文章编号:1005-0523(2023)01-0107-11

密度峰值聚类算法研究综述

王 森,邢帅杰,刘 琛

(华东交通大学理学院,江西 南昌 330013)

摘要:密度峰值聚类(DPC)是一种新提出的基于密度和距离的聚类算法,由于其原理简单,无需迭代和能处理形状数据集等优点,正在数据挖掘领域得到广泛应用。但 DPC 算法也有着一定的缺陷,如:对截断距离参数敏感,初始聚类中心的选择非自动化,后续标签分配存在链式问题,时间复杂度较高等。文章对 DPC 算法的研究现状进行了总结与整理,首先介绍了 DPC 的算法原理和流程;其次,针对 DPC 算法的不足对 DPC 算法的优化进行概括和分析,指出了优化算法的核心技术以及优缺点;最后,对 DPC 算法未来可能面对的挑战和发展趋势进行展望。

关键词:聚类算法;密度峰值;截断距离;初始聚类中心;微簇合并;时间复杂度

中图分类号:TP391 **文献标志码**:A

本文引用格式:王森,邢帅杰,刘琛. 密度峰值聚类算法研究综述[J]. 华东交通大学学报,2023,40(1):106-116.

Survey of Density Peak Clustering Algorithm

Wang Sen, Xing Shuaijie, Liu Chen

(School of Science, East China Jiaotong University, Nanchang 330013, China)

Abstract: Density peak clustering(DPC) is a novel clustering algorithm based on density and distance. It is widely used in the field of data mining because of its simple principle, no iteration and the ability to process shape datasets. However, DPC algorithm also has some defects, including the sensitive cutoff distance parameter, non-automatic selection of initial clustering center, the chain problem in subsequent allocation and high time complexity. This paper summarizes and arranges the research status of DPC algorithm. Firstly, it introduces the principle and process of DPC algorithm. Secondly, in view of the deficiencies of DPC algorithm, the optimization of DPC algorithm is summarized and analyzed, and the core technology, advantages and disadvantages of the optimization algorithm are pointed out. Finally, the possible challenges and development trend of DPC algorithm in the future are concluded.

Key words: clustering algorithm; density peak; cutoff distance; initial cluster center; micro-cluster merge; time complexity

Citation format: WANG S, XING S J, LIU C. Survey of density peak clustering algorithm[J]. Journal of East China Jiaotong University, 2023, 40(1): 106-116.

随着互联网和信息技术的快速发展,数据量每天都在以惊人的速度增加,人类社会已步入大数据时代,如何从海量数据中挖掘潜在的有价值的信息是一个迫切的问题。聚类分析是数据挖掘和机器学

收稿日期:2022-04-02

基金项目:江西省自然科学基金项目(20192ACBL20010)

习领域一种处理大数据的重要方法，无需先验知识[1]。其主要目的是根据数据集的某些特征将数据集划分为几个不同的潜在的簇，并且使簇内数据对象的相似性尽可能高，簇间的相似性尽可能低[2]。各种聚类算法已经成功应用于许多领域，如：图像处理[3-4]、短文本聚类[5]、损伤诊断[6]、模式识别[7]、生物信息学[8]、信息检索[9]等。

目前得到广泛使用的聚类算法非常多，并且研究人员仍在寻找新的聚类算法以应对各种不同的聚类任务。在过去的几十年里，涌现出了许多不同类型的聚类算法，主要包括基于划分的 K-Means[10] 和 K-Medoid[11]算法，基于密度的 DBSCAN[12]和 OPTICS[13]算法，基于层次的 CURE[14]和 BIRCH[15]算法，基于网格的方法[16]，基于图论的方法[17]，基于模型的方法[18]以及基于深度学习的方法[19]等。

2014 年，Rodriguez 和 Laio[20]在 SCIENCE 提出了一个基于密度和相对距离的新算法（clustering by fast search and find of density peaks，DPC）。其主要基于两个假设：① 聚类簇的中心的密度要比其周围数据对象的密度高；② 聚类中心到比它密度更高的数据对象的距离较远。由于 DPC 算法可解释性强，无需迭代，算法简单且能处理不同类型的数据集而备受瞩目。但 DPC 算法也有着一定的局限性，比如截断距离参数的选取需要提前确定，初始聚类中心的确定受人为主观影响，后续标签分配的链式问题，算法的复杂度较高等。为了提高 DPC 的性能以及克服上述缺点，各个领域的学者已经做了很多工作并提出了一系列的优化算法。

本文首先简述传统 DPC 算法的原理以及算法流程，然后针对 DPC 算法的不足，按照 4 个优化方向分类讨论了不同优化算法的关键技术与优缺点，最后对 DPC 算法的未来发展所可能面临的挑战及优化方向进行了展望。

1　传统密度峰值聚类算法

1.1　密度峰值聚类算法原理

DPC 算法是一种基于距离和密度的无监督的聚类算法，且无需提前输入簇的真实个数。DPC 算法首先寻找密度峰值点作为初始聚类中心，之后按照距离将剩余非中心点分配至最近的聚类中心所在的簇。寻找簇中心（密度峰值点）作为算法最重要的一步，主要思想基于两个假设：① 簇的中心局部密度较高且被密度比它低的数据所包围；② 簇中心到比其密度更大的数据对象的距离相对较大。针对以上假设，DPC 的作者提出了两个重要参数描述每一个数据对象：局部密度 ρ 和与密度更高点的相对距离 δ。

对于数据集 $X=\{x_1,x_2,\cdots x_n\}$，采用欧式距离计算两点之间的相似性

$$\mathrm{dist}(x_i,x_j)=\sqrt{\sum_{i=1}^{m}(x_{il}-x_{jl})^2} \tag{1}$$

式中：m 为数据的维度；x_{il} 是数据对象 x_i 在第 l 个维度下的取值。

DPC 算法使用截断距离 d_c 或者高斯核计算数据对象的局部密度 ρ_i，对于数据集 X 每一个数据对象 x_i，其局部密度 ρ_i 的计算为

$$\rho_i=\sum_{x_i\neq x_j}\varphi(\mathrm{dist}(x_i,x_j)-d_c) \tag{2}$$

式中：$\varphi(x)$ 为判断函数，当 $x\geqslant 0$ 时，$\varphi(x)=1$；当 $x<0$ 时，$\varphi(x)=0$。

$$\rho_i=\sum_{x_i\neq x_j}\exp\left(-\left(\frac{\mathrm{dist}(x_i,x_j)}{d_c}\right)^2\right) \tag{3}$$

式中：$\mathrm{dist}(x_i,x_j)$ 表示数据对象 x_i 和 x_j 之间的欧氏距离，$d_c>0$。式(2)计算的局部密度即为以数据对象 x_i 为中心，以截断距离 d_c 为半径的邻域内包含的点的个数，或者可以认为是与数据对象 x_i 的距离小于 d_c 的数据对象的个数。式(3)计算的则是所有数据对象到该点的高斯距离之和。

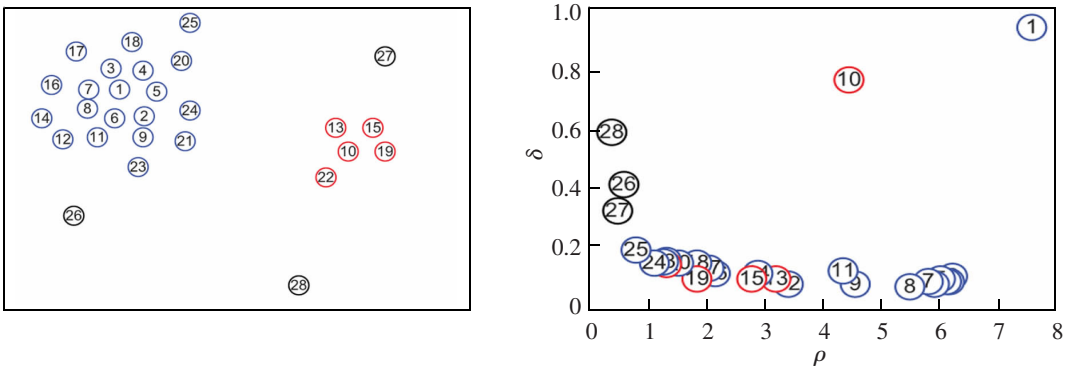
相对距离 δ_i 定义为

$$\delta_i=\min_{x_j:\rho_j>\rho_i}(\mathrm{dist}(x_i,x_j)) \tag{4}$$

对于具有最大局部密度的数据对象，其相对距离 δ_i 为

$$\delta_i=\min_{x_i\neq x_j}(\delta_j) \tag{5}$$

DPC 算法认为，当局部密度 ρ 和相对距离 δ 的值都比较大时，此时对应的数据对象即为初始聚类中心（密度峰值点）。通过绘制出关于 ρ 和 δ 的二维决策图，初始聚类中心点与非聚类中心点明显分开，故通过人为选择位于右上角区域（较大的 ρ 和 δ）的点作为初始聚类中心点，且挑选出的点的个数即为最终聚类簇的个数。图 1(a)为数据实际分布情况，图 1(b)为投影至二维空间得到的决策图。密度峰值点即为决策图右上角的突出的点 1 和点 10，参考图 1(a)，这两点位于每一个簇的中心位置，表明



(a) Point distribution (b) Decision graph

图1 数据分布与决策图

Fig.1 Data distribution and decision graph[20]

选择的初始聚类中心是合适的。点7虽然也有较高的密度，但其相对距离 δ 的值很小，这说明在点7的周围存在密度比其更大的点(点1)，点7不能作为初始聚类中心。另外点28具有较高的值但密度较小，这表明点28是远离聚类簇的噪声点。

另外也可以通过决策值 $\gamma_i=\rho_i\delta_i$ 选择聚类中心，选择具有较大决策值的点作为初始聚类中心。最后再将所有的非聚类中心点分配至距离最近的聚类中心点所在的簇中，完成聚类过程。

1.2 传统DPC算法流程

密度峰值聚类算法的主要步骤如下：

1) 输入数据集 $X=\{x_1,x_2,\cdots,x_n\}$；

2) 根据式(1)计算 n 个数据对象之间的欧式距离矩阵 $D^{n\times n}$；

3) 根据式(2)或(3)计算数据对象局部密度参数 ρ_i；

4) 根据式(4)或(5)计算数据对象 x_i 到具有更高密度的数据点的最近距离 δ_i；

5) 根据密度参数 ρ_i 和相对距离参数 δ_i 绘制决策图；

6) 根据决策图或决策值 γ 挑选合适的初始聚类中心；

7) 将剩下的非聚类中心点分配至具有更高的密度 ρ_i 中距离最近的点所在的簇；

8) 输出最终聚类结果 $\varphi=\{C_1,C_2,\cdots,C_m\}$。

2 密度峰值聚类算法的优化

2.1 实现DPC算法的自动化

提前确定合适的截断距离参数与密度峰值点的选取阻碍了算法的自动化。DPC算法的核心即寻找密度峰值点，而决定密度参数大小的因素主要取决于 d_c。DPC算法对输入 d_c 的值异常敏感，过大或者过小的 d_c 值可能产生完全不同的聚类结果，聚类精度波动大，但提前确定合适的 d_c 的值是非常困难的。另外，密度峰值点的选取是人为通过决策图选取的，当决策图复杂时难以选取合适的峰值点，且操作人员得到的峰值点受主观影响较大，同一个数据集可能会产生不同的选择结果。

Liu等[21]在ADPC-KNN中引入K-最近邻的思想自动计算截断距离 d_c 以及基于K-最近邻的密度 ρ_i 计算方式

$$\rho_i=\sum_{j\in KNN}\exp\left(\frac{d_{ij}^2}{d_c^2}\right) \tag{6}$$

$$d_c=\mu^K+\sqrt{\frac{1}{n-1}\sum_{i=1}^{N}(\delta_i^K-\mu^K)^2} \tag{7}$$

$$\mu^K=\frac{1}{N}\sum_{i=1}^{N}\delta_i^K \tag{8}$$

式中：d_{ij} 为两点间的欧氏距离；δ_i^K 为数据点 i 与第 K 个最近邻之间的距离；N 为整个数据集的容量；μk为不同数据点的 δ_i^k 的均值。ADPC-KNN能自适应地找到合适的 d_c 值和初始聚类中心。虽然算法减少了对 d_c 的依赖性，但同时引入了需要提前确定的K-最近邻参数。另外，算法仍然需要通过人工确定初始聚类中心的个数。

王芙银等[22]提出一种结合鲸鱼优化算法的WOA-DPC算法，建立ACC目标函数，利用鲸鱼优化算法迭代寻找使指标最大时所对应的 d_c 值即最优的截断距离参数。另外，根据加权的决策值斜率的变化情况确定初始聚类中心。该算法实现了自动化，降

低了对截断参数的依赖性。但 WOA-DPC 算法的时间复杂度高于传统 DPC 算法，在针对大型数据集时耗时较长。

Wang 等[23]引入物理学中的数据场的势能熵，自动确定计算密度参数时的截断参数 d_c，避免了人为确定的随机性。场函数中数据对象的潜在势能 φ_i 即数据分布的局部密度，越密集的对象 φ_i 的值越大，能很好地反应数据分布的情况。熵用来描述数据集的不稳定性，最小的熵取值对应于 σ 的最优取值。最后根据高斯分布的 3σ 准则，得到最优 d_c 的值为 $\frac{\sqrt{3}}{2}\sigma$。实验结果表明，针对小型数据集，该算法的聚类效果优于传统算法且能自动确定截断参数 d_c 的取值，但面对大型数据集时的有效性还有待研究。

王洋等[24]引入基尼指数描述数据的不纯度，自适应地确定截断距离参数 d_c 同时自动获取初始聚类峰值点。当基尼指数的值最小时所对应的 σ 的取值即为截断距离参数 d_c。按 γ 对数据点降序排列，并绘制二维图像，横坐标为数据点的排列序号，纵坐标为 γ 值。然后通过寻找决策值的临界值 p 自动选择初始聚类中心，如果 p 满足

$$p=\max\{i \mid ||k_i|-|k_{i+1}|| \geqslant \beta, i=1,2,3,\cdots,n-2\} \quad (9)$$

$$\beta=a(j)/(n-2) \quad (10)$$

$$a(j)=\sum_{j=1}^{n-2}||k_j|-|k_{j+1}|| \quad (11)$$

式中：k_i 为 γ 排序图上相邻两点之间的斜率；$a(j)$ 为 γ 序列图中相邻两点之间斜率差的和，β 为数据集中 $a(j)$ 的平均值。

Flores 等[25]提出一种通过检测决策值 γ 在连续数据点之间的差距 gap 进而自动确定聚类中心的方法。首先将所有数据对象 $\{x_1,x_2,\cdots,x_n\}$ 按照 γ 值降序排列为 $P=\{p_1,p_2,\cdots,p_n\}$。并定义点距离 $d_i=|\gamma_i+\gamma_{i+1}|$，平均点距离为 $\bar{d}=\sum_{a\in P}\frac{d_a}{|P|}$，则最后一次出现的 $d_i \geqslant \bar{d}$ 所对应的点 i 即为阈值点，比点 x_i 的 γ 值大的数据点将被自动视为初始中心。

Ding 等[26]提出新的判断初始聚类中心的参数，提出了 DPC-GEV 和 DPC-CI 自动选择初始聚类中心。首先采用“最小最大化”将局部密度 ρ 和相对距离 δ 标准化，DPC-GEV 认为新的判断参数 $\theta=\min(\rho^*,\delta^*)$ 大致遵循广义极值分布，利用式(12)确定分位数 $\hat{x}_p$，$\theta>\hat{x}_p$ 的点被视为初始中心。

$$\hat{x}_p=\begin{cases}\hat{\mu}-\frac{\hat{\sigma}}{\hat{\xi}}(1-y_p^{-\xi}), & \hat{\xi}\neq 0\\ \hat{\mu}-\hat{\sigma}\log y_p, & \hat{\xi}\neq 0\end{cases} \quad (12)$$

$$y_p=-\log p \quad (13)$$

式中：$\hat{\mu},\hat{\sigma},\hat{\xi}$ 的值均通过最大似然估计确定。DPC-CI 借助切比雪夫不等式，如果点 i 满足式(14)则被视为初始聚类中心。

$$\theta_i>(\mu+\varepsilon\times\sigma),\forall\varepsilon>0 \quad (14)$$

式中：μ 和 σ 分别为判断指标 θ 的均值和标准差。实验结果表明，DPC-GEV 与 DPC-CI 可以选择出合理的初始中心，保证了 DPC 的连续性。但由于新的判断指标 θ 需参考局部密度参数，对于截断参数 d_c 的依赖仍然存在。

关于上述 DPC 自动化的优化策略的对比分析如表 1 所示。

2.2　优化密度参数 ρ 和相对距离 δ

尽管 DPC 算法能处理大部分形状数据集，但局部密度的定义过于简单，当面对形状结构复杂的数据集时聚类精度下降。例如一个簇内存在多个密度峰会导致簇被分割，数据分布的稠密性差异较大时 DPC 会忽视稀疏簇，流形数据集中位于不同簇但距离较近的数据对象被误分配标签等。需要对 DPC 算法中的密度 ρ 和相对距离 δ 进行优化以适应不同类型的数据集。

Guo 等[27]提出的 NDPC 认为无论是在密集或者稀疏的簇中，相比于其他点，聚类中心被包含在更多的 K-最近邻中。将数据对象的局部密度定义为其邻居包含该点的数据对象的个数，从而公平对待密度分布差异较大的区域。基于此定义，初始聚类中心的挑选不会造成稀疏簇的遗漏。另外，若一个簇中存在多个密度峰值，提出 NDPC 结合凝聚层次聚类算法的 NDPC-AC 将被分割的小簇迭代合并，直到最终簇的个数等于真实簇的个数，但 NDPC-AC 算法的凝聚策略需要提前知道真实簇的个数。刘娟等[28]提出一种借助自然反向最近邻的密度峰值聚类算法，利用反向最近邻计算密度参数，利用密度自适应距离计算初始中心的距离，之后挑选更为合适的初始聚类中心(密度峰值点)。Du 等[29]提出的 DPC-KNN-PCA 中将 K-最近邻的思想引入密度峰

表1 DPC算法的自动化

Tab.1 The automation of DPC algorithm

Improve algorithm	Core idea	The aspects of improvement				The Shortcoming
		Determine d_c	Determine initial center	Improve accuracy	Improve speed	
ADPC-KNN	KNN	Automatic	Non-automatic	No	No	A new parameter K, and determine the number of clusters manually
WOA-DPC	Whale optimization algorithm	Non-automatic	Automatic	Yes	No	Increase the time complexity and it has poor clustering effect on high-dimensional datasets
Literature[23]	The potential entropy of data field	Automatic	Non-automatic	Yes	No	The clustering effect of large datasets is poor
ADPC	Gini index	Automatic	Automatic	No	No	Increase the time complexity of algorithm
GB-DPC	The difference of decision value	Non-automatic	Automatic	No	No	The parameters need to be determined in advance
DPC-GEV&DPC-CI	Generalized extreme value distribution & Chebyshev inequality	Non-automatic	Automatic	No	No	The parameters need to be determined in advance

值聚类算法并改进局部密度的定义,同时结合主成分分析(PCA)降维处理高维数据。

Hou 等[30]认为造成 DPC 算法性能不佳的原因是算法假设与实现过程的不一致性。假设基于数据对象间的相对密度关系,而算法实施过程中密度参数的计算却是绝对密度关系,故作者提出了一个新的基于相对密度关系的初始聚类中心识别准则。首先,作者提出从属点和直接上级点的定义,并认为数据点与其上级点存在于同一个簇中。如图2所示,箭头由从属点指向直接上级点。

如果 x_2 是 x_1 最近的高密度点,即 $\delta_1=d(x_1,x_2)$,那么 x_1 是 x_2 的直接从属点,x_2 是 x_1 的直接上级点。为防止两个独立簇的合并,满足 $\delta>T$ 的点被视为聚类中心,聚类中心点不会作为任意点的从属点。其次,数据对象 x_1 的局部密度 ρ_i' 通过只考虑直接从属点的个数 η_i 确定

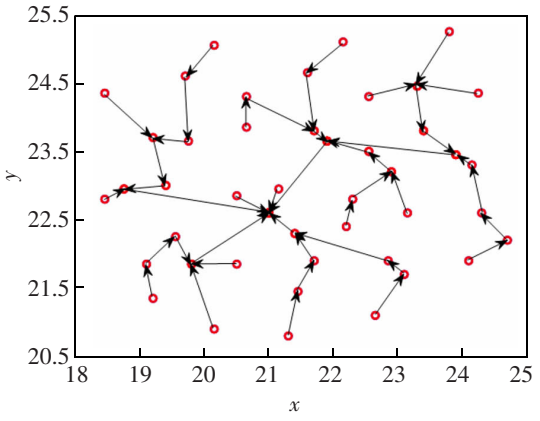


图2 从属点与上级点

Fig.2 Subordinate and superior points

$$\rho_i'=\sum_{j\in S,j\neq i}\zeta(s_i,x_j) \tag{15}$$

$$\zeta(u,v)=\begin{cases}1, & v\text{ 为 }u\text{ 的直接从属点且 }v\in S_{\eta_i}\\ 0, & \text{其他}\end{cases} \tag{16}$$

式中：S_{η_i}代表数据点 x_i 的 η_i 最近邻的集合。该算法在数据密度分布不均匀时的聚类效果优于其余聚类算法，但阈值 T 和初始聚类中心仍需要人为提前指定，且算法结果受 K-最近邻参数 K 的影响较大。

Xu 等[31]提出的 RDPC-DSS 在 DPC 算法中用密度自适应距离代替欧式距离处理流形数据集。在流形数据集中，如果两点可以通过一条穿过高密度区域的路径连接，那么称他们具有较高的相似度。数据集 $X=\{x_1,x_2,\cdots,x_n\}$ 中的数据对象可以被视为图 $G=(V,E)$ 的结点，$P=\{p_1,p_2,\cdots,p_l\}$ 为 p_1 至 p_l 的路径，p_{ij} 为连接 v_1 至 v_2 的所有路径。密度自适应距离 S_{ij} 的计算为

$$S_{ij}=\frac{1}{\min\limits_{p\in p_{ij}}\sum\limits_{k=1}^{l-1}\left(\exp\left(\rho'\cdot\mathrm{dist}\left(p_k,p_{k+1}\right)\right)-1\right)} \tag{17}$$

式中：ρ'为放缩因子。之后将密度自适应距离引入 DPC 算法，更新局部密度和相对距离的定义。RDPC-DSS 在流形数据集上的聚类效果优异，但计算密度自适应距离的时间复杂度远高于 DPC 算法，使得处理大型数据集需要消耗大量的时间。

Lotfi 等[32]提出的 DPC-DBFN 根据一种新的模糊核进行局部密度的计算，减少离群点的影响。根据决策图筛选出簇主干，簇主干能大致保持数据集的形状且距另一簇的主干距离较远，然后完成对剩余非主干点标签的传播。

第 1 步，基于模糊邻域的密度参数 ρ_i 定义为

$$\rho_i=\max\left\{1-\frac{1}{K}\left(\sum_{j\in K_{\mathrm{NN}}}d\left(x_i,x_j\right)\right),0\right\} \tag{18}$$

式中：$K_{\mathrm{NN}}(x_i)$为数据点 x_i 的 K-最近邻。

第 2 步根据决策图确定聚类中心点，簇主干点，边界点以及噪声点。score 函数值最大的 c 个数据作为初始聚类中心

$$\mathrm{score}(i)=\left(\frac{\rho_i}{\max(\rho)}\right)^2\times\frac{\delta_i}{\max(\delta)} \tag{19}$$

对于其余数据，如果 $\rho_i>\bar{\rho}$，则 x_i 为主干中的密集点；如果 $\rho_i<\bar{\rho}$ 且 $\delta_i<\lambda\sigma_\delta^2$，则 x_i 为簇的边界点；如果 $\rho_i<\bar{\rho}$ 且 $\delta_i<\lambda\sigma_\delta^2$，则 x_i 为噪声点。其中，λ 为控制参数，σ_δ^2 为相对距离的方差。

第 3 步分配标签主要包括主干点的分配以及边界点的分配。标签分配过程始于将簇中心的标签分配到附近的主干点，首先将每个簇的标签传给初始中心的 K 个最近的主干邻居，之后再将标签传递给邻居的邻居，重复此过程，直到所有主干密集点都被分配簇标签。DPC-DBFN 成功地克服了初始 DPC 算法以及大部分优化算法带来的问题，该算法能有效地识别聚类中心，并且能对形状复杂的数据进行聚类，并且对于噪声点具有鲁棒性。但该算法中仍存在两个参数需要确定，控制参数和最近邻参数 K，且其对于聚类性能有着较大的影响。

基于共享最近邻的快速搜索密度峰值聚类算法 SNN-DPC[33]提出了共享最近邻的概念，充分考虑数据集的结构和形状提出了新的两个数据对象的相似性度量。并采用补偿值的方式来处理簇间密度差异较大的数据集，为改善 DPC 算法后续非中心点分配过程的链式问题，提出了两步分配的策略。共享最近邻 SNN 定义为 K-最近邻的交集 S_{NN}，基于 SNN 的相似性 $\mathrm{sim}(x_i,x_j)$定义为

$$\mathrm{sim}(x_i,x_j)=\begin{cases}\dfrac{\left|S_{\mathrm{NN}}(x_i,x_j)\right|^2}{\sum_{p\in S_{NN}(x_i,x_j)}\left(\left(d(x_i,p)+d(p,x_j)\right)\right)}, & \\ \quad 若\ x_i,x_j\in S_{\mathrm{NN}}(x_i,x_j) & \\ 0, \quad 其他 & \end{cases} \tag{20}$$

式中：局部密度 ρ 定义为具有最大 S_{NN} 相似性的 K 个数据对象的相似性之和。当数据分布的稀疏程度差异较大时，为避免忽视稀疏簇，提出基于补偿值的相对距离 δ 的定义。SNN-DPC 在面对复杂形状数据集时取得了优异的聚类结果，且对噪声点具有鲁棒性。但缺点在于需要提前确定 K-最近邻的参数 K，以及初始聚类中心仍需要通过人工选取，打断了算法的连续性。

本小节基于 DPC 优化密度参数和相对距离的不同策略，分析了优化算法的核心技术以及优缺点，优化算法的对比分析如表 2 所示。

2.3　微簇合并避免链式问题

DPC 算法在面对流形数据集时，受算法假设的影响，即使挑选了合适的初始聚类中心，但在后续非中心点标签的分配过程中可能会产生链式问题：一个误分配点可能会导致大范围数据标签的错误传播。凝聚层次聚类算法将数据对象看作单独簇，每次合并相似度最高的两个簇，直至合并到真实簇的个数。研究人员提出了一些类似于层次聚类的算法，先将数据集划分成多个微簇，对数据对象进行局部范围内的标签的分配，避免链式问题。微簇的个数往往大于真实簇的个数，之后根据提出的簇间

表2 优化密度参数和相对距离

Tab.2 Optimizing density parametersand relative distance

Improve algorithm	Core idea	The aspects of improvement				Shortcoming
		Determine d_c	Determine initial center	Improve accuracy	Improve speed	
NDPC-AC	Optimize the density with KNN, the aggregation hierarchical clustering	Non-automatic	Non-automatic	Yes	No	A new parameter K is introduced, and the number of clusters needs to be determined in advance
Literature[28]	The natural inverse nearest neighbor; Density adaptive distance	Non-automatic	Non-automatic	Yes	No	The initial clustering center needs to be selected manually through the decision graph, and the time complexity is high
DPC-KNN-PCA	KNN & PCA	non-automatic	non-automatic	Yes	No	When the dataset appears vertical stripes, the clustering effect is poor
Literature[30]	The subordination of data points and optimize the definition of local density	Non-automatic	Non-automatic	Yes	No	The threshold parameter and the parameter K need to be determined, and the clustering results are greatly affected by the value of K
RDPC-DSS	The density adaptive distance	Non-automatic	Non-automatic	Yes	No	The time consumption of calculating density adaptive distance is much greater than DPC, and it is unable to process large-scale datasets
DPC-DBFN	The cluster backbone	Non-automatic	Non-automatic	Yes	No	Control parameter need to be determined, which has a great impact on the clustering performance; The initial center be selected manually
SNN-DPC	The shared nearest neighbor , a two-step allocation strategy	Non-automatic	Non-automatic	Yes	No	The number of clusters and parameter K need to be known. The complexity is higher than DPC

相似性不断地将微簇进行合并以形成最终的聚类结果。

Zhang 等[34]受自然界衰减现象的启发,提出基于密度衰减图的 DGDPC 算法,通过密度衰减图形成初始簇，之后根据同一衰减现象下的数据对象属于同一个簇完成聚类过程。Li 等[35]定义一种局部密度差异处理高维且数据分布不均匀的数据集,考虑了数据对象与其邻居的密度差异,公平对待稠密程度不同簇的确定核心点。之后在 K 最近邻图上寻找潜在的交叉簇的边，并删除连接了噪声的边以及距离过远的边，最后根据得到的新的 K-最近邻图完成聚类。Bie 等 [36] 提出了 Fuzzy-CFSFDP 算法借助模糊规则对不同的密度峰挑选不同的聚类中心，并且将具有相近的内部模式的密度峰进行合并，最后通过一种启发式的方式挑选合适的初始中心。

Cheng 等[37]将自然最近邻的概念引入密度峰值聚类算法,提出了一种无需参数的基于局部核心密集成员的 DLORE-DPC 算法,其本质是先寻找微簇核心再进行微簇间的合并。通过挑选局部核心代表整个数据集,形成不同的微簇核心,之后在局部核心点之上通过图距离完成 DPC,最后根据代表点完成对剩余点的标签的分配。数据点 x_i 局部密度 den (x_i)为

$$\mathrm{den}(x_i)=\frac{k}{\sum_{x_j\in S_{\mathrm{NN}}(x_i,x_j)}\mathrm{dist}(x_j,x_i)} \tag{21}$$

式中:k 为 x_i 的自然最近邻数。如果数据对象 p 是 q 的局部邻域里具有最大 den 的点,则 p 称作数据对象 q 的代表点,记作 $\mathrm{REP}(q)=p$。在此基础上,如果 $\mathrm{REP}(p)=p$,数据对象 p 被认为是一个局部核心点。将局部核心点的集合视为一个较小数据集,对其执行 DPC 算法,两个局部核心点 p,q 之间基于图的距离 $D_{\mathrm{G}}(p,q)$定义为:

$$D_{\mathrm{G}}(p,q)=\sum_{k=1}^{m-1}D_{\mathrm{S}}(p_k,p_{k+1}) \tag{22}$$

$$D_{\mathrm{S}}(p,q)=\begin{cases}\dfrac{d(p,q)}{|\mathrm{dlsore}(p,q)|\times\sum_{o\in\mathrm{dlsore}(p,q)}\mathrm{den}(o)}, & \text{若}|\mathrm{SNN}(p,q)|\neq 0\\ \max d, & \text{其他}\end{cases} \tag{23}$$

式中:p_k,p_{k+1} 为最短路径上相邻的局部核心点;m 为连接核心点的节点数;$|\mathrm{dlsore}(p,q)|$为局部核心点之间的共享最近邻。实验结果表明,DLORE-DPC 算法无需人为输入参数能发现不同形状的聚类簇,由于该算法只计算局部核心之间的图距离且只对核心点执行 DPC 算法,算法的时间复杂度得到大大降低,但自然最近邻搜索算法受噪声点的影响较大且核心点集上的初始聚类中心的选取仍未自动化。

Ni 等[38]提出了一种寻找突出峰值点的优化算法 PPC,其通过将数据集划分为多个潜在的簇,然后将密度峰值不突出的簇合并,从而获得准确的聚类结果。首先根据统计学中标准差和均值可以衡量数据的中心化趋势,将满足 $\delta_i>\bar{\delta}+S$ 的数据点视为潜在的簇中心,$\bar{\delta}$ 和 S 分别代表相对距离参数的均值和标准差,此时得到的簇的个数多于真正簇的个数。然后使用两个密度峰之间的密度差确定峰值是否突出。如果此峰的密度差异大于阈值 th,则足够突出,就是一个聚集中心;否则,它与密度较高的邻近峰在同一簇中。

Chen 等[39]认为对于具有变密度分布(VDD)数据集,继续使用统一的密度函数会导致稀疏簇的丢失,故提出了针对数据密度分布不均匀的优化算法 DADC。首先定义了一种基于 K-最近邻(KNN)的区域自适应密度,以自适应地检测不同密度区域的密度峰值。区域自适应密度 ρ_i 定义为

$$\rho_i=\begin{cases}\partial_i\times\max\limits_j(d_{ij}), & \text{若}\partial_i=\partial_{\max}\\ \partial_i\times\min\limits_{j:\partial_j>\partial_j}(d_{ij}), & \end{cases} \tag{24}$$

$$\partial_i=\mathrm{den}_k(i)+\sum_{j\in K_{\mathrm{NN}}(j)}(\mathrm{den}_k(j)\times\omega_i) \tag{25}$$

$$\mathrm{den}_k(i)=\frac{k}{\sum_{j\in K_{\mathrm{NN}}(j)}d_{ij}} \tag{26}$$

其中:$\omega_i=1/d_{ij}$ 为数据与其 K-最近邻的权重值;$\mathrm{den}_k(i)$,$\mathrm{den}_k(j)$为中间变量。最后根据簇间融合度是否大于阈值决定微簇的合并。

基于微簇合并避免链式问题的优化 DPC 算法的对比分析如表 3 所示。

2.4　提升算法速度

DPC 算法的时间复杂度较高,DPC 算法的时间复杂度主要由三方面组成:第一,计算数据点之间的欧氏距离矩阵需要消耗 $O(N^2)$;第二,对于每个数据对象的局部密度和相对距离的计算都需要遍历数据集中的每一个点,时间复杂度为 $O(N^2)$;第三,对于非聚类中心点的分配过程的时间复杂度为 $O(N^2)$。因此,DPC 算法整体的时间复杂度为 $O(N^2)$,算法的时间消耗较大,处理大型数据集时的效率不高。

Chen 等[40]提出的 FastDPeak 算法基于降低原始 DPC 算法中计算局部密度和对距离的时间复杂度。首先采用 Cover-tree 寻找每个数据对象的 K-最近邻,并利用 KNN 密度代替原始密度。其次对于相对距离的计算,将数据对象分为局部密度峰值和一般点(非局部密度峰值)。局部密度峰值点是数据对象在其 K-最近邻具有最大的局部密度,否则为一般点。一般数据点的相对距离的计算只需要遍历其 K-最近邻,而对于局部峰值点,通过逐渐扩大 K 值的方式计算其相对距离,大大降低了算法计算密度的时间消耗,时间复杂度约为。Wu 等[41]提出了 DGB 聚类算法,DGB 运用了网格化的思想减少数据空间中不必要的欧式距离的计算。首先将数据空间进行网格化得到一些小单元格 Cell,将原本每个数据点之间的欧氏距离替换为计算少量的非空单元格 Cell 之间的距离,这大大地提升了算法的计算速度。Sami 等[42]提出一种优化的快速密度峰值聚类算法 FastDP,在对聚类质量没有明显影响的情况下实现了算法的加速。其使用一种快速且通用性强的 K-

表3 微簇合并避免链式问题

Tab.3 Micro cluster merge to avoid chain problem

Improve algorithm	Core idea	The aspects of improvement				Shortcoming
		Determine d_c	Determine initial center	Improve accuracy	Improve speed	
DGDPC	The density decay patterns form micro clusters and then merged	Non-automatic	Non-automatic	Yes	No	Parameter needs to be determined in advance, and the clustering effect of complex datasets is poor
LGD	The density with KNN graph, and clustering based on connection relationship	Non-automatic	Non-automatic	Yes	No	The nearest neighbor parameter K and threshold are introduced
DLORE-DPC	The natural nearest neighbor, local core points, graph distance	Non-automatic	Non-automatic	Yes	No	The clustering effect is greatly affected by noise, and the clustering center when performing DPC on micro clusters needs to be determined manually
PPC	Merge the micro clusters with non-prominent peak	Automatic	Non-automatic	Yes	No	Parameters and threshold are introduced, and the optimization algorithm is greatly affected by outliers
DADC	Domain adaptive density and micro cluster merging	Non-automatic	Automatic	Yes	No	The threshold parameter for judging micro clustering is introduced

最近邻图计算局部密度和相对距离参数,传统DPC算法二次时间复杂度的问题得到消除,并允许对大规模的数据集进行聚类。

表4对比分析了基于提升速度的优化DPC算法。

表4 提升DPC算法速度

Tab.4 Increasing the speed of DPC

Improve algorithm	Core idea d_c	The aspects of improvement				Shortcoming
		Determine	Determine initial center	Improve accuracy	Improve speed	
FastDpeak	Using cover-tree to find KNN and KNN density	Non-automatic	Automatic	No	Yes	The number of clusters needs to be determined in advance. When the K value of the nearest neighbor parameter is larger, the storage space needs to be larger
DGB	Using grid to reduce unnecessary calculation of Euclidean distance	Non-automatic	Non-automatic	No	Yes	With the increase of data space dimension, the grid method may be invalid
FastDP	A fast and generic construction of the KNN graph	Non-automatic	Automatic	No	Yes	The K value of the nearest neighbor parameter needs to be determined in advance

3　结束语

本文主要将 DPC 的优化算法基于 4 大类进行了详细阐述和分析:实现 DPC 的自动化,优化密度参数和相对距离,微簇合并避免链式问题与提升算法速度,并指出优化算法的核心技术及优缺点。

参考文献:

[1] XU R,WUNSCH D. Survey of clustering algorithms[J]. IEEE Transactions on Neural Networks,2005,16(3):645–678.

[2] JAIN A K,Murty M N,FLYNN P J. Data clustering:a review [J]. ACM Computing Surveys(CSUR),1999,31(3):264–323.

[3] HOU J,LIU W,XU E,et al. Towards parameter–independent data clustering and image segmentation[J]. Pattern Recognition,2016,60:25–36.

[4] SULAIMAN S N,ISA N A M. Adaptive fuzzy–K–means clustering algorithm for image segmentation[J]. IEEE Transactions on Consumer Electronics,2010,56(4):2661–2668.

[5] YIN J,WANG J. A dirichlet multinomial mixture modelbased approach for short text clustering[C]//Association for Computing Machinery:Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining,2014.

[6] 韩庆华,马乾,刘名,等. 温度变化下基于固有频率聚类分析的空间网格结构损伤诊断[J]. 华东交通大学学报,2021,38(4):8–17.
HAN Q H,MA Q,LIU M,et al. Damage diagnosis of space grid structure based on natural frequency clustering analysis under varying temperature effects[J]. Journal of East China Jiaotong University,2021,38(4):8–17.

[7] HAMZA A B. Graph regularized sparse coding for 3D shape clustering[J]. Knowledge–Based Systems,2016,92:92–103.

[8] TRUONG H Q,NGO L T,PEDRYCZ W. Granular fuzzy possibilistic C–means clustering approach to DNA microarray problem[J]. Knowledge Based Systems,2017,133:53–65.

[9] BORDOGNA G,PASI G. A quality driven hierarchical data divisive soft clustering for information retrieval[J]. Knowledge–based systems,2012,26:9–19.

[10] MACQUEEN J. Some methods for classification and analysis of multivariate observations[C]//Berkelay:Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. 1967,1(14):281–297.

[11] KAUFMAN L,ROUSSEEUEW P J. Finding groups in data:an introduction to cluster analysis[M]. London:John Wiley & Sons,2009.

[12] ESTER M,KRIEGEL H P,SANDER J,et al. A density–based algorithm for discovering clusters in large spatial databases with noise[J]. Data Base System and Logic,1996,96(34):226–231.

[13] ANKERST M,BREUNIG M M,KRIEGEL H P,et al. OPTICS:Ordering points to identify the clustering structure[J]. ACM Sigmod Record,1999,28(2):49–60.

[14] GUHA S,RASTOGI R,SHIM K. CURE:An efficient clustering algorithm for large databases[J]. ACM Sigmod Record,1998,27(2):73–84.

[15] ZHANG T,RAMAKRISHNAN R,LIVNY M. BIRCH:an efficient data clustering method for very large databases[J]. ACM Sigmod Record,1996,25(2):103–114.

[16] WANG W,YANG J,MUNTZ R. STING:A statistical information grid approach to spatial data mining[C]. Programming,1997,97:186–195.

[17] VON L U. A tutorial on spectral clustering[J]. Statistics and Computing,2007,17(4):395–416.

[18] DEMPSTER A P,LAIRD N M,RUBIN D B. Maximum likelihood from incomplete data via the EM algorithm[J]. Journal of the Royal Statistical Society:Series B(Methodological),1977,39(1):1–22.

[19] 章永来,周耀鉴. 聚类算法综述[J]. 计算机应用,2019,39(7):1869–1882.
ZHANG Y L,ZHOU Y J. Review of clustering algorithms [J]. Journal of Computer Applications,2019,39(7):1869–1882.

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

[21] LIU Y H,MENG E M,YANG F. Adaptive density peak clustering based on K–nearest neighbors with aggregating strategy[J]. Knowledge–Based Systems,2017,133:208–220.

[22] 王芙银,张德生,张晓. 结合鲸鱼优化算法的自适应密度峰值聚类算法[J]. 计算机工程与应用,2021,57(3):94–102.
WANG F Y,ZHANG D S,ZHANG X. Adaptive density peaks clustering algorithm combining with whale optimization algorithm[J]. Computer Engineering and Applications,2021,57(3):94–102.

[23] WANG S,WANG D,LI C,et al. Clustering by fast search and find of density peaks with data field[J]. Chinese Journal of Electronics,2016,25(3):397–402.

[24] 王洋,张桂珠. 自动确定聚类中心的密度峰值算法[J]. 计算机工程与应用,2018,54(8):137–142.
WANG Y,ZHANG G Z. Automatically determine density

of cluster center of peak algorithm[J]. Computer Engineering and Applications,2018,54(8):137-142.
[25] FLORES K G,GARZA S E. Density peaks clustering with gap-based automatic center detection[J]. Knowledge Based Systems,2020,206:106350.
[26] DING J,HE X,YUAN J,et al. Automatic clustering based on density peak detection using generalized extreme value distribution[J]. Soft Computing,2018,22(9):2777-2796.
[27] GUO Z,HUANG T,CAI Z,et al. A new local density for density peak clustering[C]//Springer Cham:Pacific-Asia Conference on Knowledge Discovery and Data Mining,2018.
[28] 刘娟,万静. 自然反向最近邻优化的密度峰值聚类算法[J]. 计算机科学与探索,2021,15(10):1888-1899.
LIU J,WAN J. Optimized density peak clustering algorithm by natural reverse nearest neighbor[J]. Journal of Frontiers of Computer Science and Technology,2021,15(10):1888-1899.
[29] DU M,Ding S,JIA H. Study on density peaks clustering based on K-nearest neighbors and principal component analysis[J]. Knowledge-Based Systems,2016,99:135-145.
[30] HOU J,ZHANG A,QI N. Density peak clustering based on relative density relationship[J]. Pattern Recognition,2020,108:107554.
[31] XU X,DING S,WANG L,et al. A robust density peaks clustering algorithm with density-sensitive similarity[J]. Knowledge-Based Systems,2020,200:106028.
[32] LOTFI A,MORADI P,BEIGY H. Density peaks clustering based on density backbone and fuzzy neighborhood[J]. Pattern Recognition,2020,107:107449.
[33] LIU R,WANG H,YU X. Shared-nearest-neighbor-based clustering by fast search and find of density peaks[J]. Information Sciences,2018,450:200-226.
[34] ZHANG Z,ZHU Q,ZHU F,et al. Density decay graph-based density peak clustering[J]. Knowledge-Based Systems,2021,224:107075.
[35] LI R,YANG X,QIN X,et al. Local gap density for clustering high-dimensional data with varying densities[J]. Knowledge Based Systems,2019,184:104905.
[36] BIE R,MEHMOOD R,RUAN S,et al. Adaptive fuzzy clustering by fast search and find of density peaks[J]. Personal and Ubiquitous Computing,2016,20(5):785-793.
[37] CHENG D,ZHANG S,HUANG J. Dense members of local cores-based density peaks clustering algorithm[J]. Knowledge Based Systems,2020,193:105454.
[38] NI L,LUO W,ZHU W,et al. Clustering by finding prominent peaks in density space[J]. Engineering Applications of Artificial Intelligence,2019,85:727-739.
[39] CHEN J,PHILIP S Y. A domain adaptive density clustering algorithm for data with varying density distribution[J]. IEEE Transactions on Knowledge and Data Engineering,2019,33(6):2310-2321.
[40] CHEN Y,HU X,FAN W,et al. Fast density peak clustering for large scale data based on K_{NN} [J]. Knowledge Based Systems,2020,187:104824.
[41] WU B,WILAMOWSKI B M. A fast density and grid based clustering method for data with arbitrary shapes and noise [J]. IEEE Transactions on Industrial Informatics,2016,13(4):1620-1628.
[42] SIERANOJA S,FRANTI P. Fast and general density peaks clustering[J]. Pattern Recognition Letters,2019,128:551-558.

第一作者:王森(1969—),男,教授,硕士生导师,研究方向为计算机算法与应用。E-mail:515613251@qq.com。

通信作者:邢帅杰(1998—),男,硕士研究生,研究方向为聚类分析与数据挖掘。E-mail:xsj980414@163.com。

(责任编辑:姜红贵)