文章编号: 1001-0920(2018)11-1921-10 DOI: 10.13195/j.kzyjc.2017.0822

一种基于相对密度和决策图的聚类算法

周世波[1,2], 徐维祥[1†]

(1. 北京交通大学 交通运输学院，北京 100044；2. 集美大学 航海学院，福建 厦门 361021)

摘 要: 聚类是数据挖掘领域的一个重要研究方向,针对复杂数据集中存在的簇间密度不均匀、聚类形态多样、聚类中心的识别等问题,引入样本点k近邻信息计算样本点的相对密度,借鉴快速搜索和发现密度峰值聚类(CFSFDP)算法的簇中心点识别方法,提出一种基于相对密度和决策图的聚类算法,实现对任意分布形态数据集聚类中心快速、准确地识别和有效聚类. 在7类典型测试数据集上的实验结果表明,所提出的聚类算法具有较好的适用性,与经典的DBSCAN算法和CFSFDP等算法相比,在没有显著提高时间复杂度的基础上,聚类效果更好,对不同类型数据集的适应性也更广.

关键词: 聚类；相对密度；决策图；密度峰值；k-近邻；数据挖掘

中图分类号: TP273 **文献标志码:** A

A novel clustering algorithm based on relative density and decision graph

ZHOU Shi-bo[1,2], *XU Wei-xiang*[1†]

(1. School of Traffic and Transportation，Beijing Jiaotong University，Beijing 100044，China；2. Navigation College，Jimei University，Xiamen 361021，China)

Abstract: Clustering is an important research domain in data mining. For some knotty problems in clustering complex datasets, such as uneven densities among clusters, miscellaneous patterns of clusters and the identification of the centers, a clustering method is proposed based on relative density and decision graph, which introduces the idea of k-nearest neighbors to compute the relative densities of data points, and uses the clustering by fast search and find of density peaks(CFSFDP) algorithm for identifying central points, which can identify central points quickly and accurately and cluster datasets of arbitrary distribution effectively. The experimental results on seven typical test datasets show that the proposed clustering algorithm has good feasibility and performance. Compared with the classical density-based spatial clustering of application with noise(DBSCAN) algorithm and CFSFDP algorithm, the proposed algorithm has better clustering effect and accuracy, and has a wider range of adaptation.

Keywords: clustering；relative density；decision graph；density peaks；k-nearest neighbors；data mining

0 引 言

聚类是一种重要的数据分析技术,聚类分析的目的就是使得同一类中的对象之间具有很高的相似度,不同类中的对象高度相异. 国内外学者根据不同的分析视角提出了不同类型的聚类算法[1],这些算法在模式识别、图像处理、风险管理和生物信息学等领域[2-5]中得到了广泛的应用. 其中,基于密度的聚类方法把簇看成数据空间中稀疏区域分开的稠密区域,具有深厚的理论基础,是聚类算法研究的一个重要分支. 一般而言,当数据集中各个簇的密度相对均匀时,经典的基于密度的聚类算法可以取得较好的聚类效果,但是当数据集中各个簇的密度相差较大时,就会遇到困难. 以DBSCAN算法[6]为例,在图1中,假定邻域半径$\varepsilon = 5$,核心对象邻域包含的最少样本数MinPts$= 9$,左边稠密的样本点将作为一个簇,而右边稀疏的样本点则可能作为噪声点处理,这样在聚类时就会出现误判,把本应归于一个簇的样本点作为噪声处理.

由于变密度的数据集广泛存在,为解决这类数据集聚类的问题,本文提出一种新的样本点密度度量尺度,在计算样本点密度时,引入k近邻思想,将样本点局部密度与其k近邻邻域内样本点局部密度的平均

收稿日期: 2017-06-26；**修回日期:** 2017-11-05.

基金项目: 国家自然科学基金项目(61672002, 61272029, 41501490)；福建省自然科学基金项目(2016J01243).

责任编委: 刘民.

作者简介: 周世波(1978−), 男, 副教授, 博士生, 从事数据挖掘及其应用等研究；徐维祥(1964−), 男, 教授, 博士生导师, 从事数据挖掘等研究.

†通讯作者. E-mail: wxxu@bjtu.edu.cn

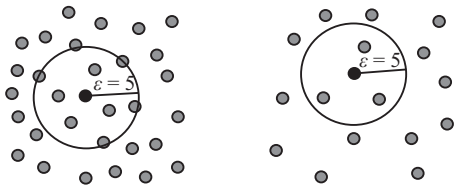


图1 变密度示意图

值之比作为样本点的相对密度,从而揭示各样本点与其相邻样本点的局部信息特征,有效处理数据集中簇间密度不均匀对聚类结果产生的不利影响,同时,借鉴CFSFDP算法[7]中的决策图思想搜索簇中心点,采用相对密度作为样本点密度的度量尺度,解决数据集中簇间密度差别较大时CFSFDP算法对簇中心点选取困难的问题,以便快速、准确地发现簇中心. 最后通过相关实验验证所提出算法的效果和适用性.

1 相关工作

1.1 聚类算法研究进展

根据不同的理论以及针对不同的应用,聚类算法可以分为基于划分的方法、基于层次的方法、基于网格的方法和基于密度的方法[8].

基于划分的方法的核心思想是将数据集划分为不同的簇,使得簇内样本点间的距离尽可能小,而簇间样本点间的距离尽可能大,其特点是可以发现球形、互斥的簇. k-means算法是这类方法的典型代表,它具有效率高、伸缩性好的优点,但是需要事先确定簇个数k,对初始聚类中心敏感,并且只能对凸分布数据集做线性划分,这些不足得到了后续相关研究人员的改进[9-12].

基于层次的聚类算法主要是通过自上向下的分裂或自下向上的合并实现聚类的. BIRCH(Balanced iterative reducing and clustering using hierarchies)算法[13]和CURE(Clustering using representatives)算法[14]是经典的基于层次的聚类算法,BIRCH算法利用树结构实现快速聚类,效率高,适合于大数据集,但只适用于簇的分布呈凸形及球形的情况. CURE算法采用随机取样的方法定义簇,可以高效处理大量数据. 但是,CURE算法的伸缩方式隐含地依赖于球形簇的假设,因此在处理特殊形状的簇时比较困难[12].

基于网格的聚类方法是将样本点映射到已经划分好的不同的网格单元中,计算每个网格单元的密度,由足够稠密的网格形成簇. STING(Statistical information grid)算法[15]以及CLIQUE(Clustering in quest)算法[16]是典型的基于网格的聚类算法. 基于网格的方法的优点是执行效率高,可以处理任意类型的数据,缺点是无法处理不规则分布的数据.

基于密度的聚类算法一般不需要预先指定簇的个数,具有对噪声点不敏感、能发现任意形状和大小的簇等优点,因而在数据挖掘中得到了广泛的应用. DBSCAN算法是基于密度的聚类算法的典型代表,该算法将簇定义为密度相连的点的最大集合,把具有足够高密度的区域划分为簇,其核心思想是利用样本点邻域半径参数ε内的最小样本数MinPts作为密度,在参数ε和MinPts设置得当时,能快速发现包含噪声的任意形状的簇,但是该算法需要在没有先验知识的情况下输入参数,并且当数据集中各个簇的密度相差较大时,聚类效果不理想. 针对这些问题,一些学者提出了相应的改进方法,其中OPTICS(Ordering points to identify the clustering structure)算法[17]是最著名的改进算法. 该算法通过对象排序识别聚类结构,并不显式地产生结果簇,它为自动和交互的聚类分析计算一个簇次序,这个次序代表了数据基于密度的聚类结构,从而屏蔽了输入参数的敏感性问题. 文献[18]将支配集算法与DBSCAN算法结合,提出了一种DSets-DBSCAN聚类算法,使得聚类结果不依赖于输入参数;文献[19]将模糊集理论应用于DBSCAN算法,优化DBSCAN算法输入参数难以选择的问题;文献[20]采用模糊理论优化DBSCAN算法,解决了DBSCAN算法对变密度数据集聚类效果不理想的缺点;文献[21]改进了DBSCAN算法的扩展策略,从区域密度最大点开始向外扩展,直到由密度比例因子决定的边缘区域为止,实现了对密度不均匀数据集的聚类. 密度估计是基于密度的聚类算法的一个核心问题,在基于DBSCAN的聚类算法中,通过邻域半径ε计算密度,这种密度估计方法对参数ε比较敏感. 为解决这一问题,一些学者采用了核密度估计的方法,其中DENCLUE(Density based clustering)算法[22]是这类方法的典型代表. 该算法采用高斯核估计样本点的密度,将密度函数的局部最大点作为簇中心,使用步进式爬山过程把非簇中心点分配到各个簇中心. 文献[23]提出的DENCLUE 2.0对DENCLUE进行了扩展,使用一个新的用于高斯核的爬山过程自动调整步长,提高了算法的效率;文献[24]使用模拟退火和遗传算法替代DENCLUE算法的爬山过程,优化了聚类效果和效率;文献[25]用梯度计算优化爬山过程,显著地提高了DENCLUE算法的运行效率;文献[26]用相似密度函数替代DENCLUE算法中高斯函数,解决了DENCLUE算法对密度不均匀数据集聚类效果差的问题,并采用动态阈值法替代爬山法来降低算法复杂度. 与经典的DBSCAN算法、DENCLUE算法等采用全局密度作为密度度量尺度的方法不同,CFSFDP算法采用局部密度作为密度的度量尺

度. 该算法首先计算每个样本点的局部密度ρ和到局部密度比它大的样本点的距离δ,然后通过构造ρ和δ的决策图来确定簇的中心(选择ρ和δ值都较大的样本点作为簇中心),簇中心找到后,剩余的非中心点被归属到距离其最近邻的且拥有高密度值样本点的簇中. CFSFDP算法能快速发现任意形状数据集的密度峰值点,并高效进行样本点分配和离群点剔除,但是当数据集中有多个密度峰值或者某一个非中心样本点分配错误时,就会出现错误的聚类结果. 针对这些问题,文献[27-30]采用簇边界划分、模糊理论、数据场、热辐射等方法优化局部密度的计算,解决多密度峰值带来的聚类中心点决策困难的不足,文献[31-33]采用k近邻、密度比率、主成分分析等手段改进了CFSFDP算法非中心点聚类策略,取得了比较理想的效果,而文献[34-35]通过增量聚类和MapReduce分布式计算的方法提高了CFSFDP算法的运行效率.

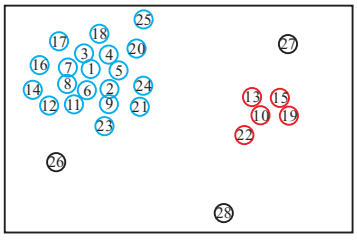
1.2　决策图

决策图是文献[7]提出的一种新颖的识别数据集簇中心的方法. 该方法通过构造数据集中各样本点的局部密度ρ和距离δ的决策图来确定簇的中心,只有当一个样本点的密度值ρ和距离值δ都较大时,该点才可能是簇中心点,其中ρ和δ的定义如下:

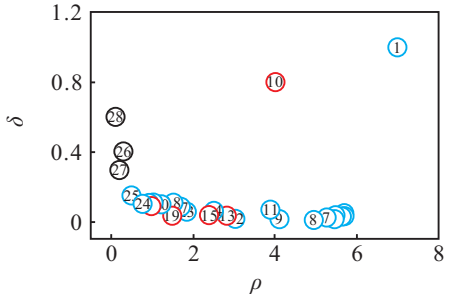
$$\rho_i = \sum_j \chi(d_{ij} - d_c), \tag{1}$$

$$\delta_i = \min(d_{ij}). \tag{2}$$

这里:d_{ij}为样本点i与样本点j之间的距离;d_c为截断距离,一般需要人工指定;$\chi(x)$为0-1函数,当$x < 0$时,$\chi(x) = 0$,当$x \geqslant 0$时,$\chi(x) = 1$.



(a) 按照密度降序排列的样本点



(b) 各样本点的决策图

图 2　决策图

根据样本点的局部密度和距离的值,聚类中心可以很直观地被选取,文献[7]以图2所示的数据集说明了决策图选取簇中心的过程. 图2(a)中一共有28个按照密度降序排列的样本点,这些样本点可以分为两个簇;图2(b)是分别以ρ为横轴、δ为纵轴绘制的决策图,由决策图可知,1号和10号样本点位于决策图的最右上角,局部密度和距离均较大,是簇中心点.

2　基于相对密度和决策图的聚类算法

为有效处理数据集中各个簇之间密度差别大对聚类结果产生的不利影响,需要充分考虑各样本点蕴涵的局部信息特征,用以揭示样本点与其相邻样本点的相对紧密程度及类属关系,从而解决数据集中簇间密度相差较大时聚类困难的问题. 因此,在阐述本文算法之前,先给出相对密度的相关定义.

2.1　相关定义

定义1　样本点p的k距离:k为任意正整数,样本点p的k距离是p到它的k最近邻的最大距离,记为$k_\mathrm{dist}(p)$.

定义2　样本点p的k距离邻域:p为数据集D中的一个样本点,$k_\mathrm{dist}(p)$为样本点p的k距离,样本点p的k距离邻域包含与p的距离不超过$k_\mathrm{dist}(p)$的样本点,记作$N(p)$.

定义3　样本点p相对于样本点o的可达距离:k为任意正整数,样本点p相对于样本点o的可达距离为$\mathrm{reach_dist}(p,o) = \max\{k_\mathrm{dist}(o), \mathrm{dist}(p,o)\}$,其中$\mathrm{dist}(p,o)$为样本点$p$与$o$之间的欧氏距离.

定义4　样本点p的局部密度:$|N(p)|$为样本点p的k距离邻域内所包含的样本点的数量,样本点p的局部密度是样本点p的k距离邻域中各样本点的平均可达密度的倒数,记作$\mathrm{ld}(p)$. 根据定义可得

$$\mathrm{ld}(p) = |N(p)| \Big/ \Big(\sum_{o \in N(p)} (\mathrm{reach\_dist}(p,o)) \Big).$$

对于给定的k值,样本点p周围的样本点越密集,其k近邻距离越小,局部密度越大,因此,局部密度从另一个侧面刻画了样本点的密度. 为分析样本点与其相邻样本点的相对疏密程度,引入数据集各样本点的k近邻信息,定义样本点的相对密度.

定义5　样本点p的相对密度:样本点p的局部密度与其k距离邻域内所有样本点的局部密度平均值之比,称为样本点p的相对密度,记作$\rho_r(p)$,即$\rho_r(p) = \mathrm{ld}(p) \times |N(p)| \Big/ \Big(\sum_{q \in N(p)} \mathrm{ld}(q) \Big)$.

定义6　核心样本点:对于样本点p,如果p的相对密度大于其k距离邻域内样本点的相对密度的均值,则称样本点p为核心样本点.

采用相对密度作为度量各个样本点密度的尺度,核心样本点就不一定是样本点密集区域内的点. 一个样本点是否是核心样本点与该点k距离范围内样本点的相对紧密程度有关,以图3为例(图中实心星号所示为采用相对密度计算得到的核心样本点,空心圆为非核心样本点),左侧圆形内样本点明显比右边菱形内样本点稀疏,如果从全局密度考量,核心样本点基本上都会位于菱形内,而采用相对密度作为密度的度量尺度,就可以揭示数据集的内在特征,在样本点稀疏的区域(圆形区域)也同样会发现核心样本点.

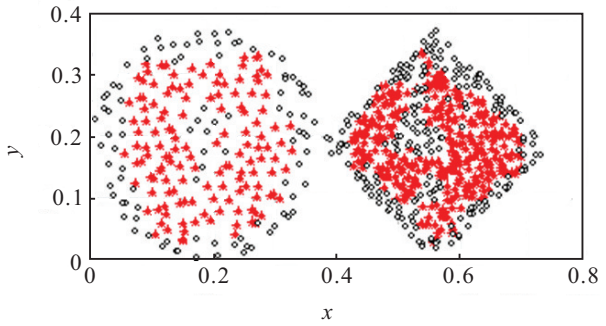
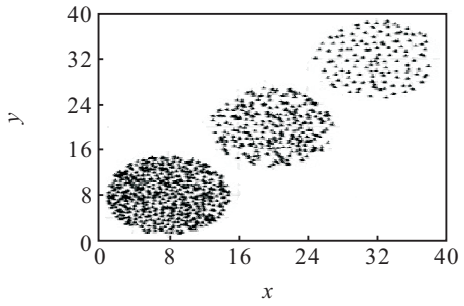
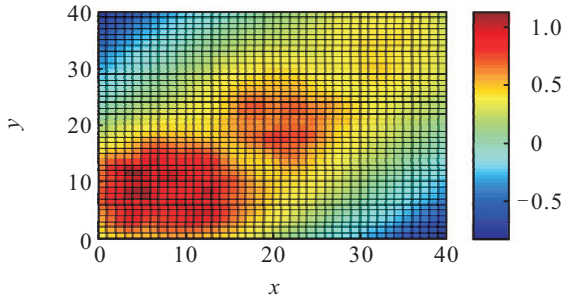


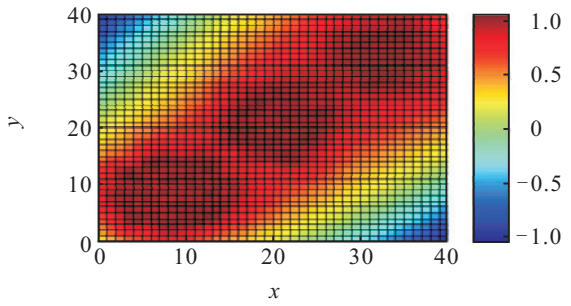
图3 采用相对密度计算的核心样本点



(a) 密度不同的3个圆形簇



(b) 各样本点局部密度



(c) 各样本点相对密度

图4 相对密度和局部密度

一般而言,全局密度或局部密度是从整个数据集统一或局部的角度来反映某一个样本点周围数据特征,而不能反映出一个样本点与其周围其他样本点的相对密度关系,而相对密度可以用来刻画数据集中各样本点与周围其他样本点的相对疏密程度. 以图4为例来说明相对密度和局部密度的区别. 图4(a)是3个样本点疏密程度不同的圆形簇. 图4(b)是采用CFSFDP算法定义的局部密度计算得出的各样本点的局部密度图,从图4(b)中可以看出,局部密度基本上是各个样本点与其周围样本点疏密的客观反映,样本点稠密的簇,局部密度值就大,反之就小,在采用密度的方法聚类时,容易出现由于簇间密度不均衡而产生的聚类不准确的问题. 图4(c)是采用定义5计算得到的各样本点的相对密度图,可以看出,虽然3个圆形簇中各样本点的疏密程度不尽相同,但是各个簇中各样本点相对密度值的分布基本一致,这揭示出了数据集中样本点分布的相对疏密特征,3个簇中各样本点的相对密度基本均等,这也说明相对密度可以作为统一度量数据集疏密的尺度,在采用基于密度的方法聚类时,就能很好地区分不同密度的簇,实现对变密度数据集的正确聚类.

2.2 算法描述

根据2.1节的定义,提出一种新颖的基于相对密度和决策图的聚类算法RDCA(Relative density-based clustering algorithm by using decision graph). RDCA算法首先计算数据集中各个样本点的相对密度ρ_r和到相对密度比它大的样本点的距离δ,通过构造相对密度ρ_r和距离δ的决策图确定聚类的中心点,在中心点确立后,由中心点开始向外扩展核心样本点进行簇的聚集,在扩展过程中,根据样本点的k近邻信息和相对密度判断样本之间的紧密程度,同时根据样本点的类别判别簇的边界,从而实现所有样本点的聚类. RDCA算法的具体步骤如下(其中k为输入参数).

Step 1: 计算各样本点之间的距离,并根据定义1~定义4计算各个样本点的k距离、k距离邻域、可达距离和局部密度;

Step 2: 根据定义5和式(2)计算各样本的相对密度ρ_r和到相对密度比该点大的样本点的距离δ值;

Step 3: 绘制决策图,根据ρ_r和δ值,利用决策图确定簇中心;

Step 4: 根据定义5和定义6对样本点进行分类(中心点、核心点、非核心点),并将所有样本点标记为未访问;

Step 5: 从中心点集合中选出一个簇中心o_i,创建一个簇C_i;

Step 6: 针对每个中心样本点o_i开始进行扩展,将o_i的k近邻邻居纳入簇C_i,并判断其k近邻邻居是否为核心点,并记作已访问,如果是核心样本点,则执行Step 6,如果是非核心样本点,则扩展到该点结束;

Step 7: 对于Step 6执行完之后仍然标记为未访问的样本点,统计其邻域样本点的簇归属,该样本点的类属与包含样本点最多的簇一致,并记作已访问,若该样本点邻域内所有样本点均标记为未访问的样本点,则从其邻域中选择一个样本点重复Step 7,直至所有标记为未访问的样本点标记为已访问.

2.3　算法分析

RDCA算法以相对密度作为度量密度的尺度,每个样本点密度值的计算范围限制在其k近邻范围内的样本点,其值仅仅与其k个近邻样本有关,这样更能揭示样本点与其k近邻样本点的相对紧密程度,反映出每个样本点及其k近邻范围内样本点的局部信息,从而使得RDCA算法不仅适应簇间密度相对均匀的数据集,也适应簇间密度差别大的数据集.

RDCA算法时间复杂度由4部分组成: 计算样本点之间的距离、相对密度ρ_r和距离δ值,以及非中心样本点的分配. 在数据集规模为n的情况下: 1) 计算样本点之间的距离,时间复杂度为$O(n^2)$; 2) 在计算ρ_r时,需要寻找每个样本点的k个近邻,时间复杂度为$O(n)$,计算n个样本点的k个近邻的总的时间复杂度为$O(n^2)$; 3) 计算每个样本点的距离δ,时间复杂度为$O(n^2)$; 4) 分配非中心点,采用深度优先的方法,总时间复杂度为$O(n^2)$. 因此,RDCA算法的时间复杂度的量级为$O(n^2)$.

3　实验与结果分析

3.1　实验数据集和评价标准

使用4个典型的人造数据集和3个真实数据集来检验RDCA算法的有效性,并与DBSCAN算法、CFSFDP算法和文献[31]改进的CFSFDP算法(KNN-DPC算法)进行对比,其中CFSFDP算法计算d_c时,数据集DS2取值为所有样本点之间距离值按照升序排列的5%位置处的距离数值,其他数据集为2%.

实验数据集的属性见表1.

表 1　实验数据集属性表

数据集	样本点数量	簇个数/数据来源
DS1	788	7/文献[36]
DS2	240	2/文献[37]
DS3	800	4/本文生成
DS4	1 585	2/本文生成
Iris数据集	150	3/文献[38]
Wine数据集	178	3/文献[38]
Seed数据集	210	3/文献[38]

表1中: DS1～DS4为人造数据集,Iris数据集、Wine数据集和Seeds数据集为UCI数据库[38]中的真实数据集. 这7个数据集在样本点的密度分布、规模等方面有比较大的差异,选取这些数据集的主要目的是为了更好地验证RDCA算法对不同类型数据集的适应性和聚类效果.

采用两种方法评价实验结果: 一是可视化聚类效果,二是聚类准确性的量化分析. 可视化聚类效果可以直观、清晰地展示聚类结构和数据分布情况,是一种常用的聚类分析手段. 聚类准确性的量化分析可以精确地分析聚类效果,有很多量化指标可以采用,由于本文实验数据是有标记的数据集,选取各数据集聚类结果的纯度值(purity)作为衡量聚类算法性能的指标,其计算公式如下:

$$\text{purtiy} = \frac{1}{N}\sum_{i=1}^{c} a_i. \tag{3}$$

其中: N为样本点的数量,c为簇的个数,a_i为数据集的各个簇中对数据某一类分类正确的数量,评价指标的范围在0-1之间,一个较大的纯度值代表较好的聚类性能.

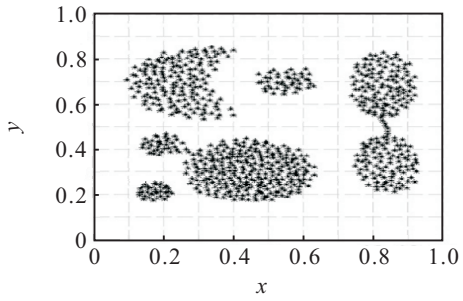
3.2　人造数据集实验结果对比分析

表1中DS1～DS4为人造数据集,其几何形状见图5. 数据集DS1和DS2来源于文献[36-37]中使用的数据集,数据集DS3和DS4为本文生成的两个二维数据集. 在这4个数据集中,DS1代表簇间密度相对均匀并且簇间相连的数据集种类,DS2代表簇间数据集相对均匀、簇间混叠并且有局部离群点的数据集种类,DS3代表簇间密度相差较大并且形状多变的数据集种类,DS4代表簇间密度相差较大并且簇相连接的数据集种类.

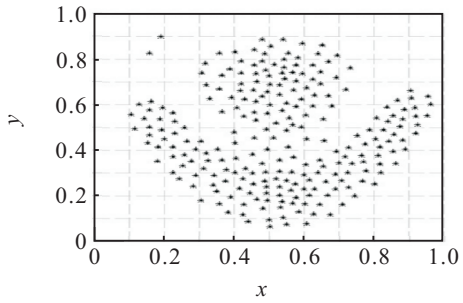
1) 聚类效果的可视化对比与分析

对二维人造数据集的聚类结果,采用不同形状的图形进行直观地展示. 图6～图9分别为RDCA算法、CFSFDP算法、KNN-DPC算法和DBSCAN算法对4个人造数据的聚类效果展示.

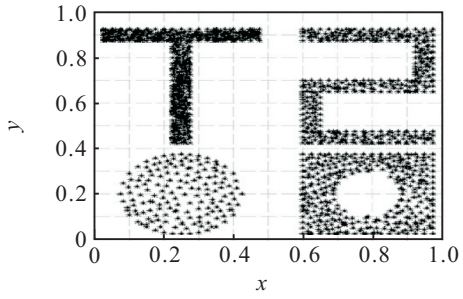
i) 图6是RDCA算法对数据集DS1～DS4的聚类结果. 由于RDCA算法的核心思想之一是发现数据集的簇中心,图形展示的结果中标出了簇中心点(实体正方形代表簇中心),以及选取中心点的决策图(决策图的坐标进行了归一化处理). 从图6中可以看出,RDCA算法能对各数据集给出符合直观判断和真实聚类的结果,并能有效处理这4类测试数据集中包含的非凸分布、疏密度不一、簇间混叠、簇间密度不均衡等特殊情况. 在聚类中心的选择上,从决策图中可以看出,采用相对密度作为度量样本点密度的尺



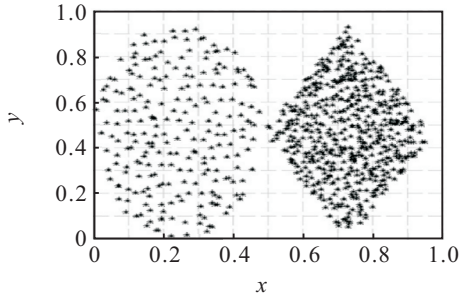
(a) 数据集 DS1(Aggregation 数据集)



(b) 数据集 DS2(Flame 数据集)



(c) 数据集 DS3

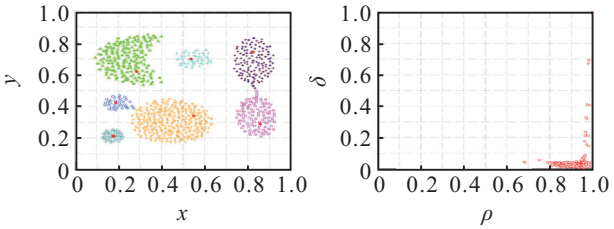


(d) 数据集 DS4

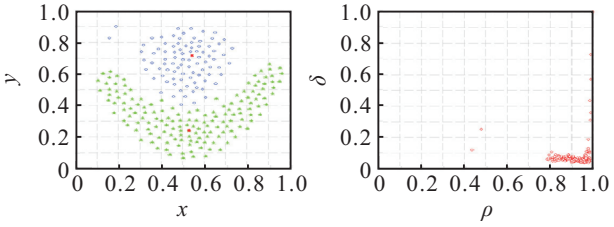
图 5 人造实验数据集的二维展示

度,正确识别出了各个数据集的簇中心. 相对密度能反映出样本点与其周围其他样本点的相对疏密程度,刻画了样本点所在位置的局部特征,从而反映出数据集中密度不同的子簇的特性,较好地处理了簇间密度差别过大带来的聚类中心决策困难的问题.

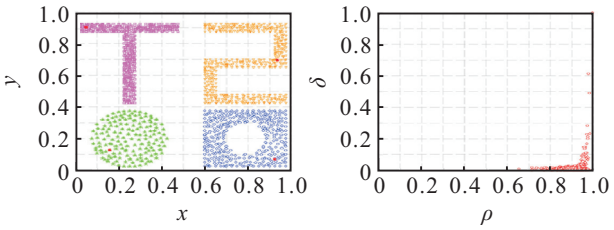
ii) 图 7 是 CFSFDP 算法对数据集 DS1 ~ DS4 的聚类结果. 从图 7 中可以看出,对于簇间密度相对均匀的数据集 (DS1 和 DS2),聚类效果较好,基本能够反映出各样本点的真实聚集情况,而对于簇间密度差异较大的数据集 (DS3 和 DS4),CFSFDP 算法没能正确识别聚类中心点,这主要是由于 CFSFDP 算法是选择局部密度较大的样本点作为簇中心,从而将局部密度小的簇排除在中心点选择范围之外,导致聚类错误. 对比图 6 可以看出,对于簇间密度相对均匀的数据集,RDCA 算法和 CFSFDP 算法均能取得较理想的聚类效果,而对于簇间密度差别较大的数据集,RDCA 算法的聚类效果明显优于 CFSFDP 算法.



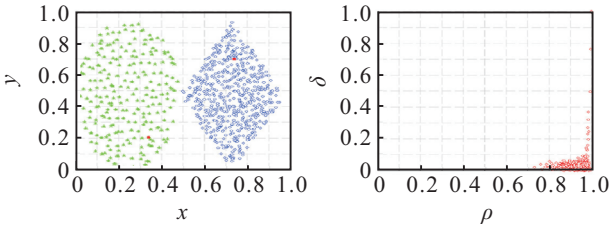
(a) 数据集 DS1



(b) 数据集 DS2



(c) 数据集 DS3



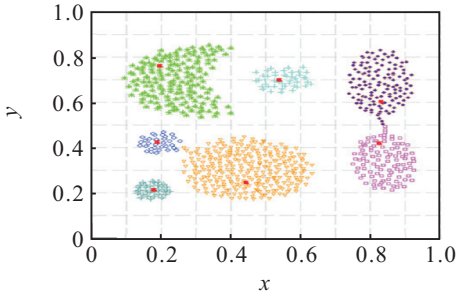
(d) 数据集 DS4

图 6 RDCA 算法在 DS1 ~ DS4 上聚类效果

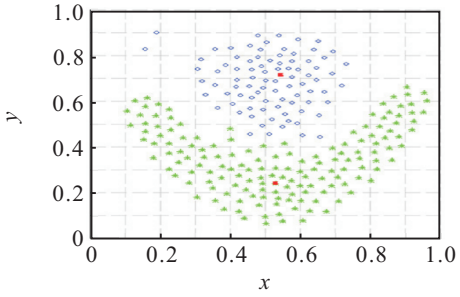
iii) 图 8 是文献 [31] 提出的 KNN-DPC 算法对数据集 DS1 ~ DS4 聚类的结果. 从图中可以看出,该算法对数据集 DS1 和 DS2 的聚类效果较好,但是对于簇间密度分布不均匀的数据集 (DS3 和 DS4) 的聚类效果较差. KNN-DPC 算法优化了 CFSFDP 算法的局部密度计算方法和非中心点的分配策略,但未解决簇间密度差别大对聚类结果不利影响的问题,因此对簇间密度分布不均匀类型数据集的聚类结果不理想.

iv) 图 9 显示了经典的 DBSCAN 算法对数据集 DS1 ~ DS4 的聚类结果 (图 9(c) 左下方和图 9(d) 左侧样本点是被 DBSCAN 算法作为噪声处理的样本点). 从图中可以看出,DBSCAN 算法对数据分布相对均匀的数据集 (DS1 和 DS2) 聚类效果较好,能正确识别

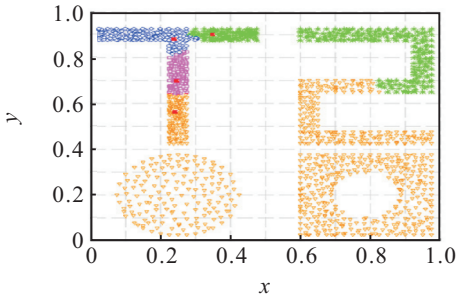
各个簇,而对簇间密度分布不均匀类型数据集(DS3和DS4)的聚类结果不理想,这主要是由于DBSCAN采取全局密度作为度量样本点密度的尺度,使得核心点集中在样本点稠密的簇中,而样本点稀疏的簇就被看作噪声点处理.



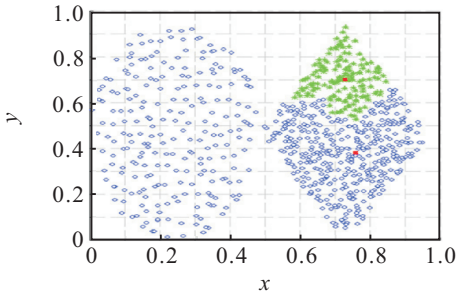
(a)　数据集 DS1



(b)　数据集 DS2



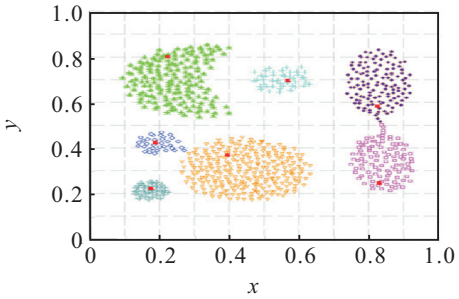
(c)　数据集 DS3



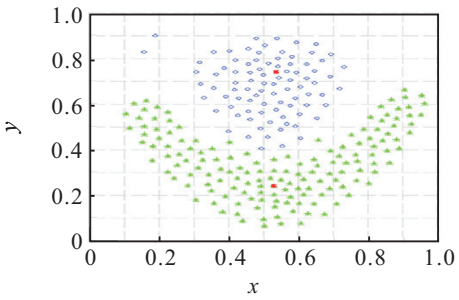
(d)　数据集 DS4

图 7　CFSFDP算法在DS1～DS4上聚类效果

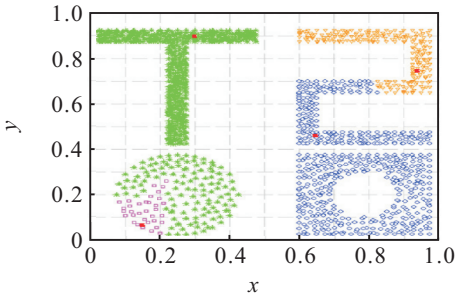
从图6～图9中各算法对不同类型数据集的聚类结果可以看出,对于簇间密度相对均匀的数据集,RDCA、CFSFDP、KNN-DPC和DBSCAN算法基本上能取得较好的聚类效果,4种算法聚类效果基本一致,而当数据集中簇间密度差别较大时,CFSFDP及其改进的KNN-DPC算法、DBSCAN算法的聚类结果不理想,而RDCA算法可以实现这类数据集的正



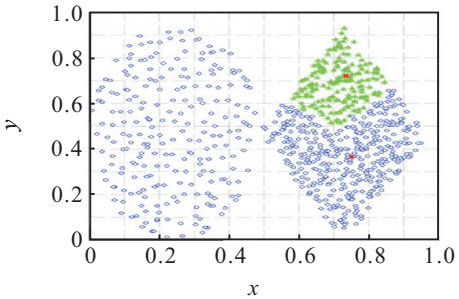
(a)　数据集 DS1(Aggregation 数据集)



(b)　数据集 DS2(Flame 数据集)



(c)　数据集 DS3



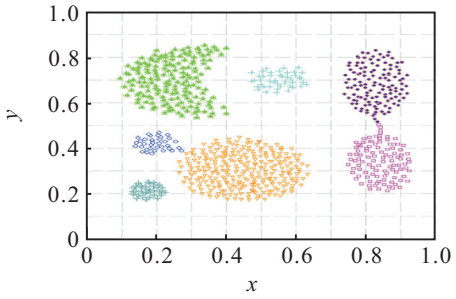
(d)　数据集 DS4

图 8　KNN-DPC算法在DS1～DS4上聚类效果

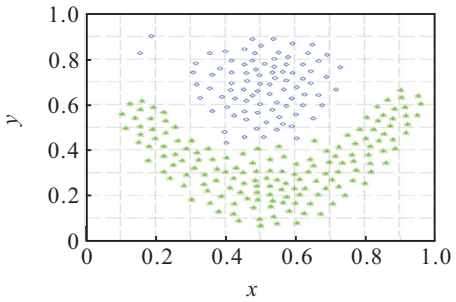
确聚类,因此,RDCA算法在聚类效果上优于经典的DBSCAN、CFSFDP和KNN-DPC算法,对不同类型数据集的适应性更广.

2) 聚类准确性量化指标的对比.

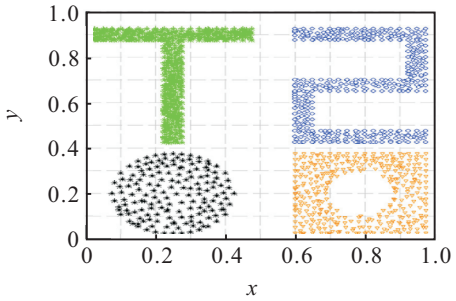
表2是4种对比算法对4个人造数据集聚类结果的purity评价指标值,由于CFSFDP算法和KNN-DPC算法对数据集DS3和DS4聚类错误,表中CFSFDP算法和KNN-DPC算法没有purity评价指标值的统计值. 从表2中可以看出,在簇间密度相对均匀的数据集(DS1和DS2)上,各算法聚类结果的纯度指标值基本相当,但是,对于簇间密度相差较大的数据集(DS3和DS4),RDCA算法明显优于其他算法.



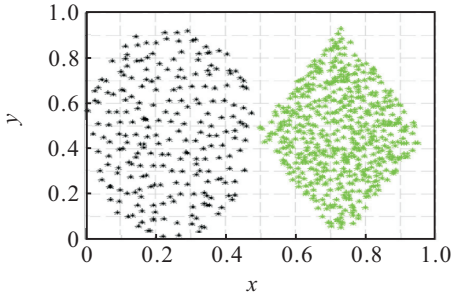
(a) 数据集 DS1



(b) 数据集 DS2



(c) 数据集 DS3



(d) 数据集 DS4

图 9 DBSCAN算法在DS1～DS4上聚类效果

表 2 人造数据集聚类准确性对比

数据集	RDCA	CFSFDP	KNN-DPC	DBSCAN
DS1	0.995	0.997	0.997	0.995
DS2	0.992	1.000	0.983	0.987
DS3	1.000	—	—	0.896
DS4	1.000	—	—	0.703

3.3 真实数据集实验结果对比分析

实验数据集中的Iris数据集、Wine数据集以及Seeds数据集是UCI数据库专门用于测试机器学习、数据挖掘算法性能的常用数据集,这3个数据集中的样本点都有确定的分类,因此可以通过聚类结果和标准的分类对比来精确地计算聚类结果的准确性. 表3是各算法在这3个真实数据集上聚类结果的purity评价指标值.

表 3 真实数据集聚类准确性对比

数据集	RDCA	CFSFDP	KNN-DPC	DBSCAN
Iris	0.980	0.940	0.973	0.793
Wine	0.944	0.883	0.932	0.665
Seeds	0.921	0.895	0.915	0.715

Iris数据集也称为鸢尾花数据集,包含3类植物(Setosa、Versicolour 和Virginica),每类植物各50个样本,共150个样本,该数据集以花瓣的长度、宽度及萼片的长度和宽度4个属性作为不同种类鸢尾花的特征. 数据集中Setosa簇中的样本点与其他簇中的样本点线性可分,Versicolour簇和Virginica簇中样本点非线性可分,从表3可以发现,RDCA算法、CFSFDP算法和KNN-DPC算法可将这3个簇分开,但RDCA算法效果最好,准确性为0.980(Virginica簇中有3个样本点分配到了Versicolour簇中),优于其他算法.

Wine数据集是对在意大利同一地区生产的3种不同品种葡萄酒进行大量分析所得出的数据,一共包含178个样本,每个样本的13个属性是葡萄酒的13种化学成分. 就聚类结果纯度值的指标而言,RDCA算法和KNN-DPC算法聚类精度相当,而DBSCAN算法聚类效果最差,主要是因为该数据集中存在样本点稀疏的簇,使得聚类精度偏低.

Seeds数据集是小麦种子数据集,用7个属性表征小麦种子的特性,该数据集包含3类不同的小麦种子,每一类种子70个样本点,一共210个样本. 从表3的实验结果可知,RDCA算法聚类精度优于KNN-DPC算法和CFSFDP算法,DBSCAN算法的聚类精度最低.

通过4种算法对UCI真实数据集的聚类准确性的对比分析可以发现,本文提出的RDCA算法具有良好的聚类性能,能发现真实数据集的簇分布特征,聚类性能优于CFSFDP算法、KNN-DPC算法和DBSCAN算法.

3.4 时间复杂度对比分析

1) 在数据集规模为n的情况下:i) 根据2.3节分析可知,RDCA算法的时间复杂度为$O(n^2)$. ii) CFSFDP算法的时间开销主要是在3个方面:一是计算数据集中样本点之间的距离,时间复杂度为$O(n^2)$;二是计算样本点i的局部密度ρ,需要考虑除样本点i之外的其他所有样本点,则搜索n个样本点的时间复杂度为$O(n^2)$;三是计算每个样本点的距离,时间复杂度为$O(n^2)$. 因此,CFSFDP算法的总的时间复杂度为$O(n^2)$. iii) DBSCAN算法中,密度可达对象的获取是通过不断执行区域查询实现的,在最坏情况下,算

法时间复杂度为$O(n^2)$. iv) KNN-DPC算法的时间复杂度为$O(n^2)$[31].

2) RDCA算法与CFSFDP算法和KNN-DPC算法相比,均需要计算样本间的距离,在这一点上,时间开销一致. 在计算样本点的密度时,CFSFDP算法中每个样本点局部密度的计算需要考虑除该点外的其他所有样本点,RDCA算法需要考虑样本点i的k个近邻样本点,在此基础上计算样本点i的局部可达密度,KNN-DPC算法也需要考虑样本点i的k个近邻样本点. 因此,在密度的计算上,RDCA算法时间开销要略大于CFSFDP算法,但是与KNN- DPC算法基本一致. 非中心样本点分配时,CFSFDP算法是将每个非中心样本点分配到距离其最近邻的且拥有高密度值样本点所在的类簇中,类簇分配只需一步即可完成,RDCA算法和KNN-DPC算法采用基于样本点k近邻信息的非中心样本点分配策略. 因此,RDCA算法分配非中心点的时间开销要高于CFSFDP算法,但与KNN-DPC算法时间开销大致相同. 总体而言,RDCA算法的实际时间消耗与KNN-DPC算法基本一致,但是要略微大于CFSFDP算法.

3) RDCA算法与DBSCAN算法相比,DBSCAN算法时间复杂度主要取决于区域查询的次数,这与输入参数密切相关,在最坏情况下的时间复杂度为$O(n^2)$. RDCA算法需要计算样本点之间距离、样本点相对密度和到相对密度比该点大的样本点的距离,除此之外还有非中心样本点的分配,各计算部分的时间复杂度均为$O(n^2)$,因此RDCA算法的时间复杂度要高于DBSCAN算法.

4) 从以上对各算法的时间复杂度的对比分析可以看出,RDCA算法、CFSFDP算法、KNN-DPC算法和DBSCAN算法的时间复杂度均为$O(n^2)$,虽然总体时间复杂度量级一致,但是RDCA算法在计算样本点相对密度时增加了时间开销,总体的时间开销与KNN-DPC算法基本一致,但是要略大于CFSFDP算法和DBSCAN算法.

4 结 论

变密度的数据集在实际应用中广泛存在,处理形状多样、密度不均匀数据集的聚类问题是基于密度的聚类方法所面临的一个难题,为此,本文从数据集所蕴涵的局部信息出发,引入样本点的k近邻信息,采用相对密度作为度量样本点密度的尺度,借鉴CFSFDP算法的决策图作为确定聚类中心点的方法,提出了一种基于相对密度和决策图的聚类算法. 在不同类型数据集上的实验表明,本文提出的算法能够对任意形状、密度不均匀的数据集有效聚类,聚类效果和聚类准确性均优于经典的CFSFDP算法、KNN-DPC算法和DBSCAN算法,并且对不同类型数据集的适应性更广. 但是,在算法的时间复杂性方面,虽然RDCA算法和CFSFDP算法、KNN-DPC算法和DBSCAN算法在同一个量级上,但是在相同条件下,RDCA算法在聚类过程的耗时上要略微大于CFSFDP算法和DBSCAN算法,如何降低算法的复杂性和参数k的自动选择问题是今后研究的重要工作.

参考文献(References)

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

[2] He H, Tan Y. Automatic pattern recognition of ECG signals using entropy-based adaptive dimensionality reduction and clustering[J]. Applied Soft Computing, 2017, 55(1): 238-252.

[3] Peng C, Kang Z, Xu F, et al. Image projection ridge regression for subspace clustering[J]. IEEE Signal Processing Letters, 2017, 24(7): 991-995.

[4] Röthlisberger V, Zischg A P, Keiler M. Identifying spatial clusters of flood exposure to support decision making in risk management[J]. Science of the Total Environment, 2017, 598(11): 593-603.

[5] Suzuki S, Kakuta M, Ishida T, et al. Faster sequence homology searches by clustering subsequences[J]. Bioinformatics, 2015, 31(8): 1183-1190.

[6] Ester M, Kriegel H P, Xu X. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise[C]. Int Conf on Knowledge Discovery and Data Mining. Portland: AAAI Press, 1996: 226-231.

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

[8] Han J, Kamber M, Pei J. Data mining: Concepts and techniques[M]. 3rd ed. Morgan Kaufman, 2011: 448-449.

[9] 李晓瑜, 俞丽颖, 雷航, 等. 一种k-means改进算法的并行化实现与应用[J]. 电子科技大学学报, 2017, 46(1): 61-68.
(Li X Y, Yu L Y, Lei H, et al. The parallel implementation and application of an improved k-means algorithm[J]. J of University of Electronic Science and Technology of China, 2017, 46(1): 61-68.)

[10] 李武, 赵娇燕, 严太山. 基于平均差异度优选初始聚类中心的改进K-均值聚类算法[J]. 控制与决策, 2017, 32(4): 759-762.
(Li W, Zhao J Y, Yan T S. Improved K-means clustering algorithm optimizing initial clustering centers based on average difference degree[J]. Control and Decision, 2017, 32(4): 759-762.)

[11] Oliveira G V, Coutinho F P, Campello R J G B, et al. Improving k-means through distributed scalable metaheuristics[J]. Neurocomputing, 2017, 246(7): 45-57.

[12] 雷小锋, 谢坤青, 林帆, 等. 一种基于K-Means局部最优性的高效聚类算法[J]. 软件学报, 2008, 19(7): 1683-1692.

(Lei X F, Xie K Q, Lin F, et al. An efficient clustering algorithm based on local optimality of K-Means[J]. J of Software, 2008, 19(7): 1683-1692.)

[13] Zhang T, Ramakrishnan R, Livny M. BIRCH: A new data clustering algorithm and its applications[J]. Data Mining and Knowledge Discovery, 1997, 1(2): 141-182.

[14] Guha S, Rastogi R. CURE: An efficient clustering algorithm for large database[J]. Information Systems, 2001, 26(1): 35-58.

[15] Wang W, Yang J, Muntz R R. STING: A statistical information grid approach to spatial data mining[C]. Int Conf on Very Large Data Bases. Athens: Morgan Kaufmann Publishers Inc, 1997: 186-195.

[16] Agrawal R, Gehrke J, Gunopulos D, et al. Automatic subspace clustering of high dimensional data for data mining applications[C]. Acm Sigmod Record. Washington: ACM, 1998: 94-105.

[17] 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.

[18] Hou J, Gao H, Li X. DSets-DBSCAN: A parameter-free clustering algorithm[J]. IEEE Trans on Image Processing, 2016, 25(7): 3182-3193.

[19] Smiti A, Elouedi Z. Fuzzy density based clustering method: Soft DBSCAN-GM[C]. The 8th Int Conf on Intelligent Systems. Sofia: Institute of Electrical and Electronics Engineers Inc, 2016: 443-448.

[20] Ienco D, Bordogna G. Fuzzy extensions of the DBScan clustering algorithm[J]. Soft Computing, 2016, 22(5): 1-12.

[21] 范敏, 李泽明, 石欣. 一种基于区域中心点的聚类算法[J]. 计算机工程与科学, 2014, 36(9): 1817-1822.
(Fan M, Li Z M, Shi X. A clustering algorithm based on local center object[J]. Computer Engineering & Science, 2014, 36(9): 1817-1822.)

[22] Hinneburg A, Keim D A. A general approach to clustering in large databases with noise[J]. Knowledge & Information Systems, 2003, 5(4): 387-415.

[23] Hinneburg A, Gabriel H H. DENCLUE 2.0: Fast clustering based on kernel density estimation[C]. Advances in Intelligent Data Analysis VII. Ljubljana: Springer Heidelberg, 2007: 70-80.

[24] Idrissi A, Rehioui H, Laghrissi A, et al. An improvement of DENCLUE algorithm for the data clustering[C]. The 5th Int Conf on Information & Communication Technology and Accessibility. Marrakech: Institute of Electrical and Electronics Engineers Inc, 2016: 1-6.

[25] Rehioui H, Idrissi A, Abourezq M, et al. DENCLUE-IM: A new approach for big data clustering[C]. The 7th Int Conf on Ambient Systems, Networks and Technologies. Madrid: Elsevier B V, 2016: 560-567.

[26] 朱亮, 李东波, 何非, 等. 采用改进型DENCLUE和SVM的电子皮带秤故障诊断[J]. 哈尔滨工业大学学报, 2015, 47(7): 122-128.
(Zhu L, Li D B, He F, et al. Fault diagnosis of belt weigher using the improved DENCLUE and SVM[J]. J of Harbin Institute of Technology[J]. 2015, 47(7): 122-128.)

[27] 贾培灵, 樊建聪, 彭延军. 一种基于簇边界的密度峰值点快速搜索聚类算法[J]. 南京大学学报: 自然科学, 2017, 53(2): 368-377.
(Jia P L, Fan J C, Peng Y J. An improved clustering algorithm by fast search and find of density peaks based on boundary samples[J]. J of Nanjing University: Natural Sciences, 2017, 53(2): 368-377.)

[28] Mehmood R, Bie R, Dawood H, et al. Fuzzy clustering by fast search and find of density peaks[C]. The 4th Int Conf on Identification, Information, and Knowledge in the Internet of Things. Beijing: Institute of Electrical and Electronics Engineers Inc, 2016: 258-261.

[29] Wang S L, Wang D K, Li C Y, et al. Clustering by fast search and find of density peaks with data Field[J]. Chinese J of Electronics, 2015, 25(3): 397-402.

[30] Mehmood R, Zhang G, Bie R, et al. Clustering by fast search and find of density peaks via heat diffusion[J]. Neurocomputing, 2016, 208(6191): 210-217.

[31] 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 Informations, 2016, 46(2): 258-280.)

[32] Zhou Y, Ting K M, Carman M J. Density-ratio based clustering for discovering clusters with varying densities[J]. Pattern Recognition, 2016, 60(11): 983-997.

[33] 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(5): 135 -145.

[34] Zhang Q C, Zhu C S, Yang L T, et al. An incremental CFS algorithm for clustering large data in industrial internet of things[J]. IEEE Trans on Industrial Informatics, 2017, 13(3): 1193-1201.

[35] Zhang Y F, Chen S M, Yu G. Efficient distributed density peaks for clustering large data sets in MapReduce[J]. IEEE Trans on Knowledge and Data Engineering, 2016, 28(12): 3218-3230.

[36] Gionis A, Mannila H, Tsaparas P. Clustering aggregation[J]. ACM Trans on Knowledge Discovery from Data (TKDD), 2007, 1(1): 1-30.

[37] Fu L, Medico E. FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data[J]. BMC Bioinformatics, 2007, 8(1): 1-15.

[38] Lichman M. UCI machine learning repository[EB/OL]. (1998-08-16)[2017-11-05]. http://archive.ics.uci.edu/ml/datasets.html.

(责任编辑：闫 妍)