基于自然和加权共享最近邻的密度峰值聚类算法

王　森，陈　翔，詹小秦，徐 璐，吴启正

（华东交通大学理学院，江西 南昌 330013）

摘　要：密度峰值聚类（DPC）作为一种高效且不需要迭代的聚类算法得到广泛应用。研究发现，该算法使用在非球形簇和密度不均匀的聚类时，DPC 很难选择正确的簇中心，且该算法受截断距离参数的影响较大。【目的】为了解决 DPC 算法在密度分布不均匀的数据集上效果不佳的问题，【方法】提出了一种基于自然和加权共享最近邻的密度峰值聚类算法。该算法首先引入自然最近邻计算加权值，再根据一阶和二阶共享最近邻的定义重新计算数据对象之间的相似度，然后通过融合共享最近邻相似度的定义和自然最近邻权重值计算相对密度和相对距离，最后还设计了新的分类型簇中心扩散分配策略。【结果】在 8 个不同类型的数据集上的实验结果表明，本文所提出算法的聚类性能要明显优于其余 4 个对比算法。【结论】该方法在密度不均匀的数据集上对簇中心也有较好的识别效果，很好地解决了上述问题。

关键词：聚类算法；密度峰值聚类；自然最近邻；共享最近邻；簇中心扩散

中图分类号： TP391　　**文献标志码：**A

Density Peak Clustering Algorithm based on Natural and Weighted Shared Nearest Neighbors

Wang Sen, Chen Xiang, Zhan Xiao Qin, Xu Lu, Wu Qi Zheng

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

Abstract: Density Peak Clustering (DPC) has been widely used as an efficient and non-iterative clustering algorithm. However, studies have found that DPC struggles to select correct cluster centers, especially in datasets with non-spherical clusters and non-uniform density. Moreover, the algorithm is heavily influenced by the truncation distance parameter. 【Objective】In order to address the issue of poor performance of DPC on datasets with uneven density distributions, 【Method】we propose a density peak clustering algorithm based on natural and weighted shared nearest neighbors. This algorithm first introduces natural nearest neighbor computations to calculate weights, then redefines the similarity between data objects based on the definitions of first-order and second-order shared nearest neighbors. Subsequently, by fusing the definitions of shared nearest neighbor similarity and natural nearest neighbor weights, relative density and relative distance are calculated. Finally, a novel strategy for distributing cluster centers is designed. 【Result】Experimental results on sixs different types of datasets demonstrate that the proposed algorithm outperforms four other comparative algorithms significantly in terms of clustering performance. 【Conclusion】The method achieves better cluster center identification on datasets with non-uniform density, effectively addressing the aforementioned issues.

Key words: Cluster algorithm; Density peak clustering; Natural nearest neighbor; Shared nearest neighbor; Cluster center diffusion.

聚类是机器学习中的一种分类方法，旨在根据数据集的某些特征将其划分为多个潜在的簇。该方法致力于确保簇内的数据对象具有较高相似性，而簇间的相似性则尽可能低[1]。目前各领域的科学家和学者已经提出了多种不同的聚类算法，包括基于划分的算法[2]、基于密度的算法[错误!未找到引用源。]、基于层次[4]的算法、基于网格[5]的算法、基于图论[6]以及基于模型[错误!未找到引用源。]的算法。理想的聚类算法应当具

备高效性、鲁棒性等特点，且应无需过多参数就能产生有价值的聚类结果。

【研究意义】在 2014 年，Rodriguez 和 Laio[8]在《SCIENCE》杂志上提出了一种名为密度峰值聚类（Clustering by Fast Search and Find of Density Peaks，简称 DPC）的新算法。由于 DPC 算法具有较强的可解释性、并且其无需迭代、简单且能够识别具有形状的数据集，因此受到了各科学领域的广泛关注。然而，传统的 DPC 算法存在以下三个不足之处：其一，人为设置的截断参数值对聚类效果会产生较大影响；其二，对于密度分布不均匀的数据集，该算法在识别簇中心时容易忽略稀疏簇的中心点；其三，DPC 算法进行非中心点标签分配时，一个点的分配错误会导致后续点也分配错误的链式反应。为了高效准确地用 DPC 算法进行聚类分析，对 DPC 算法进行相关的改进具有重大研究意义。

【研究进展】DPC 算法自问世后也提出了一系列改进算法。Liu 等人[错误!未找到引用源。]提出的 ADPC-KNN 算法的核心思想是在计算截断距离时通过 KNN 自动计算参数值，可以自适应地找到了合适的截断距离，从而实现了参数选择的自动化，该算法虽然自适应地找到了合适的截断距离值，但是却增加了新的参数值 K；Guo 等人[10]提出了一种基于 K 近邻集的 NDPC 算法，其核心思想是簇中心点往往被低密度的数据对象包围，所以也被以该点为邻居的所有数据对象包围，计算所有视该点为邻居的数据对象的个数即可得到该点的相对密度，该方法在密度不均匀的数据集上使用效果较好；Cheng 等人[11]提出了一种基于局部核心点的 DLORE-DPC 算法，该算法结合了自然最近邻的思想先寻找微簇的核心点作为微簇的代表点，计算代表点间的相似度，再对微簇进行合并，最后完成聚类任务。Liu 等人[12]提出了一种基于共享最近邻的 SNN-DPC 算法，该算法根据共享最近邻的概念重新定义了数据对象间的相似度，计算相对距离时提出了基于 K 近邻的补偿值，最后在分配非中心点时则使用两步分配策略。

【创新特色】针对 DPC 算法的三点不足之处，本文提出一种基于自然和共享最近邻的密度峰值聚类算法（DPC-NN-WSNN）。该算法首先引入自然最近邻的概念，无需输入任何参数就能找到每个数据对象的自然最近邻权重值，然后用基于一阶和二阶共享最近邻的融合相似度公式乘上该权重值，计算相对密度值，再用相同的方法计算相对距离，然后得出决策图，选取合适的簇中心。最后使用分类型簇中心扩散分配策略，基于两种不同密度的簇中心设置不同的参数进行标签分配。该算法无需设置截断距离参数，并且在对稀疏簇的中心识别时添加了基于自然最近邻权重值的补偿值，最后设计的分类型簇中心扩散策略也减小了链式错误产生的影响，该算法对 DPC 算法的三点缺陷都进行了合理地改进，可以有效完成聚类任务。

【关键问题】研究 DPC 算法各种不同类型的数据集上的缺陷，然后针对不足之处进行改进，有助于提升算法的鲁棒性，对于数据挖掘和聚类分析有较大现实意义。

1 DPC 密度峰值聚类算法

1.1 密度峰值聚类算法核心思想

DPC 算法其核心思想基于两个假设：首先每个簇的中心密度相对较高且被低密度区域所包围，其次每个簇中心之间的距离会相对较远[13]。为了识别满足这两个假设的簇中心，引入了相对密度 ρ_i 和相对距离 δ_i 这两个定义对应这两个假设。数据对象 x_i 的相对密度记作 ρ_i，定义为：

$$\rho_i = \sum_{x_i \neq x_j} \chi\left(\mathrm{dist}(i,j) - d_c\right) \tag{1}$$

公式（1）中，$\mathrm{dist}(i,j)$ 表示数据对象 x_i 和 x_j 之间的欧氏距离，d_c 是需要人为提前确定的截断距离参数，当 $u \geq 0, \chi(u)=1$；当 $u<0, \chi(u)=0$。该公式也被称为截断核密度公式。除了截断核密度公式之外，还提出了高斯核密度公式，定义为：

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

公式（2）计算的是所有数据对象到 x_i 的高斯核函数值之和。当数据集规模较大时一般使用截断核相对密度公式，数据集规模较小时则使用高斯核相对密度公式，d_c 值一般是基于数据集中数据对象分布距离的 1%到 4%设定的。

数据集中每个数据对象到其密度更高的且最近的数据对象的相对距离 δ_i 定义为：

$$\delta_i = \min_{x_j:\rho_j>\rho_i} \left(\mathrm{dist}(i,j)\right) \tag{3}$$

公式（3）表示，对于数据对象 x_i 先找到所有密度比它大的数据对象，然后在这些数据对象里找到距离 x_i 最近的数据对象 x_j，计算二者的欧式距离即为该数据对象相对距离。

对于数据集里相对密度最大的数据对象 x_i，相对距离 $\delta_{\max}$ 的计算公式定义为：

$$\delta_{\max} = \max(\delta_i) \tag{4}$$

DPC 算法认为当相对密度 ρ_i 和相对距离 δ_i 的值都较大时，对应的数据对象被选取为簇中心（即密度峰值点）。之后将剩余非中心点分配至距离最近且密度比其大的簇，从而完成聚类任务。

2 优化算法 DPC-NN-WSNN

2.1 基于自然和共享最近邻的加权相对密度

K 最近邻算法因其简单无需先验知识被广泛应用于各种科学领域。但是其缺点是算法结果受 K 值

影响非常大，实验时难以选择合适的 K 值。为了找到最合适的 K 值，Zhu 等人[14]提出了自然最近邻的概念，该定义自适应地根据数据对象的特点寻找自身的自然最近邻，无需设置参数值 K 就能自适应找到每个数据对象的自然最近邻，可以降低参数值对聚类结果产生的影响。

定义 1 如果点 x_i 是点 x_j 的邻居，同时点 x_j 是点 x_i 的邻居，此时称 x_i 与 x_j 互为彼此的自然最近邻，则 x_i 的自然最近邻 $NN(x_i)$ 定义为：

$$x_j \in NN(x_i) \Leftrightarrow (x_i \in KNN(x_j)) \cap (x_j \in KNN(x_i)) \quad (5)$$

自然最近邻搜索算法：K 最近邻居集的 K 值从 1 开始增加，每次加 1。当所有数据集中所有数据对象都拥有了至少一个确定的自然最近邻时，则此时数据集达到了自然稳定状态。记下 K 值，依据此 K 值，根据公式（5）即可自动得到每个数据对象的自然最近邻集。

基于自然最近邻的思想，将提出自然最近邻权重值的定义。

定义 2 数据对象 x_i 的自然最近邻权重值记作 $W_n(i)$，定义为：

$$W_n(i) = \frac{\sum_{m \in NN(i)} \mathrm{dist}(i,m)}{\left|NN(i)\right|} \quad (6)$$

公式（6）中，$\left|NN(i)\right|$ 为点 x_i 的自然最近邻的个数。自然最近邻权重值无需参数值 K 就能根据每个数据对象的自然最近邻计算其参数值，对于处于密度较稀疏的簇的数据对象来说权重值会较大，对于密度较大的簇中的数据对象其权重值则会较小，该特性很好地对稀疏簇的数据对象进行了补偿，从而得到更真实的相对密度和相对距离值。

Liu 等人在 SNN-DPC 算法中提出了共享最近邻的概念，其通过共享最近邻来定义两个点之间的相似度，再通过点的邻居里最大的前 K 个共享最近邻居相似度之和来定义该数据对象的相对密度值。

定义 3 数据对象 x_i 和 x_j 的一阶共享最近邻相似度记作 $Sim(i,j)$，定义为：

$$Sim(i,j) = \frac{\left|SNN(i,j)\right|^2}{\sum_{p \in SNN(i,j)} \left(\mathrm{dist}(p,i) + \mathrm{dist}(p,j)\right)} \quad (7)$$

公式（7）中，$\left|SNN(i,j)\right|$ 为 x_i 和 x_j 的一阶共享最近邻数量，$SNN(i,j)$ 为 x_i 和 x_j 的共享最近邻集合，一阶共享最近邻即 x_i 和 x_j 共同拥有的一阶 K 最近邻，该式基于一阶共享最近邻定义了 x_i 和 x_j 之间的相似度。特别的，当点 x_i 和点 x_j 没有一阶共享最近邻时，一阶共享最近邻相似度将设置为 0。

在计算孤立点的相对密度时，由于孤立点远离簇中心，在有些情况下找不到其一阶共享最近邻也难以计算其一阶共享最近邻相似度，并且一阶共享最近邻的信息在计算过程中是不够精确的，高阶邻居的信息对计算结果也能产生较大的影响。针对一阶共享最近邻的缺陷，本文提出二阶共享最近邻的定义，并且提出基于二阶共享最近邻的相似度定义。

定义 4 数据对象 x_i 和 x_j 的二阶共享最近邻记作 $SSNN(i,j)$，定义为：

$$SSNN(i,j) = KNN_2(i) \cap KNN_2(j) \quad (8)$$

公式（8）中，$KNN_2(i)$ 是数据对象 x_i 的二阶邻居集合，即 x_i 的 K 最近邻居的 K 最近邻居集合，$KNN_2(j)$ 同理。

定义 5 数据对象 x_i 和 x_j 的二阶共享最近邻相似度记作 $Ssim(i,j)$，定义为：

$$Ssim(i,j) = \frac{\left|SSNN(i,j)\right|^2}{\sum_{p \in SSNN(i,j)} \left(\mathrm{dist}(p,i) + \mathrm{dist}(p,j)\right)} \quad (9)$$

公式（9）中，$SSNN(i,j)$ 是点 x_i 和点 x_j 共同拥有的二阶邻居集合，$\left|SSNN(i,j)\right|$ 为点 x_i 和点 x_j 的二阶共享最近邻数量。特别的，当点 x_i 和点 x_j 没有二阶共享最近邻时，二阶共享最近邻相似度设置为 0。

定义 6 数据对象 x_i 和 x_j 的融合相似度记作 $Fsim(i,j)$，定义为：

$$Fsim(i,j) = \alpha Sim(i,j) + (1-\alpha) Ssim(i,j) \quad (10)$$

公式（10）中，α 为可以调整的平衡值。考虑到一阶共享最近邻相似度对数据对象的相似度贡献值较大，后续实验将 α 的值设置为 0.8。该公式计算了数据对象的融合相似度，其中包括了多阶的融合共享最近邻信息，对于数据集中的边界点或者孤立点，有效地提升了其相似度的计算精度。

定义 7 数据对象 x_i 的相对密度记作 $\rho_r(i)$，定义为：

$$\rho_r(i) = \sum_{j \in L(i)} Fsim(i,j) * W_n(i) \quad (11)$$

公式（11）中，$L(i)$ 是与数据对象 x_i 的融合相似度最高的 K 个点的集合，该式表示对于数据对象 x_i 计算其他所有点与它的融合相似度，找到最高的 K 个点然后相加，就能得到融合相似度之和，再乘上该点的自然最近邻权重值 $W_n(i)$，即为该数据对象的相对密度。

本文提出的相对密度 ρ_r 能够自适应的根据簇的稀疏程度通过自然最近邻权重值调整数据对象的相对密度值，对于处于稀疏簇的数据对象，其相对密度值会因此放大，得到更为真实的相对密度值。

2.2 基于自然最近邻的加权相对距离

在 DPC 算法中，相对距离 δ_i 的计算方式是数据对象 x_i 到密度比它大的数据对象的最短距离，但是这样计算忽略了数据对象的周围环境信息，对于密度稀疏的点没有权重补正[15]，因此本小节提出基于自然最近邻的加权相对距离公式。

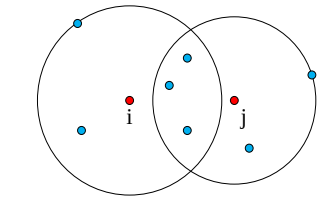
定义 **8** 数据对象 x_i 的加权相对距离公式记作 $\delta_n(i)$，定义为：

$$\delta_n(i)=\left[\min_{j:\rho_j>\rho_i}\operatorname{dist}(i,j)\right]*\left(W_n(i)+W_n(j)\right) \quad (12)$$

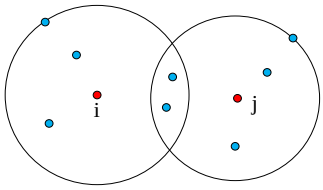
公式（12）对于稀疏簇来说，依然会有效放大其相对距离值，有助于识别正确地簇中心[16]。

2.3 分类型簇中心扩散策略

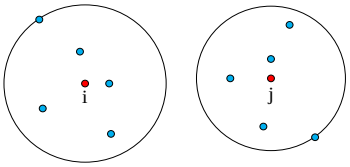
传统DPC算法的分配过程只要有一个标签分配错误，后面就会出现链式错误。针对这个问题，本小节中提出了分类型簇中心扩散策略[17]，该策略基于图1对从属点的解释图，对稀疏和密集簇中心点进行区分，设计了不同的识别从属点阈值。



(a) Inevitable Subordinate Point



(b) Possible Subordinate Point



(c) Impossible Subordinate Point

图 1 对于从属点的解释图

Fig. 1 Explanation of subordinate points

定义 9 必然从属点（Inevitable Subordinate Point）：假设数据对象 x_i 的标签已经被分配，当且仅当 x_j 满足如下条件时：

$$\left|SNN(i,j)\right|\geq\frac{k}{n} \quad (13)$$

x_j 是 x_i 的一个必然从属点。

定义 10 可能从属点（Possible Subordinate Point）：假设数据对象 x_i 的标签已经被分配，当且仅当 x_j 满足如下条件时：

$$0<\left|SNN(i,j)\right|<\frac{k}{n} \quad (14)$$

x_j 是 x_i 的一个可能从属点。

定义 11 不可能从属点（Impossible Subordinate Point）：假设数据对象 x_i 的标签已经被分配，当且仅当 x_j 满足如下条件时：

$$\left|SNN(i,j)\right|=0 \quad (15)$$

x_j 是 x_i 的一个不可能从属点。

公式（13）、（14）、（15）中，k 是最近邻的数量，n 是任意正整数（后续实验取值为 2 或者 3），如果数据对象 x_i 和 x_j 共享最近邻数量高于阈值 $\frac{k}{n}$，那么 x_j 是 x_i 的必然从属点，x_j 和 x_i 的标签将会保持一致，反之 x_j 是 x_i 的可能从属点。特别的，共享最近邻数量为 0 时，x_j 是 x_i 的不可能从属点。

对于密度较高的簇，数据对象分布更密集，会识别出更多的共享最近邻，所以在进行簇中心扩散分配策略时，处于密度较高簇的中心点更容易识别出更多的从属点，从而影响标签分配的准确率。针对这个问题，提出两个定义对两种密度差别较大的簇中心进行分类，然后分类使用不同的从属点的判断阈值以解决这一问题[18]。

定义 12 相对高密度簇中心：假设数据对象 x_i 是被决策图选取出的簇中心，当且仅当 x_i 满足如下条件时：

$$0\leq W_n(i)<\frac{\sum_{p\in L(c_i)}W_n(p)}{\left|L(c_i)\right|} \quad (16)$$

x_i 是相对高密度簇中心。

定义 13 相对低密度簇中心：假设数据对象 x_i 是被决策图选取出的簇中心，当且仅当 x_i 满足如下条件时：

$$W_n(i)\geq\frac{\sum_{p\in L(c_i)}W_n(p)}{\left|L(c_i)\right|} \quad (17)$$

x_i 是相对低密度簇中心。

公式（16）和（17）中，$L(c_i)$ 是所有被识别出的簇中心集合，右式计算的是簇中心的自然最近邻权重的平均值，因为自然最近邻权重越大，密度相

对越低，所以所有簇中心里自然最近邻权重值大于平均值的簇中心都视为相对低密度簇中心，反之为相对高密度簇中心。对于密度较低的簇中心，设置更大的 n 值，从而减小阈值，保证低密度簇在进行标签分配时也能识别更多的必然从属点，从而可以进行更精确的点的分配[19]。

分类型簇中心扩散策略如下：

输入：数据集 X 的初始簇中心 $C(X)=\{c_1,c_2,\ldots,c_m\}$；

Step1：初始化创建队列 Q，将 $C(X)$ 里的簇中心索引加入队列中；

Step2：检查所有簇中心的自然最近邻权重值，其权重值小于 $\dfrac{\sum_{p\in L(c_i)} W_n(p)}{|L(c_i)|}$ 的簇中心设置为相对高密度簇中心，反之则设置为低密度簇中心；

Step3：对于队列中的相对高密度簇中心 n 设定为 2，低密度簇中心 n 设定为 3，并找到其邻居点 x_i 并检查其是否已经被分配过标签且和其一阶共享最近邻的数量是否大于 $\dfrac{k}{n}$，如果都满足则将 x_i 分配给该簇，并且将 x_i 加入队列 Q，并且重复 Step3 迭代扩展队列 Q；

Step4：初始化创建数组 A，将未被分配的点的索引加入数组中，检查剩下未被分配的点的邻居点的分配情况，查询每个簇在邻居点里的数量，找到数量最多的簇，将其分配到数量最多的簇里；

Step5：更新数组 A，去除已经分配的点，如果仍然有未被分配的点，重复 Step4；

Step6：如果依然有未被分配的点，则 $k+1$；

输出：聚类结果 $\phi=\{C_1,C_2,\ldots,C_m\}$（$m$ 为簇的个数）。

2.4 DPC-NN-WSNN 算法步骤

DPC-NN-WSNN 算法步骤如下：

算法输入：数据集 $X=\{x_1,x_2,\ldots,x_n\}$，$\alpha=0.8$，高密度聚类中心的 n 值设置为 2，低密度聚类中心的 n 值设置为 3；

Step1：对数据集 X 中的原始数据执行归一化；

Step2：计算 n 个数据对象之间的距离矩阵 $D^{n\times n}$；

Step3：利用自然最近邻的概念计算每个数据对象的自然最近邻权重值 W_n；

Step4：根据公式（11）和公式（12）计算每个数据对象的相对密度 ρ_r 和相对距离 δ_n；

Step5：通过新的密度和相对距离公式计算出的 ρ_r 和 δ_n 的决策图找到合理的簇中心；

Step6：调用算法 2 的分类型簇中心扩散策略完成标签分配；

输出：最终聚类结果 $\phi=\{C_1,C_2,\ldots,C_m\}$。

3 对比实验

为了验证本文提出的 DPC-NN-WSNN 算法的有效性，将在 6 个数据集上进行对比实验。这些数据集具有凸型、球形、均匀密度和变密度等特征。与此同时，还将选取 DBSCAN、DPC 和 SNN-DPC 这 3 个聚类算法作为对比实验对象。表 1 中提供了这 6 个数据集的基本信息。在本次实验中，将采用 3 个常用的聚类算法评价指标进行算法性能评估：调整后的互信息（AMI）、调整后的兰德系数（ARI）和 Fowlkes-Mallows 指标（FMI）。

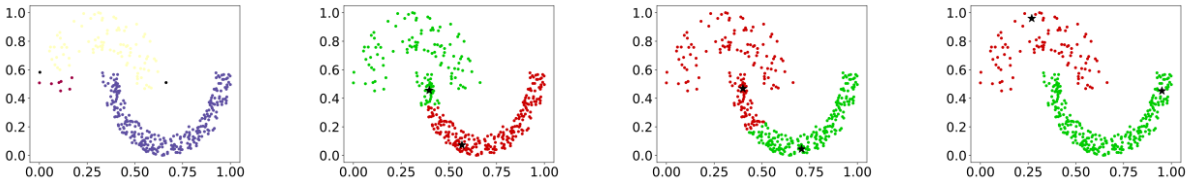
表 1 实验数据集

Tab.1 The experimental datasets

	Datasets	Capacity	The number of clusters
datasets	Jain	373	2
	Spiral	312	3
	Pathbased	300	3
	Compound-2	349	5
	Zelnik6	238	3
	Aggregation	788	7

3.1 实验结果

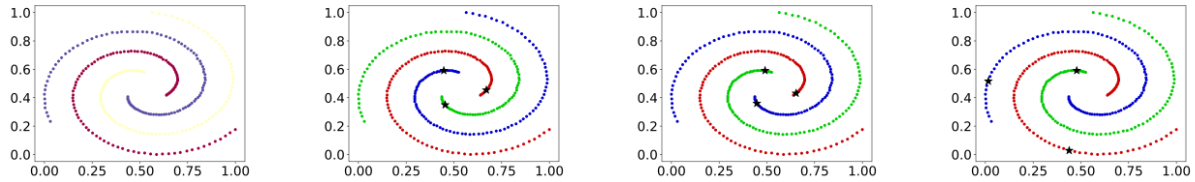
本小节将展示本文算法 DPC-NN-WSNN 在具有复杂结构和形状的数据集上相较于 DBSCAN，DPC，SNN-DPC 算法的优势，图 2 至图 7 分别可视化了在 6 种数据集上聚类算法的结果图，表 2 列出了 6 组对比实验的聚类评价指标。



(a) DBSCAN (b) DPC (c) SNN-DPC (d) DPC-NN-WSNN

图 2 4 种算法在 Jain 的聚类结果对比

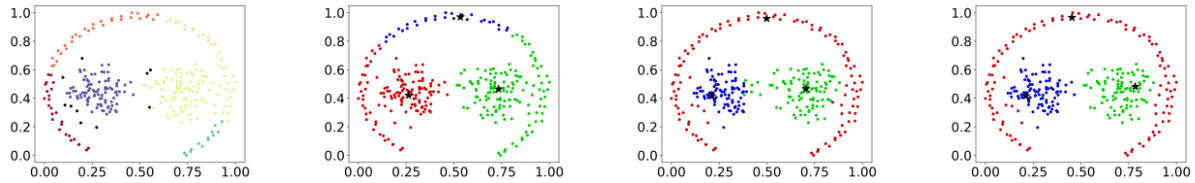
Fig. 2 The comparison of clustering results of 4 algorithms in Jain



(a) DBSCAN (b) DPC (c) SNN-DPC (d) DPC-NN-WSNN

图 3　4 种算法在 Spiral 的聚类结果对比

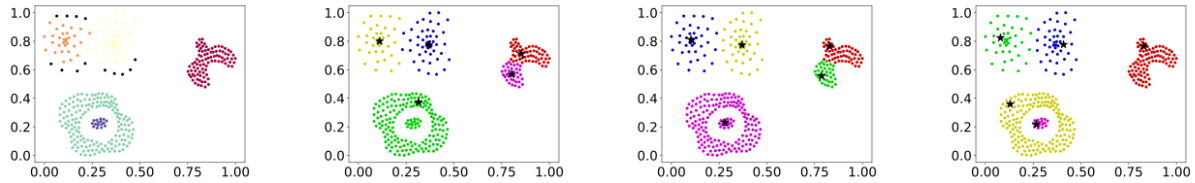
Fig. 3 The comparison of clustering results of 4 algorithms in Spiral



(a) DBSCAN (b) DPC (c) SNN-DPC (d) DPC-NN-WSNN

图 4　4 种算法在 Pathbased 的聚类结果对比

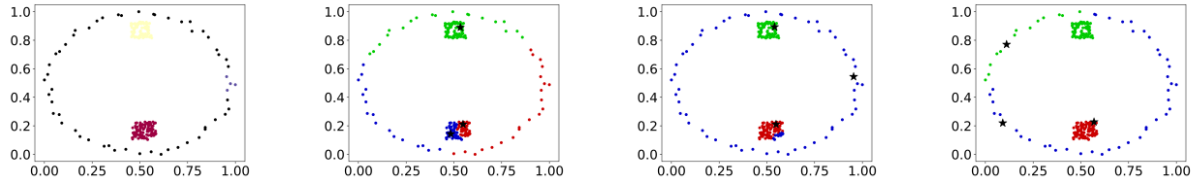
Fig. 4 The comparison of clustering results of 4 algorithms in Pathbased



(a) DBSCAN (b) DPC (c) SNN-DPC (d) DPC-NN-WSNN

图 5　4 种算法在 Compound-2 的聚类结果对比

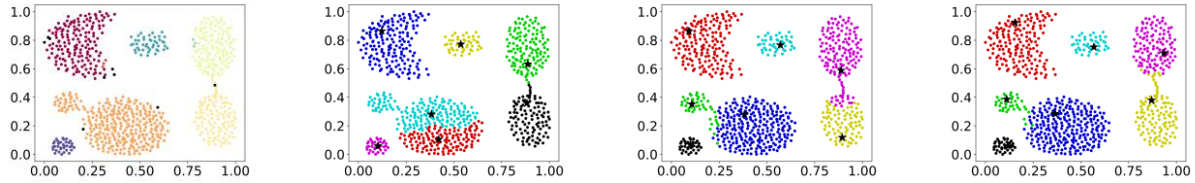
Fig. 5 The comparison of clustering results of 4 algorithms in Compound-2



(a) DBSCAN (b) DPC (c) SNN-DPC (d) DPC-NN-WSNN

图 6　4 种算法在 Zelnik6 的聚类结果对比

Fig. 6 The comparison of clustering results of 4 algorithms in Zelnik6



(a) DBSCAN (b) DPC (c) SNN-DPC (d) DPC-NN-WSNN

图 7　4 种算法在 Aggregation 的聚类结果对比

Fig. 7 The comparison of clustering results of 4 algorithms in Aggregation

表 2　4 种算法在 6 个数据集的评价指标对比

Tab.2 The Comparison of evaluation indicators of 4 algorithms in 6 datasets

Datasets	Algorithms	AMI	ARI	FMI
Jain	DBSCAN	0.9276	0.9758	0.9906
	DPC	0.5761	0.6183	0.8386
	SNN-DPC	0.4299	0.3856	0.7259
	DPC-NN-WSNN	**1.0000**	**1.0000**	**1.0000**
Spiral	DBSCAN	**1.0000**	**1.0000**	**1.0000**
	DPC	**1.0000**	**1.0000**	**1.0000**
	SNN-DPC	**1.0000**	**1.0000**	**1.0000**
	DPC-NN-WSNN	**1.0000**	**1.0000**	**1.0000**
Pathbased	DBSCAN	0.6905	0.6444	0.7542
	DPC	0.5360	0.4530	0.6585
	SNN-DPC	0.8979	0.9077	0.9103
	DPC-NN-WSNN	**0.9024**	**0.9294**	**0.9533**
Compound-2	DBSCAN	0.9416	0.9738	0.9817
	DPC	0.8607	0.8188	0.8746
	SNN-DPC	0.8775	0.8242	0.8783
	DPC-NN-WSNN	**1.0000**	**1.0000**	**1.0000**
Zelnik6	DBSCAN	**1.0000**	**1.0000**	**1.0000**
	DPC	0.4909	0.4564	0.6442
	SNN-DPC	0.8736	0.8743	0.9173
	DPC-NN-WSNN	0.8566	0.8661	0.9143
Aggregation	DBSCAN	0.9344	0.9417	0.9436
	DPC	0.8921	0.7550	0.8074
	SNN-DPC	0.9314	0.9272	0.9428
	DPC-NN-WSNN	**0.9548**	**0.9594**	**0.9681**

3.2 分析与评估

Jain 数据集是典型的密度分布不均匀的数据集，本文提出的 DPC-NN-WSNN 对稀疏簇中心的识别进行了有效地改进，因此在密度分布不均匀的 Jain 数据集上该算法取得了最好的效果，并且效果显著优于其他三个算法。

Spiral 数据集由三个螺旋形的非线性簇组成，簇形状较为明显，也没有明显的孤立点。本文提出的 DPC-NN-WSNN 和 DBSCAN、DPC 和 SNN-DPC 这三种算法一同取得了最优的聚类结果，说明该算法在改进之后也很好的保留了原 DPC 算法可以准确识别形状明显的数据集的优点。

Pathbased 数据集由中间的两个类球形簇和边缘的半环形簇组成，但是簇间有明显连接部分，且有少数数据对象是较远的边界点。在该数据集上 DPC-NN-WSNN 成为效果最好的算法，因为其融合了多阶邻居信息，并且加入了孤立点信息，效果显著优于原 DPC 算法。

Compound-2 数据集由五个簇组成，存在嵌套和连接部分，且簇的密度不是均匀分布的。本文提出的 DPC-NN-WSNN 对密度的定义里加入了多层次的共享最近邻信息，并采用分类型簇中心扩散策略，因此对连接部分和嵌套部分都进行了几乎准确的识别，效果在四个算法中最好。

Zelnik6 数据集由一个环形簇加两个方形簇组成，两个方形簇密度较高，环形簇密度较低，有少数较远边界点。本文提出的 DPC-NN-WSNN 未能识别出准确的簇中心，环形簇上半区域的一部分被分配给了方形簇，聚类效果显著降低。这个数据集里，反而效果最好的是 DBSCAN 算法。

Aggregation 数据集由七个簇组成，密度较为均匀，但是其中有两组两两相连的簇，且不同簇数据对象数量和密度差距较大，无较远边界点和孤立点。DPC-NN-WSNN 由于其对稀疏簇的补偿机制依然以微弱优势取得了最好的效果。

根据可视化的聚类结果图和聚类评价指标的表格，本文提出的优化算法 DPC-NN-WSNN 在其中五个数据集上有着相等甚至更好的聚类效果，只在一个数据集 Zelnik6 上差于 DBSCAN，而在簇间密度差异较大的 Compound-2 和 Jain 数据集上聚类准确性有着显著提升。故相比于其余三个聚类算法，本文提出的基于自然和加权共享最近邻的密度峰值聚类算法不仅能发现并利用数据集潜在的结构和形状信息，避免了稀疏簇的遗漏，同时也能更为准确地识别簇中心并且完成聚类后续标签的分配。

4 总结

（1）为了使 DPC 算法能在识别不同形状和不同密度的簇时有更好的表现，在传统 DPC 算法的基础上提出了一种基于自然和加权共享最近邻的密度峰值聚类算法；

（2）实验结果表明，DPC-NN-WSNN 算法在应对复杂形状和簇间稀疏程度差异大的数据上具有明显的优势，聚类精度更高；

（3）但是该算法在处理包含噪声点和识别簇间有连接部分的数据集时聚类效果有所下降，如何去除噪声点的影响和对连接部分的数据点进行准确的分配可以作为未来继续探索研究的方向[20]。

参考文献

[1] 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.DOI:10.16749/j.cnki.jecjtu.20230209.006
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.

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

[3] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]//kdd. 1996, 96(34): 226-231.

[4] Guha S, Rastogi R, Shim K. Cure: an efficient clustering algorithm for large databases[J]. Information systems, 2001, 26(1): 35-58.

[5] Wang W, Yang J, Muntz R. STING: A statistical information grid approach to spatial data mining[C]//Vldb. 1997, 97: 186-195.

[6] Shi J, Malik J. Normalized cuts and image segmentation[J]. IEEE Transactions on pattern analysis and machine intelligence, 2000, 22(8): 888-905.

[7] Dempster A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm[J]. Journal of the royal statistical society: series B (methodological), 1977, 39(1): 1-22.

[8] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.

[9] Yaohui L, Zhengming M, Fang Y. Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy[J]. Knowledge-Based Systems, 2017, 133: 208-220.

[10] Guo Z, Huang T, Cai Z, et al. A new local density for density peak clustering[C]//Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III 22. Springer International Publishing, 2018: 426-438.

[11] Cheng D, Zhang S, Huang J. Dense members of local cores-based density peaks clustering algorithm[J]. Knowledge-Based Systems, 2020, 193: 105454.

[12] 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.

[13] 王芙银，张德生，肖燕婷 .基于加权共享近邻与累加序列的密度峰值算法［J］.计算机工程，2022，48（4）：61-69.
WANG F Y，ZHANG D S，XIAO Y T. Density peak algorithm based on weighted shared nearest neighbor and accumulated sequence［J］.Computer Engineering，2022，48（4）：61-69.

[14] Zhu Q, Feng J, Huang J. Natural neighbor: A self-adaptive neighborhood method without parameter K[J]. Pattern Recognition Letters, 2016, 80: 30-36.

[15] 徐晓，丁世飞，丁玲，密度峰值聚类算法研究进展.软件学报，2022，33(5): 1800－1816.
Xu X，Ding S F，Ding L. Survey on Density Peaks Clustering Algorithm. Ruan Jian Xue Bao/Journal of Software,2022, 33(5)：1800–1816 (in Chinese).

[16] 孙林，秦小营，徐久成，等。基于 K 近邻和优化分配策略的密度峰值聚类算法[J]. 软件学报, 2022, 33(04): 1390-1411.
SUN L, QIN X Y, XU J C, et al. Density peak clustering algorithm based on K-nearest neighbors and optimized allocation strategy[J]. Journal of Software, 2022, 33(04): 1390-1411.

[17] 吕莉, 朱梅子, 康平, 等. 面向密度分布不均数据的混

合近邻密度峰值聚类算法. 控制理论与应用, 2023, 40
LYULi, ZHU Meizi, KANG Ping, et al. Multiplex neighbor density peaks clustering for uneven density data sets. Control Theory & Applications, 2023, 40.

[18] 丁志成, 葛洪伟. 优化分配策略的密度峰值聚类算法. 计算机科学与探索, 2020, 14(05): 792 – 802.
DING Zhicheng, GE Hongwei. Density peaks clustering with optimized allocation strategy. Journal of Frontiers of Computer Science and Technology, 2020, 14(05): 792-802.

[19] 赵嘉，姚占峰，吕莉，等。基于相互邻近度的密度峰值聚类算法[J]. 控制与决策, 2021, 36(3): 543-552.
ZHAO J, YAO Z F, LV L, et al. Density peaks clustering based on mutual neighbor degree[J]. Control and Decision, 2021, 36(3): 543-552.

[20] 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.控制与决策: 1–9[2022–11–05]. DOI:10.13195/j.kzyjc. 2021.1286.
WU Rubxiu, YING Shihao, ZHAO Jia, et al. Density peaks clustering based on relative density estimating and multi cluster merging. Control and Decision, 1-9[2022-11-05]. DOI:10.13195/j.kzyjc.2021.12 86.

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

通信作者：陈翔（1998—），男，硕士研究生，研究方向为聚类分析与数据挖掘。
E-mail: 2022088085410004@ecjtu.edu.cn。