doi:10.3772/j.issn.1002-0470.2023.01.009

基于数据相似度和引力理论的密度峰聚类算法①

詹 康② 王逸文 何熊熊

(浙江工业大学信息工程学院 杭州 310023)

摘 要 本文针对密度峰聚类算法(DPC)中存在的参数敏感、算法不连续和聚类分块化问题,提出一种基于数据相似度和引力理论的密度峰聚类算法(SLDPC)。该算法基于数据相似度确定局部密度,采用引力理论扩大簇心与非簇心数据点之间的差异,通过设定阈值自动确定簇心,通过基于边缘分布的合并策略对聚类分块化进行合并。实验共采用 16 个数据集,并与 DPC、K-means、基于密度的噪声应用空间聚类算法(DBSCAN)及 DPC 改进算法进行了对比。实验结果表明,本方法具有优异的聚类准确性和良好的稳定性。

关键词 聚类分析; 密度峰值; 数据相似度; 引力理论; 聚类合并

0 引 言

聚类分析是一种基本的数据分析方法,它可以发现数据的内在隐藏模式,在图像处理、网络安全、模式识别、生物信息学、微阵列分析和社交网络等多个领域有着广泛的应用[1-3]。

K-means[4]作为常用的聚类算法,在大部分数据集上都能取得较好的聚类效果,且算法实现较容易,但 K-means 需要聚类个数作为算法输入,且多次执行 K-means 可能产生不一样的聚类结果。基于密度的噪声应用空间聚类算法(density-based spatial clustering of applications with noise, DBSCAN)[5]能够识别任意形状的簇且能够抑制噪声点的影响,但 DBSCAN 中存在 2 个参数:半径 *eps* 和密度阈值 *MinPts*,参数的设定对于算法有较大的影响。

文献[6]提出了密度峰聚类(density peak clustering, DPC)算法,该算法不仅能够在没有先验知识的情况下识别出聚类数量,而且能够识别出任意形状的簇。

尽管 DPC 算法具有计算简单、快速的优势,但还是存在着一些不足[7-9]。(1)数据对象属性值的计算依赖于截断距离 d_c 的设定;(2)人工选取簇心的聚类结果受主观性的影响,并且在数据点相似的情况下难以正确确定簇心;(3)对于一些多密度峰的数据集,容易造成聚类分块化的问题。

为了进一步提升 DPC 算法的聚类效果,国内外学者提出了许多 DPC 改进算法,改进方向主要为计算局部密度、确定簇心以及分块化数据的合并。文献[10-16]基于 K 最近邻(K-nearest neighbor,KNN)对 DPC 算法进行了改进,避免了截断距离 d_c 的计算,但却引入了另一个参数 K,不同的 K 值对聚类结果有着不同的影响。文献[17-19]通过不同的方法来自动确定簇心,但在面临数据点相似的情况时也很难正确确定簇心。文献[20-22]分别使用了数据点之间的独立性、簇间相似性关系及密度共享区域来确定两个子集能否进行合并,但这些方法通常需要考虑子集中的所有数据点,具有较高的复杂度。

基于上述讨论,本文提出一种基于数据相似度和引力理论的密度峰聚类算法(L1-norm based data similarity density peak clustering, SLDPC)。首先,考虑数据点之间的差异性以及全局数据点对于密度的

① 国家自然科学基金(61873239,61675183)和浙江省重点研发计划(2020C03074)资助项目。
② 男,1996 年生,硕士生;研究方向:数据挖掘,推荐系统;联系人,E-mail: 826825031@qq.com。
(收稿日期:2021-04-15)

影响,采用基于 L1 范数的数据相似度对数据点的局部密度进行计算。其次,采用基于引力理论的簇心评价方法扩大簇心与非簇心数据点之间的差异,并通过设定阈值自动确定簇心。然后,通过基于边缘分布的合并策略对聚类分块化进行合并。最后,通过实验验证了本文所提方法的有效性。

1　SLDPC 算法

1.1　基于数据相似度的无参数密度计算方法

在 DPC 算法中,计算 ρ 和 δ 的关键在于截断距离 d_c 的设定,合适的 d_c 可以更好地反映数据分布的真实情况,使算法效果达到最好。相反,不合适的 d_c 会使算法产生不良的效果。因此,解决参数敏感问题对于整个算法性能的提升至关重要。

L1 范数是指向量中各个元素的绝对值之和,也称为“稀疏规则算子”。在模型拟合中,L1 范数误差的线性增长速度使其对大噪声不敏感,从而能够对不良作用进行抑制。将 L1 范数应用到聚类中,可以降低噪声对于数据点密度的影响,并且能够保证每个数据点对密度的贡献都为正。以数据点之间的相似度作为其余数据点对当前数据点密度贡献的权重,可以更好地体现数据之间的差异性。

基于 L1 范数确定数据相似度为

$$s_{XY} = \frac{\sum_i \max(|x_i|, |y_i|)}{(\|X\|_1 + \|Y\|_1)/2} \tag{1}$$

其中,$|x_i|$、$|y_i|$ 为数据点 X、Y 中第 i 个元素的绝对值,$\|X\|_1$、$\|Y\|_1$ 为数据点 X、Y 的 L1 范数。

将 L1 范数用于相似度度量可以避免一些无用特征对于相似度的影响。相比基于 L2 范数的余弦相似度,基于 L1 范数的相似度度量是在不执行任何乘法的情况下进行计算,属于低功率相似性度量方法[23]。

用数据点之间的相似度作为数据点密度计算的权重,最终密度计算公式为

$$\rho_i = \sum_{j=1}^{n} e^{-s_{ij} \times d_{ij}} \tag{2}$$

其中,s_{ij} 为数据集中第 i 个数据点和第 j 个数据点之间的相似度,d_{ij} 为 2 个数据点之间的欧氏距离,n 为数据集中数据点的数量。

DPC 算法在 d_c 不同的情况下会产生不同的聚类结果,而对比式(2)和 DPC 密度计算方法可以发现,本文所提密度计算方法无需任何参数的输入,因此聚类效果不受参数影响,使得整个算法具有更好的稳定性。

1.2　基于引力理论的簇心评价方式

人工选取簇心会破坏算法的连续性,且在数据点相似的情况下难以正确地确定簇心。为了解决此问题,本文提出一种基于引力理论的簇心评价方法。

根据牛顿第三定律,2 个物体之间的吸引力可以表示为

$$\vec{\boldsymbol{F}}_{ij} = G\frac{m_i m_j}{D_{ij}^2}\vec{\boldsymbol{D}}_{ij} \tag{3}$$

其中 $\vec{\boldsymbol{D}}_{ij}$ 为吸引力方向的单位向量,m_i、m_j 为物体的质量,D_{ij} 为两个物体之间的距离。当 2 个物体之间的距离在某个局部区域内没有明显变化时,吸引力可以简化为

$$\vec{\boldsymbol{F}}_{ij} = Gm_i m_j\vec{\boldsymbol{D}}_{ij} \tag{4}$$

物体与其 k 近邻之间的合力为邻域内其他物体对其吸引力的矢量和,具体表示为

$$\overrightarrow{\boldsymbol{LRF}}(i, k) = \sum_{j=1}^{k}\vec{\boldsymbol{F}}_{ij} = Gm_i\sum_{j=1}^{k}m_j\vec{\boldsymbol{D}}_{ij} \tag{5}$$

一般情况下,合力的大小与物体的质量成反比,质量大的点对相邻点的影响较大,而质量小的点受相邻点的影响较大[24]。因此,式(5)可以简化为

$$\overrightarrow{\boldsymbol{LRF}}(i, k) = \frac{1}{m_i}\sum_{j=1}^{k}\vec{\boldsymbol{D}}_{ij} \tag{6}$$

其中 m_i 定义为[24]

$$m_i = \frac{1}{\sum_{j=1}^{k}D_{ij}} \tag{7}$$

如图 1 所示,当近邻的数量 k 变化时,数据点的合力方向以及大小都会发生不同程度的变化。本文中只考虑合力大小的变化,其变化程度表示为

$$\triangle LRF(i, k) = \|\overrightarrow{\boldsymbol{LRF}}(i,k)\| - \|\overrightarrow{\boldsymbol{LRF}}(i,k+1)\| \tag{8}$$

当 k 达到设定的最大值时,合力大小的变化为每一次 k 变化产生的合力大小变化的总和,即:

$$\theta LRF(i,k) = \sum_{k=1}^{k_{\max}-1}\triangle LRF(i,k) \tag{9}$$

数据集中异常点、边界点以及内部点的合力变化程度都不一样,其中异常点的合力变化程度最

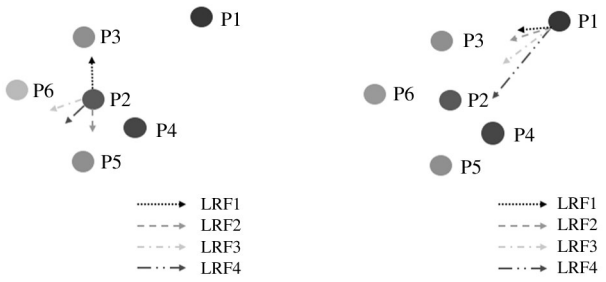


图 1　内部点和异常点在邻域从 1 到 4 的变化情况

大[24]。决策图中的数据点可以被抽象为具体的物体,其质量根据式(7)计算。在决策图中,簇心点一般距离非簇心点较远且分布稀疏,可以被认为是决策图中的异常点。通过对决策图中数据点合力变化程度的分析,选取合力变化大的数据点作为簇心。同时考虑数据点密度的影响,降低密度较小数据点作为簇心的可能性,最终簇心评价方式为

$$\theta LRF(i,k) = \left(\sum_{k=1}^{K_{\max}-1} \triangle LRF(i,k)\right) \times \rho'_i \tag{10}$$

其中 ρ'_i 为归一化后的密度值,具体表示为

$$\rho'_i = \frac{\rho_i - \min(\rho)}{\max(\rho) - \min(\rho)} \tag{11}$$

用式(10)来评估数据点能否作为簇心,设定阈值为 $th = mean(\theta LRF(i,k)) + \alpha \times std(\theta LRF(i,k))$ 来自动提取簇心。其中 $mean(\theta LRF(i,k))$ 为所有数据点 θLRF 值的均值, $std(\theta LRF(i,k))$ 为所有数据点 θLRF 值的方差, α 为可调参数。对于不同的数据集, α 的值不同。

本文所提簇心评价方法能够在数据点相似的情况下扩大数据点之间的差异,也能够对簇心明显的数据集进行正常聚类。

对于 Flame 数据集,在使用本文所提的密度计算方法后,各个数据点的 θLRF 值如图 2 所示。θLRF 值表示该数据点作为簇心的评估分数,分数越大则表示该数据点作为簇心的可能性越大。对于该数据集, θLRF 特别大的 2 个数据点就是簇心,对应图 3 中形状为五角星的数据点。选取这 2 个数据点作为簇心进行聚类,结果如图 3 所示。可以看出该簇心评价方式能够正确地评估一个数据点是否可以作为簇心。

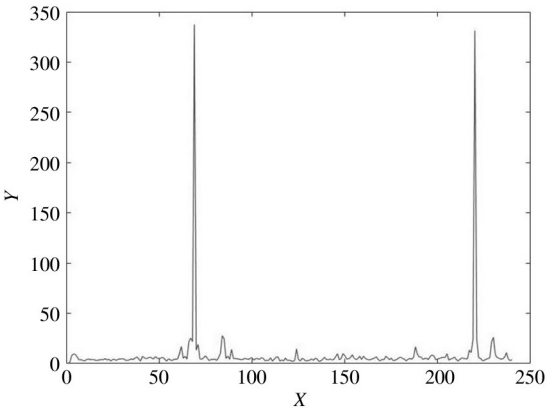


图 2　Flame 中各个数据点的 θLRF 值

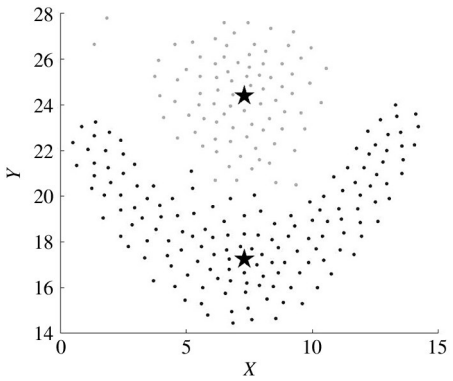


图 3　Flame 的聚类结果

1.3　基于边缘分布的合并策略

当数据集中存在多密度峰时,决策图中会出现多个可能为簇心的数据点,从而导致一个类中存在 2 个或多个高 ρ 值和高 δ 值的数据点。在确定簇心时将这些数据点作为簇心,最终会造成聚类分块化。为了解决此问题,本文提出一种基于边缘分布的合并策略。

定义 1　子类。本应在一个类中,但是由于分块化导致的多个小类。

定义 2　2 个子类之间的边缘数据集合。对于 2 个子类 C_i 和 C_j, 边缘数据集合的计算方式为

$$\forall x \in C_i, y \in C_j, \text{若 } d_{ij} < 0.5 \times d_{m_i m_j}, \text{则 } x, y \in D_{ij} \tag{12}$$

其中, m_i 和 m_j 是第 i 类和第 j 类的簇心。

边缘数据集合的平均密度为

$$\bar{\rho}_{ij} = \frac{\sum_{k \in c_{ij}} \rho_k}{\| D_{ij} \|} \tag{13}$$

其中 $\| D_{ij} \|$ 为该集合中元素的个数。

通过比较 2 个子类之间边缘数据集合的平均密度与 2 个子类簇心的密度是否满足 $\bar{\rho}_{ij} > \beta \times (\rho_{m_i} + \rho_{m_j})$ 来对 2 个子类进行合并。其中 β 为可调参数,取值范围为 0 ~ 1。

对于数据集 Jain,在使用本文提出的密度计算方法后,选取图 4 中 1 ~ 10 号作为簇心,聚类结果如图 5 所示,很明显出现了聚类分块化的问题。使用本文提出的合并策略对子类进行合并,设置 β 为 0.2,如图 6(a)所示,聚类数量由 10 个合并为 2 个,对比图 6(b)正确聚类结果可知本文所提方法能够正确地将子类进行合并。

该合并策略通过 2 个子集边缘数据集合与簇心数据的分布情况进行子类合并,让子类数据合并后更符合原数据分布的规律。同时参数也在较小的范围内,通过调整参数就能满足各种出现聚类分块化数据集的合并需求,从而更好地提升算法性能。

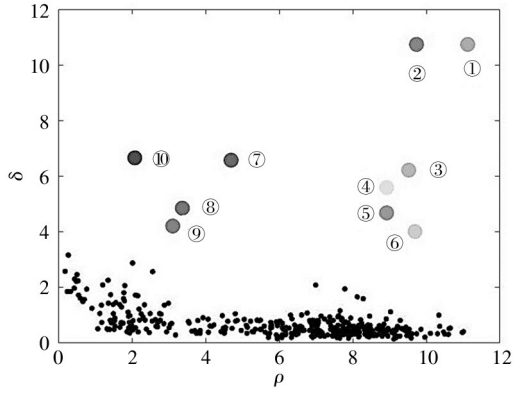


图 4　Jain 的决策图

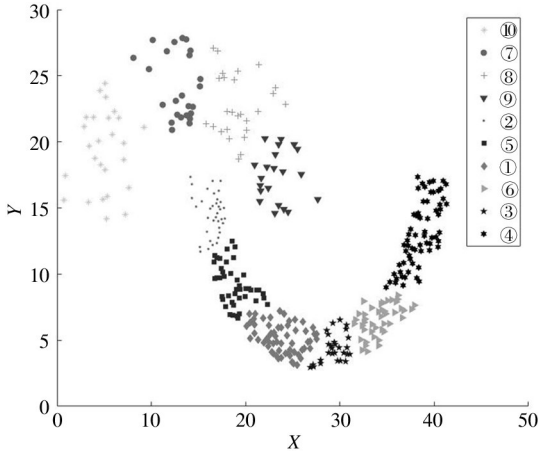
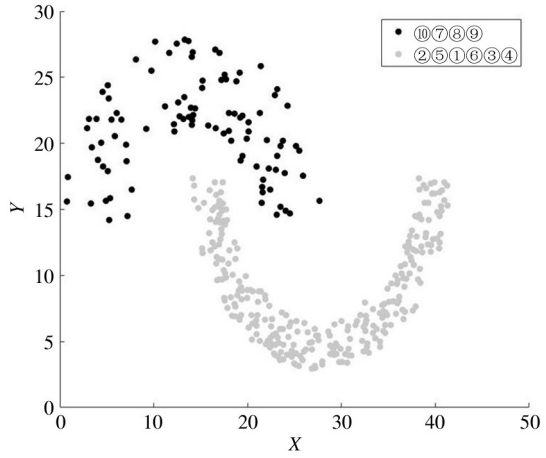
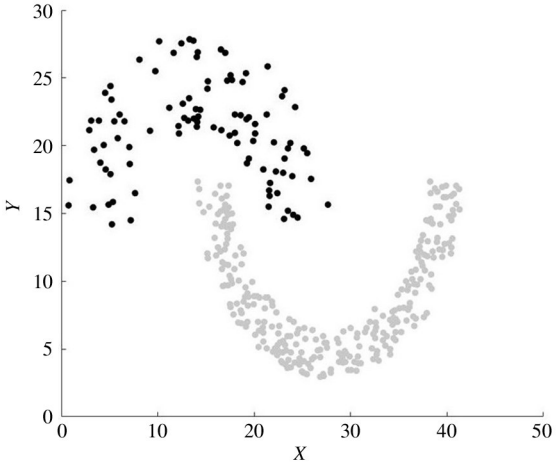


图 5　Jain 的聚类结果



(a) SLDPC 对 Jain 的聚类结果



(b) Jain 的正确聚类结果

图 6　Jain 的聚类结果对比

1.4　SLDPC 算法时间复杂度分析

假设 n 是数据集中所有数据点的数量, m 是子类的数量,SLDPC 的时间复杂度可以分为以下几个部分:(1)计算数据点之间的距离的复杂度为 $O(n^2)$;(2)计算数据点之间相似度的复杂度为 $O(n^2)$;(3)计算数据点 θLRF 值的复杂度为 $O(n^2)$;(4)数据分配过程的复杂度为 $O(n)$;(5)子集合并过程的复杂度为 $O(m^2n^2)$。所以整个算法的时间复杂度为 $O((m^2+3)n^2+n)$。

1.5　SLDPC 算法流程

SLDPC 算法流程图如图 7 所示。首先根据数据点之间的相似度计算每个数据点的属性值 ρ 和 δ,然后计算每个数据点作为簇心的可能性,即 θLRF 值。通过设定阈值自动确定簇心后将剩下非簇心点分配至对应簇中。最后对出现聚类分块化的数据集

进行合并可能性计算。如果满足合并条件,则对相应子类进行合并。

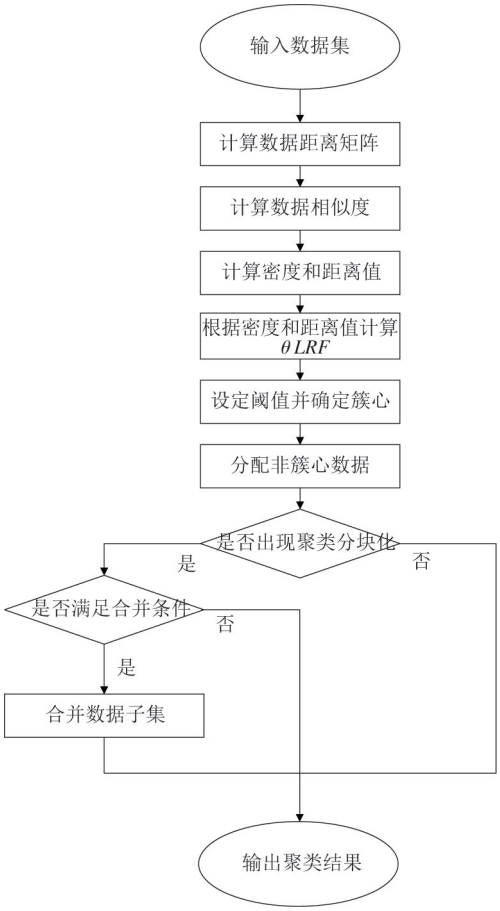


图 7 SLDPC 算法流程图

2 实验结果与分析

本文共采用 16 个人工数据集和真实数据集进行实验,选取 K-mean[4]、DBSCAN[5] 和 DPC[6] 算法进行对比。由于密度估计在整个 DPC 算法中是最重要的一个环节,另取基于加权局部密度序列和最近邻分配的密度峰聚类算法(DPC based on weighted local density sequence and nearest neighbor assignment, DPCSA)[16]进行对比。

本文中采用 2 个聚类评价指标 ACC[25] 和 F1-measure[26]。2 个指标的具体定义如式(14)和式(17)所示。其中,ACC 和 F1-measure 的取值范围均为[0,1],数值越大则聚类质量越高。

$$ACC = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n} \tag{14}$$

$$P = \frac{TP}{TP + FP} \tag{15}$$

$$R = \frac{TP}{TP + TN} \tag{16}$$

$$F1\text{-}measure = \frac{2 \times P \times R}{P + R} \tag{17}$$

为了能够更好地对比实验结果,SLDPC 中的 k 值固定为 50,DPC 中的算法比例 p 设置为 2%,K-means 中 K 值为正确聚类个数,且 DBSCAN 和 K-means 均取最优聚类结果。

2.1 各算法在人工数据集上的实验结果

本节中共选取 8 个人工数据集进行实验来验证 SLDPC 的有效性,数据集的具体信息如表 1 所列。

表 1 实验所用 8 个人工数据集

数据集	大小	维度	类数
Flame	240	2	2
Spiral	312	2	3
Twenty	1000	2	20
Sticks	512	2	4
Square1	1000	2	4
Pathbased	300	2	3
Jain	373	2	2
Agg	788	2	7

图 8 展示了 SLDPC 在部分人工数据集上的聚类结果。从中可以看出,SLDPC 能够正确地处理环形簇、块状簇和非均匀簇。

5 种算法在人工数据集上的聚类指标都汇总于表 2 和表 3 中,可以看出,相比于 DPC、K-means、DBSCAN 和 DPCSA,SLDPC 在人工数据集上都展现了极大的优越性。在数据集 Jain 上,所提取的 2 个簇心实际在一个类中,其他算法无法很好地处理该数据集,但是 SLDPC 将 Jain 数据集的决策图中的所有可能簇心都选取,然后将所有能够合并的子类进行合并,最终达到 100% 的准确率,验证了本文所提合并策略的有效性。在 Sticks 数据集上,由于 DPC 没有考虑数据点的全局分布情况,导致 Sticks 数据集的 4 个簇心分布在 2 个类中,无法很好地做到对 Sticks 数据集的聚类,而 SLDPC 正确地考虑了数据点的分布情况,在选取 4 个簇心完成聚类后的准确率达到了 100%。

2.2　各算法在真实数据集上的实验结果

为了进一步验证算法的有效性,本节中共选取8个真实数据集进行实验,数据集的具体信息如表4所示。

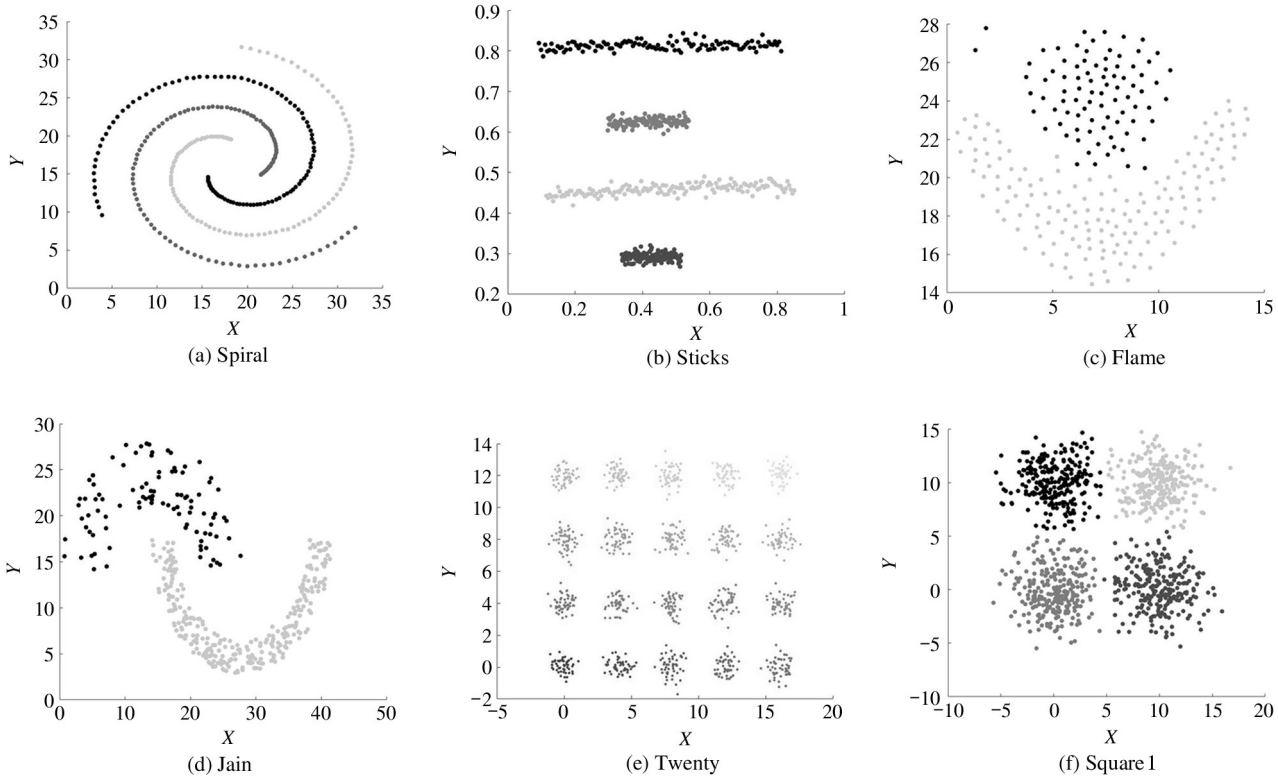


图8　SLDPC在人工数据集上的聚类结果

表2　各算法在人工数据集的性能指标值(ACC)

	SLDPC	DPCSA	DPC	K-means	DBSCAN
Flame	**1.000**	**1.000**	**1.000**	0.842	0.892
Spiral	**1.000**	**1.000**	**1.000**	0.346	**1.000**
Twenty	**1.000**	**1.000**	**1.000**	0.868	**1.000**
Sticks	**1.000**	0.894	0.414	0.721	0.293
Square1	**0.992**	0.989	**0.992**	0.990	0.946
Pathbased	**0.893**	0.823	0.733	0.743	0.613
Jain	**1.000**	0.624	0.861	0.775	0.984
Agg	**0.998**	0.951	**0.998**	0.868	0.977

表3　各算法在人工数据集的性能指标值(F1-measure)

	SLDPC	DPCSA	DPC	K-means	DBSCAN
Flame	**1.000**	**1.000**	**1.000**	0.741	0.931
Spiral	**1.000**	**1.000**	**1.000**	0.328	**1.000**
Twenty	**1.000**	**1.000**	**1.000**	0.885	**1.000**
Sticks	**1.000**	0.851	0.445	0.643	0.401
Square1	**0.984**	0.978	**0.984**	0.980	0.928
Pathbased	**0.881**	0.749	0.654	0.659	0.536
Jain	**1.000**	0.591	0.787	0.686	0.974
Agg	**0.998**	0.958	**0.998**	0.791	0.982

表4　实验所用8个真实数据集

数据集	大小	维度	类数
Iris	150	4	3
Pima	768	8	2
Segmentation	210	19	7
Water	527	38	2
Bupa	345	6	2
Air	359	64	3
Breast	277	9	2
Vote	435	16	2

SLDPC与对比算法在真实数据集上的聚类指标均汇总于表5和表6中。可以看出,相比于DPC、DBSCAN、K-means和DPCSA,SLDPC能够适用于大部分的真实数据集。尽管SLDPC在小部分的数据集中没有DPC以及其他算法的效果好,但也不会产生很差的结果。比如在Vote数据集上,DPC算法能够取得87.5%的准确率,而SLDPC也能够达到86.9%的准确率,仅有0.6%的差距。SLDPC不需要调整参数d_c来达到优异的聚类效果,同时也能自

动提取簇心。由于目前没有一个很好的办法来自适应地获得 DPC 算法的最优参数,本文所提的无参数密度计算方法能够很好地解决 DPC 算法中的参数敏感问题。

表 5 各算法在真实数据集的性能指标值(ACC)

	SLDPC	DPCSA	DPC	K-means	DBSCAN
Iris	0.953	**0.966**	0.906	0.887	0.713
Pima	**0.669**	0.649	0.516	0.660	0.647
Segmentation	0.485	0.509	0.419	**0.552**	0.219
Water	0.521	0.521	0.521	0.533	**0.546**
Bupa	**0.579**	0.556	**0.579**	0.551	**0.579**
Air	**0.417**	0.415	0.415	0.415	0.415
Breast	**0.661**	0.531	0.617	0.516	0.498
Vote	0.869	0.866	**0.875**	0.867	0.614

表 6 各算法在真实数据集的性能指标值(F1-measure)

	SLDPC	DPCSA	DPC	K-means	DBSCAN
Iris	0.911	**0.935**	0.841	0.811	0.663
Pima	0.614	**0.688**	0.521	0.628	0.557
Segmentation	0.415	0.474	0.419	**0.486**	0.171
Water	**0.667**	**0.667**	0.499	0.528	0.603
Bupa	**0.667**	0.536	**0.667**	0.622	0.656
Air	**0.508**	**0.508**	**0.508**	0.484	**0.508**
Breast	0.611	0.553	**0.635**	0.538	0.524
Vote	0.777	0.773	**0.787**	0.774	0.688

2.3 参数分析

SLDPC 中有 2 个参数,分别为 α 和 β。其中 β 为固定范围参数,且参数范围值较小,在本节中不做讨论。下面进行 α 的分析。

图 9 ~ 图 14 展示了 α 的范围为[1,10]时,数据集Flame、Spiral和Sticks聚类效果的变化情况。综合分析 α 从 1 ~ 10 变化时图中 3 个数据集聚类效果的变化情况可知,当 α 为 3 时 3 个数据集都取得了最好的聚类结果。由此可见, α 一般可设置为 3。

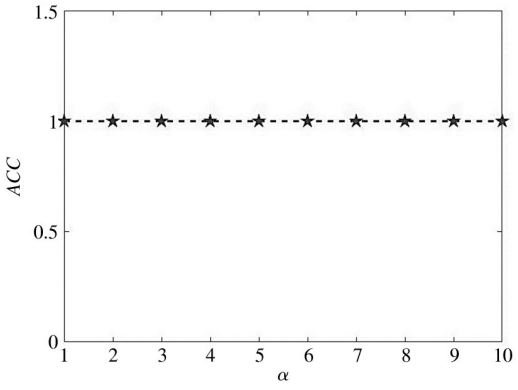


图 9 α 从 1 ~ 10 变化,Flame 数据集的 ACC 值

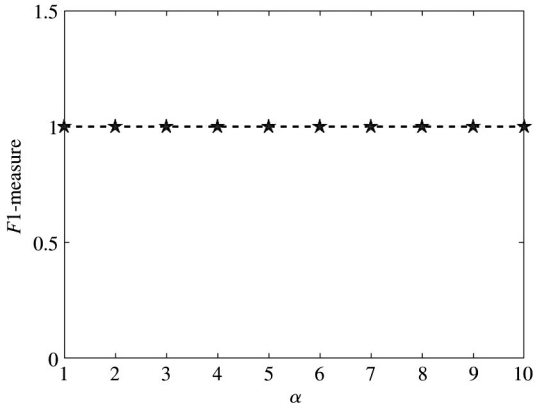


图 10 α 从 1 ~ 10 变化,Flame 数据集的 F1-measure 值

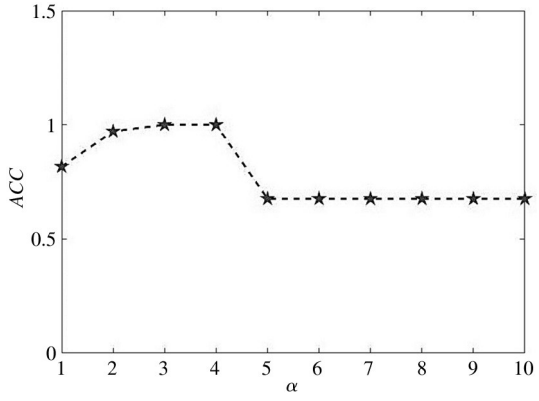


图 11 α 从 1 ~ 10 变化,Spiral 数据集的 ACC 值

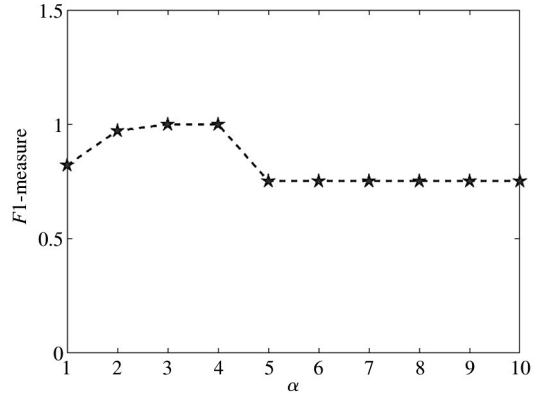


图 12 α 从 1 ~ 10 变化,Spiral 数据集的 F1-measure 值

由图 9 和图 10 可知,对于 Flame 数据集, α 从 1 ~ 10 变化时对于整个算法的聚类结果并无影响,且算法一直处于效果最优的状态。由图 13 和图 14 可知,当 α 从 1 ~ 7 变化时,聚类效果并没有受到影响,

而当 α 从 8 ~ 10 变化时,聚类效果开始降低,侧面证明了本文所提的基于引力理论的簇心评价方式能够扩大决策图中数据点之间的差异,也证明了此方法的有效性。

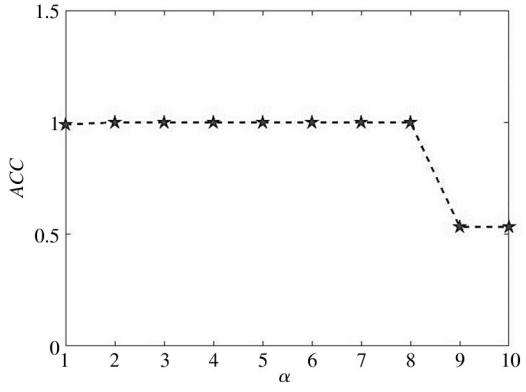


图 13　α 从 1 ~ 10 变化,Sticks 数据集的 ACC 值

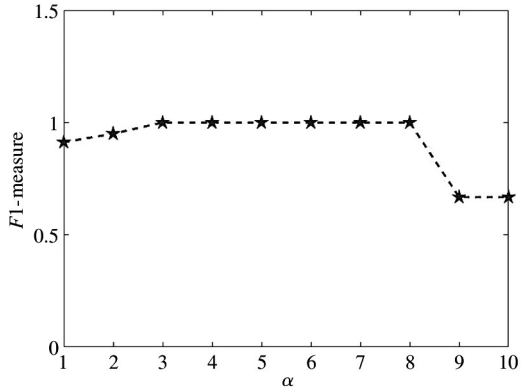


图 14　α 从 1 ~ 10 变化,Sticks 数据集的 F1-measure 值

3　结　论

本文针对 DPC 中存在的参数敏感、算法不连续和聚类分块化问题,提出了一种基于数据相似度和引力理论的密度峰聚类算法(SLDPC)。首先,通过基于数据相似度的密度计算方法,消除了 d_c 对 DPC 聚类效果的影响。其次,采用基于引力理论的簇心评价方法解决了人工选取簇心过程中存在的问题。同时采用基于边缘分布的合并策略对聚类分块化进行合并。最后,通过 16 个数据集对本文所提方法的有效性进行了验证,并且对算法中存在的参数进行了分析。实验结果表明,SLDPC 在实验数据集上总体表现优于其他 4 个算法,且算法中存在的参数更容易确定,证明了本文所提方法的有效性。

参考文献

[1] 余文凯, 章政, 付雪画, 等. 基于地图预处理及改进 A * 算法的路径规划[J]. 高技术通讯, 2020, 30(4): 383-390.

[2] 张志龙, 李爱华, 李楚为. 基于密度峰值搜索聚类的超像素分割算法[J]. 计算机学报, 2020, 43(1): 1-15.

[3] 江超, 邢科新, 林叶贵, 等. 未知环境下移动机器人静态与动态实时避障方法研究[J]. 高技术通讯, 2019, 29(10): 1012-1020.

[4] JAIN A K. Data clustering: 50 years beyond K-means [J]. Pattern Recognition Letters, 2010, 31(8): 651-666.

[5] ESTER M, KRIEGEL H P, SAMDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// Data Mining and Knowledge Discovery. Portland: AAAI Press,1996: 226-231.

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

[7] DU M, DING S, XUE Y. A robust density peaks clustering algorithm using fuzzy neighborhood[J]. International Journal of Machine Learning and Cybernetics, 2018, 9(7): 1131-1140.

[8] LIU T, LI H, ZHAO X. Clustering by searching in descending order and automatic find of density peaks[J]. IEEE Access, 2019, 7: 133772-133780.

[9] ZHUO L, LI K, LIAO B, et al. HCFS: a density peak based clustering algorithm employing a hierarchical strategy[J]. IEEE Access, 2019, 7: 74612-74624.

[10] CHEN J, YU P. 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.

[11] XIE J, GAO H, XIE W, et al. Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors[J]. Information Sciences, 2016, 354: 19-40.

[12] 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, 9: 135-145.

[13] LIU Y, MA Z, YU F. Adaptive density peak clustering

based on K-nearest neighbors with aggregating strategy [J]. Knowledge-Based Systems, 2017, 133: 208-220.

[14] GENG Y, LI Q, ZHENG R, et al. RECOME: a new density-based clustering algorithm using relative KNN kernel density[J]. Information Sciences, 2018,436:13-30.

[15] DU M, DING S, XUE Y. A robust density peaks clustering algorithm using fuzzy neighborhood[J]. International Journal of Machine Learning and Cybernetics, 2018, 9(7): 1131-1140.

[16] YU D, LIU G, GUO M, et al. Density peaks clustering based on weighted local density sequence and nearest neighbor assignment[J]. IEEE Access, 2019,7:34301-34317.

[17] 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.

[18] YAN H, WANG L, LU Y. Identifying cluster centroids from decision graph automatically using a statistical outlier detection method[J]. Neurocomputing, 2019, 329: 348-358.

[19] 陈晋音, 何辉豪. 基于密度的聚类中心自动确定的混合属性数据聚类算法研究[J]. 自动化学报, 2015, 41(10): 1798-1813.

[20] WANG G, WEI Y, TSE P. Clustering by defining and merging candidates of cluster centers via independence and affinity[J]. Neurocomputing, 2018, 315: 486-495.

[21] MEHMOOD R, El-ASHRAM S, BIE R, et al. Clustering by fastsearch and merge of local density peaks for gene expression microarray data[J]. Scientific Reports, 2017, 7(1): 1-7.

[22] HOU J, ZHANG B. Cluster merging based on a decision threshold [J]. Neural Computing and Applications, 2018, 30(1): 99-110.

[23] AKBAS C E, GUNAY O, TASDEMIR K, et al. Energy efficient cosine similarity measures according to a convex cost function[J]. Signal, Image and Video Processing, 2017, 11(2): 349-356.

[24] XIE J, XIONG Z, DAI Q, et al. A local-gravitation-based method for the detection of outliers and boundary points [J]. Knowledge-Based Systems, 2020, 192: 105331-105331.

[25] FEDELE M, VERNIA C. Inverse problem for multispecies ferromagneticlike mean-field models in phase space with many states[J]. Physical Review E, 2017,96(4): 1-11.

[26] LU J, ZHU Q. An effective algorithm based on density clustering framework[J]. IEEE Access, 2017, 5: 4991-5000.

Density peak clustering algorithm based on data similarity and gravity theory

ZHAN Kang, WANG Yiwen, HE Xiongxiong

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023)

Abstract

In this paper, a L1-norm based data similarity density peak clustering (SLDPC) algorithm is proposed to solve the problems of parameter sensitivity, algorithm discontinuity and clustering fragmentation in density peak clustering (DPC). The local density is determined based on the data similarity, and the gravity theory is used to enlarge the difference between cluster centers and non-cluster centers. The cluster centers are determined automatically by setting a threshold value. The clustering partitioning is merged by a merging strategy based on edge distribution. A total of 16 data sets are used in the experiment, and compared with DPC, K-means, density-based spatial clustering of applications with noise (DBSCAN) and the improved DPC algorithm. The experimental result shows that the proposed method has excellent clustering accuracy and good stability.

Key words: clustering analysis, density peak, data similarity, gravity theory, cluster merging