智能系统学报
CAAI TRANSACTIONS ON INTELLIGENT SYSTEMS

面向混合数据的对称邻域和微簇合并密度峰值聚类算法

陈威, 吕莉, 肖人彬, 谭德坤, 潘正祥

引用本文:
陈威, 吕莉, 肖人彬, 等. 面向混合数据的对称邻域和微簇合并密度峰值聚类算法[J]. 智能系统学报, 2025, 20(1): 172-184.
CHEN Wei, LYU Li, XIAO Renbin, et al. Density peak clustering algorithm based on symmetric neighborhood and micro-cluster merging for mixed datasets[J]. *CAAI Transactions on Intelligent Systems*, 2025, 20(1): 172-184.

在线阅读 View online: https://dx.doi.org/10.11992/tis.202311005

DOI: 10.11992/tis.202311005
网络出版地址: https://link.cnki.net/urlid/23.1538.tp.20240919.1445.002

面向混合数据的对称邻域和微簇合并密度峰值聚类算法

陈威[1,2], 吕莉[1,2], 肖人彬[3], 谭德坤[1,2], 潘正祥[4]

（1. 南昌工程学院 信息工程学院, 江西 南昌 330099; 2. 南昌工程学院 南昌市智慧城市物联感知与协同计算重点实验室, 江西 南昌 330099; 3. 华中科技大学 人工智能与自动化学院, 湖北 武汉 430074; 4. 山东科技大学 计算机科学与工程学院, 山东 青岛 266590）

摘　要: 混合数据是指包含密度分布不均和流形特征的数据集。密度峰值聚类算法局部密度定义方式易忽略密度分布不均数据集类簇间样本的疏密差异, 导致误选聚类中心; 分配策略依据欧氏距离进行样本分配, 不适用于流形数据集同一类簇样本相距较远的情况, 致使样本被错误分配。针对这些问题, 本文提出一种面向混合数据的对称邻域和微簇合并密度峰值聚类算法。该算法引入对称邻域概念, 采用对数倒数累加方法重新定义局部密度, 有效提升了聚类中心的识别度; 同时, 提出了一种基于密度差的微簇个数选取方法, 使微簇个数的选取处于合理范围; 此外, 设计了一种微簇间相似性度量方法进行微簇合并, 避免了分配时产生的连带错误。实验表明, 相较于对比算法, 本文算法在混合数据集、UCI 数据集和图像数据集上均取得较好的聚类效果。

关键词: 密度峰值聚类; 密度分布不均; 流形数据; K 近邻; 逆近邻; 对称邻域; 微簇间相似性; 微簇合并
中图分类号: TP301　　**文献标志码:** A　　**文章编号:** 1673−4785(2025)01−0172−13

中文引用格式: 陈威，吕莉，肖人彬，等．面向混合数据的对称邻域和微簇合并密度峰值聚类算法 [J]．智能系统学报，2025, 20(1): 172–184.
英文引用格式: CHEN Wei, LYU Li, XIAO Renbin, et al. Density peak clustering algorithm based on symmetric neighborhood and micro-cluster merging for mixed datasets[J]. CAAI transactions on intelligent systems, 2025, 20(1): 172–184.

Density peak clustering algorithm based on symmetric neighborhood and micro-cluster merging for mixed datasets

CHEN Wei[1,2], LYU Li[1,2], XIAO Renbin[3], TAN Dekun[1,2], PAN Zhengxiang[4]

(1. School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China; 2. Nanchang Key Laboratory of IoT Perception and Collaborative Computing for Smart City, Nanchang Institute of Technology, Nanchang 330099, China; 3. School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China; 4. School of Computer Science and Engineering, Shandong University Of Science And Technology, Qingdao 266590, China)

Abstract: Mixed data refers to datasets containing uneven density distribution and streaming features. The local density definition of density peak clustering algorithm is apt to ignore the sparsity difference of samples between clusters of uneven density distribution dataset, which leads to misselection of clustering centers; the allocation strategy is based on the Euclidean distance for the allocation of the samples, which is not applicable to the streaming dataset with the same type of clusters in the case of the samples far away, resulting in the samples being misallocated. In this paper, we propose a density peak clustering algorithm based on symmetric neighborhood and micro-cluster merging for mixed datasets algorithm (DPC-SNMM). The algorithm introduces the concept of symmetric neighborhood and redefines the local density by using the logarithmic inverse cumulative method, which effectively improves the identification of clustering centers; at the same time, it proposes a method of selecting the number of micro-clusters based on the difference of densities, which puts the selection of micro-clusters in a reasonable range; moreover, it designs an inter-micro-cluster similarity metric to perform the micro-cluster merging, which avoids the cascading errors generated during the allocation. Experiments show that compared with comparison algorithms, the algorithm in this paper achieves better clustering results on mixed datasets, UCI datasets and image datasets.

Keywords: density peaks clustering; uneven density; manifold data; K near neighbour; inverse close neighbor; symmetric neighborhood; similarity between micro-clusters; micro-cluster merging

聚类作为数据挖掘领域的一种重要技术, 由于不依赖于预定义的类别以及数据样本的标签, 因此被称为无监督学习[1]。如今, 聚类在图像处

收稿日期: 2023−11−05.　**网络出版日期:** 2024−09−19.
基金项目: 国家自然科学基金项目 (62066030); 江西省教育厅科技项目 (GJJ190958).
通信作者: 吕莉. E-mail: lvli623@163.com.

理[2]、模式识别[3]、市场分析[4]、网络安全[5] 及医学研究[6] 等领域具有重要的应用价值。

聚类的主要目的是将数据分成若干类簇, 使得同一类簇内的样本具有较高的相似性, 不同类簇间的样本相似性较低。然而, 目前并没有统一的相似性度量标准, 导致聚类算法难以在不同的数据类型上均具有良好的聚类效果。传统的聚类算法主要分为基于划分[7]、基于层次[8]、基于网格[9]、基于模型[10] 和基于密度[11] 的聚类算法。这 5 类算法各具优缺点, 适用于不同类型的数据集。

2014 年, Rodriguez 等[12] 提出通过快速搜索和寻找密度峰值聚类 (clustering by fast search and find of density peaks, DPC) 算法。DPC 算法利用样本的局部密度和相对距离属性快速确定聚类中心, 可以聚类任意形状的类簇。但它存在一些不足: 1)算法对样本密度的依赖较强, 在面对样本分布与密度相关性较弱的数据类型时, 易错过一些类簇的密度峰值; 2)算法沿密度较低的方向进行链式样本分配, 一旦某个样本被错误地分配, 可能导致大量相关样本被错误地分配到其他类簇。

密度分布不均数据指的是类簇间样本密度差异显著, 包含多个密集和稀疏区域的数据集。对于此类数据, DPC 算法局部密度未充分考虑不同类簇间样本的疏密差异, 导致类簇中心集中分布在密集区域。在样本分配上, 偏向于将局部密度较大的样本优先分配, 易使稀疏类簇的样本被错误分配至密集类簇。为此, 陈蔚昌等[13] 通过融合逆近邻和 K 近邻信息重定义局部密度, 从而提升对稀疏区域类簇中心判定的精确度。在样本分配策略上, 引进共享近邻的概念构建样本相似性矩阵, 增强同类样本的内聚性, 降低分配错误率。吕莉等[14] 采用自然近邻概念确定样本局域密度, 调和密集与稀疏区域的密度偏差, 准确识别类簇中心。同时, 通过共享信息和自然近邻强化同类簇样本间的相似性, 有效避免稀疏样本的误分配。吴润秀等[15] 基于 K 近邻计算样本局部密度, 并引入微簇相似性判别标准, 采取多簇合并策略进行样本划分, 从而有效防止多米诺现象, 增强了在密度分布不均数据集上的聚类效能。

流形数据指的是具有复杂非线性结构的数据, 这种结构通常包含线条状和圆环状的类簇。DPC 算法处理该类数据集时, 其局部密度估计不足以准确反映数据的结构特征, 可能导致某些流形类簇出现多个密度峰值, 从而影响类簇中心的准确识别。分配策略优先考虑对类簇中心周围的样本进行链式分配, 若某个样本分配错误, 可能造成距离类簇中心较远的样本被错误地归类到其他流形类簇。为此, 赵嘉等[16] 整合 K 近邻与测地距离优化局部密度定义, 有效突显密度峰值与非密度峰值的差异。同时, 构建基于余弦互逆近邻的样本相似度矩阵, 以确保流形类簇的样本能够准确分配。Tao 等[17] 引入一种具有指数项和比例因子的流形距离来估计样本的局部密度, 从而更准确地反映数据结构的全局和局部一致性, 对流形数据的聚类结果较优。吕莉等[18] 通过最小二阶 K 近邻定义局部密度, 使类簇中心与其他区域间的密度对比鲜明。接着, 采用 K 近邻挑选出具有代表性的局部样本, 据以界定核心点, 进而指导微簇的构建。最后, 结合最小二阶 K 近邻和共享近邻引入的微簇吸引力进行微簇合并, 减少了对类簇中心远端样本的误分配。

混合数据指的是融合密度分布不均和流形结构特征的数据集。在现实生活中, 我们可以在各个领域中观察到混合数据集的存在, 例如社交网络中的用户行为数据、城市的交通流量数据以及金融市场的交易数据等。尽管混合数据集在生活中随处可见, 但目前大部分密度峰值聚类算法的改进研究主要集中在单一的特定数据集, 它们缺乏对混合数据集的普适性研究。

针对上述问题, 本文提出面向混合数据的对称邻域和微簇合并密度峰值聚类算法 (density peak clustering algorithm based on symmetric neighborhood and micro-cluster merging for mixed datasets, DPC-SNMM)。DPC-SNMM 算法的主要创新点如下: 1)局部密度引入对称邻域概念, 采用对数倒数累加方法, 增加类簇中心和非类簇中心的区分度; 2)设计一种基于密度差的微簇个数选取方法, 使微簇个数选取在合理范围; 3)定义一种新的类簇间相似性度量准则, 增强类簇间各微簇的关联度, 并对潜在微簇进行合并, 避免样本分配时的错误连带问题。

1　DPC 算法

DPC 算法通过局部密度和相对距离检测聚类中心, 然后将非中心点分配给与其父类相同的类簇中。该算法基于两个重要假设: 1)类簇中心被较低密度的样本所包围; 2)不同类簇中心间的距离相对较远。假设数据集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$, 样本 $\boldsymbol{x}_i$ 的局部密度定义为

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

或

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

式中：$\chi(x)=\begin{cases}1, & x<0\\ 0, & x\geqslant 0\end{cases}$，$d_{ij}$是样本$\boldsymbol{x}_i$与样本$\boldsymbol{x}_j$之间的欧氏距离，$d_c$是截止距离。

相对距离δ_i可以通过搜索距离它最近且密度比它高的样本$\boldsymbol{x}_j$来确定：

$$\delta_i = \min_{j:\rho_j>\rho_i}(d_{ij}) \tag{1}$$

对于局部密度最大的样本，相对距离δ_i的定义为

$$\delta_{\max} = \max_j(d_{ij}) \tag{2}$$

DPC 算法以局部密度ρ_i为横坐标，相对距离δ_i为纵坐标绘制决策图，选取ρ_i和δ_i均较大的样本作为聚类中心。在实际情况下，为了更准确地识别聚类中心，定义参数γ_i：

$$\gamma_i = \rho_i\cdot\delta_i \tag{3}$$

确定聚类中心后，将剩余样本分配给密度比它高且距离最近的样本所属类簇。

2　DPC-SNMM 算法

在聚类算法中，K 近邻与逆近邻在表示密度方面具有重要意义。K 近邻能够有效地反映样本在空间中的局部分布特点。另一方面，逆近邻从全局角度审视其邻域，数据分布的波动会对聚类结果造成不同程度的影响。因此，本文首先引入对称邻域概念，重新定义局部密度；其次，利用密度差与平均密度变化确定微簇个数；最后，设计了一种新的簇间相似性度量准则，并据此对潜在微簇进行合并，直到微簇个数与实际类簇数相等。

2.1　对称邻域的局部密度

定义 1　**K 近邻**。给定数据集$\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^n$，$n$为样本个数。$\boldsymbol{x}_i^k$代表$\boldsymbol{x}_i$的第$k$个近邻，定义为

$$\mathrm{KNN}(\boldsymbol{x}_i)=\left\{\forall\boldsymbol{x}_j\in\boldsymbol{X}\middle|d(\boldsymbol{x}_i,\boldsymbol{x}_j)\leqslant d\left(\boldsymbol{x}_i,\boldsymbol{x}_i^k\right)\right\} \tag{4}$$

定义 2　**逆近邻**[19]。$\boldsymbol{x}_i$在$\boldsymbol{x}_j$的 K 近邻集中，那么$\boldsymbol{x}_j$是$\boldsymbol{x}_i$的逆近邻，定义为

$$\mathrm{RNN}(\boldsymbol{x}_i)=\left\{\boldsymbol{x}_j\in X\middle|\boldsymbol{x}_i\in\mathrm{KNN}(\boldsymbol{x}_j)\right\} \tag{5}$$

定义 3　**对称邻域**[20]。K 近邻和逆近邻相交的邻域被称为对称邻域，表示为

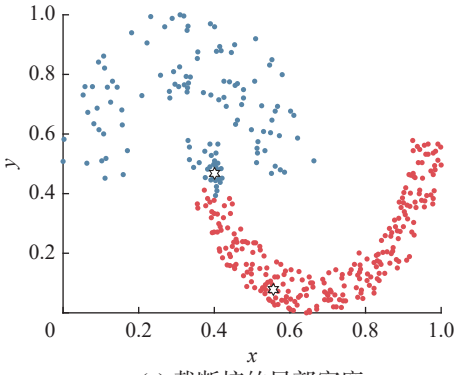
$$\mathrm{SN}(\boldsymbol{x}_i)=\{o|o\in(\mathrm{KNN}(\boldsymbol{x}_i)\cap\mathrm{RNN}(\boldsymbol{x}_i))\} \tag{6}$$

定义 4　**对称邻域的局部密度**。基于对称邻域思想，局部密度定义为

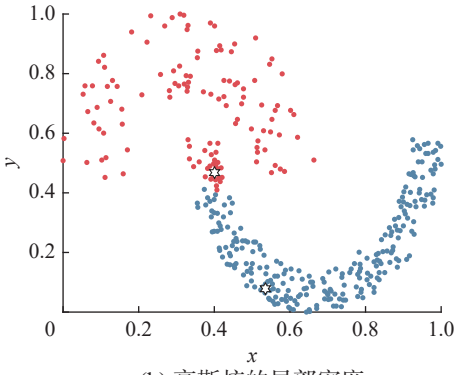
$$\rho_i=\sum_{\boldsymbol{x}_j\in\mathrm{SN}(\boldsymbol{x}_i)}\frac{1}{1+\log|\mathrm{SN}(\boldsymbol{x}_i)|} \tag{7}$$

式中$|\mathrm{SN}(x_i)|$为对称邻域中样本$\boldsymbol{x}_i$的 K 近邻与逆近邻相交的个数，该值越大，表明样本$\boldsymbol{x}_i$与其周围联系越紧密。式 (7) 局部密度设计的优势在于，通过 K 近邻和逆近邻相结合的方式充分考虑样本的整体分布，能够较好地平衡样本的局部一致性和全局一致性。采用对数倒数的方法，可以调整不同分布类簇样本局部密度的大小，提高稀疏类簇样本的局部密度，有助于定位稀疏类簇的聚类中心，从而提高密度差异较大数据集的聚类结果。

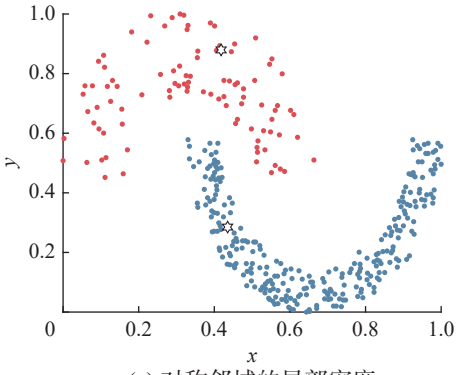
为验证对称邻域局部密度在密度差异较大的数据集中可准确找到类簇中心，在 Jain 数据集上进行实验。图 1 为不同局部密度定义下识别的 Jain 数据集聚类中心，“白色六角星”代表类簇中心。



(a) 截断核的局部密度



(b) 高斯核的局部密度



(c) 对称邻域的局部密度

图 1　不同局部密度定义下识别的 Jain 数据集聚类中心

Fig. 1　Jain dataset clustering center identified under different local density definitions

由图 1(a) 和 (b) 可知, DPC 算法的局部密度未考虑类簇间样本的疏密差异, 致使类簇中心都落在密集区域。图 1(c) 中, 对称邻域的局部密度能够正确找到密度与稀疏类簇的中心。因此, 采用对称邻域的局部密度定义方法能够提高在疏密差异较大的数据集中识别类簇中心的准确性。

2.2　微簇合并策略

微簇合并策略在处理大规模数据集、提升聚类精度、增强鲁棒性和灵活性等方面展现出显著优势。首先, 该策略通过将规模较大的数据集分解成微簇, 再进行合并, 有效提升了大规模数据的处理效率; 其次, 它能够识别并处理噪声和离群点, 这些样本通常形成低密度微簇, 在合并过程中被忽略, 从而增强聚类结果的鲁棒性; 此外, 微簇合并策略能准确地揭示数据集的微观结构, 提升聚类精度; 最后, 该策略具有高度灵活性, 可以根据实际需求调整微簇的数量和大小, 以适应不同的数据集和应用场景。

定义 5　样本间相似度。通过样本间的距离度量表达相似度, 其定义为

$$\omega(\boldsymbol{x}_i,\boldsymbol{x}_j)=\begin{cases}\mathrm{e}^{-\frac{d^2(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\mu}}, & \boldsymbol{x}_j\in \mathrm{SN}(i)\\ 0, & \text{其他}\end{cases}\tag{8}$$

$$\mu=\frac{\sum_{i=1}^{N}\sum_{\boldsymbol{x}_j\in \mathrm{SN}(\boldsymbol{x}_i)} d^2(\boldsymbol{x}_i,\boldsymbol{x}_j)}{N\cdot|\mathrm{SN}(\boldsymbol{x}_i)|}$$

式中: μ为归一化因子, 当μ较大时, 样本间的联系较松散, 反之则紧密; N为总样本数; $\omega(\boldsymbol{x}_i,\boldsymbol{x}_j)$是样本$\boldsymbol{x}_i$与样本$\boldsymbol{x}_j$的相似度, 两个样本的距离越近, 相似度就越高, 并且样本$\boldsymbol{x}_i$和对称邻域以外样本的相似度为 0, 这确保了样本间相似性仅与其对称邻域关联, 从而降低了不相关数据的影响。

定义 6　样本与微簇的邻近度。通过样本间的相似度, 定义样本到微簇的邻近度$P_{x_i\to C_{x_j}}$。

$$P_{\boldsymbol{x}_i\to C_{\boldsymbol{x}_j}}=\frac{\sum_{v\in C_{\boldsymbol{x}_j}}\omega(\boldsymbol{x}_i,v)+\sum_{v\in C_{\boldsymbol{x}_j}}\omega(v,\boldsymbol{x}_i)}{\left|C_{\boldsymbol{x}_j}\right|}\tag{9}$$

式中: $C_{\boldsymbol{x}_j}$代表属于微簇$\boldsymbol{x}_j$的样本集合, $\left|C_{\boldsymbol{x}_j}\right|$是微簇$\boldsymbol{x}_j$的样本数。分子前半部分为样本到微簇的邻近度, 后半部分为微簇到样本的邻近度。

定义 7　微簇间的邻近度。微簇与微簇间的邻近度为

$$P_{C_{\boldsymbol{x}_i}\to C_{\boldsymbol{x}_j}}=\sum_{\boldsymbol{v}\in C_{\boldsymbol{x}_i}}P_{\boldsymbol{v}\to C_{\boldsymbol{x}_j}}\tag{10}$$

式中$P_{C_{\boldsymbol{x}_i}\to C_{\boldsymbol{x}_j}}$代表两微簇间的邻近度。样本$\boldsymbol{v}\in C_{\boldsymbol{x}_i}$到微簇$C_{\boldsymbol{x}_j}$的邻近度越大, 则微簇$C_{\boldsymbol{x}_i}$与微簇$C_{\boldsymbol{x}_j}$的邻近度越大。

在微簇合并策略的研究中, 微簇个数的选取是一个关键步骤, 它直接影响到聚类结果的质量和稳定性。传统的方法通常需要人为设定微簇个数, 这不仅增加了参数选择的难度, 也可能导致聚类结果受到参数选择的影响。为解决该问题, 本文设计了一种基于密度差的微簇个数选取方法。首先, 通过式 (4)~(7) 计算数据集中样本的局部密度, 将密度进行升序排序; 其次, 计算样本的平均密度与排序后相邻样本的密度差; 最后, 确定微簇个数的计算公式为

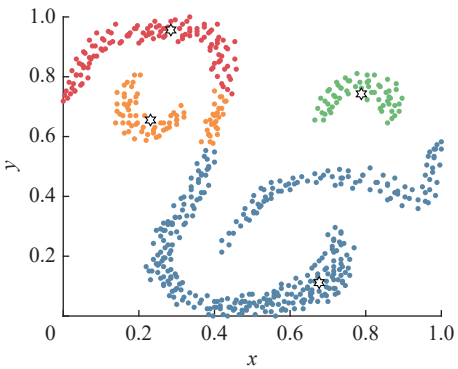
$$M=\sum\sigma(\Delta\rho_i-\bar{\rho})\tag{11}$$

式中: $\sigma(x)=\begin{cases}1, & x<0\\ 0, & x\geqslant 0\end{cases}$, $\Delta\rho_i=\rho_{i+1}-\rho_i$, $\bar{\rho}$代表平均密度。

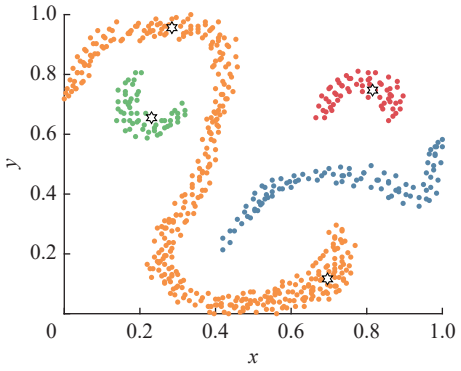
该方法能够自适应数据的实际分布, 从而确定微簇个数, 无需人为设定。其核心思想是通过计算相邻样本的密度差来判断是否形成新的微簇, 只有当密度差超过平均密度时, 才会被视为新的微簇。这种策略有效防止了在密度变化较小的区域内过度划分微簇, 从而避免选取过多的微簇。同时, 由于微簇个数的确定是基于数据的实际密度分布, 因此它能够自适应不同的数据分布, 避免选取过少的微簇。此外, 这种方法在处理离群点方面具有一定优势, 可以有效地将离群点从微簇中剔除, 避免因离群点导致微簇个数过度增加。

微簇合并的过程如下, 首先, 根据式 (10) 生成多个微簇; 其次, 计算每个不含聚类中心的微簇与含有聚类中心的微簇之间的邻近度; 然后, 每次将相似度最高的一个不含聚类中心的微簇与一个含有聚类中心的微簇合并, 直到所有不含聚类中心的微簇分配完毕为止。

为证明本算法分配策略的有效性, 在 Db 数据集上进行实验。图 2(a) 和 (b) 给出了利用截断核局部密度定义方法, 不同分配策略取得的聚类结果。



(a) DPC算法分配策略取得的结果



(b) 微簇合并分配策略取得的结果

图 2　不同分配策略在 Db 数据集上的聚类结果

Fig. 2　Clustering results of different allocation strategies on Db dataset

2.3　算法步骤

输入　数据集 $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^{n}$，近邻数 k

输出　聚类结果 C

1）预处理数据，数据归一化；

2）计算样本间的欧氏距离，并依据式 (7) 计算样本的局部密度 ρ_i，式 (1) 和式 (2) 计算相对距离 δ_i；

3）根据式 (3) 计算样本的决策值 γ_i，选取最终聚类中心的集合 C_n，式 (11) 确定微簇个数，初始生成微簇聚类中心的集合 C_m，依据 DPC 的分配策略将剩余样本分配给密度比它高且距离其最近的样本所在类簇；

4）根据式 (8) 计算样本间相似度；

5）若 $C_m \neq C_n$，根据式 (9) 计算样本与微簇的邻近度，式 (10) 计算微簇间的邻近度，将邻近度最高的一个包含最终聚类中心的微簇与一个不含最终聚类中心的微簇进行合并；

6）若 $C_m = C_n$，输出聚类结果，否则转至步骤 5。

2.4　时间复杂度分析

在样本规模为 n，近邻数为 k，微簇数为 m 的设定下，DPC 算法的时间复杂度为 $O(n^2)$[21]。DPC-SNMM 算法的时间复杂度主要由 5 个部分组成：1）计算样本间欧几里得距离的复杂度 $O(n^2)$；2）计算样本的局部密度 $O(n^2)$；3）计算样本相对距离的复杂度 $O(n^2)$；4）计算微簇个数的复杂度 $O(n)$；5）合并潜在微簇，合并过程需要计算样本间相似性 $O(n^2)$、样本与微簇的邻近度 $O(mn)$ 和微簇间的邻近度 $O(m^2n)$。本文算法受微簇数的直接影响。在样本规模较小的情况下，由于数据特性和微簇设计的相互作用，实际识别的微簇数会远小于样本规模，从而使算法的复杂度近似于 DPC 算法。然而，当样本规模增大时，即使微簇数选择得当，其数量的平方仍有可能超过样本数量，从而导致了算法的时间复杂度相对 DPC 算法有所提升。综上，DPC-SNMM 算法的时间复杂度为 $O(m^2n)$。

3　实验结果与分析

3.1　实验设置

为验证 DPC-SNMM 算法的性能，本文在密度分布不均及流形数据集、UCI 真实数据集和图像数据集进行实验。选取的对比算法是 SDPC（sampling-based density peaks clustering）[22]、DPC-CE（density peak clustering with connectivity estimation）[23]、IDPC-FA（improved density peaks clustering based on firefly algorithm）[24]、DPC-DBFN（density peaks clustering based on density backbone and fuzzy neighborhood）[25] 和 DPC[12] 算法。其中，DPC-SNMM、DPC-DBFN 和 DPC 算法需要进行参数调优，DPC-SNMM、DPC-DBFN 近邻数 k 的选取在 1~100，步长为 1；原始 DPC 算法截断距离 d_c 的取值为所有样本间距离升序排序集合的前 1%~2% 位置所对应的距离值，通过实验发现该范围的 d_c 取值不具有普适性，本文将该范围调整为 0.1%~5%，步长为 0.1%。

本文采用调整互信息（adjusted mutual information, AMI）[26]、Fowlkes-Mallows 指数（Fowlkes-Mallows index, FMI）[26] 和调整兰德系数（adjusted Rand index, ARI）[27] 来评估聚类效果。这一选择的出发点在于全面评估聚类效果的需求，相较于精确度和召回率等传统评价指标，AMI、FMI 和 ARI 指标的优势在于它们可以更好地掌握群体结构，它们度量的是实际群体与预测群体之间的相似度或一致性，而不仅局限于单个点的分类对错。AMI 和 ARI 通过衡量共享信息或正确分类的对数来评估集群的相似性，而 FMI 则使用基于成对的点的方法。当所有数据点被完美地聚类，即预测的群体和真实的群体完全一致时，这些指标的值将接近 1，而精确度和召回率主要关注单个样本的分类正确性，对于聚类这样的问题来说，可能无法提供全面准确的评价。实验环境为 Win10 64 bit 操作系统，AMD Ryzen 75800H with Radeon Graphics 3.20 GHz 处理器，16.0 GB 内存。

3.2　混合数据集的实验结果与分析

表 1 给出了密度分布不均及流形数据集的基本特征。其中，Jain、Compound、Flame 是密度分布不均数据集，Circle、Twomoons、Db 流形数据集，Lineblobs、Cmc、Ring、Sticks 既是密度分布不均也是流形数据集。

表 2 给出了 6 种算法在密度分布不均匀及流

形数据集上的聚类结果, DPC-SNMM 算法在 10 个数据集上均取得最优的聚类结果, 其次是 DPC-CE、IDPC-FA 和 SDPC 算法, DPC-DBFN 和 DPC 算法聚类效果不佳。

表 1　密度分布不均及流形数据集的基本特征

Table 1　Uneven density distribution and the basic characteristics of manifold datasets

数据集	数据来源	样本规模	维度	类簇数
Jain	文献[28]	373	2	2
Circle	文献[16]	1 897	2	3
Twomoons	文献[29]	1 502	2	2
Lineblobs	文献[29]	266	2	3
Cmc	文献[13]	1 002	2	3
Ring	文献[13]	1 200	2	2
Db	文献[16]	630	2	4
Compound	文献[13]	399	2	6
Flame	文献[30]	240	2	2
Sticks	文献[31]	512	2	4

表 2　6 种算法在密度分布不均及流形数据集上的聚类结果

Table 2　Clustering results of the six algorithms on the uneven density distribution and manifold datasets

数据集	算法	AMI	ARI	FMI	Arg-	数据集	算法	AMI	ARI	FMI	Arg-
Jain	DPC-SNMM	**1.0000**	**1.0000**	**1.0000**	14.0	Circle	DPC-SNMM	**1.0000**	**1.0000**	**1.0000**	17.0
	SDPC	**1.0000**	**1.0000**	**1.0000**	0.1		SDPC	0.4266	0.4258	0.6817	0.1
	DPC-CE	**1.0000**	**1.0000**	**1.0000**	—		DPC-CE	0.5290	0.2555	0.6279	—
	IDPC-FA	**1.0000**	**1.0000**	**1.0000**	—		IDPC-FA	0.4629	0.4385	0.7652	—
	DPC-DBFN	0.6816	0.7984	0.9278	43.0		DPC-DBFN	0.2491	0.1512	0.4863	31.0
	DPC	0.6183	0.7146	0.8819	0.8		DPC	0.3596	0.3015	0.6048	0.3
Twomonns	DPC-SNMM	**1.0000**	**1.0000**	**1.0000**	10.0	Lineblobs	DPC-SNMM	**1.0000**	**1.0000**	**1.0000**	10.0
	SDPC	0.6645	0.7596	0.8996	0.1		SDPC	0.8649	0.8635	0.9105	0.1
	DPC-CE	**1.0000**	**1.0000**	**1.0000**	—		DPC-CE	**1.0000**	**1.0000**	**1.0000**	—
	IDPC-FA	0.5171	0.6106	0.8458	—		IDPC-FA	**1.0000**	**1.0000**	**1.0000**	—
	DPC-DBFN	0.4048	0.4843	0.8049	95.0		DPC-DBFN	0.6500	0.6192	0.7486	81.0
	DPC	0.6671	0.7621	0.9005	4.7		DPC	0.8375	0.8375	0.8375	4.2
Cmc	DPC-SNMM	**1.0000**	**1.0000**	**1.0000**	15.0	Ring	DPC-SNMM	**1.0000**	**1.0000**	**1.0000**	5.0
	SDPC	0.6011	0.5744	0.7778	0.1		SDPC	0.2299	0.1595	0.6505	0.1
	DPC-CE	0.6694	0.7362	0.8352	—		DPC-CE	0.0902	0.0286	0.6605	—
	IDPC-FA	0.8093	0.8421	0.9027	—		IDPC-FA	0.1333	0.0886	0.6362	—
	DPC-DBFN	0.3524	0.3321	0.6491	40.0		DPC-DBFN	0.0239	0.0012	0.6897	6.0
	DPC	0.3857	0.2661	0.5377	5.0		DPC	0.2073	0.1815	0.6431	0.06
Db	DPC-SNMM	**1.0000**	**1.0000**	**1.0000**	27.0	Compound	DPC-SNMM	**0.8483**	**0.8577**	**0.9001**	22.0
	SDPC	0.6447	0.4699	0.6812	0.1		SDPC	0.7486	0.6074	0.6988	0.1
	DPC-CE	0.6758	0.5588	0.7395	—		DPC-CE	0.8082	0.6141	0.7060	—
	IDPC-FA	0.6526	0.5033	0.6999	—		IDPC-FA	0.7922	0.8327	0.8815	—
	DPC-DBFN	0.4461	0.3181	0.5879	24.0		DPC-DBFN	0.7870	0.8087	0.8645	5.0
	DPC	0.5185	0.2794	0.5853	4.0		DPC	0.7754	0.5910	0.6876	4.0
Flame	DPC-SNMM	**1.0000**	**1.0000**	**1.0000**	37.0	Sticks	DPC-SNMM	**1.0000**	**1.0000**	**1.0000**	11.0
	SDPC	0.9267	0.9666	0.9845	0.1		SDPC	**1.0000**	**1.0000**	**1.0000**	0.1
	DPC-CE	**1.0000**	**1.0000**	**1.0000**	—		DPC-CE	**1.0000**	**1.0000**	**1.0000**	—
	IDPC-FA	**1.0000**	**1.0000**	**1.0000**	—		IDPC-FA	**1.0000**	**1.0000**	**1.0000**	—
	DPC-DBFN	0.9318	0.9666	0.9848	27.0		DPC-DBFN	0.9177	0.9397	0.9548	81.0
	DPC	**1.0000**	**1.0000**	**1.0000**	2.8		DPC	0.8094	0.7534	0.8235	2.0

注: 表中加粗代表最优结果, “Arg-”为各算法的最优参数取值。“—”表示不含参数。

Friedman 检验[32]是利用秩实现对多个总体分布进行非参数检验，以判断是否存在显著差异的方法。该方法可以更精确地呈现不同算法间评价指标的差异，秩均值越高则算法的聚类效果越优。从表 3 可以发现，DPC-SNMM 算法在密度分布不均及流形数据集中 3 种评价指标的秩均值均是最优的，且秩均值都大于 5.6。

表 3　6 种算法在密度分布不均及流形数据集上的秩均值

Table 3　Rank mean values of the six algorithms on the uneven density distribution and manifold datasets

算法	DPC-SNMM	SDPC	DPC-CE	IDPC-FA	DPC-DBFN	DPC
AMI	5.40	3.10	4.50	4.05	1.50	2.45
ARI	5.40	3.20	4.00	4.25	1.80	2.35
FMI	5.40	3.10	4.30	4.05	2.20	1.95

受篇幅限制，本文选取了 3 个具有代表性的数据集。图 3、图 4 和图 5 分别给出了 6 种算法在 Compound、Db 和 Cmc 数据集上的聚类结果。图中各颜色表示不同的类簇，聚类中心用“六角星”标识。

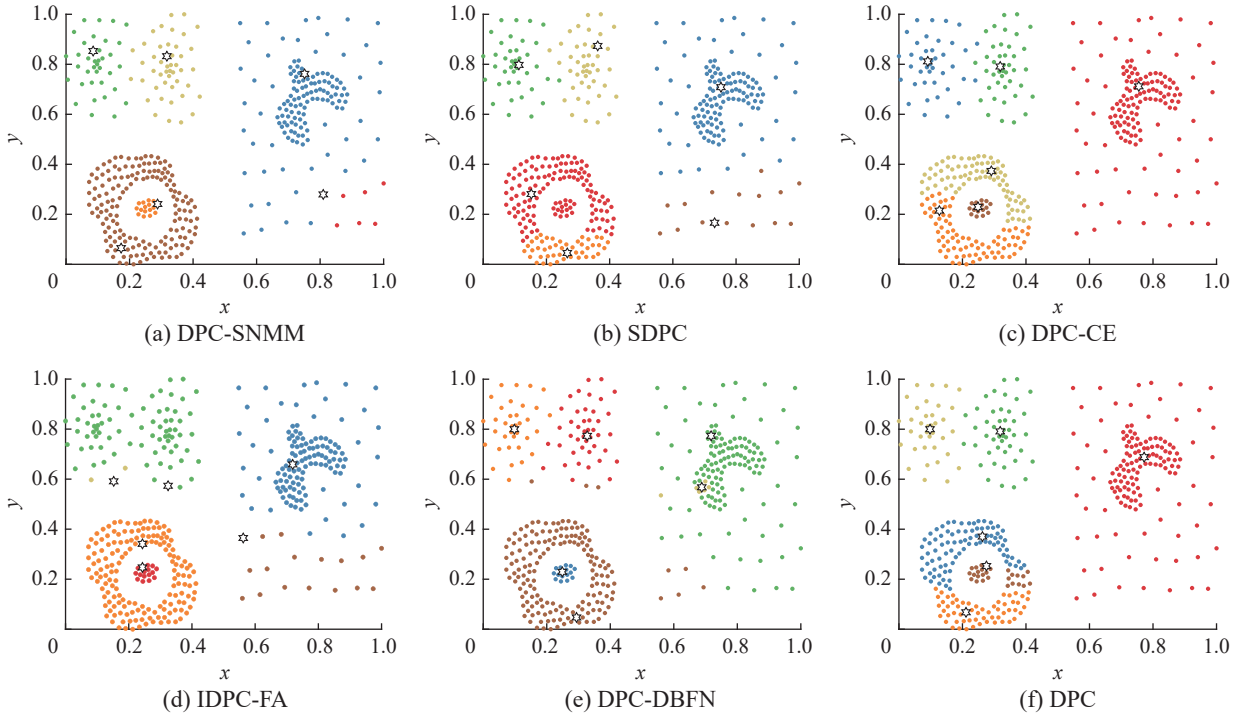
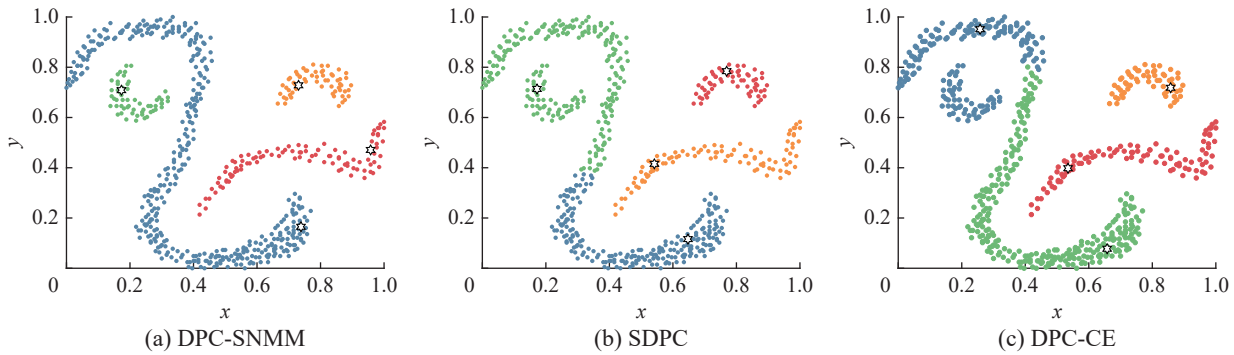


图 3　6 种算法在 Compound 数据集上的聚类结果

Fig. 3　Clustering results of six algorithms on Compound dataset

Compound 数据集由圆状和不规则状类簇构成，是典型的密度分布不均数据集。从图 3 可知，DPC-SNMM 算法的局部密度计算公式充分考虑了密度分布不均数据的分布特征，可以找到正确的聚类中心，DPC-DBFN、DPC-CE、SDPC 和 DPC 算法由于在较密集簇中选择多个聚类中心，从而导致后续的分配错误。IDPC-FA 算法在部分区域误选了聚类中心，导致分配出错。



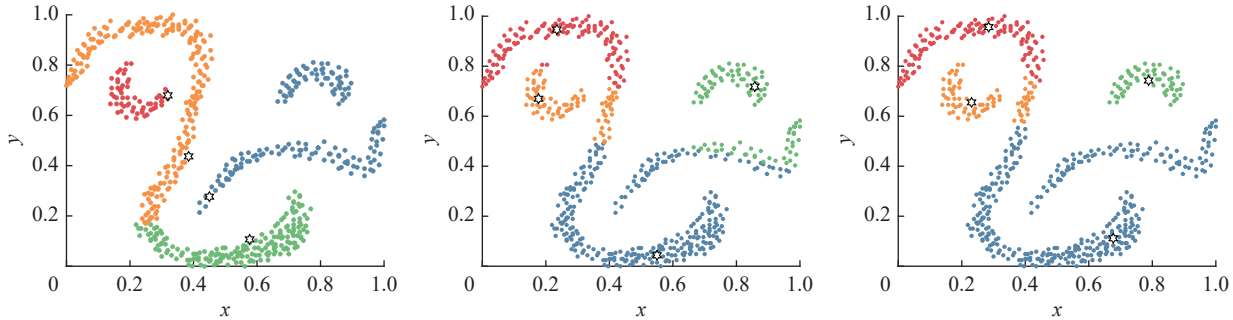


图 4　6 种算法在 Db 数据集上的聚类结果

Fig. 4　Clustering results of six algorithms on Db dataset

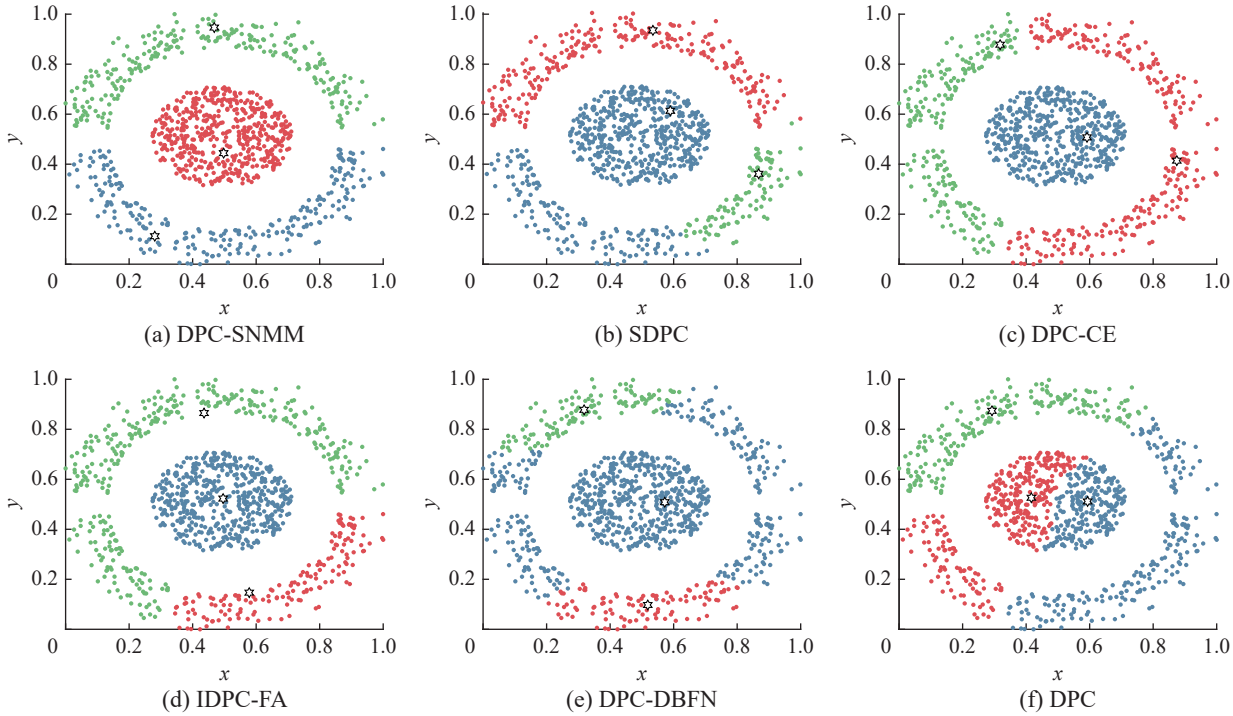


图 5　6 种算法在 Cmc 数据集上的聚类结果

Fig. 5　Clustering results of six algorithms on Cmc dataset

Db 数据集包含 4 个弧状类簇, 具备典型的流形数据集特征。图 4 中, DPC-SNMM 算法充分考虑到流形数据的分布特性, 局部密度定义正确识别了聚类中心, 微簇合并分配策略避免了多米诺效应, 取得了不错的聚类效果。SDPC 算法虽然找到正确的类簇中心, 但其样本分配策略存在不足。其余对比算法, 由于无法找到正确聚类中心, 导致聚类结果不佳。

Cmc 数据集由内侧 1 个较密集的圆状类簇和外侧 2 个稀疏的半圆弧类簇构成, 它具备密度分布不均和流形特征。从图 5 可知, DPC-SNMM 算法不仅可以准确找到类簇中心, 而且能正确地进行样本分配; IDPC-FA、DPC-DBFN、DPC-CE 和 SDPC 算法能找到正确的类簇中心, 但样本分配策略存在错误; DPC 算法无法正确找到类簇中心。

3.3　UCI 数据集的实验结果与分析

为了进一步证实 DPC-SNMM 算法的有效性, 本文选取了 8 个真实数据集, 对 6 种算法进行实验。数据集的基本特征如表 4 所示。

表 4　UCI 数据集的基本特征

Table 4　Basic characteristics of UCI dataset

数据集	数据来源	样本规模	维度	类簇数
Iris	文献[33]	150	4	3
Wine	文献[33]	178	13	3
Seeds	文献[34]	210	7	3
Ecoli	文献[33]	336	8	8
Inonsphere	文献[35]	351	34	2
Libras	文献[36]	360	90	15
Dermatology	文献[33]	366	33	6
Glass	文献[20]	214	9	6

表 5 给出了 6 种算法在 UCI 数据集上的聚类结果。从表 5 可以发现，Seeds 数据集中，DPC-SNMM 算法的聚类效果不及 IDPC-FA、DPC-DBFN、SDPC 和 DPC 算法。Ecoli 数据集中，DPC-SNMM 算法的聚类效果略低于 IDPC-FA 算法。Iris、Wine、Inonsphere、Libras、Dermatology 和 Glass 数据集中，DPC-SNMM 算法均取得较好的聚类效果。

表 5　6 种算法在 UCI 数据集上的聚类结果

Table 5　Clustering results of six algorithms on UCI dataset

数据集	算法	AMI	ARI	FMI	Arg-	数据集	算法	AMI	ARI	FMI	Arg-
Iris	DPC-SNMM	**0.8831**	**0.9038**	**0.9355**	9.0	Wine	DPC-SNMM	**0.8928**	**0.9121**	**0.9417**	52.0
	SDPC	**0.8831**	**0.9038**	**0.9355**	0.1		SDPC	0.8463	0.8806	0.9208	0.1
	DPC-CE	0.7277	0.6634	0.7824	—		DPC-CE	0.5841	0.5362	0.6945	—
	IDPC-FA	**0.8831**	**0.9038**	**0.9355**	—		IDPC-FA	0.7675	0.7713	0.8283	—
	DPC-DBFN	0.8337	0.8510	0.9001	21.0		DPC-DBFN	0.8192	0.8465	0.8988	46.0
	DPC	0.8606	0.8857	0.9233	0.2		DPC	0.7065	0.6724	0.7835	2.0
Seeds	DPC-SNMM	0.7156	0.7395	0.8257	63.0	Ecoli	DPC-SNMM	0.6165	0.7029	0.7953	10.0
	SDPC	0.7085	0.7485	0.8316	0.1		SDPC	0.6262	0.6999	0.7796	0.1
	DPC-CE	0.7144	0.7448	0.8297	—		DPC-CE	0.5065	0.4494	0.5829	—
	IDPC-FA	0.7299	**0.7670**	**0.8444**	—		IDPC-FA	**0.6638**	**0.7561**	**0.8284**	—
	DPC-DBFN	**0.7303**	0.7664	0.8439	2.0		DPC-DBFN	0.5903	0.6581	0.7553	3.0
	DPC	0.7299	**0.7670**	**0.8444**	0.7		DPC	0.4978	0.4465	0.5775	0.4
Inonsphere	DPC-SNMM	**0.2840**	**0.3869**	**0.7543**	27.0	Libras	DPC-SNMM	**0.5827**	**0.3876**	**0.4541**	15.0
	SDPC	0.1108	0.1584	0.5970	0.1		SDPC	0.5224	0.3148	0.3775	0.1
	DPC-CE	0.0704	0.1145	0.5802	—		DPC-CE	0.3516	0.1392	0.2709	—
	IDPC-FA	0.1355	0.2183	0.6432	—		IDPC-FA	0.5733	0.3816	0.4247	—
	DPC-DBFN	0.2352	0.3177	0.7409	6.0		DPC-DBFN	0.3048	0.1081	0.2405	6.0
	DPC	0.1504	0.2357	0.6491	0.5		DPC	0.5358	0.3193	0.3717	0.3
Dermatology	DPC-SNMM	0.8653	0.8563	0.8920	12.0	Glass	DPC-SNMM	**0.6126**	**0.6469**	**0.7378**	10.0
	SDPC	**0.8748**	0.7663	0.8147	0.1		SDPC	0.5660	0.5447	0.6570	0.1
	DPC-CE	0.8227	0.6768	0.7417	—		DPC-CE	0.5293	0.5461	0.6564	—
	IDPC-FA	0.8638	**0.8772**	**0.9018**	—		IDPC-FA	0.4511	0.4049	0.5552	—
	DPC-DBFN	0.5426	0.5230	0.6164	88.0		DPC-DBFN	0.4199	0.4230	0.5646	22.0
	DPC	0.7840	0.7760	0.8221	1.5		DPC	0.5565	0.5335	0.6559	0.9

注：表中加粗代表最优结果，“Arg-”为各算法的最优参数取值。“—”表示不含参数。

表 6 为 6 种算法在 UCI 数据集上评价指标的秩均值。从表 6 可知，DPC-SNMM 算法 3 种指标的秩均值均优于对比算法，其次为 IDPC-FA 算法，DPC-CE 算法的表现最差。

表 6　6 种算法在 UCI 数据集上的秩均值

Table 6　Rank mean of six algorithms on UCI dataset

算法	DPC-SNMM	SDPC	DPC-CE	IDPC-FA	DPC-DBFN	DPC
AMI	5.13	4.00	1.88	4.06	2.88	3.06
ARI	5.00	3.50	2.63	4.06	2.50	3.31
FMI	5.13	3.75	2.00	4.19	2.63	3.31

3.4　图像数据集的实验结果与分析

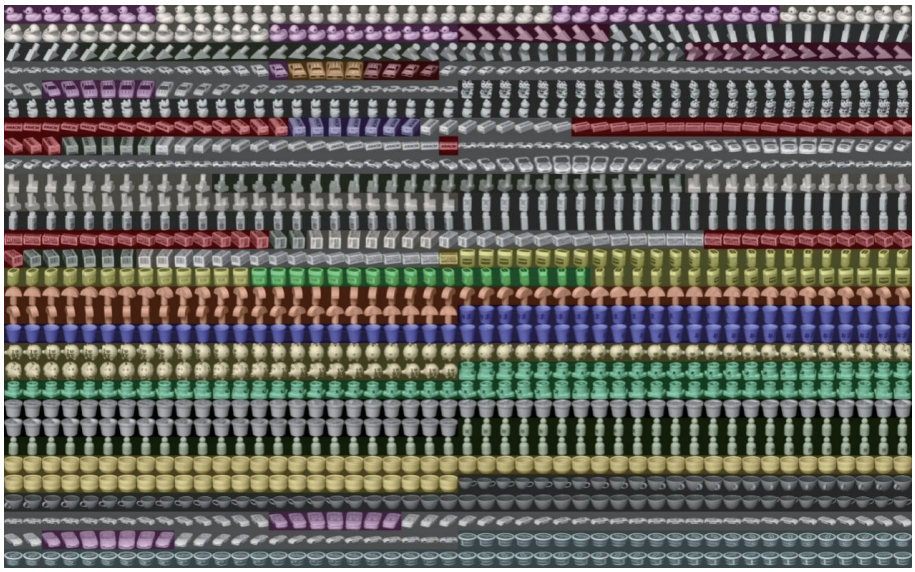
COIL-20 数据集[37]来自于哥伦比亚大学图像库，它含有 20 个物体从各个角度拍摄的照片，每隔 5°拍摄一幅图像，每个物体 72 张图像，共计 1 440 张物体图像。经过浙江大学蔡登教授的处理，每张图像被降采样为 32×32，即 1 024 维。为验证 DPC-SNMM 算法在混合数据特征图像数据集上的聚类表现，本文将 6 种算法应用于 COIL-20 数据集。从表 7 可以发现，DPC-SNMM 算法的聚类效果优于其他算法。

表 7　6 种算法在 COIL-20 数据集上的聚类结果

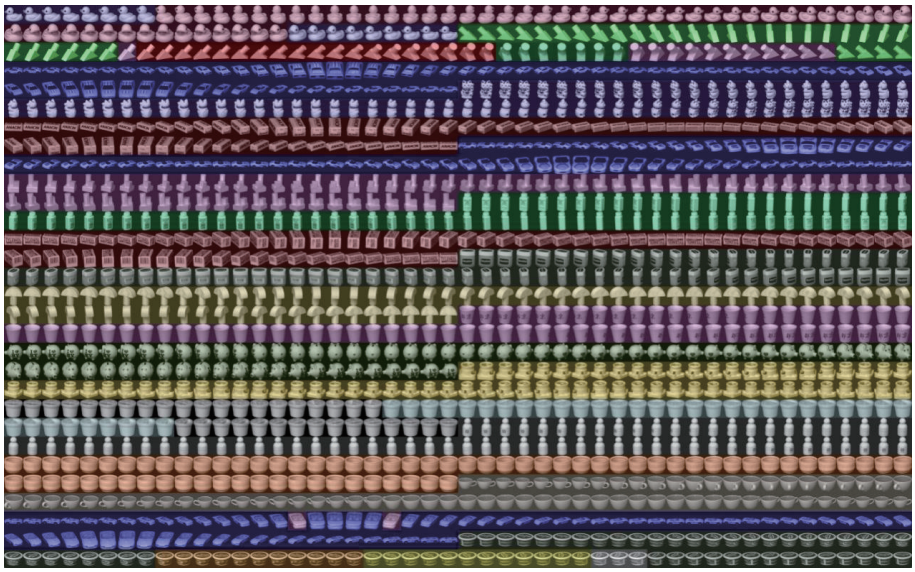
Table 7　Clustering results of six algorithms on COIL-20 dataset

算法	AMI	ARI	FMI	参数	算法	AMI	ARI	FMI	参数
DPC-SNMM	0.837 6	0.675 9	0.712 2	18.0	IDPC-FA	0.792 9	0.601 2	0.640 6	—
SDPC	0.772 6	0.591 9	0.620 4	0.1	DPC-DBFN	0.506 3	0.289 9	0.374 0	42.0
DPC-CE	0.757 7	0.568 9	0.602 4	—	DPC	0.794 4	0.601 6	0.641 3	3.2

除了 DPC-DBFN 算法外, 其他算法在 COIL-20 数据集上的聚类结果相近, 故本文选取 DPC-SNMM 和 DPC 算法进行分析。如图 6 所示, 从左到右, 从上至下, 每 72 张图像为 1 个物体, 总共 20 个物体, 分别标识为 1~20 号。由图 6(a) 可知, 第 1、2、3、5、7、9、10 和 19 号图像中, DPC 算法将同个物体至少划分成 2 个类簇, 表明在处理复杂数据时, 找到聚类中心具有一定困难, 即便算法成功地选取了正确的聚类中心, 仍然会出现样本分配错误的情况。由图 6(b) 可知, 相较于 DPC 算法, DPC-SNMM 算法能够准确识别 3、5、7、9 和 10 号物体, 提高了物体识别的准确度。综合表 7 和图 6 可知, DPC-SNMM 算法在面对复杂数据时能识别其分布特征, 得到理想的聚类效果。



(a) DPC算法



(b) DPC-SNMM算法

图 6　两种算法在 COIL-20 数据集上的聚类结果

Fig. 6　Clustering results of the two algorithms on COIL-20 dataset

3.5　算法的仿真时间与分析

仿真时间是指开展仿真实验所耗用的时间长度，它是衡量研究成果和验证实验效果的关键指标之一。为了评估实验运行的仿真时间，本文针对上述 3 类数据集进行了相应计算，具体结果如表 8 所示。

表 8　6 种算法在 3 类数据集上的仿真时间对比
Table 8　Simulation time of six algorithms on three types datasets　s

数据集	DPC-SNMM	SDPC	DPC-CE	IDPC-FA	DPC-DBFN	DPC
Jain	0.11	0.52	1.06	11.23	0.06	0.06
Circle	10.81	1.51	33.85	986.83	1.14	0.56
Twomoons	8.49	1.45	17.16	341.54	0.30	0.23
Lineblobs	0.15	0.47	0.73	6.47	0.10	0.06
Cmc	1.56	0.96	6.12	87.32	0.27	0.14
Ring	1.42	1.10	9.86	134.56	0.26	0.13
Db	0.83	0.70	2.41	28.31	0.14	0.27
Compound	0.44	0.55	1.22	14.43	0.11	0.15
Flame	0.13	0.58	0.62	5.53	0.91	0.05
Sticks	0.45	0.62	1.51	18.42	0.16	0.08
Iris	0.24	0.41	0.48	3.83	0.11	0.06
Wine	0.36	0.45	0.43	4.42	0.11	0.07
Seeds	0.46	0.44	0.55	5.23	0.12	0.07
Ecoli	0.65	0.72	1.25	14.23	0.17	0.10
Inonsphere	0.72	0.71	1.01	13.37	0.16	0.07
Libras	1.12	1.07	1.04	43.00	0.13	0.18
Dermatology	0.83	0.74	1.17	17.07	0.18	0.09
Glass	0.53	0.59	0.56	8.45	0.11	0.55
COIL-20	12.65	2.59	20.30	1 657.18	1.49	1.67
平均运行时间	2.21	0.85	5.33	179.02	0.32	0.24

从表 8 可以发现，DPC-SNMM 算法总体的聚类效率要高于 DPC-CE 和 IDPC-FA 算法，逊于 DPC-DBFN 和 DPC 算法。当样本规模不大时，聚类效率与 SDPC 算法相近。如图 7 所示，选择具有相同维度的密度分布不均及流形数据集，数据集按样本数量升序进行排列，比较各算法随样本数量变化的运行时间。IDPC-FA 算法由于其运行时间显著大于其他算法，故不进行比较。可以发现，DPC-SNMM 算法在数据集样本量规模较少时，时间增长趋势不明显。仅在数据集样本量达到一定规模后，有一定的增长趋势，且该趋势明显小于 DPC-CE 算法。这正是因为 DPC-SNMM 算法通过微簇合并策略对样本进行分配时，会受到微簇数的直接影响，遇到样本规模较大的数据集导致运行时间增长。尽管如此，与其他 5 种算法相较，本算法在总体聚类效果上表现更佳，且聚类效率处于合理区间。

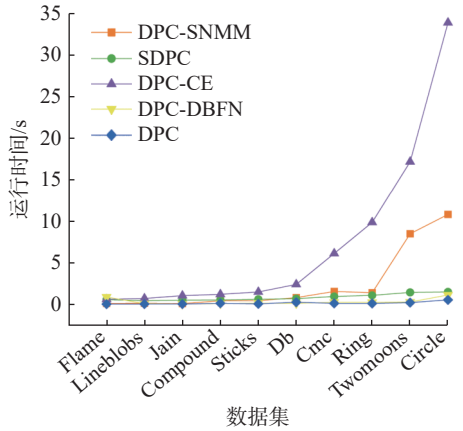


图 7　5 种算法在密度分布不均及流形数据集上的运行时间对比
Fig. 7　Runtime comparison of five algorithms on the uneven density distribution and manifold datasets

4　结束语

为解决 DPC 算法在处理密度分布不均数据

集时易错误选择聚类中心以及流形数据集时易导致样本错误分配的问题, 本文提出面向混合数据的对称邻域和微簇合并的密度峰值聚类算法。DPC-SNMM 算法引入对称邻域概念, 重新定义局部密度, 可以准确找到密度分布不均数据的聚类中心; 使用微簇合并分配策略, 充分考虑样本与微簇, 微簇与微簇间的关联性, 避免了样本分配过程中产生的链式效应, 优化了对流形数据集的聚类效果。实验结果表明, 相较于对比算法, 本文算法对混合数据集、UCI 数据集和图像数据集均可获得满意的效果。如何将近邻数k实现自适应以及采用不同的距离度量方式或降维方法提升算法在高维数据集的聚类表现是下一步的研究重心。

参考文献:

[1] ZHANG Wei, DU Lan, LI Liling, et al. Infinite Bayesian one-class support vector machine based on Dirichlet process mixture clustering[J]. Pattern recognition, 2018, 78: 56−78.

[2] ZHANG Jie, YAO Pengpeng, WU H, et al. Automatic color pattern recognition of multispectral printed fabric images[J]. Journal of intelligent manufacturing, 2023, 34(6): 2747−2763.

[3] JAIN A K, DUIN R P W, MAO Jianchang. Statistical pattern recognition: a review[J]. IEEE transactions on pattern analysis and machine intelligence, 2000, 22(1): 4−37.

[4] NIE Chunxiao, SONG Futie. Analyzing the stock market based on the structure of kNN network[J]. Chaos, solitons & fractals, 2018, 113: 148−159.

[5] LIU Xiuwen, FU Jianming, CHEN Yanjiao. Event evolution model for cybersecurity event mining in tweet streams [J]. Information sciences, 2020, 524: 254−276.

[6] HE Song, HE Haochen, XU Wenjian, et al. ICM: a web server for integrated clustering of multi-dimensional biomedical data[J]. Nucleic acids research, 2016, 44(W1): W154−W159.

[7] GAO Chenhui, CHEN Wenzhi, NIE Feiping, et al. Subspace clustering by directly solving discriminative K-means[J]. Knowledge-based systems, 2022, 252: 109452.

[8] HORNG S C, YANG Fengyi, LIN S S. Hierarchical fuzzy clustering decision tree for classifying recipes of ion implanter[J]. Expert systems with applications, 2011, 38(1): 933−940.

[9] LIN Chuan, HAN Guangjie, WANG Tingting, et al. Fast node clustering based on an improved birch algorithm for data collection towards software-defined underwater acoustic sensor networks[J]. IEEE sensors journal, 2021, 21(22): 25480−25488.

[10] CHOI J W, PARK G M, KIM J H. SR-EM: episodic memory aware of semantic relations based on hierarchical clustering resonance network[J]. IEEE transactions on cybernetics, 2022, 52(10): 10339−10351.

[11] WANG Yue fei, JIONG Yu, SU Guo ping, et al. A new outlier detection method based on OPTICS[J]. Sustainable cities and society, 2019, 45: 197−212.

[12] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492−1496.

[13] 陈蔚昌, 赵嘉, 肖人彬, 等. 面向密度分布不均数据的近邻优化密度峰值聚类算法[J]. 控制与决策, 2024, 39(3): 919−928.
CHEN Weichang, ZHAO Jia, XIAO Renbin, et al. Density peaks clustering algorithm with nearest neighbor optimization for data with uneven density distribution[J]. Control and decision, 2024, 39(3): 919−928.

[14] 吕莉, 朱梅子, 康平, 等. 面向分布不均数据的混合近邻密度峰值聚类算法[J/OL]. 控制理论与应用. (2023−11−16)[2024−01−02]. http://kns.cnki.net/kcms/detail/44.1240.TP.20231114.1333.014.html.
LYU Li, ZHU Meizi, KANG Ping, et al. Multiplex neighbor density peaks clustering for uneven density data sets [J/OL]. Control theory & applications. (2023−11−16) [2024−01−02]. http://kns.cnki.net/kcms/detail/44.1240.TP.20231114.1333.014.html.

[15] 吴润秀, 尹士豪, 赵嘉, 等. 基于相对密度估计和多簇合并的密度峰值聚类算法[J]. 控制与决策, 2023, 38(4): 1047−1055.
WU Runxiu, YIN Shihao, ZHAO Jia, et al. Density peaks clustering based on relative density estimating and multi cluster merging[J]. Control and decision, 2023, 38(4): 1047−1055.

[16] 赵嘉, 王刚, 吕莉, 等. 面向流形数据的测地距离与余弦互逆近邻密度峰值聚类算法[J]. 电子学报, 2022, 50(11): 2730−2737.
ZHAO Jia, WANG Gang, LYU Li, et al. Density peaks clustering algorithm based on geodesic distance and cosine mutual reverse nearest neighbors for manifold datasets[J]. Acta electronica sinica, 2022, 50(11): 2730−2737.

[17] TAO Xinmin, GUO Wenjie, REN Chao, et al. Density peak clustering using global and local consistency adjustable manifold distance[J]. Information sciences, 2021, 577: 769−804.

[18] 吕莉, 朱梅子, 康平, 等. 二阶 K 近邻和多簇合并的密度峰值聚类算法[J]. 吉林大学学报 (工学版), 2024, 54(5): 1417−1425.
LYU Li, ZHU Meizi, KANA Ping, et al. Density peaks clustering with second-order K-nearest neighbors and multi-cluster merging[J]. Journal of Jilin University (engineering and technology edition), 2024, 54(5): 1417−1425.

[19] BRYANT A, CIOS K. RNN-DBSCAN: a density-based clustering algorithm using reverse nearest neighbor dens-

ity estimates[J]. IEEE transactions on knowledge and data engineering, 2018, 30(6): 1109−1121.

[20] WU Chunrong, LEE Jia, ISOKAWA T, et al. Efficient clustering method based on density peaks with symmetric neighborhood relationship[J]. IEEE access, 2019, 7: 60684−60696.

[21] 徐晓, 丁世飞, 孙统风, 等. 基于网格筛选的大规模密度峰值聚类算法[J]. 计算机研究与发展, 2018, 55(11): 2419−2429.
XU Xiao, DING Shifei, SUN Tongfeng, et al. Large-scale density peaks clustering algorithm based on grid screening[J]. Journal of computer research and development, 2018, 55(11): 2419−2429.

[22] DING Shifei, LI Chao, XU Xiao, et al. A sampling-based density peaks clustering algorithm for large-scale data[J]. Pattern recognition, 2023, 136: 109238.

[23] GUO Wenjie, WANG Wenhai, ZHAO Shunping, et al. Density peak clustering with connectivity estimation[J]. Knowledge-based systems, 2022, 243: 108501.

[24] SHI Aiye, ZHAO Jia, TANG Jingjing, et al. Improved density peaks clustering based on firefly algorithm[J]. International journal of bio-inspired computation, 2020, 15(1): 24.

[25] LOTFI A, MORADI P, BEIGY H. Density peaks clustering based on density backbone and fuzzy neighborhood [J]. Pattern recognition, 2020, 107: 107449.

[26] VINH N, EPPS J, BAILEY J. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance[J]. Journal of machine learning research, 2010, 11(1): 2837−2854.

[27] FOWLKES E B, MALLOWS C L. A method for comparing two hierarchical clusterings[J]. Journal of the American statistical association, 1983, 78(383): 553−569.

[28] JAIN A K, LAW M. Data clustering: a user's dilemma [C]//Proceedings of the First International conference on pattern recognition and machine intelligence. Heidelberg: Springer, 2005: 1−10.

[29] XU Xiao, DING Shifei, WANG Lijuan, et al. A robust density peaks clustering algorithm with density-sensitive similarity[J]. Knowledge-based systems, 2020, 200: 106028.

[30] 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[J]. 软件学报, 2023, 34(12): 5629−5648.
ZHANG Qinghua, ZHOU Jingpeng, DAI Yongyang, et al. Density peaks clustering algorithm based on representative points and K-nearest neighbors[J]. Journal of software, 2023, 34(12): 5629−5648.

[31] 赵嘉, 陈磊, 吴润秀, 等. K 近邻和加权相似性的密度峰值聚类算法[J]. 控制理论与应用, 2022, 39(12): 2349−2357.
ZHAO Jia, CHEN Lei, WU Runxiu, et al. Density peaks clustering algorithm with K-nearest neighbors and weighted similarity[J]. Control theory & applications, 2022, 39(12): 2349−2357.

[32] ZIMMERMAN D W, ZUMBO B D. Relative power of the wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks[J]. The journal of experimental education, 1993, 62(1): 75−86.

[33] BLAKE C L, MERZ C J. UCI repository of machine learning database[EB/OL]. (2016−12−28)[2023−11−05]. http://archive.ics.uci.edu/ml/index.html.

[34] CHARYTANOWICZ M, NIEWCZAS J, KULCZYCKI P, et al. Complete gradient clustering algorithm for features analysis of X-ray images[M]//Advances in Intelligent and Soft Computing. Berlin: Springer Berlin Heidelberg, 2010: 15−24.

[35] SIGILLITO V G, WING S P, HUTTON L V, et al. Classification of radar returns from the ionosphere using neural networks[J]. Johns Hopkins APL technical digest (applied physics laboratory), 1989, 10(3): 262−266.

[36] DIAS D B, MADEO R C B, ROCHA T, et al. Hand movement recognition for Brazilian sign language: a study using distance-based neural networks[C]//2009 International Joint Conference on Neural Networks. Atlanta: IEEE, 2009: 697−704.

[37] NENE S, NAYAR S, MURASE H. Columbia object image library (COIL-20): CUCS 006 96[R]. Columbia: Department of Computer Science, Columbia University, 1996.

作者简介:

陈威, 硕士研究生, 主要研究方向为大数据挖掘。E-mail: chenwei9801@163.com。

吕莉, 教授, 博士, 主要研究方向为智能计算与计算智能、目标跟踪与检测、大数据与人工智。主持国家自然科学基金项目 2 项, 发表学术论文 80 余篇。E-mail: lvli623@163.com。

肖人彬, 教授, 博士, 主要研究方向为复杂系统建模与分析、群集智能。主持国家自然科学基金 11 项, 获教育部自然科学奖 1 项和湖北省自然科学奖及科技进步奖 4 项。发表学术论文 300 余篇, 出版学术专著和教材 10 余部。E-mail: rbxiao@hust.edu.cn。