计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(10):147−156 [doi: 10.15888/j.cnki.csa.009263]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

# 基于共享近邻和优化关联策略的边界剥离聚类①

冯洁净[1], 侯新民[1,2,3]

[1](中国科学技术大学 大数据学院, 合肥 230026)
[2](中国科学技术大学 数学科学学院, 合肥 230026)
[3](中国科学院吴文俊数学重点实验室 (中国科学技术大学), 合肥 230026)
通信作者: 侯新民, E-mail: xmhou@ustc.edu.cn

**摘　要**: 边界剥离聚类算法 (BP) 是一种基于密度的聚类算法, 它通过逐渐剥离边界点来揭示聚类的潜在核心, 已经被证明是一种十分有效的聚类手段. 然而, BP 算法仍存在一些不足之处: 一方面, 数据点的局部密度仅考虑了距离特征, 使得边界点的确定不够合理; 另一方面, BP 算法中的关联策略容易误判异常值, 并且在分配边界点时容易产生连带错误. 为此, 本文提出了一种基于共享近邻和优化关联策略的边界剥离聚类算法 (SOBP). 该算法使用了基于共享近邻的局部密度函数来更好地探索数据点之间的相似性, 同时优化了 BP 算法中的关联策略, 使得每次迭代中边界点不再仅与一个非边界点进行关联, 并进一步采用了边界点与非边界点、已剥离边界点之间的双重关联准则. 在一些数据集上的测试表明, 相较于其他 6 种经典算法, 该算法在评估指标上表现更佳.

**关键词**: 边界剥离聚类算法; 共享近邻; 局部密度; 关联策略

引用格式: 冯洁净,侯新民.基于共享近邻和优化关联策略的边界剥离聚类.计算机系统应用,2023,32(10):147–156. http://www.c-s-a.org.cn/1003-3254/9263.html

## Border Peeling Clustering Based on Shared Nearest Neighbors and Optimized Association Strategy

FENG Jie-Jing[1], HOU Xin-Min[1,2,3]

[1](School of Data Science, University of Science and Technology of China, Hefei 230026, China)
[2](School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, China)
[3](CAS Key Laboratory of Wu Wen-Tsun Mathematics (University of Science and Technology of China), Hefei 230026, China)

**Abstract**: The border peeling (BP) clustering algorithm is a density-based clustering algorithm. It gradually peels up border points to reveal the potential cores of clusters and has been proven to be an effective clustering algorithm. However, the BP algorithm has some limitations. On the one hand, the local density of data points only considers the distance characteristics, which can lead to the unreasonable determination of border points. On the other hand, the association strategy of the BP algorithm is prone to misjudge outliers and can generate associated errors when border points are allocated. Hence, this study proposes a BP clustering algorithm based on shared nearest neighbors and optimized association strategy (SOBP). The algorithm employs a local density function based on shared nearest neighbors to better explore the similarity between data points. Meanwhile, the association strategy of the BP clustering algorithm is optimized so that in each iteration, border points are no longer associated with only one non-border point. Furthermore, a double association criterion between border points and non-border points as well as between border points peeled up is utilized. Tests on several datasets show that the proposed algorithm outperforms six other classical algorithms in terms of evaluation indexes.

**Key words**: border peeling clustering algorithm; shared nearest neighbors; local density; association strategy

① 基金项目: 国家自然科学基金 (12071453); 量子通信与量子计算机重大项目 (2021ZD0302904)
收稿时间: 2023-03-21; 修改时间: 2023-04-20; 采用时间: 2023-05-11; csa 在线出版时间: 2023-08-21
CNKI 网络首发时间: 2023-08-22

在机器学习和数据挖掘领域, 聚类是一种被广泛使用的技术, 其根据数据之间的内在结构和关系将数据点分配至多个子集或类簇, 使得同一子集或类簇中的数据点的相似性高. 聚类技术有助于研究人员更好地理解数据, 并从中提取出有用的信息, 可用来解决图像分割[1]、社交网络[2]、生物信息学[3,4] 等领域的各种问题.

当前常见的聚类算法大致分为 5 类: 基于层次的方法、基于划分的方法, 这是由 Fraley 等人[5] 给出的分类, 后续 Han 等人[6] 补充增加了基于模型、基于网格以及基于密度的方法. 这些聚类算法被广泛应用于人工智能领域, 并受到许多研究人员的深入研究和探索 (可参见相关综述[7,8]).

最近, Averbuch-Elor 等人提出了边界剥离聚类算法 (BP), 一种基于密度的聚类方法[9]. 在以往的研究中, 许多基于密度的聚类算法假设可以通过密度推理确定聚类的核心. 然而在实际应用中, 直接定义聚类核心的结构密度非常困难. 与传统的基于密度的方法相比, BP 算法通过迭代剥离边界点来揭示聚类的潜在核心, 在许多数据集上表现良好, 并且可在图像分类以及异常值检测等领域进行应用. 但是对于具有复杂空间分布的数据集, BP 算法仍需要改进, 首先 BP 算法采用局部密度函数度量两个数据点之间的相似性, 仅考虑了欧氏距离以及数据点与其第 $k$ 个最近邻居之间的距离, 使得边界点的判定容易受到异常值的干扰, 并且无法适应相关点的邻域. 其次边界点被剥离后只与固定距离内最近的非边界点相关联, 否则会被标记为异常值, 这种关联策略会导致 BP 算法在空间分布复杂的数据集上容易进行过度划分, 并且对于一些分布不均匀的数据集容易误判异常值.

针对以上问题, 本文提出了一种基于共享近邻和优化关联策略的边界剥离聚类算法 (SOBP), 以缓解现有算法中的一些问题. 该算法主要包含两个方面的贡献: 一是局部密度的改进: 采用一个新的局部密度函数, 该函数基于共享近邻以及数据点与其 $k$ 个最近邻居之间的平均距离, 不仅考虑了相关点的邻域, 并且可以减少异常值的干扰; 二是关联策略的优化: SOBP 算法改进了连接阈值, 使得每次迭代中边界点不再仅与一个非边界点相关, 而是考虑其非边界点 $k$ 近邻在一个更合理的距离区间内与边界点建立联系, 并通过共享近邻相似性度量来构建边界点之间的联系, 从而有效减少异常值的误判.

## 1 相关工作

数据聚类是一个具有永恒魅力的研究领域. 自聚类算法问世以来, 许多学者提出了各种聚类算法并将其用于解决一些实际问题, 其中基于密度的方法一直是聚类研究的重点.

DBSCAN[10] 是最常被引用的基于密度的聚类算法之一, 其原则是: 每个类簇的密度都高于周围的密度, 噪声的密度低于任何类簇的密度, 因此它也可以用于离群点检测. DBSCAN 有两个重要参数: eps 和 minpts, 在聚类过程中, 这两个参数选择不当将会导致聚类质量的下降. HDBSCAN[11] 对 DBSCAN 进行了扩展, 它使用 $k$ 最近邻定义密度水平, 将算法参数转换为固定阈值 (邻居数 $k$). 很多研究者研究了 DBSCAN 的参数自适应. BSA-DBSCAN[12] 使用鸟群方法的全局搜索能力选择最佳 eps 参数值. Chen 等人通过 AP 算法生成一个密度列表, 以此列表作为 DBSCAN 算法的输入, 提出了一种无参数聚类算法 APSCAN[13]. 万佳等人[14] 利用去噪衰减后的数据生成了 eps 和 minpts 参数列表, 并根据去噪的水平获得初始参数值, 提出一种多密度自适应参数确定算法.

2014 年, Rodriguez 等人[15] 提出了 DPC 算法, 一种非常流行的密度聚类算法. 它使用数据集的密度峰值来找到类簇中心, 可以实现对任意形状数据的高效聚类. 近年来, 许多学者都在努力改进该算法. 例如, SNN-DPC 算法[16] 引入了共享最近邻来对局部密度进行估计; RDPC-DSS 算法[17] 使用了新的密度聚类指数 (DCI) 进行簇中心数的确定; 张清华等人[18] 通过构造 $k$ 近邻密度, 提出了新的局部密度, 并采用了一种加权的 $k$ 近邻分配策略, 提出了基于代表点与 $k$ 近邻的密度峰值聚类算法.

BP 算法是一种创新的基于密度的算法, 通过迭代确定边界点, 并逐步获得类簇的核心. 最近, Du 等人[19] 设计了一种新的连接准则, 在剥离的边界点和它们相邻的已剥离边界点之间建立联系, 并使用具有更长尾部的柯西核来衡量数据点之间的相似度, 从而减少算法的运行时间, 提出了鲁棒的基于柯西核的边界剥离聚类算法. 此外, Feng 等人[20] 提出了基于自然近邻的边界剥离聚类算法 (SANBP), 使用自然邻居获得邻域信息作为边界剥离聚类算法的输入, 实现了 BP 算法的

无参化.

## 2 算法原理及分析

### 2.1 BP 算法基本原理

给定数据集$X=\{x_1,x_2,\cdots,x_n\}\subset\mathbb{R}^d$以及输入参数$k$ (最近邻的数量), BP 算法通过对边界点进行一步一步地迭代剥离, 获得聚类中心区域进行聚类. 首先给出以下符号定义: $X^{(t)}$表示在第 $t$ 次迭代开始前还未被剥离的数据点集; $nn_k(x_i)$表示数据集中距离数据点$x_i$第$k$近的数据点; 使用$NN_k(x_i)$表示数据点$x_i$在数据集 $X$ 中的 $k$ 近邻:

$$NN_k(x_i)=\bigcup_{j=1}^{k}\left\{nn_j(x_i)\right\} \tag{1}$$

$RNN_k(x_i)$ 表示数据点$x_i$在数据集 $X$ 中的反向 $k$ 近邻:

$$RNN_k(x_i)=\{x_j|x_i\in NN_k(x_j)\} \tag{2}$$

$NN_k^{(t)}(x_i)$, $RNN_k^{(t)}(x_i)$分别表示数据点$x_i\in X^{(t)}$的 $k$ 近邻以及反向 $k$ 近邻.

对于给定的两个数据点$x_i,x_j\in X^{(t)}$, BP 算法使用高斯核函数来度量数据点之间的距离, 其具体定义如下:

$$f\left(x_i,x_j\right)=\exp\left(-\frac{d^2\left(x_i,x_j\right)}{\sigma_j^2}\right) \tag{3}$$

其中, $\sigma_j$表示在第 $t$ 次迭代中$x_j$与其距离第 $k$ 近的邻居之间的距离, 即$\sigma_j=d\left(x_j,nn_k^{(t)}\left(x_j\right)\right)$.

在考虑了一个数据点对其邻近点的局部密度的影响后, BP 算法定义了一个密度影响值$b_i^{(t)}$, 对于任意点$x_i\in X^{(t)}$, 密度影响值与其距离聚类中心的距离成正比, 计算公式为:

$$b_i^{(t)}=\sum_{x_j\in RNN_k^{(t)}(x_i)}f\left(x_i,x_j\right) \tag{4}$$

BP 算法的主要思想是迭代剥离边界点, 因此在第 $t$ 次迭代时, 通过对$b_i^{(t)}$设置截断值来进行边界点的剥离, 即首先将所有数据点的密度影响值$b_i^{(t)}$由小到大进行排列, 排列后位于前 10% 的密度影响值$b_i^{(t)}$所对应的数据点将会被判定为边界点, 并进行剥离, 这些数据点所构成的数据集为$X_B^{(t)}$. 使用$NN_{kCB}^{(t)}(x_i)$表示对于数据点$x_i\in X_B^{(t)}$, 截至当前迭代过程中已被剥离且未被识别为异常值的数据点中距离$x_i$最近的 $k$ 个邻居所构成的数据集. $D_k$则表示数据集$X$中所有数据点与其 $k$ 个邻居的所有成对距离的集合 ($D_k=\bigcup_{x_i\in X}\{d(x_i,x_j)|x_j\in NN_k(x_i)\}$). BP 算法通过设置阈值$l_i$进行边界点与非边界点之间的关联, 对于任意点$x_i\in X_B^{(t)}$, $l_i$值的定义如下:

$$l_i=\min\left\{\frac{C}{k}\sum_{x_j\in NN_{kCB}^{(t)}(x_i)}d\left(x_i,x_j\right),\lambda\right\} \tag{5}$$

其中, $\lambda=mean(D_k)+std(D_k)$.

BP 算法中经验性地将$C$设置为 3, 如果$x_i\in X_B^{(t)}$与其距离最近的非边界点$x_j\in X^{(t+1)}$之间的距离不大于$l_i$值, 则将$x_i$与$x_j$相关联, 若距离大于$l_i$值, 则$x_i$会被标记为异常值.

最后, 在迭代剥离结束后, 剩余的未剥离点是核心点. 通过迭代过程, 每个剥离点 (除了异常值) 都与核心点有传递性的关联, BP 算法采用简化的 DBSCAN 版本将核心点分配到类簇中, 并根据剥落的边界点与核心点之间的关联将已剥离数据点分配到类簇中.

### 2.2 BP 算法分析

首先, 在 BP 算法中, 边界点的识别是非常重要的. 只有合理地识别边界点才能够进一步得到有效的聚类核心区域. 边界点是由局部密度函数所定义的密度影响值确定的, 两个数据点$x_i$, $x_j\in X$之间的局部密度函数基于高斯核, 其中局部密度的尺度参数由$x_j$与其$nn_k\left(x_j\right)$ (距离数据点$x_j$的第 $k$ 近的邻居) 之间的距离决定的, 其并没有使用到 $k$ 个邻居的全部信息, 很容易受到异常值的影响. 同时局部密度的尺度参数为欧氏距离, 仅体现了数据点之间的距离特征. 但是, 两个数据点被划分到同一类簇, 不仅因为它们在距离上接近, 而且因为在大多数情况下它们属于相同的高密度区域. 因此, 边界点的识别应该考虑数据之间的空间特征, 以适应更多样化的情况.

其次, 边界点与非边界点之间的联系与变量阈值$l_i$的设置密切相关, 在 BP 算法中对于阈值$l_i$值的设置是出于经验性的设置 (采用 $C$=3). 如图 1 所示, 在 Two-circle 数据集上, 当我们改变 $C$ 的取值时, 将 $C$ 分别设置为 $C$=2, $C$=3 以及 $C$=4 (所有展示结果均是参数调优后的最好结果), 边界剥离聚类算法在 Two-circle 数据集上的聚类结果也会因为 $C$ 值的改变而受到很大的影响, 当选用 $C$=2 时, 对于内外两个环状类簇, BP 算法均进行了过度分割; 当选用 $C$=3 时, BP 算法可以完全识别外侧的环状类簇, 但针对内侧的环状类簇划分为

5 个类簇, 并将部分数据点判断为了异常值; 当选用 $C$=4 时, BP 算法没有进行异常值的误判, 并且对于内侧的环状类簇过度划分的类簇更少.

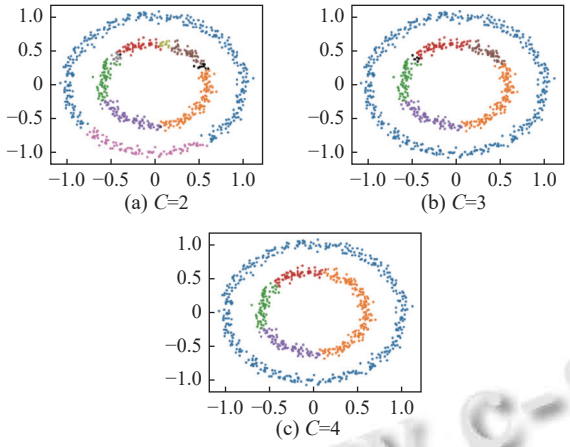


图 1 BP 算法在 Two-circle 数据集上的聚类结果

同时 BP 算法对于密度不均匀的数据集很难选择出最优的分区结构. 以图 1 为例, BP 算法在 Two-circle 数据集上的聚类结果与该数据集的实际分布有着较大差异, BP 算法过度分割了该数据集. 这可能是由于每个边界点 (除了异常值) 只连接到最近的非边界点, 如果同一类簇的两个位于内部的已剥离边界点连接到两个不同的区域, 那么在分配后续的边界点时, 也可能导致连续性的错误, 另外, 如果边界点未能在可变阈值 $l_i$内找到最近的非边界点, 那么它将被视为一个异常值, 事实上, 这方式只考虑了欧氏距离, 非常容易误判异常值, 特别是在第 1 次迭代中, 如图 2 所示, BP 算法将类簇边界上的一些点识别为异常值, 进一步影响了聚类结果的准确性.

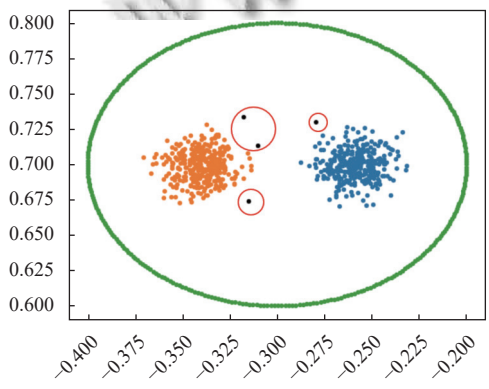

图 2 BP 算法在 Donut3 数据集上的聚类结果

### 2.3 改进的 BP 算法

本节将详细介绍我们提出的改进的 BP 算法, 即基于共享近邻和优化关联策略的边界剥离聚类算法 (SOBP). 首先, 我们将介绍一些重要的概念, 再基于这些概念描述算法的详细流程.

定义 1 (共享近邻[21]). 给定数据点$x_i, x_j \in X$, 其共享近邻$SNN\left(x_i, x_j\right)$指的是在$x_i$的 $k$ 近邻中同时是$x_j$的 $k$ 近邻的数据点构成的集合:

$$SNN\left(x_i, x_j\right)=\left\{x_r | x_r \in NN_k\left(x_i\right) \cap NN_k\left(x_j\right)\right\} \tag{6}$$

为了更好地估计两个数据点之间的相似性, 我们采用了一个基于平均距离和共享近邻的局部密度函数.

定义 2 (局部密度函数). 对于任意数据点$x_i, x_j \in X^{(t)}$, 两个点之间的局部密度被定义如下:

$$f\left(x_i, x_j\right)=\exp\left(-\frac{d^2\left(x_i, x_j\right)}{\sigma_j^2\left(\left|SNN^{(t)}\left(x_i, x_j\right)\right|+1\right)}\right) \tag{7}$$

其中, $\sigma_j^2=\frac{1}{k} \sum_{x_r \in NN_k^{(t)}\left(x_j\right)} d^2\left(x_j, x_r\right)$, $\left|SNN^{(t)}\left(x_i, x_j\right)\right|$ 表示在第 $t$ 次迭代中$x_i$与$x_j$的共享近邻的个数, 在这种度量下, 两个数据点之间的距离越小或者两个数据点之间拥有的共享邻居数量越多, 则该局部密度函数值越大.

定义 3 (密度影响值). 与式 (4) 中相同, 给定数据点$x_i \in X^{(t)}$, $x_i$的密度影响值为其反向 $k$ 近邻的局部密度函数值之和:

$$b_i^{(t)}=\sum_{x_j \in RNN_k^{(t)}\left(x_i\right)} \exp\left(-\frac{d^2\left(x_i, x_j\right)}{\sigma_j^2\left(\left|SNN^{(t)}\left(x_i, x_j\right)\right|+1\right)}\right) \tag{8}$$

定义 4 (边界分类值). 对于数据点$x_i \in X^{(t)}$, 若$x_i$的密度影响值不超过给定的截断值, 则数据点$x_i$被判定为边界点, 且其边界分类值$B_i^{(t)}$定义为 1, 用数学语言表示如下:

$$B_i^{(t)}=\begin{cases}1, & b_i^{(t)} \leqslant b_{rnd(10\%)}^{(t)} \\ 0, & \text{otherwise}\end{cases} \tag{9}$$

其中, $b_{rnd(10\%)}^{(t)}$使得每次迭代中 90% 的剩余数据点具有更大的$b_i^{(t)}$值. 在第 $t$ 次迭代中, 算法将根据边界分类值$B_i^{(t)}$进行边界点的识别, 当$B_i^{(t)}=1$时将该点剥离.

定义 5 (当前剥离点集). 当前剥离点集使用$X_{CB}^{(t)}$表

示, 指截至第 $t$ 次迭代, 被剥离的数据点中未被识别为异常值的数据点构成的集合 (包括第 $t$ 次迭代).

定义 6 (边界点 $k$ 近邻). 对于数据点$x_i \in X_B^{(t)}$, 其边界点 $k$ 近邻是指当前剥离点集中距离$x_i$最近的 $k$ 个数据点构成的集合, 使用$NN_{k_{CB}}^{(t)}(x_i)$表示.

为了建立数据点之间的联系, 并进一步判断噪声点, 算法中提出了新的阈值$l_i$.

定义 7 (连接阈值). 连接阈值$l_i$包含两部分, 可变阈值与固定阈值. 固定阈值为连接阈值的上限, 即$\lambda = mean(D_k) + std(D_k)$, $D_k = \bigcup_{x_i \in X}\{d(x_i, x_j)|x_j \in NN_k(x_i)\}$. 可变阈值考虑了数据点与当前剥离点集$X_{CB}^{(t)}$中数据点的相关性, 根据统计学中常用来检验异常值的$3-\sigma$原则, 对于任意数据点$x_i \in X_B^{(t)}$, 定义$x_i$的可变阈值为 $mean\left(D_{B,k}^{(t)}(x_i)\right) + 3std\left(D_{B,k}^{(t)}(x_i)\right)$, 其中$D_{B,k}^{(t)}(x_i) = \{d(x_i, x_j)|x_j \in NN_{k_{CB}}^{(t)}(x_i)\}$ (数据点$x_i$与其边界点 $k$ 近邻之间的距离的集合), 因此连接阈值$l_i$的具体定义如下:

$$l_i = \min\left\{mean\left(D_{B,k}^{(t)}(x_i)\right) + 3std\left(D_{B,k}^{(t)}(x_i)\right), \lambda\right\} \tag{10}$$

定义 8 (共享近邻相似度[21]). 对于数据点$x_i, x_j \in X$, 其共享近邻相似度即为两个数据点之间共享近邻的个数:

$$SNS(x_i, x_j) = \left|SNN(x_i, x_j)\right| \tag{11}$$

结合上述连接阈值$l_i$, 改进后的算法中包含了两个关联标准. 其中关联准则 (a) 建立边界点与非边界点之间的联系.

定义 9 (关联准则 (a)). 对于任意边界点$x_i \in X_B^{(t)}$, 定义$NN_{k_{NB}}^{(t)}(x_i)$(非边界点 $k$ 近邻) 为其在非边界点集合$X^{(t+1)}$中的距离最近的 $k$ 个邻居, 使用$\rho(x_i)$表示$x_i$的关联点, 给出如下关联准则:

$$\rho(x_i) = \{x_j|x_j \in NN_{k_{NB}}^{(t)}(x_i) \text{ and } d(x_i, x_j) \leqslant l_i\} \tag{12}$$

对于$x_j \in NN_{k_{NB}}^{(t)}(x_i)$, 若$x_i$与$x_j$之间的距离不超过可变阈值$l_i$, 则将$x_i$与$x_j$关联起来, 一般情况下, 关联准则 (a) 使得边界点不再仅与一个非边界点相联系, 外侧的数据点也具有修正类别的能力, 有效降低分配边界点时发生多米诺效应的概率.

定义 10 (关联准则 (b)). 考虑边界点与已剥离边界点之间的联系, 对于边界点$x_i \in X_B^{(t)}$, 定义$NN_{CB,k}^{(t)}(x_i) = \left\{x_j|x_j \in (NN_k(x_i) \cap X_{CB}^{(t)})\right\}$, $\gamma(x_i)$为$x_i$的关联点, 则$\gamma(x_i)$的具体定义如下:

$$\gamma(x_i) = \left\{x_j|x_j \in NN_{CB,k}^{(t)}(x_i) \text{ and } SNS(x_i, x_j) \geqslant \frac{k}{2}\right\} \tag{13}$$

对于$x_j \in NN_{CB,k}^{(t)}(x_i)$, 若$x_i$与$x_j$之间的共享近邻相似度大于等于$\frac{k}{2}$, 则将$x_i$与$x_j$关联起来[19], 关联准则 (b) 考虑边界点之间的空间特征, 可以弥补仅考虑距离特征时容易误判异常值的局限.

当边界点剥离过程停止时, 潜在的核心区域 (剩余未被剥离的点) 可以被很好地识别, 并且通过简单的 DBSCAN 可以将它们聚在一起, 根据关联准则 (a) 以及关联准则 (b) 分配被剥离点, 改进后的 BP 算法 (SOBP) 的具体步骤如算法 1 所示.

算法 1. SOBP 算法

输入: 数据集$X=\{x_1, x_2, \cdots, x_n\}$以及参数 $k$;
输出: 聚类标签, $Y=\{y_1, y_2, \cdots, y_n\}$.

计算所有配对点之间的共享近邻相似度及距离
$X^1 \leftarrow X$
For 迭代次数 $1 \leqslant t \leqslant T$ do
  For $x_i \in X^{(t)}$ do
    $RNN_k^{(t)}(x_i) \leftarrow \left\{x_j|x_i \in NN_k^{(t)}(x_j)\right\}$
    $SNN^{(t)}(x_i, x_j) \leftarrow \{x_r|x_r \in NN_k^{(t)}(x_i) \cap NN_k^{(t)}(x_j)\}$
    $b_i^{(t)} \leftarrow \sum_{x_j \in RNN_k^{(t)}(x_i)} \exp\left(-\frac{d^2(x_i, x_j)}{\sigma_j^2(|SNN^{(t)}(x_i, x_j)|+1)}\right)$
  End for
  $X_B^{(t)} \leftarrow \left\{x_i : B_i^{(t)} = 1 \wedge x_i \in X^{(t)}\right\}$
  $X^{(t+1)} \leftarrow X^{(t)} \backslash X_B^{(t)}$
  For $x_i \in X_B^{(t)}$ do
    $\rho(x_i) \leftarrow ASSOCIATEPOINT\left(x_i, NN_{k_{NB}}^{(t)}(x_i)\right)$ (见定义 9)
    $\gamma(x_i) \leftarrow ASSOCIATEPOINT\left(x_i, NN_{CB,k}^{(t)}(x_i)\right)$ (见定义 10)
  End for
End for
$\tilde{c} \leftarrow CLUSTERCOREPOINTS\left(X^{(t+1)}\right)$ (聚类核心点类簇)
$c \leftarrow COMPUTEFINALRESULT(X, \tilde{c}, \rho, \gamma)$ (根据计算的关联将剥离点分配给类簇)

## 3 实验

在本节中, 我们将详细介绍实验的参数设置, 并进行实验结果的展示和分析, 以证明改进后的 BP 算法 (SOBP) 的有效性.

### 3.1 实验指标和设置

本文选取了 6 种具有代表性的聚类算法进行比较, 包括 BP、K-means[22]、DBSCAN、HDBSCAN、mean-shift (MS)[23,24] 和 DPC. 其中 BP 算法基于作者提供的

源代码, DPC 算法的代码由其他作者[25] 提供, 其他算法均是在 sklearn 库和其他库的帮助下使用 Python 实现.

在后续实验中, 我们使用 sklearn 提供的函数来估计 MS 中的带宽. 在 K-means 算法中, 采用正确的聚类数量作为 K-means 算法的唯一参数. 对于其他算法, 全部的显示结果均是通过网格搜索进行参数调优后的最优结果, 具体搜索范围如下: 对于 BP 算法和 SOBP 算法, 我们选择重要的参数 $k$ 为 2 到 30 的值. DBSCAN 算法中的重要参数 eps 从 0.1 循环到 20. HDBSCAN 算法要求输入最小的类簇大小, 我们将最小尺寸设置为 2 到 15. 对于传统的 DPC 算法, 实验中使用高斯核进行密度估计, $d_c$ 表示选择的数据点占总数据点的比例, 设置 $d_c$ 从 1% 至 5%.

在评价指标方面, 本文采用调整兰德指数 (ARI)[26]、调整互信息 (AMI)[27] 和 Fowlkes-Mallows 指数 (FMI)[28] 对 7 个聚类算法的有效性进行评估, 三者的取值上限均为 1, 且聚类效果与取值成正比.

### 3.2 实验结果

表 1 给出了 7 种聚类算法在不同数据集上的实验指标值, 数据集的详细信息可见表 2. 同时, 我们将最高得分的指标以粗体标记, 图 3–图 8 更加直观地展示了人工数据集的聚类结果.

表 1 聚类指标对比

| 数据集 | 指标 | SOBP (ours) | BP | K-means | DBSCAN | HDBSCAN | MS | DPC |
|---|---|---|---|---|---|---|---|---|
| 3-spiral | ARI | **1.000 0** | 0.628 0 | −0.006 0 | **1.000 0** | **1.000 0** | 0.106 4 | **1.000 0** |
| | AMI | **1.000 0** | 0.677 2 | −0.005 5 | **1.000 0** | **1.000 0** | 0.241 9 | **1.000 0** |
| | FMI | **1.000 0** | 0.741 5 | 0.327 6 | **1.000 0** | **1.000 0** | 0.281 5 | **1.000 0** |
| Xclara | ARI | **1.000 0** | 0.927 8 | 0.992 9 | 0.982 9 | 0.990 4 | 0.993 9 | 0.996 9 |
| | AMI | **1.000 0** | 0.897 0 | 0.987 2 | 0.965 5 | 0.980 7 | 0.988 8 | 0.993 7 |
| | FMI | **1.000 0** | 0.952 0 | 0.995 3 | 0.988 7 | 0.993 6 | 0.996 0 | 0.998 0 |
| Complex8 | ARI | 0.985 3 | 0.625 4 | 0.364 0 | **0.999 6** | 0.940 1 | 0.436 5 | 0.557 1 |
| | AMI | 0.970 3 | 0.784 5 | 0.587 3 | **0.998 5** | 0.928 6 | 0.562 3 | 0.711 0 |
| | FMI | 0.987 8 | 0.688 0 | 0.460 8 | **0.999 7** | 0.950 8 | 0.563 3 | 0.628 0 |
| Complex9 | ARI | **1.000 0** | 0.345 3 | 0.413 6 | 0.999 8 | 0.694 0 | 0.488 2 | 0.406 8 |
| | AMI | **1.000 0** | 0.747 9 | 0.662 0 | 0.999 4 | 0.803 1 | 0.680 1 | 0.652 1 |
| | FMI | **1.000 0** | 0.494 5 | 0.511 1 | 0.999 8 | 0.762 7 | 0.580 1 | 0.505 8 |
| DS1 | ARI | **0.946 4** | 0.149 2 | 0.015 1 | 0.409 8 | 0.420 0 | 0.212 8 | 0.425 4 |
| | AMI | **0.917 6** | 0.484 1 | 0.211 8 | 0.642 8 | 0.675 7 | 0.475 8 | 0.662 6 |
| | FMI | **0.978 0** | 0.424 7 | 0.405 6 | 0.681 5 | 0.688 9 | 0.542 1 | 0.697 9 |
| DS2 | ARI | **1.000 0** | 0.229 7 | 0.607 4 | 0.999 6 | 0.998 8 | 0.592 3 | 0.782 0 |
| | AMI | **1.000 0** | 0.628 4 | 0.706 7 | 0.999 0 | 0.996 5 | 0.678 7 | 0.858 7 |
| | FMI | **1.000 0** | 0.393 3 | 0.679 5 | 0.999 7 | 0.999 0 | 0.667 3 | 0.824 1 |
| Dermatology | ARI | **0.851 7** | 0.843 8 | 0.693 4 | 0.782 1 | 0.768 2 | 0.836 4 | 0.766 4 |
| | AMI | **0.900 7** | 0.888 7 | 0.792 5 | 0.845 1 | 0.816 1 | 0.880 4 | 0.823 5 |
| | FMI | **0.888 1** | 0.882 3 | 0.752 0 | 0.839 0 | 0.822 2 | 0.876 5 | 0.816 7 |
| Banknote | ARI | **0.667 0** | 0.490 7 | 0.048 5 | 0.666 7 | 0.466 1 | 0.131 2 | 0.232 2 |
| | AMI | **0.647 3** | 0.512 4 | 0.029 8 | 0.647 2 | 0.483 8 | 0.118 9 | 0.347 0 |
| | FMI | **0.818 7** | 0.703 6 | 0.551 8 | 0.818 3 | 0.686 1 | 0.460 1 | 0.649 3 |
| Thy | ARI | **0.718 8** | 0.654 9 | 0.579 0 | 0.683 1 | 0.648 5 | 0.605 9 | 0.181 5 |
| | AMI | **0.558 0** | 0.524 7 | 0.488 3 | 0.503 9 | 0.462 7 | 0.421 0 | 0.202 6 |
| | FMI | 0.864 0 | 0.859 7 | 0.806 3 | **0.867 5** | 0.849 8 | 0.809 0 | 0.624 1 |
| Heart-statlog | ARI | **0.302 1** | 0.143 7 | 0.144 7 | 0.222 8 | 0.083 0 | 0.147 4 | 0.034 1 |
| | AMI | **0.214 5** | 0.104 3 | 0.101 7 | 0.145 4 | 0.052 5 | 0.103 5 | 0.079 3 |
| | FMI | **0.629 1** | 0.596 5 | 0.588 6 | 0.521 5 | 0.555 1 | 0.578 9 | 0.581 8 |

在图 3 中, 3-spiral 是一个具有 3 个类别且低密度的数据集, 数据点构成了彼此交叉缠绕的螺旋形曲线. 可以看到 BP 算法将曲线的尾部识别为异常值, 这可能与关联策略的连接阈值设置有关, 其他基于密度的聚类算法在此数据集上表现良好, 而 K-means 算法仅考虑数据点之间的距离特性, 无法处理环绕的流线型数据.

Xclara 数据集以中心密度高、边界密度低为特征, 该数据集 3 个类簇边缘数据点彼此交错, 部分数据点无法通过人为划分, 具有一定的难度. 可以看到所有算

法在此数据集上的 ARI 值都高于 0.90. 然而, 如图 4 所示, 其他算法在分类边缘点时不够准确, 只有 SOBP 算法在所有情况下都分类正确, 完美地将彼此交错的边缘数据点进行了正确划分.

在图 5 和图 6 中, Complex8 和 Complex9 都是具有不均匀密度的复杂结构数据集, 其中 Complex8 包含 8 个类别, Complex9 包含 9 个类别, 通过观察两个数据集的原始结构可以发现 Complex8 相比于 Complex9 密度更不均匀, 但 Complex9 的结构更加复杂. 在使用 7 种聚类算法对它们进行聚类时, DBSCAN 在 Complex8 数据集上具有最佳的聚类结果, 其次是 SOBP 算法. 对于 Complex9 数据集, SOBP 算法实现了最佳的聚类结果, 而 DBSCAN 将个别点识别为了离群点, 因此没有获得最佳表现. 与 BP 算法相比, SOBP 算法对于聚类复杂数据集的表现更为优异.

表 2 实验数据集

| 数据集 | 样本量 | 维度 | 类簇 |
|---|---|---|---|
| 3-spiral | 312 | 2 | 3 |
| Xclara | 3 000 | 2 | 3 |
| Complex8 | 2 551 | 2 | 8 |
| Complex9 | 3 031 | 2 | 9 |
| DS1 | 1 400 | 2 | 4 |
| DS2 | 7 236 | 2 | 6 |
| Banknote | 1 372 | 4 | 2 |
| Thy | 215 | 5 | 3 |
| Dermatology | 366 | 34 | 6 |
| Heart-statlog | 270 | 13 | 2 |

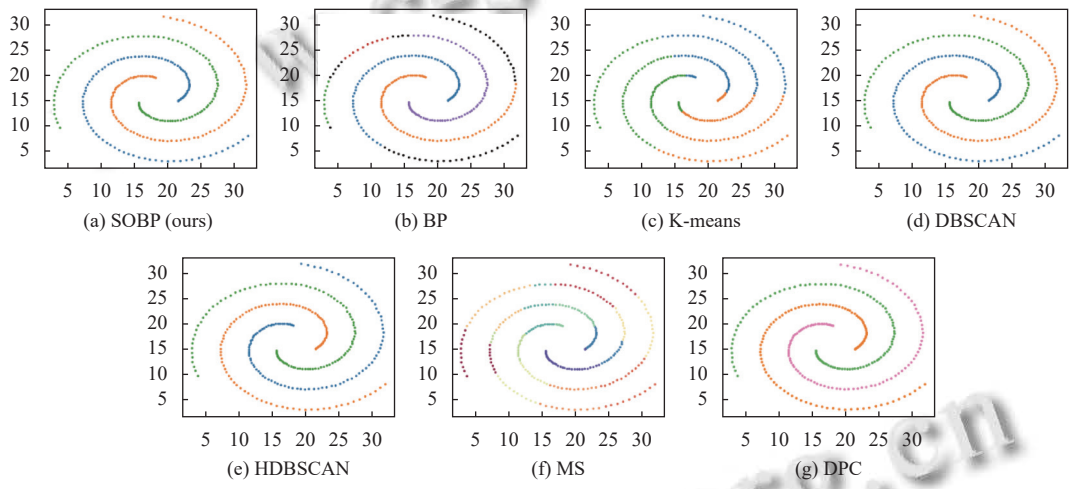


图 3 3-spiral 数据集聚类结果

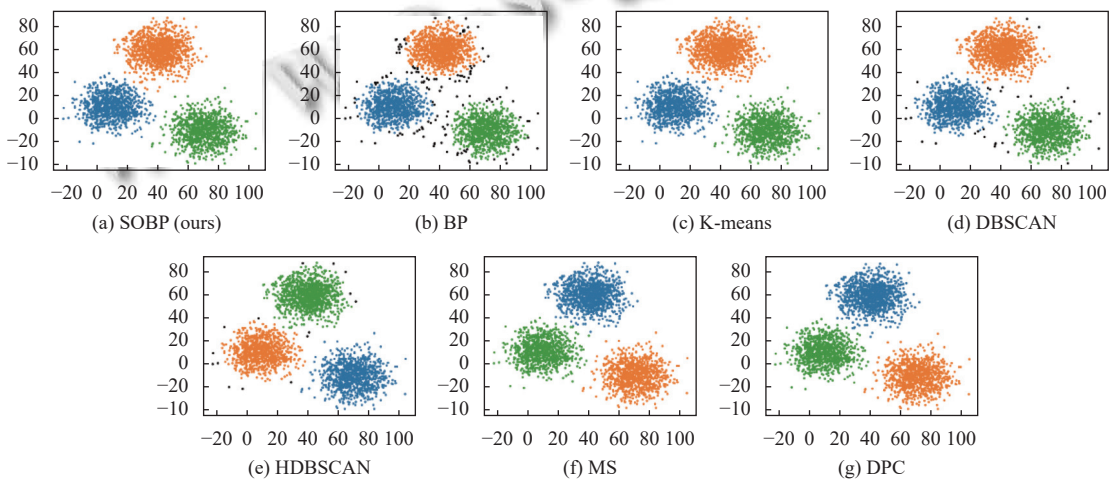


图 4 Xclara 数据集聚类结果

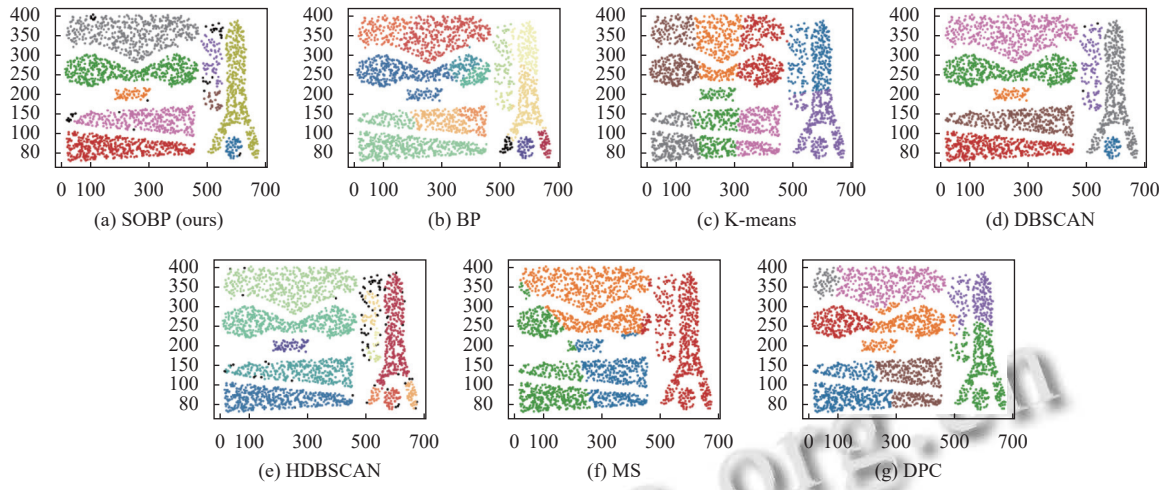


图 5 Complex8 数据集聚类结果

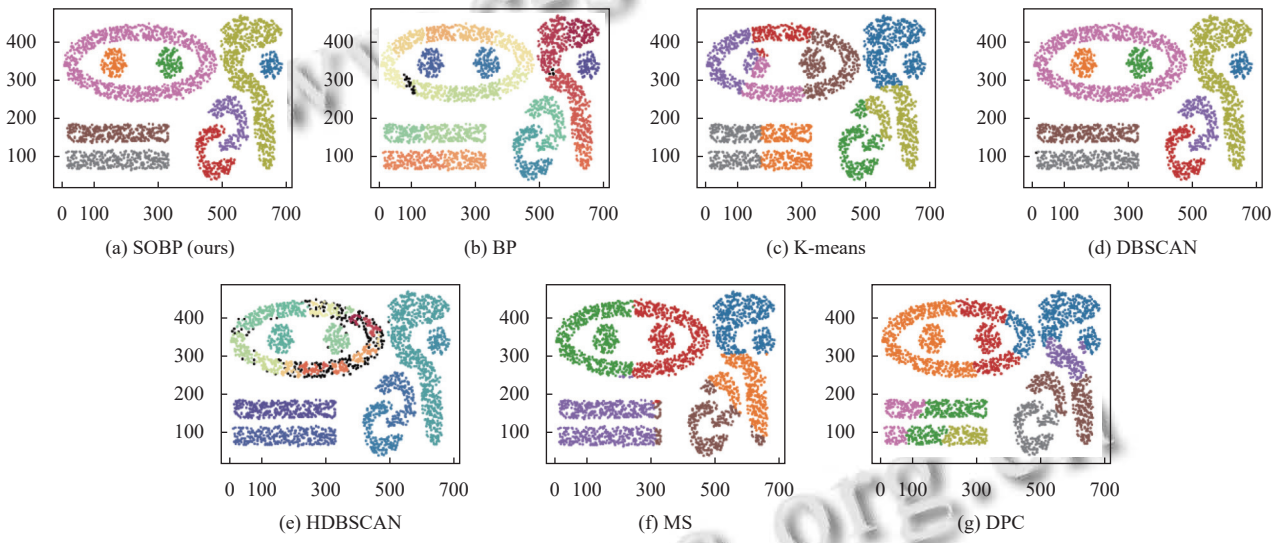


图 6 Complex9 数据集聚类结果

DS1 的主体由 1 个新月形类簇和 3 个球形类簇组成, 其中新月形类簇包围着球形类簇, 同时含有很多的噪声点, 因此比大多数数据集更具挑战性. 图 7 展示了 7 种不同的算法对 DS1 数据集的聚类结果, 可以看出所有算法的聚类结果都不完美. 相较而言, SOBP 算法在处理 DS1 数据集时表现较好, 能够大致正确地聚类出数据集中的主体结构, 而其他算法则普遍存在将新月形类簇过度划分的问题.

DS2 由不同形状的 6 个类簇组成, 这些类簇相互包围缠绕, 相对于其他数据集具有较高的密度. 在图 8 中, BP 算法将 DS2 分成更多的类簇, 远离了数据集的真实分布. DBSCAN 以及 HDBSCAN 在该数据集上表现较好, 它们的 3 项指标得分都比较高, 但是它们均将类簇的一些边界点判断成了离群点. 只有 SOBP 算法能够完全识别 DS2 的真实分布, 获得了最优的聚类效果.

为了验证 SOBP 算法在实际应用中的有效性, 本文进一步选择了若干个真实数据集进行实验, 这些数据集的样本点数量、数据维数以及聚类的类簇数目各不相同, 相较于人工数据集更加复杂. 为了保证实验的可靠性和准确性, 在实验中我们对于表 2 所列出的真实数据集进行了预处理, 其中包括数据规范化和降维等, 以消除数据中的噪声和冗余信息. 通过表 1 中的实验结果可以发现, SOBP 算法在真实数据集上的聚类效果总体优于其他 6 种聚类算法. 因此, 我们认为 SOBP 算法具有一定的有效性和实用性, 可以在实际应用中发挥作用.

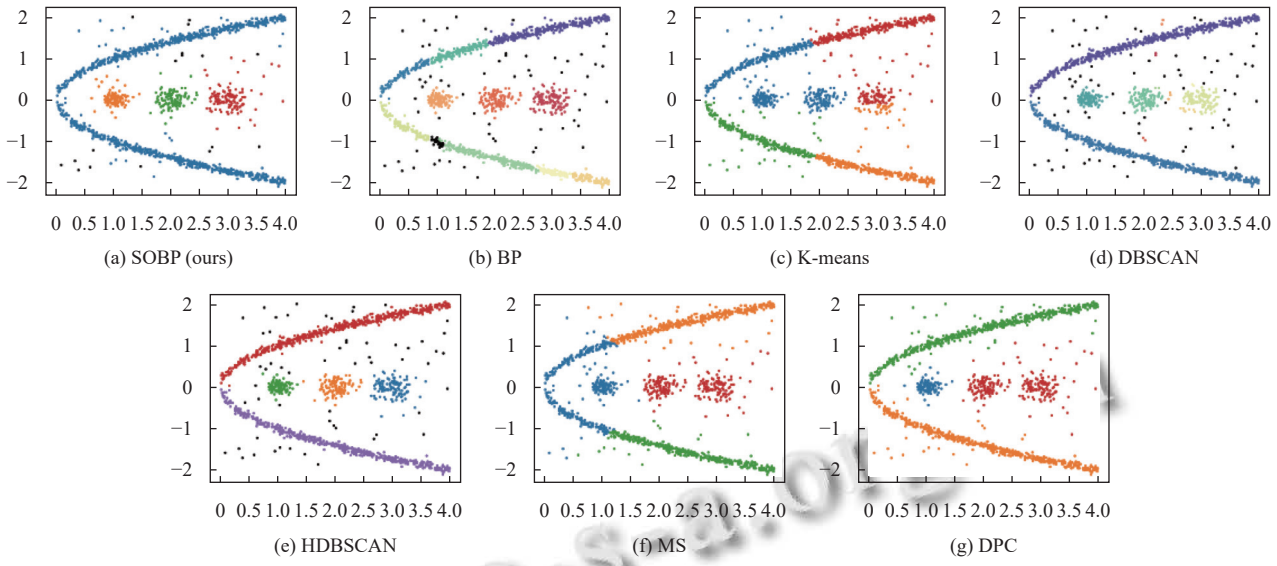


图 7 DS1 数据集聚类结果

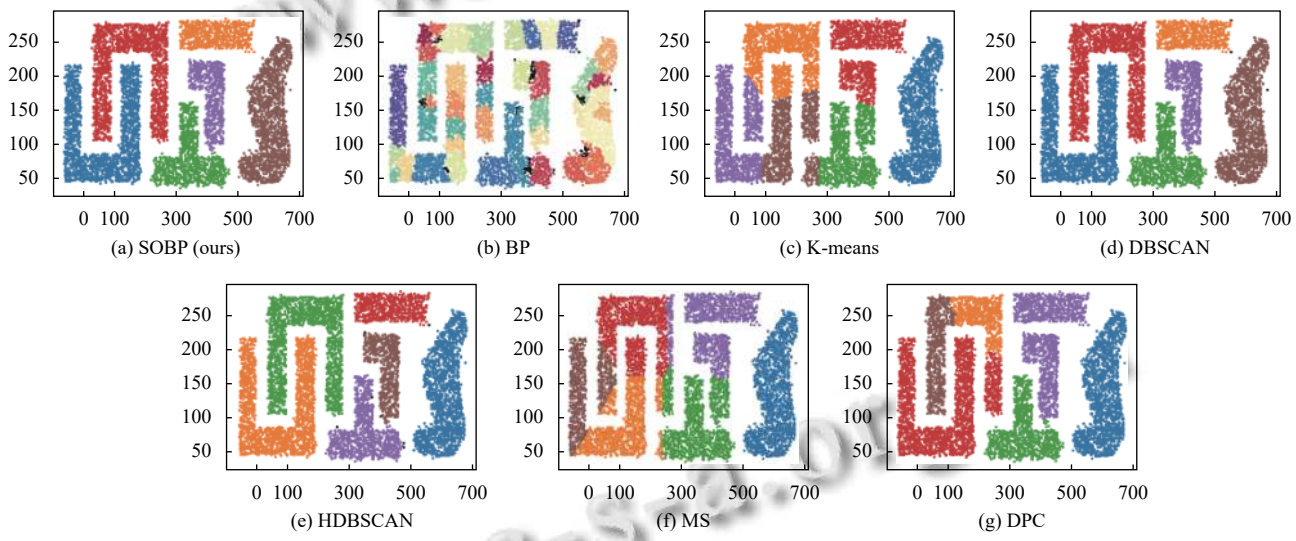


图 8 DS2 数据集聚类结果

## 4 结论

本文提出了一种基于共享近邻和优化关联策略的边界剥离聚类算法 (SOBP), 该算法采用基于共享近邻的局部密度, 考虑了每个数据点的邻域分布, 并减少了异常值对于边界点确定的影响. 针对原始算法在数据集密度不均匀、形状复杂等情况下容易过度分割且误判异常值的问题, 我们改进了关联策略, 大大降低了分配边界点时发生连带错误的概率. 实验结果表明, SOBP 算法在人工数据集和真实数据集上都表现出良好的聚类效果, 具有一定的实用价值. 然而, 该算法需要手动输入参数选择, 这对后续研究者在实际问题中应用该算法带来了一定的挑战, 因此, 在后续工作中我们将考虑输入参数的自适应.

## 参考文献

1 Li M, Sha HY, Liu HY. Microfeature segmentation algorithm for biological images using improved density peak clustering. Computational and Mathematical Methods in Medicine, 2022, 2022: 8630449. [doi: 10.1155/2022/8630449]

2 Niu YY, Kong DT, Liu LG, *et al*. Overlapping community detection with adaptive density peaks clustering and iterative partition strategy. Expert Systems with Applications, 2023, 213: 119213. [doi: 10.1016/j.eswa.2022.119213]

3 Petegrosso R, Li ZL, Kuang R. Machine learning and statistical methods for clustering single-cell RNA-sequencing data. Briefings in Bioinformatics, 2020, 21(4): 1209–1223. [doi: 10.1093/bib/bbz063]
4 Kiselev VY, Andrews TS, Hemberg M. Challenges in unsupervised clustering of single-cell RNA-seq data. Nature Reviews Genetics, 2019, 20(5): 273–282. [doi: 10.1038/s41576-018-0088-9]
5 Fraley C, Raftery AE. How many clusters? Which clustering method? Answers via model-based cluster analysis. The Computer Journal, 1998, 41(8): 578–588. [doi: 10.1093/comjnl/41.8.578]
6 Han J, Kamber M. Data Mining: Concepts and Techniques. 2nd ed. San Francisco: Morgan Kaufinann, 2006.
7 Saxena A, Prasad M, Gupta A, *et al*. A review of clustering techniques and developments. Neurocomputing, 2017, 267: 664–681. [doi: 10.1016/j.neucom.2017.06.053]
8 Ezugwu AE, Ikotun AM, Oyelade OO, *et al*. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Engineering Applications of Artificial Intelligence, 2022, 110: 104743. [doi: 10.1016/j.engappai.2022.104743]
9 Averbuch-Elor H, Bar N, Cohen-Or D. Border-peeling clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(7): 1791–1797. [doi: 10.1109/TPAMI.2019.2924953]
10 Ester M, Kriegel HP, Sander J, *et al*. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. Portland: AAAI Press, 1996. 226–231.
11 Campello RJGB, Moulavi D, Sander J. Density-based clustering based on hierarchical density estimates. Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Gold Coast: Springer, 2013. 160–172. [doi: 10.1007/978-3-642-37456-2_14]
12 Wang LM, Wang HH, Han XM, *et al*. A novel adaptive density-based spatial clustering of application with noise based on bird swarm optimization algorithm. Computer Communications, 2021, 174: 205–214. [doi: 10.1016/j.comcom.2021.03.021]
13 Chen XM, Liu WQ, Qiu HN, *et al*. APSCAN: A parameter free algorithm for clustering. Pattern Recognition Letters, 2011, 32(7): 973–986. [doi: 10.1016/j.patrec.2011.02.001]
14 万佳, 胡大裟, 蒋玉明. 多密度自适应确定 DBSCAN 算法参数的算法研究. 计算机工程与应用, 2022, 58(2): 78–85. [doi: 10.3778/j.issn.1002-8331.2012-0476]
15 Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science, 2014, 344(6191): 1492–1496. [doi: 10.1126/science.1242072]
16 Liu R, Wang H, Yu XM. Shared-nearest-neighbor-based clustering by fast search and find of density peaks. Information Sciences, 2018, 450: 200–226. [doi: 10.1016/j.ins.2018.03.031]
17 Xu X, Ding SF, Wang LJ, *et al*. A robust density peaks clustering algorithm with density-sensitive similarity. Knowledge-based Systems, 2020, 200: 106028. [doi: 10.1016/j.knosys.2020.106028]
18 张清华, 周靖鹏, 代永杨, 等. 基于代表点与 K 近邻的密度峰值聚类算法. 软件学报, 2023: 1–20. [doi: 10.13328/j.cnki.jos.006756]
19 Du MJ, Wang R, Ji R, *et al*. ROBP: A robust border-peeling clustering using Cauchy kernel. Information Sciences, 2021, 571: 375–400. [doi: 10.1016/j.ins.2021.04.089]
20 Feng J, Zhang BK, Ran RS, *et al*. An effective clustering algorithm using adaptive neighborhood and border peeling method. Computational Intelligence and Neuroscience, 2021, 2021: 6785580. [doi: 10.1155/2021/6785580]
21 Ertöz L, Steinbach M, Kumar V. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. Proceedings of the 2003 SIAM International Conference on Data Mining. 2003. 47–58.
22 MacQueen JB. Some methods for classification and analysis of multivariate observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley: University of California Press, 1967. 281–297.
23 Fukunaga K, Hostetler L. The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory, 1975, 21(1): 32–40. [doi: 10.1109/TIT.1975.1055330]
24 Cheng YZ. Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17(8): 790–799. [doi: 10.1109/34.400568]
25 Zuo WD, Hou XM. An improved probability propagation algorithm for density peak clustering based on natural nearest neighborhood. Array, 2022, 15: 100232. [doi: 10.1016/j.array.2022.100232]
26 Hubert L, Arabie P. Comparing partitions. Journal of Classification, 1985, 2(1): 193–218. [doi: 10.1007/BF01908075]
27 Vinh NX, Epps J, Bailey J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. The Journal of Machine Learning Research, 2010, 11: 2837–2854.
28 Fowlkes EB, Mallows CL. A method for comparing two hierarchical clusterings. Journal of the American Statistical Association, 1983, 78(383): 553–569. [doi: 10.1080/01621459.1983.10478008]

(校对责编: 孙君艳)