twenty-Sixth Benelux Conference on Machine Learning*. Eindhoven: Leuven University Press, 2017: 54-61.