自动化学报
ACTA AUTOMATICA SINICA

基于空间向量分解的边界剥离密度聚类

张瑞霖 郑海阳 苗振国 王鸿鹏

Density Clustering Based on the Border-peeling Using Space Vector Decomposition

ZHANG Rui-Lin, ZHENG Hai-Yang, MIAO Zhen-Guo, WANG Hong-Peng

在线阅读 View online: https://doi.org/10.16383/j.aas.c220208

基于空间向量分解的边界剥离密度聚类

张瑞霖 [1] 郑海阳 [1] 苗振国 [1] 王鸿鹏 [1, 2, 3]

摘 要 作为聚类的重要组成部分, 边界点在引导聚类收敛和提升模式识别能力方面起着重要作用, 以 BP (Border-peeling clustering) 为最新代表的边界剥离聚类借助潜在边界信息来确保簇核心区域的空间隔离, 提高了簇骨架代表性并解决了边界隶属问题. 然而, 现有边界剥离聚类仍存在判别特征不完备、判别模式单一、嵌套迭代等约束. 为此, 提出了基于空间向量分解的边界剥离密度聚类 (Density clustering based on the border-peeling using space vector decomposition, CBPVD), 以投影子空间和原始数据空间为基准, 从分布稀疏性 (紧密性) 和方向偏斜性 (对称性) 两个视角强化边界的细粒度特征, 进而通过主动边界剥离反向建立簇骨架并指导边界隶属. 与同类算法相比, 40 个数据集 (人工、UCI、视频图像) 上的实验结果以及 4 个视角的理论分析表明了 CBPVD 在高维聚类和边界模式识别方面具有良好的综合表现.

关键词 聚类, 空间向量分解, 边界剥离, 投影子空间, 高维, 密度

引用格式 张瑞霖, 郑海阳, 苗振国, 王鸿鹏. 基于空间向量分解的边界剥离密度聚类. 自动化学报, 2023, **49**(6): 1195−1213

DOI 10.16383/j.aas.c220208

Density Clustering Based on the Border-peeling Using Space Vector Decomposition

ZHANG Rui-Lin[1] ZHENG Hai-Yang[1] MIAO Zhen-Guo[1] WANG Hong-Peng[1, 2, 3]

Abstract Border points, as an essential part of density clustering, play a key role in guiding clustering convergence and improving pattern recognition ability. Indeed, the border-peeling clustering with BP (border-peeling clustering) as the latest representative ensures the spatial isolation of core region of the cluster by using intrinsic boundary information, then enhancing the cluster backbone. Nevertheless, the performance of available methods tends to be constrained by incomplete discriminant feature, single pattern and multiple iterations. To this end, this paper proposes a novel algorithm named CBPVD (density clustering based on the border-peeling using space vector decomposition). The property of CBPVD is based on the projection subspace and original space to enhance the fine-grained feature representation of the border point from the two perspectives of sparsity (compactness) and skewness (symmetry) of distribution, then reversely establishes the cluster backbone through active boundary peeling and guides the boundary membership. Finally, we compare performance of CBPVD with six state-of-the-art methods over synthetic, UCI, and image datasets. Experiments on 40 datasets and discussion cases from 4 perspectives demonstrate that our algorithm is feasible and effective in clustering and boundary pattern recognition.

Key words Clustering, space vector decomposition, border-peeling, projection subspace, high dimension, density

Citation Zhang Rui-Lin, Zheng Hai-Yang, Miao Zhen-Guo, Wang Hong-Peng. Density clustering based on the border-peeling using space vector decomposition. *Acta Automatica Sinica*, 2023, **49**(6): 1195−1213

聚类分析旨在将目标数据划分到若干互不相交的集合中, 以实现高组内相似性和低组间相似性[1], 广泛用于图像分割、推荐系统、海量数据标注等人工智能场景[2].

作为最受欢迎的聚类范式之一, 密度聚类的首要任务是搜索一批高代表性对象并建立簇的核心区域, 进而指导其余对象归属. 核心对象包含了丰富的结构信息, 通过给定的可达条件, 形成的核心区域 (目标簇骨架) 可有效表征簇的实际结构. 因此, 大多数密度聚类方法将核心对象的特征提取作为聚类主线. 例如, 以 DPC (Clustering by fast search and find of density peaks)[3] 代表的密度峰值聚类[4−7] 通过峰值最大化判别核心对象; H-DBSCAN[8] 等派生于 DBSCAN[9] 的传统方法利用全局密度描述核心对象. 总的来说, 在识别核心对象后, 许多方法[10−11] 以此为基准构建簇的骨架, 却忽视了边界对象的重要性, 仅将其视为核心对象在数据空间上的相对补.

收稿日期 2022-03-21 录用日期 2022-10-14
Manuscript received March 21, 2022; accepted October 14, 2022
广东省安全智能新技术重点实验室基础研究项目 (2022B1212010005), 深圳市基础研究专项 (JCYJ20210324132212030) 资助
Supported by Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005) and Shenzhen Fundamental Research Fund (JCYJ20210324132212030)
本文责任编委 胡清华
Recommended by Associate Editor HU Qing-Hua
1. 哈尔滨工业大学 (深圳) 计算机科学与技术学院 深圳 518071 2. 鹏城实验室 深圳 518000 3. 广东省安全智能新技术重点实验室 深圳 518000
1. School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518071 2. Peng Cheng Laboratory, Shenzhen 518000 3. Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, Shenzhen 518000

事实上, 较之特征单一的核心点, 边界点较为复杂多变, 主要位于目标簇的边缘、簇与簇之间以及远离簇的区域[12]. 在簇骨架建立时, 边界点往往会弱化簇的独立性、误合并近邻簇. 此外边界点往往具有两个或多个群体的属性特征, 在实际场景中包含了高价值的模式信息[13], 如临床中携带病毒或致病基因但未发病的人群、人脸识别中动作复杂、归属模糊的图像. 基于此, 一些研究者将密度聚类中边界被动过滤转为主动剥离, 进一步确保簇骨架之间的空间隔离. 尽管已针对性地提出一些面向边界剥离的密度聚类, 如基于统计量的 Spinver[14]、CA-CSM[15]、C-USB[16]; 基于 D-S 证据理论的 3W-DPET[17]; 基于切平面一致性的 DCUBI[18], 以及最新发表的 BP (Border-peeling clustering) 算法[13], 但现有方法的边界剥离模式仍面临着判别特征不完备、判别条件单一、倾向低维凸数据等局限.

受子空间聚类[19] 的启发, 提出了一种基于空间向量分解的边界剥离密度聚类 (Density clustering based on the border-peeling using space vector decomposition, CBPVD), 将数据点与其近邻形成的空间关系由原始高维空间射影到低维子空间中, 利用边界 (核心对象) 在投影子空间上的分布方向偏斜性 (对称性) 以及在原始空间上的分布稀疏性 (稠密性) 强化特征表示, 实现边界有效剥离, 进而通过传递闭包和密度优先级策略共同指导对象划分. 本文工作如下:

1) 引入空间向量分解理论, 从矢量视角分析并量化高维数据的局部分布;

2) 从原始数据空间和投影子空间两个维度强化边界特征表示, 提出了一种非迭代、细粒度的边界剥离方法; 利用传递闭包和密度优先级策略, 建立了一种两阶段对象关联策略;

3) 提出了基于空间向量分解的边界剥离密度聚类, 广泛的实验和深入分析验证了算法在高维聚类和边界模式信息识别方面的有效性.

1 相关工作

目前, 距离划分和密度可达是最受欢迎的两种聚类范式. 以 K-means[20] 为代表的划分聚类一般随机指定簇中心, 然后依据目标函数收敛来产生最佳聚类, 常作为目标检测、图像分割的底层算子. 但此类算法受限于凸目标函数和最近中心划分策略, 无法有效处理非凸和高维数据.

相反, 密度聚类认为簇是被低密度区域分隔的稠密连通区域, 其特点是可识别任意形状的簇. 如 DBSCAN[9] 通过阈值 Eps 和半径 $MinPts$ 搜索核心对象来建立簇骨架, 并将其可达路径上的剩余对象归为一类. DBSCAN 衍生出很多优化改进[8], 但面对高维场景时, 全局性质的密度度量、参数缺乏视觉参考以及数据量纲差异使得算法性能并不稳定. 尽管 DBSCAN 首次提出了密度聚类中边界点概念, 但没有进行深入研究.

Rodriguez 等[3] 将少数密度峰值点作为簇骨架, 提出了密度峰值聚类 DPC. 与划分聚类相比, DPC 无需迭代, 并且由决策图生成的簇中心更加合理; 与 DBSCAN 相比, DPC 以密度峰值点为基准进行对象分类, 无需多层嵌套遍历. 作为一种新的密度聚类框架, DPC 得到了广泛关注. 鉴于高可视化优势, EC[5]、SNN-DPC[21] 引入 k 近邻、互近邻、共享近邻、模糊核等优化原始决策图. 为确保聚类流的连续性, DPC-RDE[6]、CBP-EKNN[22] 通过设计评分函数/变量来量化簇中心特征, 但算法仍要提前输入簇个数. 此外, RA-Clust[23]、DPC-AHS[24]、GB-DPC[4]、CA-CSM[15] 利用统计模型 (线性回归、残差分析、正态分布曲线) 来拟合目标分布, 实现了聚类过程自动化, 但额外引入了统计参数. 优化划分策略同样是改进方向之一, 如 ADPC-KNN[25]、DenMune[26] 利用邻域结构信息、递增式划分、密度可达等要素来替代敏感的一阶段方法.

同样, 为了增强 DPC 在复杂场景下的聚类性能, SSDC[27] 利用对象间成对约束信息实现了密度峰值聚类的半监督模式; 此外, DDC[28] 开创性地将 DPC 与神经网络自编码器结合, 实现了 DPC 与自编码器的结合. 除上述理论研究外, DPC 还被嵌入到实际任务场景中, 如 ADPC[29] 借助 DPC 的泛化优势来分析医疗领域的脑电信号; AHC[30] 将密度峰值点应用在层次聚类中, 加速嵌套聚类树的构建. 然而大多数密度峰值聚类存在一个潜在约束: 聚类结果非常依赖密度峰值点, 但复杂场景下获得数量正确且高质量的密度峰值并不简单. 若算法未能正确估计簇数或度量存在偏差, 则低质量的密度峰值点往往引起划分错误连锁反应.

鉴于簇中心在构成簇骨架方面的局限, 一些学者专门对簇骨架的生成方式进行了分析研究, 旨在通过融合密度、密度峰值、隶属度等要素选取一批高质量对象. 例如, MBVC[31] 通过网格划分和均值漂移来迭代搜索簇的核心结构; DPC-DBFN[32]、CFDPC[10]、FAST-LDP-MST[33] 以部分峰值对象形成的微型簇为依据, 经过合并、剪枝操作来形成簇的骨干; 而 CLUB[11] 利用近邻结构信息直接建立簇骨架. 与传统聚类 DBSCAN 类似, 上述算法重点关注特征相对单一的核心对象或簇中心, 忽视了分布较为复杂的边界对象, 仅将其视为核心对象在数据空间上的相对补集. 因此面对复杂数据时, 聚类性能往往受到显著影响, 这一观察同样在文献 [12−13]

中得到验证.

相反, 我们没有复用“核心对象+簇骨架”模式, 而是利用多视角混合特征剥离边界, 反向确保簇骨架的空间隔离, 最大程度上保留簇的核心信息, 进而提高了面对复杂数据的鲁棒性. 相比密度峰值聚类, 主动边界剥离的引入避免了对象划分错误的连锁反应, 算法在完成聚类的同时可并行挖掘出高价值的边界模式信息.

实际上, 聚类性能通常取决于困难样本的分类结果, 即那些分布复杂、隶属模糊的边界样本. 考虑到边界点在提高聚类精度和提升模式识别能力方面的重要性, 一些研究开始借助边界的主动剥离来逐步揭示簇的结构信息, 随后通过边界之间、核心对象间、边界与核心对象间的可达关联来形成目标簇, 称为边界剥离密度聚类. 例如, Spinver[14] 利用霍普金斯统计量判别边界, CA-CSM[15]、C-USB[16] 通过三阶偏斜度描述数据的空间分布, DCUBI[18] 观察出边界对象的近邻在其切平面上的分布具有一致性; 3W-DPET[17] 借助证据理论中信任函数描述边界; Lever[13] 利用杠杆原理建模边界的邻域分布. 由于统计量自身设计局限, 上述算法很难有效处理非凸、高维数据, 如霍普金斯统计量的作用域仅为二维空间的第一象限, 偏斜度旨在描述整体数据空间的分布, 无法应用于局部邻域.

最近, Averbuch-Elor 等[12] 定义簇是由一些核心层和边界层组成的层状包围区域, 在 *T-PAMI* 上提出了 BP 算法. 作为最新代表, BP 利用密度差异迭代式剥离外部的边界点, 同时单向绑定边界点与最近未剥离点, 直至簇的核心区域有效分隔, 最后利用 DBSCAN 对核心区域进行搜索. 与以往的边界聚类不同, BP 系统地阐述了边界剥离在密度聚类中的重要性, 为边界剥离聚类提供了一种新思路. 但 BP 的 6 个超参数远高于同类算法, 并且边界判别标准相对单一: 仅通过密度大小来判定边界, 面对分布不均匀的数据, 算法很难发现最佳的划分结构, 极易造成过划分.

相反, 本文的 CBPVD 无需迭代计算和冗余连接, 不仅考虑了边界点的整体特征, 同时以矢量视角实例化边界在子空间上的潜在特征. 这种双通道细粒度特征确保了算法面对复杂分布、高维数据时的适应性, 并具有较高的聚类精度.

2　CBPVD 算法

2.1　边界剥离

文献 [9, 12−13] 指出, 相对于周围分布较为均匀、稠密的核心对象, 边界点周围分布整体上呈现明显偏斜、稀疏. 统计理论中, 变异系数、局部密度[13]、偏斜度[15], 三阶中心距[16], 霍普金斯统计量[14] 等度量可用于描述数据分布; 正如第 1 节中所述, 这些统计量存在一定的应用局限, 此外, 高维数据往往分布稀疏并且存在属性量纲差异, 而现有统计量偏向于将所有维度上的分布视为整体, 并没有考虑单一维度上的差异, 使这些统计量难以应用于高维数据. 我们对原始数据空间进行仿射变换, 将目标对象迁移变换至原点, 利用仿射变换的拓扑结构不变性, 在保持原有结构信息的同时更便于观察分析目标对象与其近邻之间的位置关系和分布情况.

从多维空间视角来看, 边界点在原始空间上的偏斜特征可表现为其邻域对象在基向量方向上围绕边界点呈现的偏态分布, 而对于聚类内部对象, 其近邻空间内的均匀性体现在基向量方向上近似的中心对称分布, 即多维对称. 基于此观察, 我们将空间向量分解定理[34] 引入对象所在的数据空间, 将其转换为向量空间, 并将对象间的成对关系视为独立的空间向量. 因此, 对象的局部分布估计转化为空间向量的位置关系判断, 存在如下描述:

假设 x_i 为数据集 $X=\{x_1, x_2, x_3, \cdots, x_n\}\in \mathbf{R}^{m\times n}$ 中任意数据点, 根据空间向量分解定理, 在 m 维数据空间 $\mathbf{R}^m$ 中, 存在唯一有序实数组 $\lambda_1, \lambda_2, \lambda_3, \cdots, \lambda_m$, 使得 $x_i=\lambda_1 e_1+\lambda_2 e_2+\lambda_3 e_3+\cdots+\lambda_m e_m$, 其中 $e_1, e_2, e_3, \cdots, e_m$ 为空间 $\mathbf{R}^m$ 的基向量. 对于 X 中任意数据点 x_i, 与数据点 x_j 形成的空间向量 $h_{i,j}$ 表示如下:

$$h_{i,j}=(\lambda_{i,1}-\lambda_{j,1}, \cdots,\ \lambda_{i,d}-\lambda_{j,d}, \cdots, \lambda_{i,m}-\lambda_{j,m})(e_1, e_2, e_3, \cdots, e_m)^{\mathrm{T}},\ 1\le i, j\le n \tag{1}$$

$\lambda_{i,d}-\lambda_{j,d}$ 表示 $h_{i,j}$ 在基向量 e_d 方向上的投影. 考虑到边界对象和核心对象的特征差异主要在于各自的邻域空间, 我们使用 k 近邻来提取邻域信息, 由此形成的邻域空间包含了丰富的邻域信息.

基于此, 任意 $x_i\in X$ 与其 k 个近邻形成 k 个空间向量, 作为描述向量位置关系最直接且有效的基本运算, 近邻向量 V_i 定义为 x_i 与其 k 个近邻 $N_k(x_i)$ 形成的向量之和, 如下:

$$V_i=\begin{cases}\sum\limits_j (h_{i,j})=[\sum\limits_j (\lambda_{i,1}-\lambda_{j,1}), \cdots, \\ \quad \sum\limits_j (\lambda_{i,m}-\lambda_{j,m})](e_1, \cdots, e_m)^{\mathrm{T}}, & \text{if } x_j\in N_k(x_i)\\ NULL, & \text{if } x_j\notin N_k(x_i)\end{cases} \tag{2}$$

其中 $\sum_j (\lambda_{i,d}-\lambda_{j,d})$ 表示 x_i 的近邻向量在基向量 e_d 上的投影坐标. 根据前文的分析可知, 聚类内部

对象的近邻相对均匀地散布在其四周, 经过空间向量分解后, 上述分布规律直接表现为每个基向量方向上, 近邻向量的投影总是关于核心对象呈现出较强的中心对称性. 相反, 边界对象的近邻向量投影具有较强的同向性. 相比于核心对象, 其近邻向量的长度明显较长. 为此, 我们可以通过向量 p-范数来判别对象, 以下是详细推导.

推论 1. 假设 $h_{i,j}$ 为对象 $x_i \in \mathbf{R}^m$ 与 $x_j \in \mathbf{R}^m$ 构成的空间向量, 给定一维特征空间 v_d $(1 \le d \le m)$, 向量 $h_{i,j}$ 在其上的投影为 $e_d(e_d^{\mathrm{T}}e_d)^{-1}e_d^{\mathrm{T}}h_{ij}$, 其中 e_d 为 v_d 的基向量.

证明.

这里使用空间向量分解定理[34] 作为引理: 假设 S 是 $\mathbf{R}^m$ 的任意子空间, $e'_1, e'_2, e'_3, e'_4, \cdots, e'_p$ 为 S 的基向量, 则对任意对象 $a \in S$, 存在 $a = y_1e'_1 + y_2e'_2 + y_3e'_3 + y_4e'_4 + \cdots + y_pe'_p$.

令矩阵 $A = [e'_1, e'_2, e'_3, e'_4, \cdots, e'_p]_{m\times p}$, 则可得 $a = Ay$, $y \in \mathbf{R}^p$. 对于空间向量 $h_{i,j}$, 令 $Proj_s h_{i,j}$ 表示 $h_{i,j}$ 在子空间 S 上的投影. 根据上方描述的空间分解定理可知, $Proj_s h_{i,j} \in S$ 且满足:

$$Proj_s h_{i,j} = Ay, \quad y \in \mathbf{R}^p \tag{3}$$

令 $Proj_{s^\perp} h_{i,j}$ 表示 $h_{i,j}$ 在子空间 S 正交补上的投影, 则满足: $h_{i,j} = Proj_{s^\perp} h_{i,j} + Proj_s h_{i,j}$. 已知子空间 S 等于矩阵 A 的列空间, 则子空间 S 正交补等于 A^{T} 的零空间. 从而, 有:

$$Proj_{s^\perp} h_{i,j} = h_{i,j} - Proj_s h_{i,j} \in Null(A^{\mathrm{T}}) \tag{4}$$

矩阵左乘 A^{T}

$$A^{\mathrm{T}}(h_{i,j} - Proj_s h_{i,j}) = 0 \tag{5}$$

即

$$A^{\mathrm{T}}(h_{i,j} - Ay) = 0 \tag{6}$$

化简后为

$$A^{\mathrm{T}}h_{i,j} - A^{\mathrm{T}}Ay = 0 \tag{7}$$

同理有

$$y = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}h_{i,j} \tag{8}$$

将式 (8) 代入式 (3)

$$Proj_s h_{i,j} = Ay = A(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}h_{i,j} \tag{9}$$

整理可得

$$Proj_s h_{i,j} = Ay = A(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}(x_i - x_j) \tag{10}$$

不失一般性, 我们假设子空间 S 实例化为一维特征空间 s_d, 则投影矩阵 A 转变为列向量 e_d (该一维空间的基向量), 投影形式化为 $e_d(e_d^{\mathrm{T}}e_d)^{-1}e_d^{\mathrm{T}}h_{ij}$. □

正交是一种特殊的向量关系, 正交分解是解析空间向量的常用模式, 主要贡献在于依据笛卡尔直角坐标系, 把复杂的空间矢量运算转化为互相垂直方向上的简单代数运算. 借助正交分解的理论观点, 我们使用标准正交基 $\{e_d\}_{1\le d\le m}(\text{s.t.}\langle e_i, e_j = 0\rangle \cap e_i^{\mathrm{T}}e_i = 1)$ 来实例化式 (10) 中的特征矩阵 A, 则对象 $x_i \in \mathbf{R}^m$ 与其近邻构成的空间向量在 m 个正交基上的投影可被表示为:

$$\begin{cases} Proj_{v_1} h_{i,j} = e_1(e_1^{\mathrm{T}}e_1)^{-1}e_1^{\mathrm{T}}h_{ij} = e_1e_1^{\mathrm{T}}x_i - e_1e_1^{\mathrm{T}}x_j = x_{i1} - x_{j1} \\ \cdots \\ Proj_{v_d} h_{i,j} = e_d(e_d^{\mathrm{T}}e_d)^{-1}e_d^{\mathrm{T}}h_{ij} = e_de_d^{\mathrm{T}}x_i - e_de_d^{\mathrm{T}}x_j = x_{id} - x_{jd} \\ \cdots \\ Proj_{v_m} h_{i,j} = e_m(e_m^{\mathrm{T}}e_m)^{-1}e_m^{\mathrm{T}}h_{ij} = e_me_m^{\mathrm{T}}x_i - e_me_m^{\mathrm{T}}x_j = x_{im} - x_{jm} \end{cases} \tag{11}$$

因此, 式 (2) 重写为:

$$\begin{aligned} V_i = & \left(\sum_j(\lambda_{i,1} - \lambda_{j,1}),\ \sum_j(\lambda_{i,2} - \lambda_{j,2}),\ \cdots,\ \sum_j(\lambda_{i,m} - \lambda_{j,m})\right)(e_1, e_2, e_3, \cdots, e_m)^{\mathrm{T}} = \\ & \left(\sum_j Proj_s h_{i,j},\ \sum_j Proj_s h_{i,j}, \cdots,\ \sum_j Proj_s h_{i,j}\right)(e_1, e_2, e_3, \cdots, e_m)^{\mathrm{T}} = \\ & \left(\sum_j e_1e_1^{\mathrm{T}}x_i - e_1e_1^{\mathrm{T}}x_j,\ \sum_j e_2e_2^{\mathrm{T}}x_i - e_2e_2^{\mathrm{T}}x_j, \cdots,\ \sum_j e_me_m^{\mathrm{T}}x_i - e_me_m^{\mathrm{T}}x_j\right)\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots \\ \cdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} = \\ & \left(\sum_j x_{i1} - x_{j1},\ \sum_j x_{i2} - x_{j2}, \cdots,\ \sum_j x_{im} - x_{jm}\right) \end{aligned} \tag{12}$$

进而, p-范数如下:

$$||v_i||_p = \left(\sum_d (|\sum_j x_{id} - x_{jd}|)^p\right)^{\frac{1}{p}}$$
$$\text{s.t. } 1 \le d \le m,\ \ x_j \in N_k(x_i) \tag{13}$$

考虑到计算成本, L_1 范数成为首选. 除了上述近邻投影范数外, 对象邻域的紧密性同样被作为判别依据. 为此, 本文将对象与其近邻之间残差平方和的指数离散作为局部密度.

定义 1[7] (局部密度). 对于 $\forall x_i \in X$, 局部密度 ρ_i 表示如下:

$$\rho_i = \exp\left(\left(-\frac{1}{k}\sum_{x_j \in N_k(x_i)} d(x_i, x_j)^2\right)\right) \tag{14}$$

$d(\cdot)$ 为欧氏距离形成的相似性度量, 聚类内部对象分布稠密, 上式取值较大, 边界对象分布相对稀疏, 其值偏小.

定义 2 (边界置信). 边界置信表征数据点隶属边界的程度. 将数据点 $x_i \in \mathbf{R}^m$ 的 $||v_i||_{p=1}$ 与密度 ρ_i 的比作为该点 x_i 的边界置信 φ_i, 表示如下:

$$\varphi_i = \frac{\|v_i\|_{p=1}}{\rho_i} \tag{15}$$

边界置信 φ_i 越大, 表明数据 x_i 位于聚类边缘的可能性越大. 由于边界点的判断不再单纯依靠密度, 而是基于客观存在的分布特征, 所以避免了由于密度度量失衡、邻域参数取值困难造成的影响.

以混合高斯数据集为例, 图 1 展示了边界置信的度量过程, 其中图 1(a) 可视化了含有两个高斯簇的测试数据集, 数据点 x_1 和 x_2 分别代表边界和核心对象; 图 1(b)、图 1(c) 显示了经过空间仿射变换后上述对象的局部邻域. 图 1(d) 和图 1(e) 展示了空间向量分解后的投影, 可以看出对象 x_1 在两个方向上的投影大部分是同向的, 而 x_2 的投影整体呈现对称状态; 图 1(f) 可视化了数据集的边界置信, 可以看出其边界置信的结果分布与实际分布相一致.

2.2　簇骨架构建和边界关联

根据式 (15) 中边界置信定义可知, 数据对象按其降序排序后, 聚类内部对象 (核心对象) 将集中分布在降序队列 L_X^{Asc} 的头部, 数据集合 X 可分割为边界集 X_{B} 和核心集 X_{core}, 如下:

$$\begin{cases} X_{\text{B}} = \{x_i \in X:\ \ \xi(x_i) = 1\} \\ \xi(x_i) = \begin{cases} 1, & \text{if } \varphi_i \le L_X^{\text{Asc}}[\lfloor n \times \tau \rfloor] \\ 0, & \text{otherwise} \end{cases} \\ X_{\text{core}} = X - X_{\text{B}} \end{cases} \tag{16}$$

其中 $\tau \in [0, 1]$ 为边界所占权重, 值越大, 表示数据集合中存在的边界对象越多, 参数详细讨论见第 4.4 节.

作为最新的边界剥离聚类代表算法, BP[12] 在逐层识别边界的同时, 建立当前边界点与上次迭代产生的历史边界点之间的隶属关系, 最后将边界点之间传递闭包按照最小距离原则关联到簇的核心区域. 然而, 后续的边界判别依赖于当前的边界剥离结果, 当前某些对象被误分后, 后续的剥离过程将被误导, 造成误差传递. 此外, 由于边界点所处区域的复杂性, 边界之间的传递闭包可能让原属于不同簇的边界建立错误的关联, 进而造成误差累积和簇合并, 因此, 边界之间的级联往往是冗余的. 为此, 提出了一种两阶段关联策略, 减少了对象间的冗余关联. 首先, 边界剥离后, 剩余对象之间具备良好的闭包性, 簇骨干可通过遍历核心点的可达近邻形成.

定义 3[15] (可达近邻). 任意核心点 x_i 和 x_j 具有可达近邻路径, 当且仅当存在链式关系 $(x_i, \cdots, x_r, x_{r+1}, \cdots, x_j)$, 其中任意相邻点对 (x_r, x_{r+1}) 满足 $d(x_r, x_{r+1}) < \max(d(x_{r+1}, \{N_k(x_r)\})$.

形式上, 我们利用 k 近邻将对象间的可达近邻模型化为有向连通图, 通过近邻矩阵 M 表示:

定义 4 (近邻矩阵). 对于 $\forall\ x_i, x_j \in X_{\text{core}}$ 满足可达近邻关系, 对应的近邻矩阵表示如下:

$$M_{n\times n}[i, j] = \begin{cases} 1, & \text{if } x_j \in N_k(x_i) \ \ \& \ \ x_j \in X_{\text{core}} \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

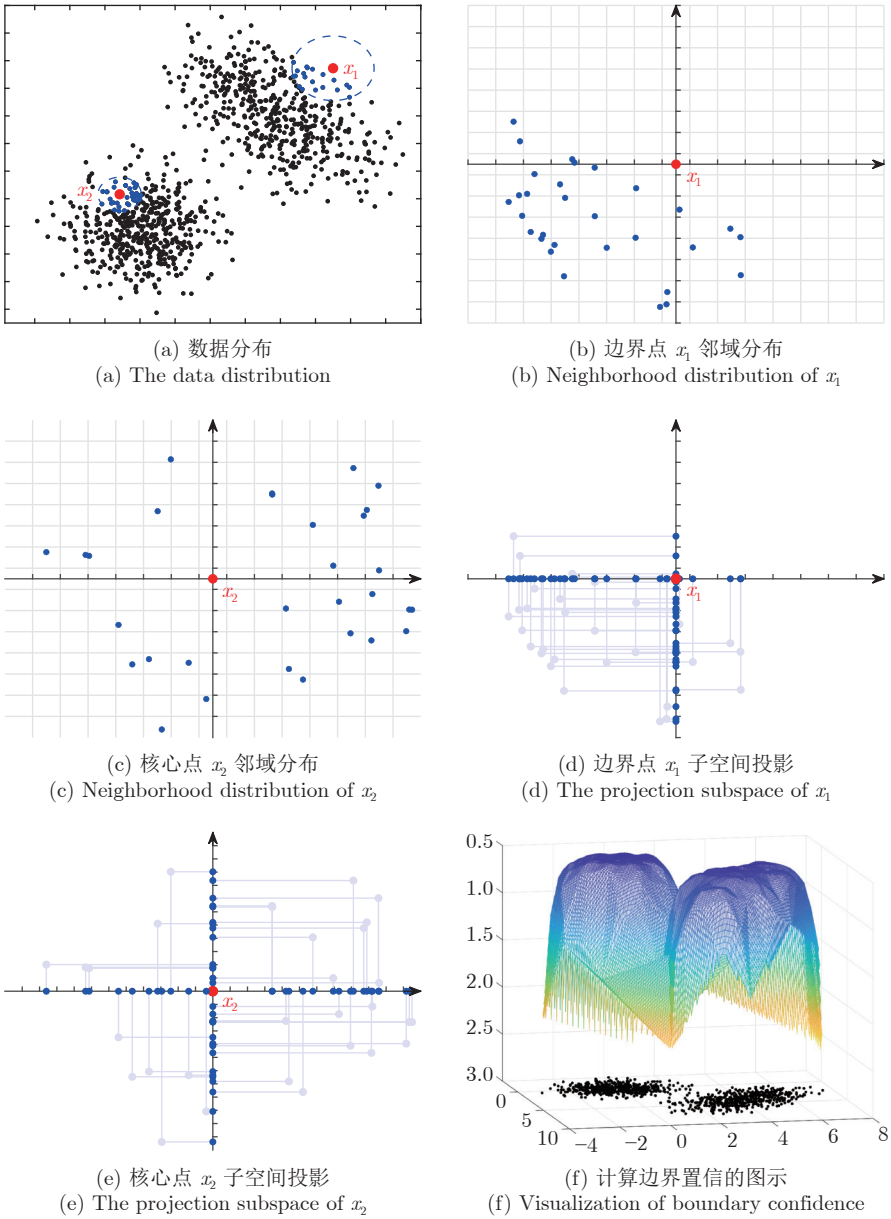
形式上, 非对称矩阵 M 中连通区域即为簇的骨架.

与 BP 不同, 在簇骨架建立后, 本算法并没有建立边界与边界之间的不稳定关联, 而是以稳定的簇骨架为基准共同指导边界隶属, 即距簇骨架中高密度核心对象的最小距离. 假设矩阵 M 中存在 K 个连通域, 则核心集 X_{core} 可表示为 $\{\sum C_{\text{core}}^{\ t} |_{t=1}^{K}\}$, $C_{\text{core}}^{\ t}$ 代表簇 t 的骨干对象. 边界点 x_i 的关联规则如下:

$$\phi_i = \arg\min_{t\in[1,\,K]} (d(x_i, x_j)) \ \text{ s.t. } x_j \in C_{\text{B}}^{(t)} \cap \rho_j > \rho_i \tag{18}$$

ϕ_i 指定了边界对象 $x_i \in X_{\text{B}}$ 的隶属信息. 总的来说, 对象划分第一阶段的关联标准是具有闭包性的可达近邻, 此时只关注簇骨架的建立, 而第二阶段的标准是密度优先级, 通过簇骨架来指导边界隶属, 旨在建立边界点与骨架之间的关联.

CBPVD 分为三步: 边界剥离、簇骨架构建、边界关联. 首先, 根据式 (14) 和 (15), 计算边界置信; 其次, 根据式 (16) 和 (17), 依据可达近邻建立簇骨



(a) 数据分布
(a) The data distribution

(b) 边界点 x_1 邻域分布
(b) Neighborhood distribution of x_1

(c) 核心点 x_2 邻域分布
(c) Neighborhood distribution of x_2

(d) 边界点 x_1 子空间投影
(d) The projection subspace of x_1

(e) 核心点 x_2 子空间投影
(e) The projection subspace of x_2

(f) 计算边界置信的图示
(f) Visualization of boundary confidence

图 1 计算边界置信的图示

Fig. 1 Graph with respect to boundary confidence calculation

架; 最后, 根据式 (18) 描述的高密度优先原则, 将边界划分到指定簇骨架中, 完成聚类. 图 2 给出了聚类可视化. 图 2(a) 显示了 Flame 的分布; 图 2(b) 展示了边界置信的热度图; 图 2(c) 中, CBPVD 将数据集分割为边界 (红色) 和核心点 (蓝色), 可以看出核心点较好地保留了簇的结构信息, 红色的边界点符合其形式化定义; 图 2(d) 显示了算法利用核心点间的强闭包性得到的簇骨架 (黄色和粉红色); 经过边界关联步骤, 图 2(e) 给出了最终聚类结果. 可以看出, CBPVD 有效识别了边界、簇骨架以及最终簇.

3 实验

除 K-means[20]、DPC[3] 外, 我们将聚类算法 (BP[12]、EC[5]、GB-DPC[4]、SNN-DPC[21]) 作为基线, 发表在 *T-PAMI* 上的 BP 算法是边界剥离聚类的最新代表, EC、GB-DPC、SNN-DPC 是 DPC 算法的最新改进. 对于 K-means、SNN-DPC, 我们提前给出正确的簇个数参数; 对于 DPC, 为了避免视觉误差, 我们利用 $\gamma = \rho \times \delta$ 降序, 选取正确数量的峰值点作为簇中心; 对于算法涉及的其他可调参数, 我们为 BP、DPC、SNN-DPC、EC、GB-DPC 设置合理的参数区间, 见表 1. 除常见合成数据集外, 我们

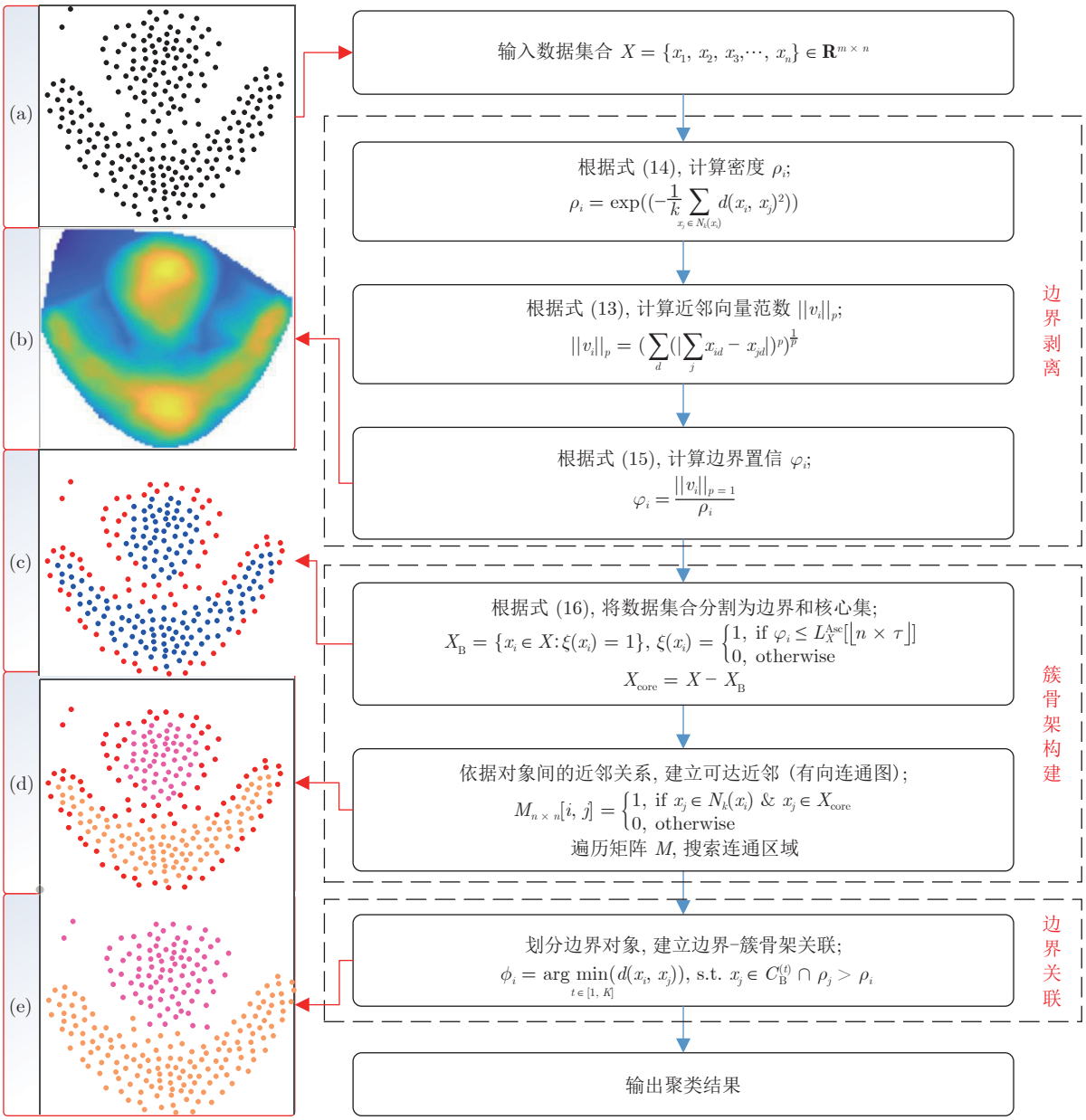


图 2　CBPVD 算法流程

Fig. 2　The algorithm flow of CBPVD

选用了 UCI 数据集、图片数据集, 涵盖多个复杂数据特征, 详细介绍见表 2. 为了综合评估算法性能, 本文使用 5 个评价指标 ACC (Accuracy), Purity, ARI (Adjusted rand index), FMI (Fowlkes-mallows index), JC (Jaccard coefficient) 来评估聚类结果.

3.1　合成数据集

从表 3 可看出, K-means 仅能处理相对独立的球形簇, 如 4k2-far, 原因在于单一的硬划分策略和一阶矩的中心计算方式限制了此类算法在交叉缠绕、流形嵌套等复杂形态下的效果. 特别地, 尽管数据

表 1　参数设置

Table 1　Hyperparameter configuration

Algorithm	Time complexity
K-means	k = The actual number of clusters
DPC	$dc \in [0.1, 20]$
SNN-DPC	$k \in [3, 70]$
GB-DPC	$dc \in [0.1, 20]$
EC	$dc \in [0.1, 20]$ or $dc \in [100, 300]$
BP	$k \in [3, 70]$, $b \in [0.1, 0.5]$, $\epsilon \in [0.1, 0.5]$, $T \in [100, 120]$, $C = 2$
CBPVD	$k \in [3, 70]$, $\tau \in [0.1, 0.4]$

表 2 数据集基本信息
Table 2 Basic information of datasets

数据集	大小	维度	簇数	特征
Compound	399	2	6	Multi-density, -Scale
R15	600	2	15	Micro, Adjoining
Flame	240	2	2	Overlapping
Parabolic	2000	2	2	Cross-winding, Multi-density
Jain	373	2	2	Cross-winding, Multi-density
4k2-far	400	2	4	Noise, Convex
D31	3100	2	31	Multiple-Micro cluster
Aggregation	788	2	7	Bridging
Spiral	240	2	3	Manifold
Heart disease	303	13	2	UCI, Clinical medicine
Hepatitis	155	19	2	UCI, Clinical medicine
German Credit	1000	20	2	UCI, Financial
Voting	435	16	2	UCI, Political election
Credit Approval	690	15	2	UCI, Credit record
Bank	4521	16	2	UCI, Financial credit
Sonar	208	60	2	UCI, Geology exploration
Zoo	101	7	16	UCI, Biological species
Parkinson	195	22	2	UCI, Clinical medicine
Post	90	8	3	UCI, Postoperative recovery
Spectheart	267	22	2	UCI, Clinical medicine
Wine	178	13	3	UCI, Wine ingredients
Ionosphere	351	34	2	UCI, Atmospheric structure
WDBC	569	30	2	UCI, Cancer
Optical Recognition	5620	64	10	OCR, Handwritten Digits
Olivetti Face	400	10304	40	Face, High-dimensional
You-Tube Faces	10000	10000	41	Video stream, Face
RNA-seq	801	20531	5	Gene expression, Nonlinear
REUTERS	10000	10000	4	Word, News, Text
G2-20	2048	2	2	Noise-20%
G2-30	2048	2	2	Noise-30%
G2-40	2048	2	2	Noise-40%
Size500	500	2	5	Gaussian
Size2500	2500	2	5	Gaussian
Size5000	5000	2	5	Gaussian
Size10000	10000	2	5	Gaussian
Dim128	1024	128	16	High-dimensional
Dim256	1024	256	16	High-dimensional
Dim512	1024	512	16	High-dimensional
Dim1024	1024	1024	16	High-dimensional
MINST	10000	784	10	OCR, high-dimensional

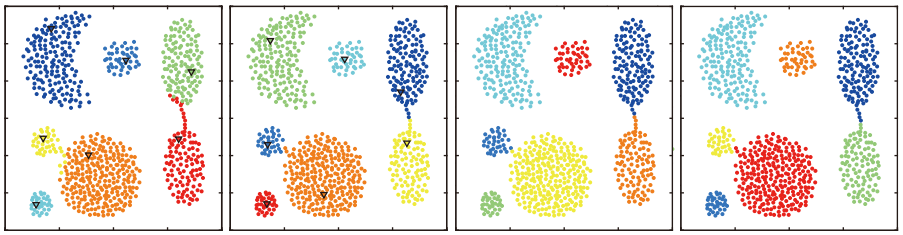
集 D31 呈现球形分布, 但簇间距过小, 在迭代优化时陷入局部最优, 导致部分对象没有正确划分, 如图 3(b) 第 1 图和表 3 第 9 行所示.

利用密度峰值思想, DPC、SNN-DPC、GB-DPC、EC 明显优于传统算法. 然而 DPC 和部分变种采用全局性质的密度度量, 即使给出了合理参数值, 仍无法有效处理多密度分布. 如 DPC 在 Compound 上识别了正确的簇个数, 但其中两个簇中心

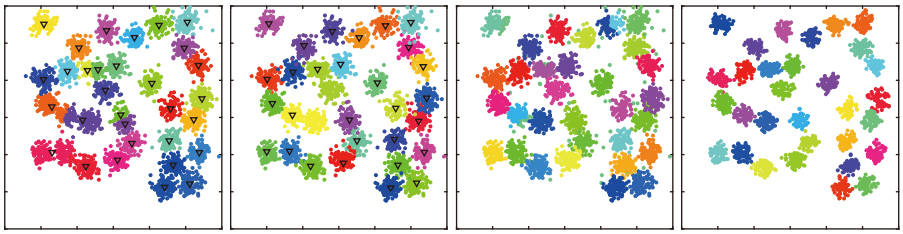
表 3 算法在合成数据集上的聚类表现

Table 3 Performance comparison of algorithms on all synthetic datasets

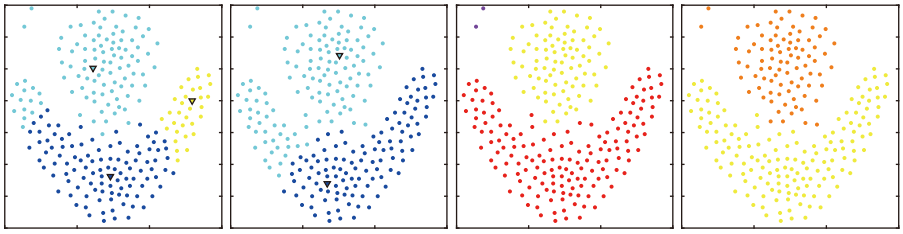
Dataset	Algorithm	Parameter	ACC	Purity	JC	ARI	FMI
4k2-far	K-means	$k=4$	**1**	**1**	0.13	**1**	**1**
	DPC	$dc=0.2168$	**1**	**1**	**1**	**1**	**1**
	GB-DPC	$dc=0.5$	**1**	**1**	0.26	**1**	**1**
	SNN-DPC	$k=10$	**1**	**1**	**1**	**1**	**1**
	EC	$\sigma=1$	**1**	**1**	**1**	**1**	**1**
	BP	—	0.98	0.99	0.01	0.97	0.98
	CBPVD	10, 0.1	**1**	**1**	**1**	**1**	**1**
Aggregation	K-means	$k=7$	0.78	0.94	0	0.76	0.81
	DPC	$k=7,\ dc=2.5$	0.91	0.95	0.22	0.84	0.87
	GB-DPC	$dc=2.5$	0.64	0.99	0.09	0.57	0.68
	SNN-DPC	$k=40$	0.98	0.98	0	0.96	0.97
	EC	$\sigma=5.5$	**1**	**1**	0	**1**	1
	BP	—	**1**	0.95	0.72	0.99	0.99
	CBPVD	16, 0.24	**1**	**1**	**1**	**1**	**1**
Compound	K-means	$k=6$	0.63	0.83	0.23	0.53	0.63
	DPC	$dc=1.25$	0.64	0.83	0.15	0.54	0.64
	GB-DPC	$dc=1.8$	0.68	0.83	0.23	0.54	0.64
	SNN-DPC	$k=12$	0.76	0.84	0.24	0.63	0.74
	EC	$\sigma=5.8$	0.68	0.86	0.68	0.59	0.69
	BP	—	0.77	**0.91**	**0.77**	0.65	0.73
	CBPVD	9, 0.08	**0.90**	**0.91**	0.13	**0.94**	**0.96**
Flame	K-means	$k=2$	0.83	0.83	0.83	0.43	0.73
	DPC	$dc=0.93$	0.84	0.84	0.16	0.45	0.74
	GB-DPC	$dc=2$	0.99	0.99	0.99	0.97	0.98
	SNN-DPC	$k=5$	0.99	0.99	0.01	0.95	0.98
	EC	$\sigma=5.4$	0.80	0.93	0.14	0.51	0.74
	BP	—	0.98	0.99	0.65	0.96	0.98
	CBPVD	3, 0.11	**1**	**1**	**1**	**1**	**1**
Spiral	K-means	$k=3$	0.35	0.35	0.33	−0.01	0.33
	DPC	$dc=1.74$	0.49	0.49	0.35	0.06	0.38
	GB-DPC	$dc=2.95$	0.44	0.44	0.36	0.02	0.35
	SNN-DPC	$k=10$	**1**	**1**	0	**1**	**1**
	EC	$\sigma=10$	0.34	0.34	0.32	0	0.58
	BP	—	0.50	0.56	0.50	0.17	0.49
	CBPVD	5, 0.32	**1**	**1**	**1**	**1**	**1**
Jain	K-means	$k=2$	0.79	0.79	0.21	0.32	0.70
	DPC	$dc=1.35$	0.86	0.86	**0.86**	0.52	0.79
	GB-DPC	$dc=1.35$	0.35	0.94	0.18	0.15	0.44
	SNN-DPC	$k=10$	0.86	0.86	0.14	0.52	0.79
	EC	$\sigma=7.65$	0.79	0.86	0.19	0.51	0.78
	BP	—	0.42	0.98	0.09	0.23	0.53
	CBPVD	13, 0.16	**1**	**1**	0	**1**	**1**
R15	K-means	$k=15$	0.81	0.86	0.03	0.80	0.81
	DPC	$dc=0.95$	0.99	0.99	0	0.98	0.98
	GB-DPC	$dc=0.2$	0.99	0.99	0.07	0.99	0.99
	SNN-DPC	$k=15$	0.99	0.99	0.99	0.99	0.99
	EC	$\sigma=1.45$	0.98	0.98	0.98	0.97	0.97
	BP	—	0.99	0.99	0	0.99	0.99
	CBPVD	9, 0.13	**1**	**1**	**1**	**1**	**1**
Parabolic	K-means	$k=2$	0.81	0.81	0.81	0.39	0.69
	DPC	$dc=1.5$	0.82	0.82	0.82	0.41	0.71
	GB-DPC	$dc=0.5$	0.94	0.94	0.06	0.77	0.89
	SNN-DPC	$k=9$	0.95	0.95	0.95	0.81	0.91
	EC	$\sigma=3.05$	0.73	0.73	0.73	0.21	0.66
	BP	—	0.19	0.98	0.03	0.13	0.36
	CBPVD	33, 0.27	**1**	**1**	**1**	**1**	**1**
D31	K-means	$k=31$	0.88	0.91	0	0.87	0.87
	DPC	$dc=1.8$	**0.97**	**0.97**	0	**0.94**	**0.94**
	GB-DPC	$dc=4$	0.46	0.46	0.02	0.32	0.45
	SNN-DPC	$k=40$	**0.97**	**0.97**	0	**0.94**	**0.94**
	EC	$\sigma=4$	0.91	0.91	0.06	0.88	0.89
	BP	—	0.94	0.95	0	0.90	0.91
	CBPVD	13, 0.15	**0.97**	**0.97**	**0.07**	**0.94**	**0.94**



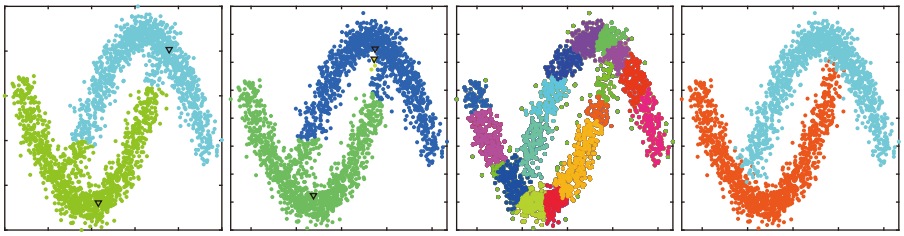
(a) Aggregation: DPC & EC & BP & CBPVD



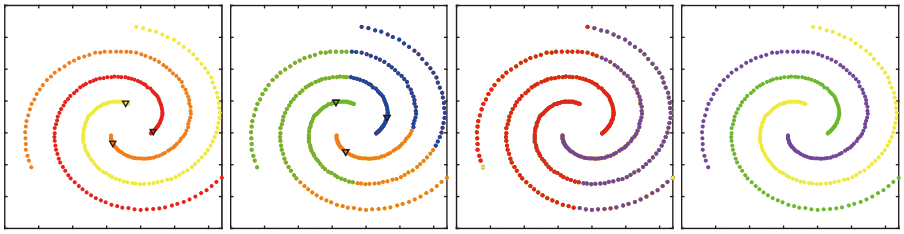
(b) D31: K-means & EC & BP & CBPVD



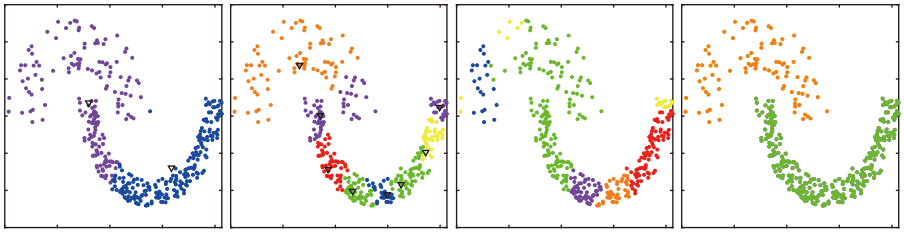
(c) Flame: EC & DPC & BP & CBPVD



(d) Parabolic: SNN-DPC & GB-DPC & BP & CBPVD



(e) Spiral: SNN-DPC & GB-DPC & BP & CBPVD



(f) Jain: K-means & GB-DPC & BP & CBPVD

图 3 合成数据集的聚类可视化结果

Fig. 3 Visualized results of algorithms on synthetic datasets

位于同一簇内, 完全忽略了相邻的低密度区域. 其次, 参数 dc 较为敏感, 即使微小的变化也容易造成聚类结果的大幅波动, 如图 3(c) 和表 3 中 Flame, 当 dc 为 0.93 (经验法则) 时, DPC 将底部簇中的部分对象分配到顶部区域, 然而 $dc = 1.07$ 时 (仅差 0.14), 其聚类达到最佳 (ACC = 1). 事实上, DPC 的实际效果往往低于表 3 中的实验数据, 因为实际场景下决策图极易产生视觉误差, 造成簇中心选取困难. 相反, SNN-DPC 表现较均衡, 这得益于 SNN-DPC 在聚类时不仅考虑了距离信息, 还融合了近邻和共享近邻信息, 并利用这些结构信息提出了新的相似性度量. 因此 SNN-DPC 具有更优的聚类表现, 尤其在 Parabolic 和 Spiral 上, 如图 3(d)、3(e), 但算法涉及复杂的多重嵌套, 高计算成本限制了其应用场景.

作为 DPC 的最新改进, GB-DPC 和 EC 均实现了过程自动化, 其中 EC 利用活动邻域内密度极值点的唯一性来自动筛选簇中心, 而 GB-DPC 依据统计量降序后的最大间隔来自动定位簇中心与其他对象的界限. 但从表 3 中可以看出, 两个算法并不稳定, 因为 EC 和 GB-DPC 均继承了原始 DPC 的密度度量, 在面对复杂分布时, 固定邻域的密度度量削弱了簇中心的特征表示, 造成选取偏差 (代表性低、数量过多、数量过少). 如包含两个簇的 Jain, 参数值 $dc = 1.35$ 与 DPC 相同, 但 GB-DPC 将数据集识别为 8 个簇 (如图 3(f) 中第 2 图), 精度 (ACC = 0.35) 远低于 DPC、SNN-DPC、EC; 相似地, EC 将 Flame 聚类为 3 个簇 (如图 3(c) 中第 1 图), 而实际为 2 个簇.

对于 Parabolic、Spiral、Jain 等非凸分布, 仅靠密度特征进行边界剥离的 BP 算法并没有准确识别簇的基本形状, 如图 3(d)、3(e)、3(f). 造成这一问题的因素有两个: 1) 以高斯核密度为标准的判别方式过于单一, 尤其面对分布不均的簇时, BP 往往将稀疏区域误判为边界, 进而导致稀疏簇演化为多个子簇, 即过划分, 如 BP 将原本仅含 2 个流形簇的 Parabolic 归为 17 个微簇; 2) 迭代式的逐层剥离往往会引起误差累积, 由于后续剥离要依据当前的剥离结果, 当某些对象被错分后, 后续的剥离过程将受到误导, 导致误差传递多米诺效应, 如 Spiral.

从图 3 和表 3 可以发现, 一些算法实际表现良好但某些指标值却很低, 甚至为 0. 如关于 JC 指标, K-means 在 4k2-far, DPC 在 R15, GB-DPC 在 Parabolic 上取值均很低. 这是由于该指标自身评判标准的偏好性: 通过比较每个对象的标签在数值上是否与 ground truth 严格相同来评价聚类, 而聚类结果的评价思想主要基于簇内对象的整体一致性. 以 Jain 两个流形簇为例, 数据对象 1 到 97, 标签为 1; 数据对象 98 到 373, 标签为 2. 然而 CBPVD 将对象 1 到 97 归为簇 2, 对象 98 到 373 归为簇 1. 因此, JC = (0+0)/373 = 0. 总的来说, 本文方法在 9 个合成数据集中的 7 个达到最佳聚类, 在剩余数据集上的指标与最佳聚类效果相差很小, 说明 CBPVD 在聚类复杂分布特征的数据时是有效的.

3.2 UCI 数据集

表 4 给出了算法在 UCI 数据集上的聚类结果. 对于实际场景下产生的数据, 每个属性有其明确且固定的物理意义. 依据一阶矩产生的簇中心并非实际对象, 因此, 性能严格依赖簇中心的硬划分算法 K-means 在多数情况下无法达到满意效果. 例如, K-means 在 Post 数据集上取得了最低的 ACC 分数 (0.425).

SNN-DPC 通过融合各种近邻信息确保了面对真实数据时的稳定性. 此外, 从 SNN-DPC 的实验数据可以看出, 密度估计中近邻采样比固定半径更有效. 例如, SNN-DPC 在 Spectheart、Bank 数据集上分别比原始 DPC 精度上提高了 0.353 和 0.17. 但 SNN-DPC 要预先指定实际的簇个数, 而大多数情况下很难得到数据先验信息. BP 算法共有 6 个超参: 最大迭代次数、边界判断阈值、近邻数、离群点判断阈值、迭代停止阈值、阈值可信度. 从表中可以看出, 其性能在大多数情况下略高于或等价于 SNN-DPC、GB-DPC, 从侧面验证了 BP 算法在高维情况下是有效的. 但其边界剥离过程需要迭代优化, 并且最终需要调用 DBSCAN 完成核心对象划分, 其效率相对较低.

理论上, DPC: dc、EC: dc、GB-DPC: dc 的参数范围均为正实数空间, 对于低维数据, 算法可通过其直观分布进行粗略估计, 而对于分布稀疏且无法直观显示的高维数据, 超参数取值缺少有效的视觉参考. 伴随高维数据分布相对稀疏这一特征, 类似截断距离 dc 这种粗糙度量难以描述对象的密度差异, 更不用说在此基础上形成的簇中心判别标准, 因此, 三个算法的聚类结果并不稳定. 例如, DPC 在 Parkinson、Wine、Optical Recognition 上表现最差; GB-DPC 在 9 个数据集上的聚类精度高于 DPC, 但同样存在 3 个数据集低于 DPC. 由于 EC 的簇中心搜索策略严格依赖于阈值参数, 密度失衡将会放大这种消极影响, 其整体表现低于 GB-DPC, 如 German 和 Hepatitics 数据集. 但不可否认, GB-DPC 和 EC 在实际场景中的应用性显

表 4 算法在 16 个真实数据集 (UCI) 上的聚类表现

Table 4 Performance comparison of algorithms on 16 real-world datasets

Dataset	Algorithm	Parameter	ACC	Purity	JC	ARI	FMI
Heart disease	K-means	$k = 2$	0.57	0.57	**0.57**	0.02	0.52
	DPC	$dc = 19.4424$	0.55	0.55	0.45	0.01	0.51
	GB-DPC	$dc = 19.4424$	0.54	0.54	0.54	0	0.71
	SNN-DPC	$k = 65$	0.59	0.59	0.41	0.03	0.54
	EC	$\sigma = 100$	0.54	0.54	0.46	−0.001	0.71
	BP	—	0.53	0.54	0.47	−0.002	0.68
	CBPVD	0.27, 26	**0.68**	**0.68**	0.32	**0.12**	**0.77**
Hepatitis	K-means	$k = 2$	0.66	0.84	0.66	−0.02	0.67
	DPC	$dc = 1$	0.63	0.84	0.01	−0.11	0.61
	GB-DPC	$dc = 10.2$	0.73	0.70	0.28	−0.01	0.72
	SNN-DPC	$k = 45$	0.70	0.84	0.30	−0.07	0.71
	EC	$\sigma = 5.8$	0.01	**1**	0.01	**0**	0.01
	BP	—	0.83	0.84	**0.83**	−0.02	0.84
	CBPVD	10, 0.2	**0.84**	0.84	0.76	**0**	**0.85**
German	K-means	2	0.67	0.70	0.33	0.05	0.66
	DPC	$dc = 53.9814$	0.61	0.70	0.61	0.03	0.58
	GB-DPC	$dc = 53.9814$	0.61	0.70	0.61	0.03	0.58
	SNN-DPC	$k = 30$	0.62	0.70	0.39	0.01	0.61
	EC	$\sigma = 100$	0.15	0.72	0.01	0.01	0.20
	BP	—	0.14	0.70	0.07	0.001	0.20
	CBPVD	4, 0.39	**0.83**	**0.83**	**0.83**	**0.43**	**0.74**
Voting	K-means	$k = 2$	0.51	0.61	0.51	−0.002	0.51
	DPC	$dc = 1$	0.81	0.81	0.19	0.39	0.7
	GB-DPC	$dc = 1.7$	0.87	0.87	**0.87**	0.54	0.78
	SNN-DPC	$k = 60$	**0.88**	0.88	0.12	0.57	**0.79**
	EC	$\sigma = 2$	0.75	**0.89**	0.75	0.42	0.68
	BP	—	0.86	0.91	0.05	0.59	**0.79**
	CBPVD	66, 0.33	**0.88**	0.88	0.12	**0.68**	**0.79**
Credit	K-means	$k = 2$	0.55	0.55	0.45	0.003	0.71
	DPC	$dc = 1$	0.68	0.68	0.68	0.13	0.60
	GB-DPC	$dc = 7$	0.55	0.55	0.45	0	0.71
	SNN-DPC	$k = 50$	0.61	0.61	0.61	0.05	0.53
	EC	$\sigma = 800$	0.56	0.59	0	0.02	0.68
	BP	—	0.33	0.69	0.26	0.06	0.35
	CBPVD	31, 0.33	**0.85**	**0.85**	**0.85**	**0.49**	**0.74**
Bank	K-means	$k = 2$	0.82	**0.88**	0.11	−0.002	0.82
	DPC	$dc = 2.39$	0.64	**0.88**	0.14	**0.04**	0.65
	GB-DPC	$dc = 10$	0.76	0.74	0.24	−0.02	0.76
	SNN-DPC	$k = 3$	0.81	**0.88**	**0.81**	0.01	0.81
	EC	$\sigma = 300$	0.82	0.82	0	0.02	0.82
	BP	—	0.24	**0.88**	0.09	0.01	0.29
	CBPVD	24, 0.2	**0.88**	**0.88**	0.12	0	**0.89**
Sonar	K-means	$k = 2$	0.54	0.54	0.34	**0.50**	0.50
	DPC	$dc = 2.82$	0.58	0.58	0.42	0.02	0.66
	GB-DPC	$dc = 1.4$	0.51	0.53	0.51	−0.004	0.51
	SNN-DPC	$k = 19$	0.50	0.53	0.50	−0.01	0.51
	EC	$\sigma = 1.6$	0.54	0.57	0.07	0.01	0.66
	BP	—	0.51	0.53	0.51	−0.004	**0.68**
	CBPVD	9, 0.66	**0.66**	**0.66**	**0.66**	0.10	0.60
ZOO	K-means	$k = 7$	0.76	0.84	**0.62**	0.6	0.69
	DPC	$dc = 2.4$	0.70	0.79	0.36	0.59	0.68
	GB-DPC	$dc = 3.6$	0.66	0.75	0.03	0.48	0.60
	SNN-DPC	$k = 5$	0.56	0.56	0.12	0.31	0.53
	EC	$\sigma = 2.3$	0.80	0.81	0.08	0.65	0.73
	BP	—	0.59	0.59	0.23	0.4	0.62
	CBPVD	10, 0.15	**0.86**	**0.86**	0.01	**0.93**	**0.94**

表 4 算法在 16 个真实数据集 (UCI) 上的聚类表现 (续表)

Table 4 Performance comparison of algorithms on 16 real-world datasets (continued table)

Dataset	Algorithm	Parameter	ACC	Purity	JC	ARI	FMI
Parkinson	K-means	$k=2$	0.72	0.75	0.28	0	0.74
	DPC	$dc=1.3$	0.66	0.75	0.34	0.05	0.63
	GB-DPC	$dc=3$	0.71	0.71	0.29	−0.05	0.75
	SNN-DPC	$k=80$	0.72	0.75	0.28	0.11	0.69
	EC	$\sigma=135$	0.70	0.75	0.7	0.14	0.66
	BP	—	0.19	**0.98**	0.03	0.13	0.36
	CBPVD	13, 0.16	**0.82**	0.82	**0.82**	**0.25**	**0.81**
POST	K-means	$k=3$	0.43	0.71	0.43	−0.002	0.45
	DPC	$dc=1$	0.53	0.71	0.53	−0.01	0.52
	GB-DPC	$dc=2.7$	0.61	0.71	0.38	−0.03	0.62
	SNN-DPC	$k=60$	0.61	0.71	0.61	0.02	0.60
	EC	$\sigma=6$	0.70	0.72	0.05	0.04	0.74
	BP	—	0.62	0.72	0.09	0.04	0.61
	CBPVD	10, 0.01	**0.79**	**0.79**	**0.79**	**0.25**	**0.78**
Spectheart	K-means	$k=2$	0.64	**0.92**	0.64	−0.05	0.69
	DPC	$dc=1.4142$	0.52	**0.92**	0.48	−0.01	0.65
	GB-DPC	$dc=1.1$	0.52	**0.92**	0.08	0	**0.92**
	SNN-DPC	$k=80$	0.87	**0.92**	0.13	**0.11**	0.87
	EC	$\sigma=4$	0.92	**0.92**	0.08	0	**0.92**
	BP	—	0.91	**0.92**	**0.91**	−0.01	0.91
	CBPVD	15, 0.26	**0.92**	**0.92**	0.08	0	**0.92**
Wine	K-means	$k=4$	0.66	0.70	0.11	0.32	0.54
	DPC	$dc=0.5$	0.55	0.58	0.43	0.15	0.57
	GB-DPC	$dc=5.6$	0.60	0.71	0.35	0.27	0.50
	SNN-DPC	$k=3$	0.62	0.66	0.51	0.34	0.63
	EC	$\sigma=250$	0.66	0.66	0.66	0.37	0.66
	BP	—	0.68	0.71	0.21	0.34	0.56
	CBPVD	4, 0.03	**0.91**	**0.95**	**0.75**	**0.8**	**0.87**
Ionosphere	K-means	$k=2$	0.71	0.71	0.71	0.18	0.61
	DPC	$dc=3.7$	0.65	0.65	0.35	0.02	0.73
	GB-DPC	$dc=3.7$	0.65	0.65	0.35	0.02	0.73
	SNN-DPC	$k=34$	0.67	0.67	0.67	0.11	0.57
	EC	$\sigma=5$	0.65	0.67	0	0.05	0.73
	BP	—	0.80	0.80	0.80	0.34	0.76
	CBPVD	6, 0.51	**0.83**	**0.83**	**0.87**	**0.42**	**0.77**
WDBC	K-means	$k=2$	0.74	0.89	0.22	0.54	0.76
	DPC	$dc=5$	0.67	0.67	**0.67**	0.10	0.60
	GB-DPC	$dc=3.9$	0.63	0.63	0.63	0	0.73
	SNN-DPC	$k=3$	0.81	0.81	0.19	0.36	0.75
	EC	$\sigma=350$	0.82	0.87	0	0.49	0.78
	BP	—	0.44	0.88	0.12	0.25	0.52
	CBPVD	3, 0.6	**0.95**	**0.95**	0.05	**0.81**	**0.91**
RNN-seq	K-means	$k=5$	0.75	0.75	0.17	0.72	0.79
	DPC	$dc=159.6$	0.70	0.73	0.39	0.62	0.76
	GB-DPC	$dc=159.6$	0.73	0.73	0.54	0.63	0.77
	SNN-DPC	$k=30$	0.73	0.73	0.001	0.51	0.71
	EC	$\sigma=240$	0.38	0.38	0.17	0	0.49
	BP	—	0.78	0.74	0.002	0.63	0.72
	CBPVD	10, 0.4	**0.996**	**0.996**	**0.81**	**0.99**	**0.99**
REUTERS	K-means	$k=4$	0.50	0.58	0.22	0.15	0.41
	DPC	$dc=3.5$	0.43	0.43	0.28	0.10	0.46
	GB-DPC	$dc=3.5$	0.35	0.55	0	0.14	0.41
	SNN-DPC	$k=40$	0.49	0.50	0.49	**0.24**	0.54
	EC	$\sigma=300$	0.40	0.40	0.40	0	**0.55**
	BP	—	0.39	0.41	0.38	0.01	0.50
	CBPVD	20, 0.1	**0.61**	**0.61**	**0.61**	0.23	0.47

著高于 DPC 和 SNN-DPC.

总的来说, 从表 3 和表 4 中可以看出, 本文算法在 28 个数据集的 140 个评价指标值中有 120 个达到了最高值, 表明 CBPVD 在大多数情况下优于同类算法, 理论分析详见第 4 节. 相对于 GB-DPC、SNN-DPC、EC 等密度峰值聚类的最新改进, 我们可以有效处理尺度不一、嵌套弯曲、流形、多源噪声等复杂分布. 这一优势主要得益于细粒度的多维混合特征, 避免了边界在聚类过程中的干扰. 特别地, 对于高维异构数据, 如含有 20 531 个属性的生物数据集 RNN-seq, CBPVD 的聚类结果达到了最佳 (ACC = 0.996), 而其他算法表现一般, 这主要由于 CBPVD 的边界剥离策略是针对维度层面的, 可以提取到高维空间中更细粒度的边界特征, 这一优势同样体现在文本类型的 RELETER 数据集上. 此外, 相比于密度峰值聚类中的簇个数参数, CBPVD 参数具有更高的鲁棒性, 详见第 4.4 节. 另一方面, 对于含有 6 个参数的边界剥离聚类代表 BP, CBPVD 参数更少且无需冗余迭代, 详见第 4.1 节. 此外, 多视角的边界特征确保了算法具有良好的高维边界模式识别能力, 详见第 4.2 节.

3.3 图像数据集

为了检验算法在高维、实际场景下的聚类效果, 本文选用 Olivetti 人脸数据集、Optdigits 光学识别数据集、You-Tube Faces (YTF) 视频数据集进行实验. 图 4 可视化了 CBPVD 算法在 Olivetti 数据集的聚类结果, 表 5 分别展示了三个图像数据集的聚类定量评估.

Olivetti 数据集采集了 40 个受试者在不同俯仰角度、光照、表情、时间、配饰、穿戴下的 400 张脸部图像 (92×112), 其规模为 $400\times10\,304$. 实际聚类中, EC 算法产生过划分现象, 共产生了 131 个簇, 其中多个簇中仅含有一个对象. 这一结果的原因或许是样本量 (400) 与簇数量 (40) 处于同一量级, 造成密度参数 dc 失效. BP 算法将 400 张人脸图像划分为 3 类, 与实际情况相差很大, 推断这一情况主要是由于数据维度较高 (10 304), 适用于凸形簇的 BP 最初无法得到有效的边界信息, 随着迭代式的剥离过程, 划分错误被不断传递和积累. 从表 5 和图 4 中可以看出, 本文算法在 6 个指标中的 5 个达到了最佳. 虽然聚类精度未能达到 100%, 但 CBPVD 算法分类错误的图像数量是最少的. 这同样在 You-Tube Faces 数据集上可以看出, 由于视频内容的复杂性, DPC、SNN-DPC、EC 等算法的聚类精度均浮动在 0.5 左右, 甚至 GB-DPC 为 0.31, 而本算法的聚类达到了 0.65.

在光学识别领域, Optdigits 数据集收录了 43 人共计 1 797 张 8×8 手写数字图像, 由于个人习惯和偏好不同, 字体在笔画、线条粗细、大小、形状上均存在差异, 并且出现连笔、断笔、涂改、潦草等情况, 表 5 给出了算法和 6 个对比基线的聚类结果. 可以看出, 面对高维的图像数据, 基于密度峰值的聚类算法 (DPC、SNN-DPC、GB-DPC) 的表现效果均不理想, 本文算法的聚类效果显著好于其他算法 (ACC = 0.93).

4 讨论

4.1 性能分析

根据图 2 可知, 算法复杂度主要存在于边界置信 φ 计算、边界剥离、近邻矩阵 M 建立、连通区域遍历、边界隶属. 算法使用 KD 树 ($O(n\log_2 n)$) 来确定对象近邻, 则密度度量 (式 (14)) 需要 $O(k*n)$; 对于近邻向量的子空间投影和范数计算 (式 (12), 式 (13)), 需要在原始数据空间中按基向量方向逐一进行分解, 耗时 $O(mkn)$. 因此, 计算边界置信 (式 (15)) 需要 $O(n\log_2 n+kn(m+1))$; 随后, 边界剥离过程需要将对象按边界置信排序 (式 (16)), 耗时

图 4 CBPVD 在 Olivetti 上的聚类结果

Fig. 4 The clustering results on Olivetti faces by CBPVD

表 5　图像数据集的聚类结果

Table 5　Performance comparison of algorithms on image datasets

Dataset	Algorithm	Parameter	ACC	Purity	JC	ARI	FMI
Olivetti	K-means	$k=40$	0.64	0.67	0.01	0.517	0.54
	DPC	$dc=0.922$	0.59	0.65	0.02	0.523	0.56
	GB-DPC	$dc=0.65$	0.65	0.73	**0.05**	0.577	0.59
	SNN-DPC	$k=40$	0.66	0.74	0	0.585	0.61
	EC	$\sigma=3\,700$	0.44	0.58	0.02	0.22	0.32
	BP	—	0.03	0.03	0.03	0	0.15
	CBPVD	4, 0.14	**0.75**	**0.78**	0	**0.646**	**0.68**
Optical	K-means	$k=10$	0.71	0.73	0.04	0.58	0.63
	DPC	$dc=1.1$	0.60	0.62	0.09	0.475	0.56
	GB-DPC	$dc=10.5$	0.61	0.62	0.02	0.468	0.56
	SNN-DPC	$k=10$	0.71	0.73	0.20	0.629	0.69
	EC	$\sigma=30$	0.69	0.69	0.17	0.596	0.67
	BP	—	0.80	0.85	0	0.717	0.75
	CBPVD	4, 0.45	**0.93**	**0.95**	**0.30**	**0.889**	**0.90**
You-Tube Faces	K-means	$k=41$	0.52	0.63	0.02	0.51	0.53
	DPC	$dc=6.5$	0.53	0.62	0.02	0.48	0.51
	GB-DPC	$dc=6.5$	0.31	0.31	0	0.25	0.35
	SNN-DPC	$k=59$	0.57	0.69	0.03	0.47	0.50
	EC	$\sigma=100$	0.51	0.56	0.01	0.40	0.46
	BP	—	0.52	0.62	**0.04**	0.19	0.32
	CBPVD	20, 0.1	**0.66**	**0.88**	0.01	**0.62**	**0.64**

$O(n\log_2 n)$; 建立近邻矩阵 M (式 (17)) 的时间复杂度为 $O(kn)$, 随后 M 中联通区域的遍历涉及 $O(k+n)$. 最后, 算法依据密度优先级将边界点划分至对应的簇骨架中 (式 (18)), 复杂度为 $O(\tau n(1-\tau)n)$, 形成最终聚类. 总的来说, 算法时间复杂度可约简为 $O(2n\log_2 n+kn(m+2)+k+n+\tau n(1-\tau)n)\sim O(n^2)$. 如表 6 所示, 本文算法的复杂度与大多数算法处于同一量级.

表 6　复杂度对比

Table 6　The time complexity of algorithms

Algorithm	Time complexity
DBSACN	$O(n^2)$
DPC	$O(n^2)$
GB-DPC	$O(n\log_2 n)$
SNN-DPC	$O(n^2)$
DPC-RDE	$O(n^2)$
RA-Clust	$O(n\sqrt{n})$
EC	$O(n^2)$
BP	$O(n^2)$
CBPVD	$O(n^2)$

除了定性分析, 我们分别从数据量和维度方面测试了 CBPVD 与 BP 的运行效率. 所用数据集为表 2 中 Size 和 Dim 系列数据集 (Size: 500, 2 500, 5 000, 10 000; Dim: 128, 256, 512, 1 024), 实验结果如图 5. 虽然 CBPVD 与 BP 算法处于相同的复杂量级, 但运行效率却存在一定差异. 由于 BP 算法的边界剥离过程为迭代优化, 耗时相对较长, 此外, BP 在处理 Size10 000 数据集时难以收敛, 时间明

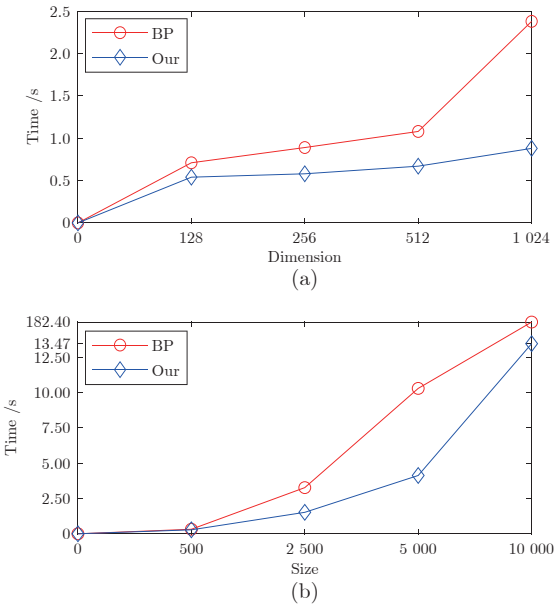


图 5　运行时间测试

Fig. 5　Running time test

显增加. 相反, 本文算法 CBPVD 无需迭代, 其主要耗时为连通区域的遍历和边界隶属.

4.2 边界模式识别

在实际场景中, 聚类边界指的是那些隶属明确但又具备脱离本类属性特征的一类特殊数据, 如在 OCR (Optical character recognition) 领域, 由于书写习惯和偏好的影响, 聚类边界表征那些属于同一类但异于常态 (难以肉眼区分) 的字体. 对具有边界特征的字体识别, 有助于进一步获取模式识别和决策判别信息.

以 MNIST 为例, 共计 10 000 张 28×28 大小的图像, 图 6 展示了本文算法的边界识别结果. 图 6(a) 显示了根据边界置信降序排序中前 400 张图像, 可以看出这些数字书写非常潦草, 具有模糊、断笔、空洞、连笔等边界特征, 难以辨认, 与正常字体形态具有较大的差异. 图 6(b) 展示了边界置信降序排序 8 400 ~ 8 800 的图像信息, 很明显, 这些字体较为工整、清晰, 易于区分. 因此, 本文算法 CBPVD 提出的边界置信概念较好地描述了边界特征, 在边界模式识别方面是有效的.

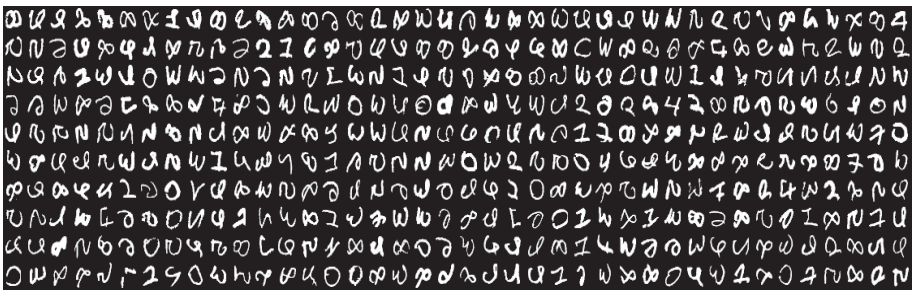
4.3 统计测试

为了更直观、科学地判断不同算法的性能, 我们结合 Friedman 检验和 Nemenyi 后续检验进一步分析了表 3、表 4、表 5 中的量化数据. 为此, 零假设表示所有算法性能一致, 备择假设推断算法存在性能差异. 经过统计, 算法在 28 个数据集上的平均序 (Average rank) 分别为 K-means: 4.7, DPC: 4.52, GB-DPC: 4.80, SNN-DPC: 4.27, EC: 4.36, BP: 4.29, CBPVD: 1.07. 数据上, 算法的综合性能排序为 CBPVD > SNN-DPC > BP > EC > DPC > K-means.

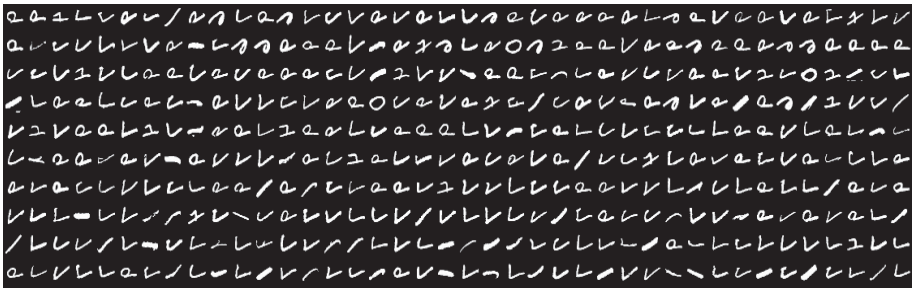
假设算法数量为 Q, 数据集数量为 N, 则统计变量 τ_F 服从自由度为 $Q-1$ 和 $(N-1)\times(Q-1)$ 的 F 分布, 公式如下:

$$\begin{cases}\tau_F=\dfrac{(N-1)\tau_{\chi^2}}{N(Q-1)-\tau_{\chi^2}}\\ \tau_{\chi^2}=\dfrac{12N}{Q(Q+1)}\left(\displaystyle\sum_{i=1}^{Q}r_i^2-\frac{Q(Q+1)^2}{4}\right)\end{cases}\tag{19}$$

此时, τ_F 的自由度为 $7-1=6$ 和 $28-1=27$. 在置信水平 0.05 的情况下, 临界值 $F(6,27)=2.154948$, $\tau_F=15.606858$ 远大于临界值. 因此, 零假设不成立. 在 Nemenyi 后续检验中, 临界差异 (Critical difference) $CD=q_\alpha\sqrt{K(K-1)/6N}=1.70$, 其中 q_α 为检验临界值, 可在统计教材中查询, 此处 q_α 为 2.95 (置信水平为 0.05 时). Nemenyi 后



(a) 聚类边界 (top 400)
(a) The identified border points (top 400)



(b) 聚类内部对象 (8 400 ~ 8 800)
(b) The identified core points (8 400 ~ 8 800)

图 6 在手写体数据集上识别的边界信息
Fig. 6 The boundary information extraction on MNIST

续检验认为若算法间平均序差大于 CD, 则这两种算法的聚类表现明显不同. 图 7 给出了 Nemenyi 后续检验的可视化结果, 其中方点表示平均序值, 线段是长度为 $CD = 1.70$ 的置信区间, 垂线代表本文算法的上置信水平. 可以看出, 本文算法的蓝色垂线与其他算法的红线没有重叠 (本文算法与其他算法的平均序差值远高于 CD 值). 因此, 本文算法优于其他算法这一结论具有统计学意义.

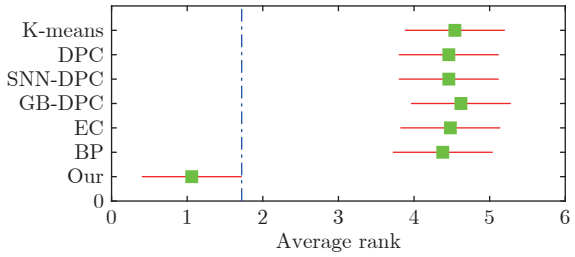


图 7　Nemenyi 检验结果

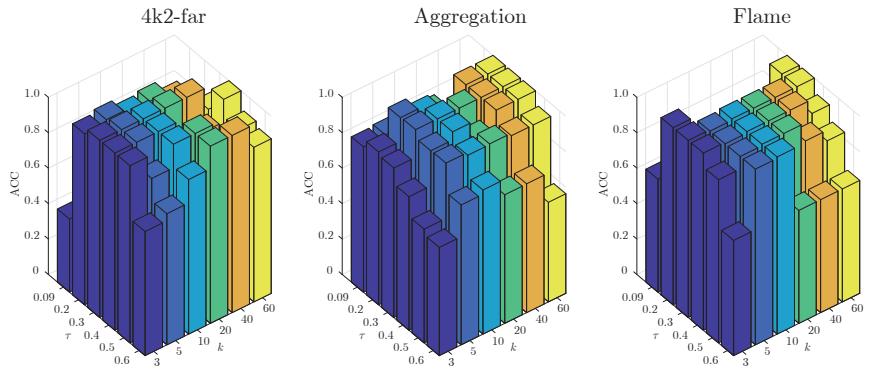
Fig. 7　The Nemenyi test result

4.4　鲁棒性分析

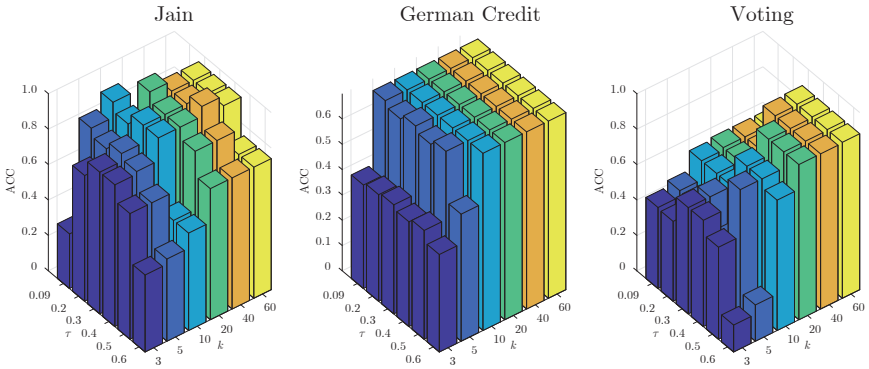
本算法存在两个超参: 近邻 k 旨在提取局部信息, 边界阈值 τ 用于剥离边界. 从表 3 和表 4 的结果来看, 本算法的参数值整体上分布在 $k \in [5, 40]$, $\tau \in [0.01, 0.4]$, BP 最大迭代次数 T 和近邻参数 k 取值范围较小, 但涉及 6 个超参 (最大迭代次数、边界判断阈值、近邻数、离群点判断阈值、迭代停止阈值、阈值可信度). 直观上, 本文参数更加稳定且易于确定.

图 8 展示了复杂分布、高维场景、噪音干扰下的鲁棒性结果. 如图所示, 算法的聚类效果并没有异常波动. 详细来说, 当 k 较小时, 算法无法获得充足的局部结构信息, 边界在原始空间和投影子空间上的特征并不明显, 无法完全判定边界对象, 如图 8(b) 第 3 图; 随着 k 增长, Olivetti 中边界和核心对象的特征差异逐渐明显, 其精度呈现不断上升并逐渐稳定的趋势; 当 k 过于大时, 算法可能在密度度量和边界剥离时意外包含了其他簇的对象, 导致精度下降, 如图 8(a) 第 1 图. 特别地, 图 8(d) 显示了噪音场景下的鲁棒性结果, 三个数据集分布一致且噪音含量依次增加. 尽管存在大量噪音 (G2-40 数据集已无法从视觉上判别簇结构), 算法仍达到了理想且稳定的聚类精度, 这主要得益于主动边界剥离的引入.

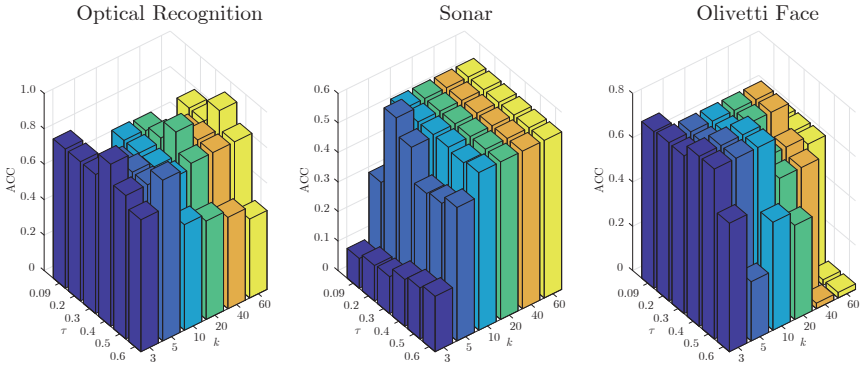
边界阈值 τ 的波动规律与 k 相似, 随着阈值增大, 边界点被不断识别, 形成的簇骨架有效表征簇的核心结构. 然而继续增大时, 会将部分核心对象错误剔除, 进而导致聚类精度的降低. 总的来说, 当 $k \approx \sqrt{n}$, $\tau \approx 0.15$ 时, 可达到理想聚类.



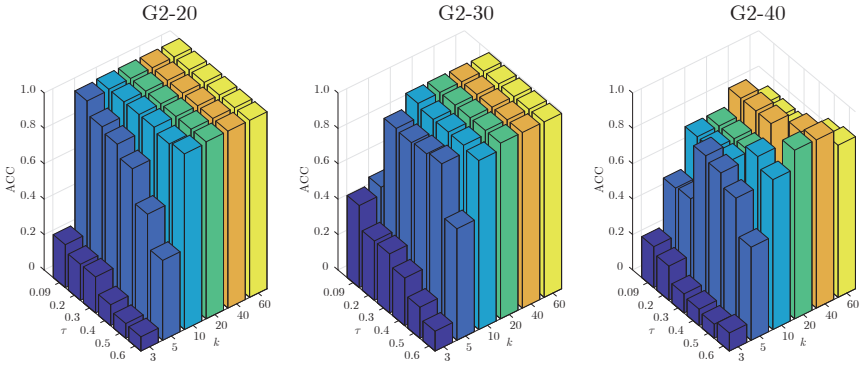
(a) 4k2-far & Aggregation & Flame



(b) Jain & German Credit & Voting



(c) Optical Recognition & Sonar & Olivetti Face



(d) G2-20 & G2-30 & G2-40

图 8 鲁棒性分析

Fig. 8 Robustness analysis

5 结论

本文通过融合边界点 (核心点) 在投影子空间上的维度偏斜性 (对称性) 和原始数据空间上的空间稀疏性 (紧密性) 主动剥离聚类边界, 提出了基于空间向量分解的边界剥离密度聚类 CBPVD, 不仅解决了现有边界剥离聚类中边界判别标准单一、嵌套迭代、倾向于分布均匀、球形簇的局限, 同时提升了算法在复杂分布、高维数据下的表现. 算法的有效性可归为如下因素, 首先, 边界点的判断不再单一依靠密度大小, 而是以原始数据空间和投影子空间为基准从分布大小和分布方向两个视角强化边界的细粒度特征表示; 其次, 提出的两阶段对象关联策略避免了现有算法中边界之间的冗余级联, 减少了对象划分错误的闭包传递; 此外, CBPVD 无需迭代优化, 其超参数少于 BP 算法且易于确定; 最后, 从理论分析和多维实验上对比了 CBPVD 与 EC、DPC、BP、K-means、GB-DPC、SNN-DPC 的聚类表现, 累计 40 个数据集 (合成、UCI、图像) 的实验结果和 4 个维度的深入分析 (鲁棒性、统计排名、性能、边界模式识别) 表明 CBPVD 在高维聚类和边界模式信息提取方面的有效性. 如何将边界剥离聚类思想嵌入深度神经网络以实现参数推荐和扩大应用场景是下一步工作.

致谢

感谢鹏城实验室、广东省安全智能新技术重点实验室在理论研究方面给予的指导和帮助.

References

1 Zhu Ying-Wen, Chen Song-Can. High dimensional data stream clustering algorithm based on random projection. *Journal of Computer Research and Development*, 2020, **57**(8): 1683−1696 (朱颖雯, 陈松灿. 基于随机投影的高维数据流聚类. 计算机研究与发展, 2020, **57**(8): 1683−1696)

2 Xia S Y, Peng D W, Meng D Y, Zhang C Q, Wang G Y, Giem E, et al. Ball k k-means: Fast adaptive clustering with no bounds. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2022, **44**(1): 87−99

3 Rodriguez A, Laio A. Clustering by fast search and find of density peaks. *Science*, 2014, **344**(6191): 1492−1469

4 Flores K G, Garza S E. Density peaks clustering with gap-based automatic center detection. *Knowledge-Based Systems*, 2020, **206**: Article No. 160350

5 Wang S L, Li Q, Zhao C F, Zhu X Q, Yuan H N, Dai T R. Extreme clustering —— A clustering method via density extreme points. *Information Sciences*, 2021, **542**: 24−39

6 Hou J, Zhang A H, Qi N M. Density peak clustering based on relative density relationship. *Pattern Recognition*, 2020, **108**: Article No. 107554

7 Xu X, Ding S F, Wang Y R, Wang L J, Jia W K. A fast density peaks clustering algorithm with sparse search. *Information Sciences*, 2021, **554**: 61−83

8 Weng S Y, Gou J, Fan Z W. h-DBSCAN: A simple fast DB-

SCAN algorithm for big data. In: Proceedings of Asian Conference on Machine Learning. New York, USA: PMLR, 2021. 81−96

9 Ester M, Kriegel H, Sander J, Xu X W. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of Knowledge Discovery and Data Mining. New York, USA: ACM, 1996. 226−231

10 Fang F, Qiu L, Yuan S F. Adaptive core fusion-based density peak clustering for complex data with arbitrary shapes and densities. *Pattern Recognition*, 2020, **107**: Article No. 107452

11 Chen M, Li L J, Wang B, Cheng J J, Pan L N, Chen X Y. Effectively clustering by finding density backbone based-on kNN. *Pattern Recognition*, 2016, **60**: 486−498

12 Averbuch-Elor H, Bar N, Cohen-Or D. Border peeling clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, **42**(7): 1791−1797

13 Cao X F, Qiu B Z, Li X L, Shi Z L, Xu G D, Xu J L. Multidimensional balance-based cluster boundary detection for high-dimensional data. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, **30**(6): 1867−1880

14 Qiu B Z, Cao X F. Clustering boundary detection for high dimensional space based on space inversion and Hopkins statistics. *Knowledge-Based Systems*, 2016, **98**: 216−225

15 Zhang R L, Song X H, Ying S R, Ren H L, Zhang B Y, Wang H P. CA-CSM: A novel clustering algorithm based on cluster center selection model. *Soft Computing*, 2021, **25**(13): 8015−8033

16 Li X L, Han Q, Qiu B Z. A clustering algorithm using skewness-based boundary detection. *Neurocomputing*, 2018, **275**: 618−626

17 Yu H, Chen L Y, Yao J T. A three-way density peak clustering method based on evidence theory. *Knowledge-Based Systems*, 2021, **211**: Article No. 106532

18 Tong Q H, Li X, Yuan B. Efficient distributed clustering using boundary information. *Neurocomputing*, 2018, **275**: 2355−2366

19 Zhang S Z, You C, Vidal R, Li C G. Learning a self-expressive network for subspace clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York, USA: IEEE, 2021. 12393−12403

20 MacQueen J. Classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symp. Math. Statist. Probability. Berkeley, USA: University of California Press, 1967. 281−297

21 Liu R, Wang H, Yu X M. Shared-nearest-neighbor-based clustering by fast search and find of density peaks. *Information Sciences*, 2018, **450**: 200−226

22 Gong C Y, Su Z G, Wang P H, Wang Q. Cumulative belief peaks evidential K-nearest neighbor clustering. *Knowledge-Based Systems*, 2020, **200**: Article No. 105982

23 Qiu Bao-Zhi, Zhang Rui-Lin, Li Xiang-Li. Clustering algorithm for mixed data based on residual analysis. *Acta Automatica Sinica*, 2020, **46**(7): 1420−1432
(邱保志, 张瑞霖, 李向丽. 基于残差分析的混合属性数据聚类算法. 自动化学报, 2020, **46**(7): 1420−1432)

24 Zhang R L, Miao Z G, Tian Y, Wang H P. A novel density peaks clustering algorithm based on Hopkins statistic. *Expert Systems with Applications*, 2022, **201**: Article No. 116892

25 Liu Y H, Ma Z M, Yu F. Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy. *Knowledge-Based Systems*, 2017, **133**: 208−220

26 Abbas M, El-Zoghabi A, Shoukry A. DenMune: Density peak based clustering using mutual nearest neighbors. *Pattern Recognition*, 2021, **109**: Article No. 107589

27 Ren Y Z, Hu X H, Shi K, Yu G X, Yao D Z, Xu Z L. Semi-supervised denpeak clustering with pairwise constraints. In: Proceedings of Pacific Rim International Conference on Artificial Intelligence. Cham, Switzerland: Springer, 2018. 837−850

28 Ren Y Z, Wang N, Li M X, Xu Z L. Deep density-based image clustering. *Knowledge-Based Systems*, 2020, **197**: Article No. 105841

29 Gao T F, Chen D, Tang Y B, Du B, Ranjan R, Zomaya A Y. Adaptive density peaks clustering: Towards exploratory EEG analysis. *Knowledge-Based Systems*, 2022, **240**: Article No. 108123

30 Xu J, Wang G Y, Deng W H. DenPEHC: Density peak based efficient hierarchical clustering. *Information Sciences*, 2016, **373**: 200−218

31 Ren Y Z, Kamath U, Domeniconi C, Zhang G J. Boosted mean shift clustering. In: Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Berlin, German: Springer, 2014. 646−661

32 Lotfi A, Moradi P, Beigy H. Density peaks clustering based on density backbone and fuzzy neighborhood. *Pattern Recognition*, 2020, **107**: Article No. 107449

33 Teng Q, Yong J L. Fast LDP-MST: An efficient density-peak-based clustering method for large-size datasets. *IEEE Transactions on Knowledge and Data Engineering*, DOI: 10.1109/TKDE.2022.3150403

34 Brooks J K. Decomposition theorems for vector measures. *Proceedings of the American Mathematical Society*, 1969, **21**(1): 27−29

张瑞霖 哈尔滨工业大学 (深圳) 计算机科学与技术学院博士研究生. 主要研究方向为深度学习, 计算机视觉和数据挖掘. E-mail: zzurlz@163.com
(**ZHANG Rui-Lin** Ph.D. candidate at the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. His research interest covers deep learning, computer vision, and data mining.)

郑海阳 哈尔滨工业大学 (深圳) 计算机科学与技术学院硕士研究生. 主要研究方向为深度学习.
E-mail: 21S151085@stu.hit.edu.cn
(**ZHENG Hai-Yang** Master student at the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. His main research interest is deep learning.)

苗振国 哈尔滨工业大学 (深圳) 计算机科学与技术学院硕士研究生. 主要研究方向为深度学习.
E-mail: 20S051017@stu.hit.edu.cn
(**MIAO Zhen-Guo** Master student at the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. His main research interest is deep learning.)

王鸿鹏 哈尔滨工业大学 (深圳) 计算机科学与技术学院教授. 主要研究方向为计算机视觉, 智能机器人和人工智能. 本文通信作者.
E-mail: wanghp@hit.edu.cn
(**WANG Hong-Peng** Professor at the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. His research interest covers computer vision, intelligent robot, and artificial intelligence. Corresponding author of this paper.)