Journal of Computer Applications
计算机应用,2020,40(6):1654 - 1661
ISSN 1001-9081
CODEN JYIIDU
2020-06-10
http://www.joca.cn

文章编号:1001-9081(2020)06-1654-08
DOI:10.11772/j.issn.1001-9081.2019111881

自适应密度峰值聚类算法

吴　斌*,卢红丽,江惠君
(南京工业大学 工业工程系,南京 211816)
(*通信作者电子邮箱 wubin@njtech.edu.cn)

摘　要:密度峰值聚类(DPC)算法是一种新型的聚类算法,具有调节参数少、无需迭代求解、能够发现非球形簇等优点;但也存在截断距离无法自动调节、聚类中心需要人工指定等缺点。针对上述问题,提出了一种自适应DPC(ADPC)算法,实现了基于基尼系数的自适应截断距离调节,并建立了一种聚类中心的自动获取策略。首先,综合考虑局部密度和相对距离两种因素以重新定义簇中心权值计算公式;然后,基于基尼系数建立自适应截断距离调节方法;最后,根据决策图和簇中心权值排序图提出自动选取聚类中心的策略。仿真实验结果表明,ADPC算法可以根据问题特征来自动调节截断距离并自动获取聚类中心点,而且在测试数据集上取得了比几种常用的聚类算法和DPC改进算法更好的结果。

关键词:密度峰值;截断距离;自动聚类;基尼系数;聚类中心
中图分类号:TP181　　**文献标志码:**A

Adaptive density peaks clustering algorithm

WU Bin*, LU Hongli, JIANG Huijun
(*Industrial Engineering Department, Nanjing Tech University, Nanjing Jiangsu* 211816, *China*)

Abstract: Density Peaks Clustering (DPC) algorithm is a new clustering algorithm with the advantages such as few adjustment parameters, no iterative solution and the capacity of finding non-spherical clusters. However, there are some disadvantages of the algorithm: the cutoff distance cannot be adjusted automatically, and the cluster centers need to be selected manually. For the above problems, an Adaptive DPC (ADPC) algorithm was proposed, the adjustment of adaptive cutoff distance based on Gini coefficient was realized, and an automatic acquisition strategy of clustering centers was established. Firstly, the calculation formula of cluster center weight was redefined by taking local density and relative distance into account at the same time. Then, the adjustment method of adaptive cutoff distance was established based on Gini coefficient. Finally, according to the decision graph and cluster center weight sort graph, the strategy of automatically selecting cluster centers was proposed. The simulation results show that, the ADPC algorithm can automatically adjust the cutoff distance and automatically acquire the clustering centers according to the characteristics of problem, and obtain better results than several commonly clustering algorithms and improved DPC algorithms on the test datasets.

Key words: density peak; cutoff distance; automatic clustering; Gini coefficient; clustering center

0　引言

2014年6月,《Science》发表了一种新型聚类算法——密度峰值聚类(Density Peaks Clustering, DPC)算法[1],它不需预先确定簇的数量,就能够发现并处理任何形状的数据集。DPC通过决策图人工选取聚类中心点,然后将剩余数据点分配给与它们距离较近且密度更大的数据点所在的类簇。DPC具有简单高效、调节参数少、样本点一次分配归类、能发现各种形状的簇等优势。基于上述优势,DPC算法已经在多个领域得到成功应用,如:图像分割、优化算法、文本发现、社交网络等[2]。但DPC算法还存在某些缺点:截断距离d_c仍依靠人工经验选取,聚类中心需要人工抉择,在实际应用中聚类效果差等。

自Rodriguez等[1]提出DPC算法以来,在学术界引起广泛关注,学者们对DPC算法进行深入研究,研究内容主要集中在以下方面:

1)局部密度和相对距离的定义。

谢娟英等[3]采用宽度$\delta=1$指数核函数,结合K近邻信息重新定义局部密度,使其更能反映样本的局部分布信息。Seyedi等[4]利用K近邻思想计算局部密度,并使用基于图的标签传递策略来分配剩余点的类簇,有效处理位于边界和重叠区域的样本点。Liu等[5]将共享近邻(Shared Nearest Neighbor, SNN)思想引入DPC中计算局部密度和相对距离。

2)截断距离的调整。

Liu等[6]引入K近邻思想计算截断距离。Wang等[7]基于数据场理论和信息熵理论,提出了一种提取最优截断距离的新方法,计算原始数据集在数据场中的势能熵,以获得最优的截断距离值。

3)聚类中心的获取方法。

收稿日期:2019-11-05;**修回日期:**2020-01-03;**录用日期:**2020-01-06。　**基金项目:**国家自然科学基金资助项目(71671089)。
作者简介:吴斌(1979—),男,河南郑州人,副教授,博士,主要研究方向:系统建模与仿真;　卢红丽(1995—),女,河南商丘人,硕士研究生,主要研究方向:系统建模与仿真;　江惠君(1996—),女,江苏扬州人,硕士研究生,主要研究方向:物流供应链管理。

Xu 等[8]运用线性回归方法拟合γ值，找到潜在聚类中心点，并分析γ曲线获取阶梯来构造 leading tree，再进一步获取不同层次的聚类，同时基于前导树构建集群层次结构，设计网格粒度框架使其适用于大规模和高维数据集。Ding 等[9]提出一种简单的获取聚类中心策略(γ-graph)。Yan 等[10]为了从决策图中识别聚类中心，提出一种统计异常检测方法，同时可以确定聚类中心的数量。马春来等[11]通过排序后的簇中心权值(γ)下降趋势(斜率)找出“拐点”所在位置，“拐点”之前即为聚类中心点。

4)制定新的分配规则。

薛小娜等[12]采用新的评价指标获取聚类中心后，结合K近邻与迭代思想，对样本点实现局部聚类，多类合并完成最终的聚类工作。Liu 等[5]提出两步分配方法：①通过共享近邻识别分配属于一个集群的点；②查找更多邻居所属集群分配剩余点。谢娟英等[3]提出两种基于K邻近的样本分配策略，策略 1 针对非离群点，策略 2 针对离群点和策略 1 之后还未分配的非离群点。

5)改进距离矩阵。

Zhang 等[13]结合局部敏感哈希算法与 MapReduce 模型，改进 DPC 算法中的距离矩阵计算方法，相较于欧氏距离，降低了计算代价。

综上所述，现有的改进方法中，参数及聚类中心需要人工设定与选择，缺乏自适应调节能力，算法缺乏鲁棒性。因此，本文提出了一种自适应 DPC(Adaptive DPC，ADPC)算法，实现了基于基尼系数的自适应截断距离调节，并设计一种自动获取聚类中心点的策略。

1　DPC 算法

密度峰值聚类算法核心思想如下：1)聚类中心的密度高于其邻近的样本点密度；2)聚类中心与比其密度高的数据点的距离相对较远[1]。数据集的聚类中心同时拥有较大局部密度和较大相对距离，周围都是比其局部密度低的点，并且周围这些点距离其他高密度点相对较远。ρ_i表示数据集中与数据点x_i之间的距离小于d_c的数据点总个数，ρ_i的值越大，表示与数据点x_i之间的距离小于截断距离d_c的点数量越多。当数据点x_i取最大局部密度($\max\{\rho\}$)时，相对距离δ_i表示数据集中与x_i距离最大的数据点与x_i点之间的距离，否则表示与x_i距离最小的那个数据点与x_i之间的距离。

定义 1　局部密度ρ_i[1]。

局部密度有两种计算方式：截断核为离散值的计算方式，高斯核为连续值的计算方式。

截断核为：

$$\rho_i = \sum_{j \neq i} \chi(d_{ij} - d_c) \tag{1}$$

函数$\chi(x)$定义为：

$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geqslant 0 \end{cases} \tag{2}$$

式中：ρ_i为第i个数据点对应的局部密度；$d_{ij} = \sqrt{\sum_{k=1}^{n}\left(x_{ik} - x_{jk}\right)^2}$，$d_{ij}$为数据点$x_i$和$x_j$之间的欧氏距离；$d_c$为截断距离；$i$和$j$分别为第$i$个数据点、第$j$个数据点，取值范围从 1 到数据样本点总个数$n$。

高斯核为：

$$\rho_i = \sum_{j \neq i} \exp\left(-\left(\frac{d_{ij}}{d_c}\right)^2\right) \tag{3}$$

定义 2　相对距离δ_i[1]。

$$\delta_i = \begin{cases} \min\limits_{j \in I_S^i}(d_{ij}), & I_S^i \neq \varnothing \\ \max\limits_{j \in I_S^i}(d_{ij}), & I_S^i = \varnothing \end{cases} \tag{4}$$

式中：S代表数据集$S = \{x_1, x_2, \cdots, x_n\}$，$n$为数据集样本数，$x_i$为密度峰值点；指标集$I_S^i = \{k \in I_S : \rho_k > \rho_i\}$，当$\rho_i = \max\limits_{j \in I_S}\{\rho_j\}$时，有$I_S^i = \varnothing$。

DPC 算法具体步骤如图 1 所示。首先，对样本集数据初始化及预处理，确定截断距离；其次，再计算各数据点的局部密度ρ_i和相对距离δ_i；利用绘制的决策图(以局部密度为横坐标，以相对距离为纵坐标)选出ρ_i和δ_i较大的点作为聚类中心；然后，对非聚类中心点按照分配策略归类；最后将每个簇中的数据点进一步分为核心点(cluster core)和边缘点(cluster halo)两个部分，并检测噪声点。其中，核心点是类簇核心部分，其ρ值较大；边缘点位于类簇的边界区域且ρ值较小，两者的区分界定则是借助于边界区域的平均局部密度。

图 1　DPC 算法流程

Fig. 1　Flowchart of DPC algorithm

2　ADPC 算法

标准 DPC 算法依靠人工确定截断距离和聚类中心点，导致了算法的不确定性，同时降低了鲁棒性。针对上述情况，ADPC 算法实现截断距离的自适应调整的目的，并且自动选择聚类中心以实现数据集整个过程的自动聚类。该算法综合考虑局部密度值和相对距离值，自动选择出具有较大局部密度和距离的数据点作为聚类中心点。

文献[1]中给出了一个综合考虑ρ_i值和δ_i值的计算量γ_i(式(5))，主要用于无法直接根据决策图判断聚类中心点的情况。从式(5)可知，聚类中心点的γ值往往比较大，γ值越大的点有越大概率选为聚类中心。

$$\gamma_i = \rho_i * \delta_i \tag{5}$$

式中：$\gamma_i (i \in I_S^i)$代表第i个数据点的簇中心权值，数值越大越有可能被选为聚类中心点。因此，对γ_i降序排列后，从最前面截取若干个数据点作为聚类中心。

在此基础上，本文对γ重新定义，如式(6)所示。算法首

先对局部密度和相对距离进行归一化处理，然后根据式(6)计算出各数据点的γ_i，并将它们按降序排列，聚类中心点往往是在γ_i值较大的点中获得。

$$\gamma_i = \sum_{j \neq i} \exp\left[-(d_{ij}/d_c)^2\right] * \delta_i * \delta_i \tag{6}$$

式中：$i, j \in I_S^i$。

2.1 自适应截断距离

在标准DPC算法中，截断距离d_c依靠经验选取，Rodriguez等[1]给出选择d_c方法，使各样本点平均邻居数约占数据集样本点总数的1%~2%，这一取值应用于不同规模的数据集中，导致在实际应用中鲁棒性差。截断距离不仅影响人工选择聚类中心点，还影响聚类分配中的边界区域划分，进而影响最终的聚类结果，因此需要一种方法来规避这种影响。基尼系数可做特征选择，而且可以用来表征数据不纯度，如式(7)所示。将基尼系数的定义引申到自适应DPC算法中，用基尼系数度量数据不纯度，即基尼系数G越小，数据不纯度较小，数据分布的不确定性也越小，越容易聚类。自适应DPC算法中，提出了一种基于基尼系数最小化的自适应方法，结合式(6)和式(7)，给出截断距离d_c和基尼系数G的关系式(8)，以此实现自适应截断距离。

$$Gini(D) = 1 - \sum_{i=1}^{n} (p_i)^2 \tag{7}$$

式中：$Gini(D)$代表基尼系数值，表征数据域中数据的不纯度；D表示数据集全部样本；p_i表示每种类别出现的概率。

$$G = 1 - \sum_{i=1}^{n} \left(\frac{\gamma_i}{Z}\right)^2 \tag{8}$$

式中：G代表数据集的基尼系数值大小；$Z = \sum_{i=1}^{n} \left(\delta_i * \delta_i * \sum_j \exp\left[-(d_{ij}/d_c)^2\right] \right)$(高斯核)，$Z$代表数据集总的簇中心权值大小；$\gamma_i$代表第$i$个数据点的簇中心权值。

截断距离d_c在不断的变化中，寻找使基尼系数G取得最小值时所对应的d_c，并且将优化后的截断距离作为下一步聚类的基础，代替人工选取截断距离，规避人工选取的主观性，从而达到自适应d_c的目的。

2.2 自动确定聚类中心

运用式(9)所示的最大最小方法，对ρ_i和δ_i进行数据归一化处理。归一化处理后的数据，可以消除ρ_i和δ_i量纲的不同而对实验结果产生的影响，归一化之后两者的数值均映射到区间[0,1]。

$$x^* = (x - x_{\min})/(x_{\max} - x_{\min}) \tag{9}$$

式中：x^*代表归一化处理后得到的映射数值；x代表ρ_i、δ_i的取值；$x_{\min}$表示数据(ρ_i或δ_i)的最小值；$x_{\max}$表示数据(ρ_i或δ_i)的最大值。

自动获取聚类中心主要步骤如算法1所示，对于人工数据集，γ分布满足幂次定律，即对γ取log函数，$\log(\gamma)$近似呈直线形式[1]。根据自适应优化后的截断距离计算ρ_i和δ_i，对γ_i取log函数后降序排列，然后取较大的前m个值绘制决策图。对这m个值相邻两数之间取差值，寻找差值变化最大的值，在这之前的所有数据点记为初始聚类中心点。考虑到γ_i差值突变不太明显的数据集，在获取初始聚类中心点后，对获取的点进行自动修正。对于UCI数据集，在取$\log(\gamma)$后，对前m个值降序排列后的数值进行函数拟合。在拟合函数水平线之上的点确定为潜在的聚类中心点。该方法可以尽可能地获取多数潜在聚类中心，防止后续操作时遗漏可能的正确聚类中心点。最终还需要从潜在聚类中心点中准确筛选出实际的聚类中心，进而真正地自动选取聚类中心。

算法1 自动获取聚类中心。

1)归一化处理ρ_i和δ_i(最大最小方法)，对γ_i取log函数后按照降序排列。

2)取降序排列的前m个γ值(m = 50)，绘制纵坐标为γ值(降序)，横坐标为[0, m]的决策图。人工数据集转到步骤3)，UCI数据集转到步骤4)。

3)人工随机生成的数据集获取聚类中心可分为两个步骤：

a)人工随机生成的数据集取$\log(\gamma)$后，满足幂次定律。对排序后的γ相邻两数间取差值，找差值变化最大处，在这之前的所有点，即为初始聚类中心点。

b)对获取的初始聚类中心自动修正。

设相邻两数间差值为向量D，初始聚类中心点个数为n

for i=1,2,3,4,5 then

if 第(n+i)个γ相邻两数差值>前n个γ相邻两数差值的平均值

聚类中心点个数n=n+i

end for

4)拟合排序后的γ曲线图，获取疑似聚类中心点。单独绘制疑似聚类中心点(横坐标-ρ，纵坐标-δ)，依据ρ_i和δ_i，同时结合δ_i和截断距离的大小关系，筛选获取较大ρ和δ的点为聚类中心点。

5)获取聚类中心后，标记聚类中心点，进入下一步聚类操作。

如图2(b)所示，从聚类中心到非聚类中心有一个明显陡峭的下降趋势，即非聚类中心到聚类中心的转变过程中，有一个跳跃阶段，自动获取聚类中心就是利用这一思想获取的，可知γ值较大的点为实际聚类中心点。考虑到有些数据集γ_i相邻两数差值变化不太明显(如图2(c))，可能导致遗漏聚类中心点。因此对于初步获取的聚类中心点，如果之后的簇中心权值出现差值大于之前聚类中心点的平均差值变化，则将该点之前的所有点作为新的聚类中心点，即获取过程中加上自动修正。

2.3 算法的主要步骤描述

自适应DPC算法的主要步骤如算法2所示。首先，计算数据集的距离矩阵，运用自适应截断距离方法获取d_c值；再根据优化后的d_c值计算局部密度ρ_i和相对距离δ_i，并对其进行归一化处理，同时计算γ值；然后，依据提出的自动获取聚类中心策略，自动选出局部密度和相对距离较高的点作为聚类中心点；最后，对非聚类中心点按照DPC分配策略聚类。

算法2 自适应DPC算法。

输入 数据集$S = \{x_1, x_2, \cdots, x_n\}$，数据集样本数$n$；

输出 聚类结果$C = \{c_1, c_2, \cdots, c_k\}$，聚类中心个数$k$。

1)输入数据集S，计算距离矩阵。

2)参数设置，根据式(8)优化d_c。

3)根据式(3)计算局部密度ρ。

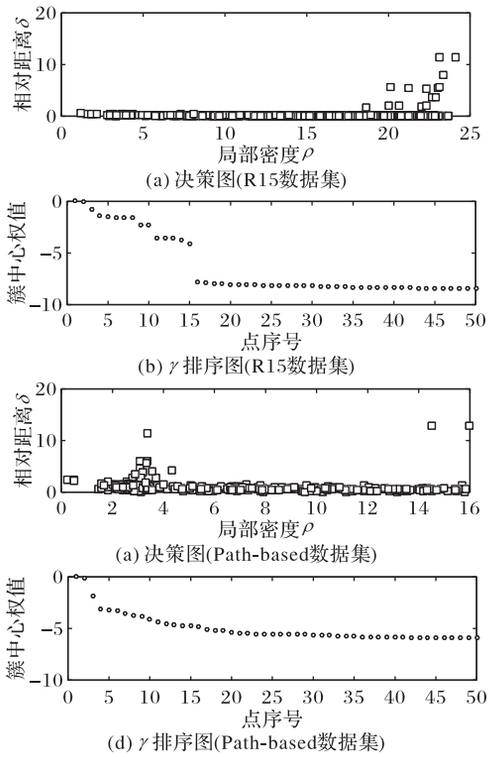
4)根据式(4)计算相对距离δ。

5)归一化处理ρ_i和δ_i(最大最小方法)，计算γ_i，γ_i取log函数后按照降序排列。

6)绘制ρ-δ决策图和γ图，前者坐标点为(ρ_i,δ_i)，后者坐标点为(i,γ_i)，γ_i是上一步处理后的值。

7)自动获取聚类中心点步骤如算法1所示。

8)将非聚类中心点进行聚类，绘制聚类结果图。



(a) 决策图(R15数据集)　(b) γ排序图(R15数据集)

(a) 决策图(Path-based数据集)　(d) γ排序图(Path-based数据集)

图2　R15、Path-based数据集决策图和γ排序图

Fig. 2　Decision graphs and γ sort graphs of R15 and Path-based datasets

3　实验与结果分析

自适应DPC算法采用Matlab R2015b编程实现，运行在Windows 7操作系统，4 GB内存，Intel Core i7-4558U CPU @ 2.80 GHz的计算机平台。测试数据集选用人工数据集和实际问题的UCI(University of California Irvine)数据集[14]，将实验结果与几种常见的聚类算法进行对比分析，这些算法分别是：基于密度的噪声应用空间聚类(Density-Based Spatial Clustering of Applications with Noise，DBSCAN)、对点排序来确定聚类结构(Ordering Points To Identify the Clustering Structure，OPTICS)、近邻传播(Affinity Propagation，AP)、*K*-均值算法(*K*-means algorithm，*K*-means)。评价指标选择调整互信息(Adjusted Mutual Information，AMI)、调整兰德系数(Adjusted Rand Index，ARI)[15]、FM指数(Fowlkes-Mallows Index，FMI)[16]。三者的取值范围分别为：$AMI \in [0, 1]$，$ARI \in [-1, 1]$，$FMI \in [0, 1]$。三种评价指标的最大值都是1，并且其数值越大代表聚类效果越好。实验中用到的数据集相关信息如表1和表2所示。

3.1　自适应DPC算法的实验分析

首先对自适应DPC算法进行分析。以Flame数据集为例，分析自适应截断距离和自动分配聚类中心对算法性能的影响，结果如图3所示，横坐标是百分比，使得选取的截断距离满足各近邻点的平均数占数据集总数的1%~10%，纵坐标是基尼系数值G，根据式(8)计算。基尼系数值G随着d_c的变化而变化，G取最小值时，数据不纯度较小，数据分布的不确定性也越小，越容易聚类。通过对大量d_c-基尼函数曲线分析，G取最小值时，满足截断距离条件的百分比在1%~6%，因此截断距离选取样本点的1%~6%。对于截断距离的选取，既缩小了选择范围、缩短了计算时间，又对其有相关的自动优化操作，目的是找到使基尼系数G达到最小值的d_c，并将它作为最优的截断距离值，然后进行下一步的聚类，最终实现截断距离的自适应操作。

表1　人工数据集

Tab. 1　Synthetic datasets

数据集	实例数	维数	类簇数
Flame[17]	240	2	2
Aggregation[18]	788	2	7
R15[19]	600	2	15
D31[19]	3 100	2	31
Spiral[20]	312	2	3
Path-based[20]	300	2	3
Jain[20]	373	2	2
S1[21]	5 000	2	15
DIM512[22]	1 024	512	16
DIM1024[23]	1 024	1 024	16

表2　UCI数据集

Tab. 2　UCI datasets

数据集	实例数	维数	类簇数
Seeds[14]	210	7	3
Libras-movement[14]	360	91	15

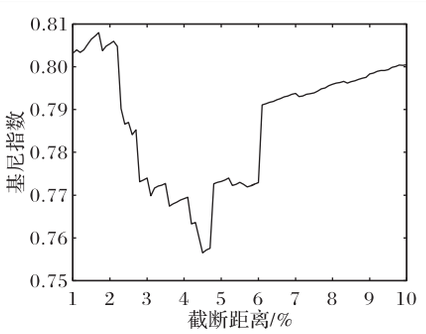


图3　d_c-基尼函数图(Flame)

Fig. 3　d_c-Gini function diagram(Flame)

从图4(a)所示的标准DPC的决策图可看出，该决策图聚类中心点与非聚类中心点界限不明显，难以直接看出点的分布规律，不易直接选取聚类中心点，也不易直接确定聚类中心点个数。利用自适应DPC算法，获取如图4(b)所示改进后的决策图，优化d_c后获得的决策图更容易正确选择聚类中心，最终得到准确的聚类结果图。

表3给出了自适应DPC算法与标准DPC算法其及改进算法的对比，改进算法包括：基于共享近邻的密度峰值快速搜索聚类(Shared-Nearest-Neighbor-based Clustering by fast search and find of Density Peaks，SNN-DPC)算法、模糊加权*K*近邻密度峰值聚类(Fuzzy weighted *K*-Nearest Neighbors Density Peak Clustering，FKNN-DPC)算法。表中评价指标值AMI、ARI、FMI结果保留小数点后四位，Par表示各算法的参数值，加粗值表示较好的实验结果，SNN-DPC、F*K*NN-DPC、DPC算法在各数据集的评价指标值来源文献[5]，表中所有结果都是参数调整后的最佳结果。表3中算法的参数说明：ADPC和DPC算法，它们有一个参数截断距离(浮点数)，表示各样本点平均邻

居数约占数据集样本点总数的Par%;SNN-DPC和FKNN-DPC算法,它们有一个参数K(整数),表示K个最近邻点的个数。

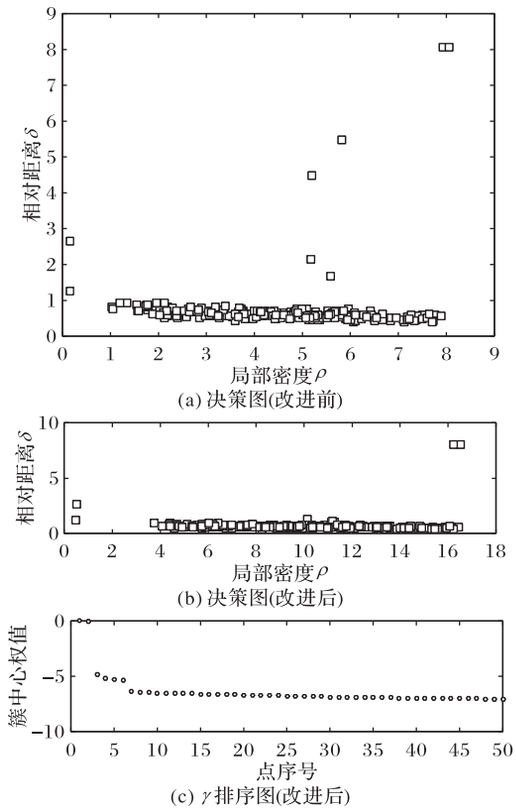


图4 决策图和γ排序图

Fig. 4 Decision graphs and γ sort graph

从表3中数据可以看出,在大多数问题上(Path-based数据集和Jain数据集除外),自适应DPC算法相较其他DPC算法取得了更好的聚类效果。自适应DPC算法提出的兼顾局部密度和相对距离的截断距离调节方法,很大程度上解决了参数敏感问题,同时自动获取聚类中心很大程度上解决了人工参与决策的主观性问题。自适应DPC算法在Path-based数据集和Jain数据集上表现一般,通过分析问题发现,适应性DPC算法在Path-based数据集成功获取了三个聚类中心点,但在最后的聚类结果上表现不理想,其主要原因是聚类的分配过程出现了偏差:某个高密度点在分配类簇时出错,导致其近邻点随之出现错误,引发一系列连锁反应。而Jain数据集中有两个具有不同密度的簇,高密度区域的点通常拥有较高的ρ和δ,易被选为聚类中心点,而且高密度区域点更易吸引更多的低密度点集聚。相对而言,处于低密度区域中的点即使有较高的δ也不易被选中为聚类中心点,导致分配结果不乐观。

自适应DPC算法在随点变化的后期出现不太满意的效果,主要是以下原因造成的:

1)对于某类问题(如Flame数据集、Path-based数据集,类簇边缘样本有重叠),簇与簇之间密度差异较小,边界区域界限较模糊,增加了聚类分配的难度。

2)DPC在选择好聚类中心点之后,对其他样本点采用贪婪分配策略。将样本点i分配给与其最近、密度更高的样本点所属类簇。这种无需迭代的一次分配策略,可以提高聚类效率,但同时也会引发“多米诺骨牌”效应。一旦某个样本点出现分配错误,将连带其邻近点的分配,进而可能影响到更多样本点的类簇分配情况。

3)同时此分配方式遇到密度分布差异大、核心点稀疏、局部密度值较小这些情况时,容易将原本属于类簇的部分核心点错误的认成噪声点。执行分配策略时,较多地考虑到局部信息而忽略了整体结构趋势走向(例如Path-based中底部开口的圆形簇)。

因此,如何在现有基础上改进分配规则将是下一步研究的重点,可从以下三个方面着重考虑:1)边界区域的平均局部密度的设定;2)样本点与聚类中心点的关联程度以及密度分布的差异性;3)样本数据集的整体趋势,样本点邻近区域范围内的整体情况,尤其关注低密度近邻点。

表3 自适应DPC算法与改进DPC算法在不同数据集上的评价指标值

Tab. 3 Evaluation index values of adaptive DPC algorithm and improved DPC algorithms on different datasets

算法	Flame				Aggregation				R15			
	AMI	ARI	FMI	Par	AMI	ARI	FMI	Par	AMI	ARI	FMI	Par
ADPC	**1.000 0**	**1.000 0**	**1.000 0**	4.5	**1.000 0**	**1.000 0**	**1.000 0**	6.0	**1.000 0**	**1.000 0**	**1.000 0**	3.2
SNN-DPC	0.897 5	0.950 2	0.976 8	5.0	0.950 0	0.959 4	0.968 1	15.0	0.993 8	0.992 8	0.993 3	10.0
FKNN-DPC	**1.000 0**	**1.000 0**	**1.000 0**	6.0	0.977 5	0.985 5	0.988 6	20.0	0.990 7	0.989 2	0.989 9	27.0
DPC	**1.000 0**	**1.000 0**	**1.000 0**	2.8	**1.000 0**	**1.000 0**	**1.000 0**	3.4	0.993 8	0.992 8	0.993 3	0.6
算法	D31				Path-based				DIM512			
	AMI	ARI	FMI	Par	AMI	ARI	FMI	Par	AMI	ARI	FMI	Par
ADPC	**1.000 0**	**1.000 0**	**1.000 0**	1.0	0.473 6	0.434 8	0.653 4	2.4	**1.000 0**	**1.000 0**	**1.000 0**	1.0
SNN-DPC	0.964 2	0.950 9	0.952 5	41.0	**0.900 1**	**0.929 4**	**0.952 9**	9.0	**1.000 0**	**1.000 0**	**1.000 0**	5.0
FKNN-DPC	0.952 2	0.927 5	0.929 8	28.0	0.834 4	0.874 4	0.916 5	9.0	**1.000 0**	**1.000 0**	**1.000 0**	20.0
DPC	0.955 4	0.936 5	0.938 5	0.6	0.521 2	0.471 7	0.666 4	3.8	**1.000 0**	**1.000 0**	**1.000 0**	0.6
算法	Spiral				Jain				DIM1024			
	AMI	ARI	FMI	Par	AMI	ARI	FMI	Par	AMI	ARI	FMI	Par
ADPC	**1.000 0**	**1.000 0**	**1.000 0**	2.5	0.559 3	0.643 8	0.850 2	1.5	**1.000 0**	**1.000 0**	**1.000 0**	1.0
SNN-DPC	**1.000 0**	**1.000 0**	**1.000 0**	5.0	**1.000 0**	**1.000 0**	**1.000 0**	12.0	—	—	—	—
FKNN-DPC	**1.000 0**	**1.000 0**	**1.000 0**	5.0	0.056 2	0.131 8	0.643 0	10.0	—	—	—	—
DPC	**1.000 0**	**1.000 0**	**1.000 0**	1.8	0.618 3	0.714 6	0.881 9	0.9	**1.000 0**	**1.000 0**	**1.000 0**	1.0

3.2 与其他算法在人工数据集上的对比分析

为了进一步验证算法的性能,将自适应DPC算法与DBSCAN、OPTICS、AP、K-means等几种常用的聚类算法进行对比,聚类结果如表4所示,DPC、DBSCAN、OPTICS、AP、

K-means在各数据集的评价指标值来源文献[5]。表中Par1、Par2代表各算法的参数，表中“—”表示没有对应值，即Par2列出现“—”表示该算法只有一个参数。表4中算法的参数说明：ADPC算法有一个参数截断距离（浮点数），表示各样本点平均邻居数约占数据集样本点总数的Par1%；DBSCAN和OPTICS算法，它们都有两个参数：邻域半径ε（浮点数）和minpts（邻域最少点数，即半径内的期望样本个数，整数）；AP算法有一个参数偏好参数Preference（浮点数），表示样本点作为聚类中心的参考度；*K*-means算法有一个参数*K*（整数），表示设定*K*个聚类中心。自适应DPC算法在数据集Flame、Aggregation、R15、D31、Path-based、Spiral、DIM512、DIM1024均比其他算法表现更好。自适应DPC算法是对标准DPC算法上的改进，它继承了DPC算法的优势，同时自适应DPC算法已将截断距离参数调至最优，避免了标准DPC算法中的参数设置的影响，这些因素使得改进后的算法效果更佳。DBCSAN算法更适用于非凸样本集聚类，它在聚类的同时还可以找出异常点，因此DBSCAN算法在Jain和Path-based数据集上的表现明显优于其他算法。图5显示了部分数据集的自适应DPC算法聚类结果，同样表明自适应DPC算法良好的聚类效果。

3.3　与其他算法在UCI数据集上的对比分析

为了验证自适应DPC算法在实际问题中的性能，采用UCI数据集中Seeds和Libras-movement两个实际问题进行测试。其中，Seeds数据集为小麦种子数据集，该数据集共210个观察值，包含3类不同的小麦种子；Libras-movement数据集为运动数据集，该数据集共360个观察值，包含15类手势移动数据，每类24个样本数据。

表5给出了本文算法和SNN-DPC、FKNN-DPC、DPC等算法的聚类结果性能指标。

表4　人工数据集上各聚类算法的评价指标值

Tab. 4　Evaluation index values of different clustering algorithms on synthetic datasets

算法	Flame					Aggregation				
	AMI	ARI	FMI	Par1	Par2	AMI	ARI	FMI	Par1	Par2
ADPC	**1.000 0**	**1.000 0**	**1.000 0**	4.50	—	**1.000 0**	**1.000 0**	**1.000 0**	6.00	—
DBSCAN	0.823 4	0.938 8	0.971 2	0.09	8	0.952 9	0.977 9	0.982 7	0.04	6
OPTICS	0.689 8	0.896 8	0.950 8	0.10	8	0.992 1	0.975 3	0.980 7	0.06	10
AP	0.498 7	0.540 3	0.749 8	-6.36	—	0.787 3	0.765 8	0.815 0	-1.21	—
K-means	0.386 3	0.453 4	0.736 4	2.00	—	0.793 5	0.730 0	0.788 4	7.00	—
算法	D31					Path-based				
	AMI	ARI	FMI	Par1	Par2	AMI	ARI	FMI	Par1	Par2
ADPC	**1.000 0**	**1.000 0**	**1.000 0**	1.00	—	0.473 6	0.434 8	0.653 4	2.40	—
DBSCAN	0.889 5	0.807 8	0.818 6	0.04	38	**0.871 0**	**0.901 1**	**0.934 0**	0.08	10
OPTICS	0.821 1	0.867 3	0.876 3	0.06	4	0.436 4	0.636 4	0.751 7	0.06	4
AP	0.836 7	0.742 5	0.766 5	0.23	—	0.519 9	0.477 5	0.657 7	-4.10	—
K-means	0.959 3	0.945 3	0.947 0	31.00	—	0.509 8	0.461 3	0.661 7	3.00	—
算法	Spiral					Jain				
	AMI	ARI	FMI	Par1	Par2	AMI	ARI	FMI	Par1	Par2
ADPC	**1.000 0**	**1.000 0**	**1.000 0**	2.50	—	0.559 3	0.643 8	0.850 2	1.50	—
DBSCAN	**1.000 0**	**1.000 0**	**1.000 0**	0.04	2	**0.865 0**	**0.975 8**	**0.990 6**	0.08	2
OPTICS	**1.000 0**	**1.000 0**	**1.000 0**	0.04	1	0.854 2	0.975 6	0.990 5	0.08	1
AP	0.293 2	0.156 9	0.340 9	-0.19	—	0.658 2	0.795 2	0.921 2	-1.77	—
K-means	-0.005 5	-0.006 0	0.327 4	3.00	—	0.491 6	0.576 7	0.820 0	2.00	—
算法	R15					DIM512				
	AMI	ARI	FMI	Par1	Par2	AMI	ARI	FMI	Par1	Par2
ADPC	**1.000 0**	**1.000 0**	**1.000 0**	3.20	—	**1.000 0**	**1.000 0**	**1.000 0**	1.00	—
DBSCAN	0.982 5	0.981 9	0.983 1	0.04	12	**1.000 0**	**1.000 0**	**1.000 0**	0.36	2
OPTICS	0.973 4	0.978 5	0.979 9	0.04	11	0.902 9	0.943 2	0.947 8	0.19	1
AP	0.990 7	0.989 1	0.989 8	-0.17	—	**1.000 0**	**1.000 0**	**1.000 0**	-1.00	—
K-means	0.993 8	0.992 8	0.993 2	15.00	—	**1.000 0**	**1.000 0**	**1.000 0**	16.00	—

表5　UCI数据集上各聚类算法的评价指标值

Tab. 5　Evaluation index values of different clustering algorithms on UCI datasets

算法	Seeds					Libras-movement				
	AMI	ARI	FMI	Par1	Par2	AMI	ARI	FMI	Par1	Par2
ADPC	**1.000 0**	**1.000 0**	**1.000 0**	1.00	—	**0.826 7**	**0.677 6**	**0.707 4**	1.00	—
SNN-DPC	0.750 9	0.789 0	0.858 9	6.00	—	0.583 4	0.392 7	0.450 7	11.00	—
FKNN-DPC	0.697 1	0.724 4	0.827 6	9.00	—	0.475 4	0.318 4	0.397 6	11.00	—
DPC	0.729 9	0.767 0	0.844 4	0.70	—	0.535 8	0.319 3	0.371 7	0.30	—
DBSCAN	0.530 2	0.529 1	0.671 1	0.24	16	0.418 3	0.196 5	0.257 0	0.90	2
OPTICS	0.380 2	0.419 0	0.635 0	0.81	5	0.137 7	0.082 8	0.212 6	0.59	1
AP	0.446 5	0.393 6	0.693 3	-2.07	—	0.148 7	0.205 6	0.197 1	4.31	—
K-means	0.670 5	0.704 9	0.802 6	3.00	—	0.523 2	0.309 4	0.361 2	15.00	—

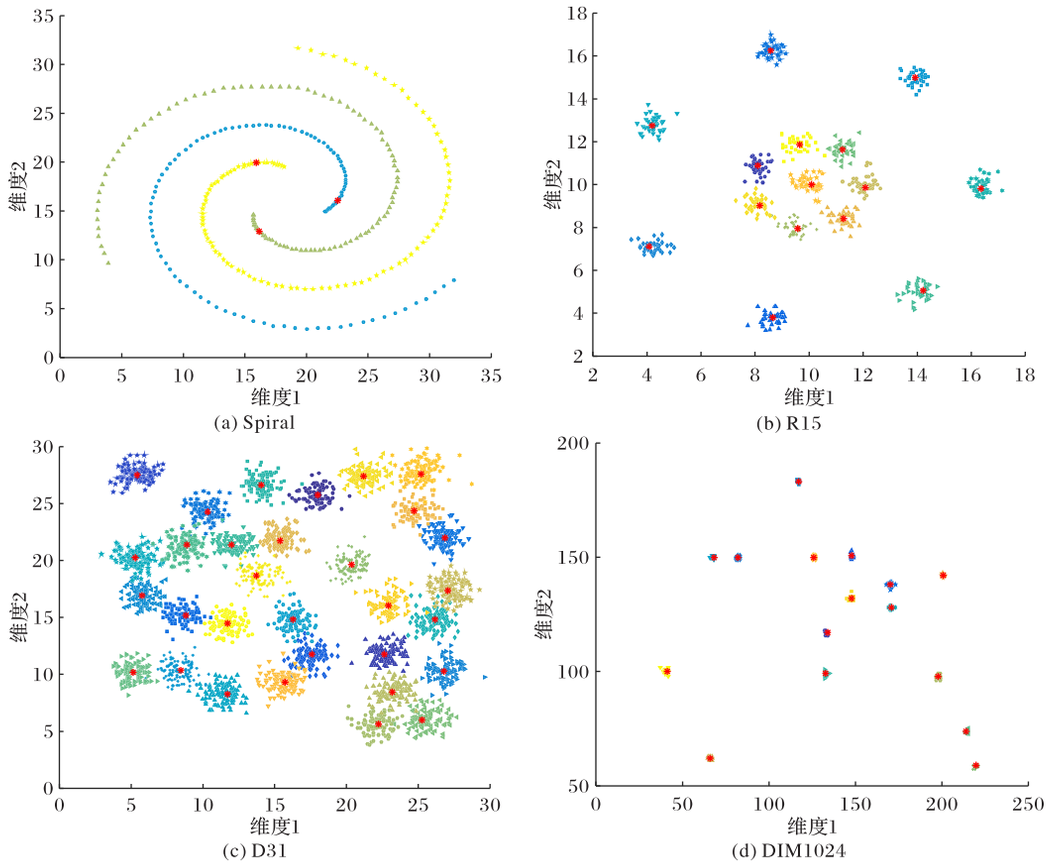


图5 部分数据集的自适应DPC算法聚类结果

Fig. 5 Clustering results of adaptive DPC algorithm for part datasets

从表5中可以看出，在AMI、ARI、FMI评价指标值上，自适应DPC算法有明显的提升，聚类效果更佳。在Seeds数据集中，DPC算法聚类效果明显优于DBSCAN、OPTICS、AP、*K*-means聚类算法，体现了DPC算法在聚类算法中的优势。自适应DPC算法在准确自动获取聚类中心的同时，其聚类性能明显优于其他算法。在Libras-movement数据集中，由于每个类别的样本数量较少，增加了算法计算密度值时的难度。所有算法在该数据集的聚类性能指标都不高，但自适应DPC算法聚类效果较其他算法仍有明显的提升。因此，自适应DPC算法在实际问题中，仍具有良好的聚类性能，能发现真实数据集的聚类中心和分布状况，且具有较强鲁棒性。

4 结语

针对密度峰值聚类算法无法自动调节参数和选择聚类中心的问题，本文提出了一种自适应密度峰值聚类算法。通过重新定义计算γ公式，基于基尼系数的思想建立自适应截断距离调节方法，同时算法还能够自动获取聚类中心点，避免了人工选取聚类中心的不确定性。通过对人工数据集和UCI数据集的测试，并与其他算法的对比分析，可以得出：自适应DPC算法可以根据不同数据集的特点自动调节截断距离d_c，自动选择聚类中心，在大多数问题都取得了比其他算法更好的性能指标。但现有的分配规则制约了其在某些问题中的性能，因此如何改进DPC算法的分配策略是今后研究的重点。同时，将DPC算法应用于实际问题的聚类分析，也是今后研究的重要方向。

参考文献（References）

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

[2] 杨洁，王国胤，庞紫玲. 密度峰值聚类相关问题的研究[J]. 南京大学学报(自然科学版)，2017，53(4)：791-801.（YANG J, WANG G Y, PANG Z L. Relative researches of clustering by fast search and find of density peaks [J]. Journal of Nanjing University (Natural Sciences), 2017, 53(4): 791-801.）

[3] 谢娟英，高红超，谢维信. *K*近邻优化的密度峰值快速搜索聚类算法[J]. 中国科学：信息科学，2016，46(2)：258-280.（XIE J Y, GAO H C, XIE W X. *K*-nearest neighbors optimized clustering algorithm by fast search and finding the density peaks of a dataset [J]. SCIENTIA SINICA Informationis, 2016, 46(2): 258-280.）

[4] SEYEDI S A, LOTFI A, MORADI P, et al. Dynamic graph-based label propagation for density peaks clustering [J]. Expert Systems with Applications, 2019, 115: 314-328.

[5] 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.

[6] 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.

[7] WANG S, WANG D, LI C, et al. Comment on "Clustering by fast search and find of density peaks" [EB/OL]. [2019-11-05]. https://arxiv.org/ftp/arxiv/papers/1501/1501.04267.pdf.

[8] XU J, WANG G, DENG W. DenPEHC: density peak based efficient hierarchical clustering [J]. Information Sciences, 2016, 373: 200-218.

[9] DING S, DU M, SUN T, et al. An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood [J]. Knowledge-Based Systems, 2017, 133: 294-313.

[10] 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.

[11] 马春来,单洪,马涛. 一种基于簇中心点自动选择策略的密度峰值聚类算法[J]. 计算机科学,2016,43(7):255-258,280. (MA C L, SHAN H, MA T. Improved density peaks based clustering algorithm with strategy choosing cluster center automatically [J]. Computer Science, 2016, 43(7): 255-258, 280.)

[12] 薛小娜,高淑萍,彭弘铭,等. 基于*K*近邻和多类合并的密度峰值聚类算法[J]. 吉林大学学报(理学版),2019,57(1):111-120. (XUE X N, GAO S P, PENG H M, et al. Density peaks clustering algorithm based on *K*-nearest neighbors and classes-merging [J]. Journal of Jilin University (Science Edition), 2019, 57(1): 111-120.)

[13] ZHANG Y, CHEN S, YU G. Efficient distributed density peaks for clustering large data sets in MapReduce [J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(12): 3218-3230.

[14] DUA D, KARRA TANISKIDOU E. UCI machine learning repository [DB/OL]. [2019-11-05]. http://archive. ics. uci. edu/ml.

[15] VINH N X, EPPS J, BAILEY J. Information theoretic measures for clusterings comparison: is a correction for chance necessary? [C]// Proceedings of the 26th Annual International Conference on Machine Learning. New York: ACM, 2009: 1073-1080.

[16] 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.

[17] FU L, MEDICO E. FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data [J]. BMC bioinformatics, 2007, 8: Article No. 3.

[18] GIONIS A, MANNILA H, TSAPARAS P. Clustering aggregation [J]. ACM Transactions on Knowledge Discovery from Data, 2007, 1(1): Article No. 4.

[19] VEENMAN C J, REINDERS M J T, BACKER E. A maximum variance cluster algorithm [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(9): 1273-1280.

[20] CHANG H, YEUNG D Y. Robust path-based spectral clustering [J]. Pattern Recognition, 2008, 41(1): 191-203.

[21] JAIN A K, LAW M H C. Data clustering: a user's dilemma [C]// Proceedings of the 2005 International Conference on Pattern Recognition and Machine Intelligence, LNCS 3776. Berlin: Springer, 2005: 1-10.

[22] FRÄNTI P, VIRMAJOKI O. Iterative shrinking method for clustering problems [J]. Pattern Recognition, 2006, 39(5): 761-775.

[23] FRÄNTI P, VIRMAJOKI O, HAUTAMAKI V. Fast agglomerative clustering using a *k*-nearest neighbor graph [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(11): 1875-1881.

This work is partially supported by the National Natural Science Foundation of China (71671089).

WU Bin, born in 1979, Ph. D., associate professor. His research interests include system modeling and simulation.

LU Hongli, born in 1995, M. S. candidate. Her research interests include system modeling and simulation.

JIANG Huijun, born in 1996, M. S. candidate. Her research interests include logistics supply chain management.