控制与决策
Control and Decision

基于相对密度估计和多簇合并的密度峰值聚类算法

吴润秀, 尹士豪, 赵嘉, 李沛武, 刘宝宏

引用本文:
吴润秀,尹士豪,赵嘉,李沛武,刘宝宏. 基于相对密度估计和多簇合并的密度峰值聚类算法[J]. 控制与决策, 2023, 38(4): 1047–1055.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2021.1286

基于相对密度估计和多簇合并的密度峰值聚类算法

吴润秀, 尹士豪, 赵 嘉†, 李沛武, 刘宝宏

(南昌工程学院 信息工程学院，南昌 330099)

摘 要: 密度峰值聚类(DPC)算法是一种新颖的基于密度的聚类算法,其原理简单、运行效率高. 但DPC算法的局部密度只考虑了样本之间的距离,忽略了样本所处的环境,导致算法对密度分布不均数据的聚类效果不理想;同时,样本分配过程易产生分配错误连带效应. 针对上述问题,提出一种基于相对密度估计和多簇合并的密度峰值聚类(DPC-RD-MCM)算法. DPC-RD-MCM算法结合K近邻和相对密度思想,定义了相对K近邻的局部密度,以降低类簇疏密程度对类簇中心的影响,避免稀疏区域没有类簇中心;重新定义微簇间相似性度量准则,通过多簇合并策略得到最终聚类结果,避免分配错误连带效应. 在密度分布不均数据集、复杂形态数据集和UCI数据集上,将DPC-RD-MCM算法与DPC及其改进算法进行对比,实验结果表明: DPC-RD-MCM算法能够在密度分布不均数据上获得十分优异的聚类效果,在复杂形态数据集和UCI数据集的聚类性能上高于对比算法.

关键词: 密度峰值聚类；密度分布不均；K近邻；相对密度；簇间关联度；多簇合并

中图分类号: TP301.6 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2021.1286 **开放科学(资源服务)标识码(OSID):**

引用格式: 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[J]. 控制与决策, 2023, 38(4): 1047-1055.

Density peaks clustering based on relative density estimating and multi cluster merging

WU Run-xiu, YIN Shi-hao, ZHAO Jia†, LI Pei-wu, LIU Bao-hong

(School of Information Engineering，Nanchang Institute of Technology，Nanchang 330099，China)

Abstract: Density peaks clustering (DPC) is a novel clustering algorithm based on density, which has simple principle and high efficiency. However, the definition of local density of samples in DPC only considers the distance between samples and ignores the environment of samples, which leads to the unsatisfactory clustering effect of the algorithm for data with uneven density distribution. At the same time, the process of sample allocation is easy to produce the joint effect of allocation errors. To solve the above problems, this paper proposes a density peaks clustering algorithm based on relative density estimating and multi cluster merging (DPC-RD-MCM). The DPC-RD-MCM defines the local density of the relative K-nearest neighbor based on the idea of the K-nearest neighbor and relative density, so as to reduce the influence of cluster density on the selection of cluster centers and avoid the absence of cluster centers in sparse regions. The similarity measure between micro clusters is redefined, and the final clustering result is obtained by the multi cluster merging strategy, which avoids the joint effect of allocation errors. The DPC-RD-MCM is compared with DPC and its improved algorithm on uneven density datasets, complex morphological datasets and UCI datasets. The experimental results show that the DPC-RD-MCM can achieve excellent clustering effect on uneven density datasets, and the clustering performance of complex morphological datasets and UCI datasets is higher than other comparison algorithms.

Keywords: density peaks clustering; uneven density; K-nearest neighbor; relative density; correlation degree of cluster; multi cluster merging

0 引 言

随着信息技术的快速发展,数据来源的多样及数据量的高速增长,如何从数据中找到特定的规则并获取有价值的信息是人们密切关注的问题. 聚类是数据挖掘领域的一个研究热点,可以在没有任何先验知识的情况下发现数据的内在隐藏模式,已广泛应用于

收稿日期: 2021-07-22；**录用日期:** 2022-02-25.

基金项目: 国家自然科学基金项目(52069014)；江西省社会科学基金项目(21JY26)；江西省教育厅科技计划项目(GJJ180940).

责任编委: 胡清华.

†通讯作者. E-mail: zhaojia925@163.com.

信息检索[1]、模式识别[2]、市场分析[3]和图像处理[4]等领域.

2014年, Rodriguez等[5]在Science上提出了通过快速搜索和寻找密度峰值的聚类(clustering by fast search and find of density peaks)方法,简称密度峰值聚类(density peaks clustering, DPC)算法. DPC算法利用样本的局部密度和相对距离表征样本的空间分布,并通过决策图寻找到类簇中心,之后将剩余样本分配给局部密度比它大,且距离它最近的样本所在类簇. DPC算法具有思想新颖、易于实现、聚类效率高等优点,在多个领域得到了广泛认可.

DPC算法简单高效,但聚类过程存在如下问题: 1)当类簇之间的密度差异较大时,密集类簇内可能出现多个密度峰值,而稀疏类簇内无密度峰值,导致发生密度峰值选取错误,降低了算法的聚类性能; 2)分配策略容错性差,如果高密度样本被错误分配,则直接影响随后的低密度样本的分配.

针对DPC算法易发生密度峰值选取错误的问题, Wu等[6]提出了基于无维和反向K近邻的自适应密度峰值聚类(ERK-DPC)算法, ERK-DPC算法采用欧拉余弦距离代替欧氏距离,使用自适应局部密度公式计算样本局部密度; Zhang等[7]提出了一种基于密度衰减图的密度峰值聚类(DGDPC)算法,该算法使用密度衰减图自动形成初始簇,然后通过简单的方法合并簇,从而避免手动选择簇中心;丁世飞等[8]提出了一种基于不相似性度量优化的密度峰值聚类(DDPC)算法,该算法引入基于块的不相似性度量计算相似度矩阵,并基于新的相似度矩阵计算样本的K近邻信息并重新定义局部密度;金辉等[9]根据自然最近邻居的概念确定样本的局部密度,依据密度峰值局部密度最高且被稀疏区域分割来确定类簇中心; Fan等[10]提出了一种基于K近邻共享的密度峰值聚类(DPC-KNNS)算法,该算法利用共享近邻与自然近邻之间的相似性定义样本的局部密度并确定类簇中心; Zhao等[11]提出了基于圆形划分和网格相似度的密度峰值聚类(DPC-CP-GS)算法,该算法将数据空间划分为圆形网格,每个网格作为一个样本,用于确定类簇个数并寻找密度峰值.

针对DPC算法分配策略容错性差的问题, Xie等[12]提出了一种基于模糊加权K近邻的密度峰值聚类(FKNN-DPC)算法. 首先, FKNN-DPC算法筛选出数据集的核心点和离群点;随后,从每个类簇中心的K个最近邻点中搜索核心点并归到相应类簇;最后,计算样本对每个类簇的隶属度来分配离群点和未分配的核心点. Seyedi等[13]提出了基于动态图标签传播的密度峰值聚类(DPC-DLP)算法, DPC-DLP算法先将类簇中心与其K近邻点形成类簇主干,再采用基于图的标签传播将类簇主干的标签传播到剩余的样本; Bie等[14]提出了快速搜索和寻找密度峰值的自适应模糊聚类(Fuzzy-CFSFDP)算法, Fuzzy-CFSFDP算法将高于局部密度平均值的点作为局部类簇中心,随后样本被分配给最近的局部类簇中心所属的类簇,将密度峰值与边缘平均密度相近的簇合并; Zhuo等[15]提出了一种基于分层策略的密度峰值聚类(HCFS)算法,给出一种度量类簇间相异度和连通性的新机制,合并高相似度和高连通性的类簇,以增加不同类簇之间的差异,并得到最终的聚类结果;赵嘉等[16]提出了一种基于相互邻近度的密度峰值聚类(DPC-MND)算法,该算法定义一种样本相互邻近度的度量准则,先计算样本间的相互邻近度,再对已分配样本寻找相互邻近度最高的未分配样本,最后将未分配样本分配给已分配样本所在的类簇; Yuan等[17]提出了一种基于K近邻自适应合并策略的密度峰值聚类(KNN-ADPC)算法,该算法使用一种基于分段聚类的类簇合并策略; Guan等[18]提出了一种基于正常邻域和自适应子簇合并的聚类(NM-DPC)算法,通过生成策略生成子簇,获得样本的局部结构信息,保证子簇中的样本属于同一簇,再根据子簇的合并力自适应地合并子簇来获得聚类结果.

以上各改进算法均有效提高了DPC算法的聚类性能,但未同时考虑数据的密度差异和内部分布特征. 在处理密度分布不均数据时, DPC算法受到类簇间密度差异的影响,识别出的类簇中心可能都在密集类簇中,同时错误地把稀疏类簇的样本分配给密集类簇,从而降低了聚类质量. 为此,本文提出一种基于相对密度估计和多簇合并的密度峰值聚类(density peaks clustering based on relative density estimating and multi cluster merging, DPC-RD-MCM)算法,从局部密度的度量和剩余点的分配两方面对DPC算法进行优化. DPC-RD-MCM算法利用样本K近邻信息定义样本的初始密度,并将样本的初始密度与其K个最近邻样本的初始密度之和相除,以此重新定义样本的局部密度. 新定义的局部密度增强了样本与其K近邻样本之间的关系,能更客观地反映样本的分布特征;依据DPC的分配策略将数据集分成多个微簇,定义簇间相似性度量准则,并依此准则对微簇进行合并,可有效避免分配错误传播的问题.

1 DPC算法

DPC算法的样本包括两个属性:局部密度ρ_i和相对距离δ_i. DPC算法基于以下假设:类簇中心的局

部密度高于同一类簇的其他样本的局部密度,并且与任何具有较高局部密度的样本有较大的距离. 对于每个样本,局部密度有两种定义方式. 大规模数据集使用截断核,即由下式计算局部密度:

$$\rho_i = \sum_j \chi(d_{ij} - d_c),\ \chi(x) = \begin{cases} 0, & x \geqslant 0; \\ 1, & x < 0. \end{cases} \tag{1}$$

小规模数据集使用高斯核,即由下式计算局部密度:

$$\rho_i = \sum_j \exp\left(-\left(\frac{d_{ij}}{d_c}\right)^2\right). \tag{2}$$

其中: d_{ij} 是样本 i 与样本 j 之间的欧氏距离; d_c 是截断距离,是唯一的输入参数. 对于每个样本 i,相对距离 δ_i 由下式定义:

$$\delta_i = \min_{j\rho_j > \rho_i}(d_{ij}). \tag{3}$$

若样本 i 具有最大的局部密度,则 $\delta_i = \max_{i \neq j}(\delta_j)$. 计算所有样本的 ρ_i 和 δ_i,用 ρ_i 作为横坐标, δ_i 作为纵坐标,建立决策图,选取 ρ_i 和 δ_i 都较大的点作为类簇中心. 另外,类簇中心也可以通过决策值 γ_i 选取, γ_i 的定义如下:

$$\gamma_i = \rho_i \times \delta_i. \tag{4}$$

类簇中心确定后,剩余的样本分配给局部密度比它大且离它距离最近样本所在类簇.

2　DPC-RD-MCM算法

2.1　相对 K 近邻的局部密度

DPC算法基于截断距离 d_c 和全局范围的样本分布计算局部密度. DPC算法的局部密度主要考虑样本截断距离范围内的样本数量,导致密集区域样本的局部密度大于稀疏区域样本的局部密度,易出现选择的类簇中心集中于密集区域,稀疏区域没有类簇中心. 为此, DPC-RD-MCM算法结合 K 近邻与相对密度思想,重新定义样本相对 K 近邻的局部密度,即

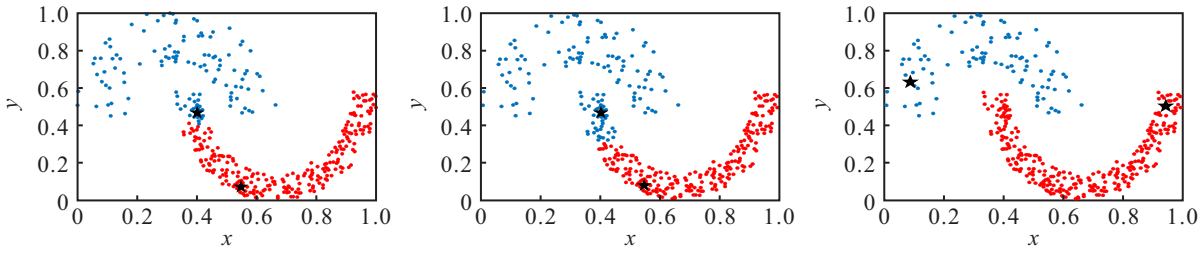
$$\overline{\rho_i} = \sum_{j \in \mathrm{knn}(i)} \exp(-d_{ij}^2), \tag{5}$$

$$\rho_i = \frac{\overline{\rho_i}}{\sum\limits_{j \in \mathrm{knn}(i)} \overline{\rho_j}}, \tag{6}$$

其中 $\mathrm{knn}(i)$ 是样本 i 的 K 个近邻构成的集合. 式(5)利用样本 K 近邻信息计算其初始密度,可以获取样本局部结构特征. 式(6)将样本的初始密度与其近邻样本初始密度之和相除得到样本的局部密度,可以减小密集区域样本的局部密度,放大稀疏区域样本的局部密度,进而提高稀疏区域样本被选为类簇中心的概率,降低类簇间密度差异对选取类簇中心的影响.

相对 K 近邻的局部密度充分考虑了 K 近邻范围内的样本对其局部密度的影响,可以很好地表征低密度区域的样本,有利于找出稀疏类簇的密度峰值,从而提高对簇间密度差别较大数据集的聚类效果.

图1显示了采用不同局部密度定义方式寻找到的Jain数据集的类簇中心,类簇中心用五角星表示. 观察图1可以发现, DPC算法选取的两个类簇中心均在下方的密集类簇上,导致聚类效果较差. 相对 K 近邻的局部密度考虑了类簇间的密度差异,可以准确找到稀疏类簇的类簇中心,提高对密度分布不均数据的聚类效果.



(a) 高斯核的局部密度　　(b) 截断核的局部密度　　(c) 相对 K 近邻的局部密度

图1　采用不同局部密度定义方式寻找到的Jain数据集的类簇中心

2.2　多簇合并策略设计

为避免DPC算法的分配错误连带效应,本文先采用DPC分配策略将数据集分成多个微簇,然后通过计算簇间关联度对这些微簇进行合并.

定义1　样本 i 与样本 j 之间的相似度 w_{ij} 为

$$w_{ij} = \frac{1}{1 + d_{ij}^2}. \tag{7}$$

样本 i 与样本 j 之间的距离越小,它们之间的关系越密切,相似度越高.

定义2　样本 i 与微簇 j 的关联度为 $R_{i \to c_j}$, $R_{i \to c_j}$ 越大,微簇对样本点的吸引程度越大.

$$R_{i \to c_j} = \frac{\sum\limits_{v \in c_j} w_{iv}}{|c_j|}. \tag{8}$$

其中: c_j 为微簇 j 的样本集合, $|c_j|$ 为微簇 j 中的样本个数.

定义 3 两微簇的关联度为 R_{c_i,c_j}，R_{c_i,c_j} 代表两个微簇属于同一类簇的可能性.

$$R_{c_i,c_j}=\sum_{v\in c_i}R_{v\to c_j}+\sum_{v\in c_j}R_{v\to c_i}. \tag{9}$$

根据式(6)计算样本的局部密度 ρ_i，式(3)计算样本的相对距离 δ_i；由式(4)计算 γ_i，并对 γ_i 排序，选择前 n 个样本作为最终生成类簇的密度峰值，选择前 $m\,(n\leqslant m)$ 个样本作为潜在的密度峰值. 将前 m 个样本标记为已分配样本，剩余样本分配给密度比它高且离它距离最近样本所在类簇，生成 m 个微簇. 为合并生成的 m 个微簇，计算生成微簇间的关联度.

针对 DPC 算法的分配策略容错性差的问题，基于簇间关联度对潜在类簇进行合并. 合并类簇的步骤为：合并 R_{c_i,c_j} 最高的两个类簇，判断合并后的类簇个数是否与最终类簇个数相等，若不相等，则重复上述操作；若相等则结束此过程，得到最终的聚类结果.

2.3 算法步骤

输入：数据集 X、参数 K；

输出：聚类 C.

step 1：数据预处理，对数据进行归一化；

step 2：计算样本间的欧氏距离，使用式(6)计算样本的局部密度 ρ_i，式(3)计算样本的相对距离 δ_i；

step 3：根据式(4)计算 γ_i，选择潜在类簇的密度峰值集合；

step 4：将潜在类簇的密度峰值标记为已分配样本，剩余的样本分配给局部密度比它大且离它距离最近样本所在类簇；

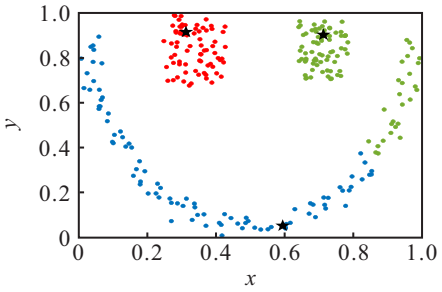
step 5：根据式(7)计算样本的相似度 w_{ij}、式(8)计算样本与微簇的关联度 $R_{i\to c_j}$；

step 6：根据式(9)计算各微簇间的关联度 R_{c_i,c_j}，建立关联度矩阵；

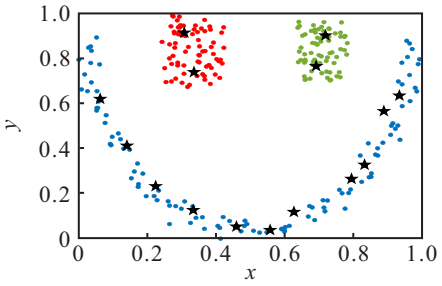
step 7：依次合并关联度最高的两个微簇，直到潜在类簇个数与真实类簇个数相等为止.

2.4 多簇合并策略的有效性分析

为验证多簇合并策略的有效性，图 2 给出了 DPC-RD-MCM 算法和 DPC 算法在 LineBlobs 数据集上的聚类结果. 图 2(a) 的 DPC 算法能准确找到类簇中心，但将右侧部分稀疏类簇样本错误分配给了中间的密集类簇. 图 2(b) 为 DPC-RD-MCM 算法的聚类结果，DPC-RD-MCM 算法选取了 15 个密度峰值，选取的密度峰值并没有集中在中间两个密集类簇，而是均匀地分布在整个数据集中，通过多簇合并能准确地将样本分配给对应类簇，说明 DPC-RD-MCM 算法的样本分配策略能避免 DPC 算法的分配错误连带效应.



(a) DPC 算法



(b) DPC-RD-MCM 算法

图 2 DPC-RD-MCM 算法和 DPC 算法在 LineBlobs 数据集上的聚类结果

3 实验结果与分析

3.1 实验设置

为验证 DPC-RD-MCM 算法的聚类性能，使用密度分布不均数据集、复杂形态数据集和 UCI 数据集对算法进行测试. 本文选用的对比算法是基于萤火虫算法的改进密度峰值聚类 (IDPC-FA)[19]、FKNN-DPC [12]、基于加权局部密度序列和最近邻分配的密度峰值聚类 (DPCSA)[20]、基于模糊邻域的鲁棒密度峰值聚类 (FNDPC)[21] 和 DPC[5] 算法. 其中 IDPC-FA、DPCSA 和 DPC 算法代码由原作者提供，FKNN-DPC 和 FNDPC 算法参考原文献编程实现. 实验环境为 Win 10 64 bit 操作系统，Matlab R2020a 软件，12.0 GB 内存，Intel(R)Core(TM)i5-10210U CPU @ 1.60 GHz 处理器.

本文选择 3 个独立于标签绝对值的评价指标来评价聚类性能. 聚类评价指标包括调整兰德系数 (adjusted rand index, ARI)[22]、调整互信息 (adjusted mutual information, AMI)[22] 和 Fowlkes-Mallows 指数 (Fowlkes-Mallows index, FMI)[23]. 3 个指标的最好结果都为 1，数值越接近 1，聚类效果越好.

除 IDPC-FA 和 DPCSA 算法，其余算法都需要进行参数调优：DPC-RD-MCM 和 FKNN-DPC 算法需预先指定一个参数 K，K 值在 1～50 之间选取；DPC 算法的截断距离 d_c 由距离矩阵的百分比确定，参数设置在 0.01～5 之间；FNDPC 算法的参数 ε 在 0.01～1 之间选取；虽然 DPC-RD-MCM、IDPC-FA、FKNN-DPC、FNDPC、DPCSA 和 DPC 算法都有确定类簇个数的方法，但是该方式无法为所有数据集正确确定类

簇个数. 因此,本文人为设定类簇个数进行聚类.

3.2　密度分布不均数据集的实验结果分析

本文选用6个不同规模的密度分布不均数据集,其基本特征由表1给出. 表2显示了6种聚类算法在密度分布不均数据集上的AMI、ARI和FMI值,其中:最佳结果以粗体、加黑显示,Arg-是每个算法的最优参数取值,“–”表示没有参数.

表1　密度分布不均数据集的基本特征

数据集	数据来源	样本规模	数据维数	类簇个数
Jain	文献[24]	373	2	2
Cth	文献[25]	1 016	2	4
Ring	文献[26]	1 000	2	2
Ls	文献[25]	1 741	2	6
Compound	文献[27]	399	2	6
LineBlobs	文献[28]	266	2	3

表2　6种聚类算法在密度分布不均数据集上的聚类性能

数据集	聚类算法	AMI	ARI	FMI	Arg-	数据集	聚类算法	AMI	ARI	FMI	Arg-
Jain	DPC-RD-MCM	**1**	**1**	**1**	25	Cth	DPC-RD-MCM	**1**	**1**	**1**	7
	IDPC-FA	**1**	**1**	**1**	–		IDPC-FA	0.848 2	0.772 9	0.835 0	–
	FKNN-DPC	0.709 2	0.822 4	0.935 9	43		FKNN-DPC	**1**	**1**	**1**	22
	DPCSA	0.216 7	0.044 2	0.592 4	–		DPCSA	0.789 1	0.653 8	0.754 7	–
	FNDPC	0.596 1	0.725 7	0.905 1	0.47		FNDPC	0.875 8	0.832 7	0.878 6	0.45
	DPC	0.618 3	0.714 6	0.881 9	0.9		DPC	0.682 0	0.501 7	0.639 7	1.1
Ring	DPC-RD-MCM	**1**	**1**	**1**	14	Ls	DPC-RD-MCM	**1**	**1**	**1**	15
	IDPC-FA	**1**	**1**	**1**	–		IDPC-FA	0.707 6	0.627 4	0.732 5	–
	FKNN-DPC	**1**	**1**	**1**	6		FKNN-DPC	0.871 9	0.817 9	0.873 5	48
	DPCSA	**1**	**1**	**1**	–		DPCSA	0.725 2	0.599 9	0.712 9	–
	FNDPC	0.550 8	0.565 1	0.789 2	0.47		FNDPC	0.756 4	0.689 8	0.780 8	0.37
	DPC	0.282 8	0.214 7	0.661 6	0.6		DPC	0.766 5	0.689 4	0.777 9	0.91
Compound	DPC-RD-MCM	**0.885 4**	**0.892 8**	**0.921 6**	8	LineBlobs	DPC-RD-MCM	**1**	**1**	**1**	29
	IDPC-FA	0.792 2	0.832 7	0.881 5	–		IDPC-FA	0.837 5	0.823 7	0.884 2	–
	FKNN-DPC	0.838 1	0.841 8	0.887 7	7		FKNN-DPC	**1**	**1**	**1**	12
	DPCSA	0.839 2	0.828 4	0.870 7	–		DPCSA	**1**	**1**	**1**	–
	FNDPC	0.723 9	0.533 7	0.644 0	0.09		FNDPC	0.638 6	0.576 9	0.721 8	0.21
	DPC	0.775 4	0.591 0	0.687 6	4.0		DPC	0.779 9	0.721 0	0.816 6	3.7

从表2可以看出:DPC-RD-MCM算法对Jain、Cth、Ring、Ls、Compound和LineBlobs数据集均能获得最佳的聚类效果;IDPC-FA算法对Jain和Ring数据集的聚类效果较好,其余数据集的聚类效果欠佳;FKNN-DPC算法对Jain和Ls数据集的聚类效果欠佳,其余数据集的聚类效果较好;DPCSA算法仅在Ring和LineBlobs数据集上的聚类性能较好,在其余数据集上的聚类效果较差;FNDPC和DPC算法在6个数据集上的聚类效果均劣于DPC-RD-MCM和FKNN-DPC算法,可见其聚类性能存在较大差距. 综合比较6种算法在6个密度分布不均数据集的聚类结果可知,DPC-RD-MCM算法在密度分布不均数据集上的聚类性能十分优异.

图3和图4为DPC-RD-MCM、IDPC-FA、FKNN-DPC、FNDPC、DPCSA和DPC算法在Jain和Cth数据集上的聚类结果.

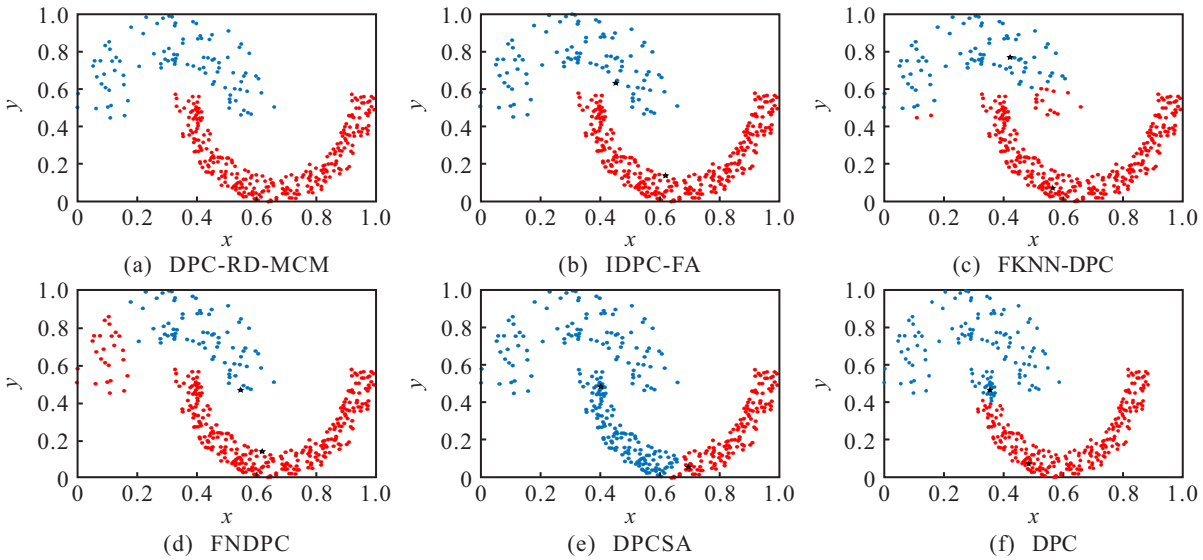


图3　6种算法在Jain数据集上的聚类结果

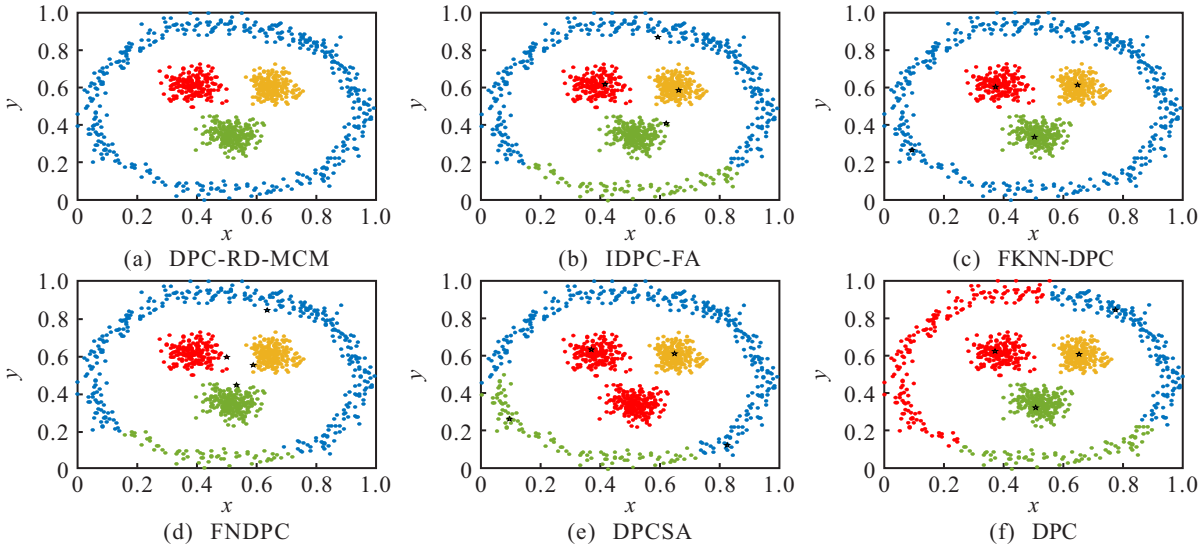


图4 6种算法在Cth数据集上的聚类结果

图3和图4中不同颜色的点代表不同的类簇,类簇中心用“五角星”表示.

图3为6种算法对Jain数据集的聚类结果. Jain数据集由两个新月形类簇组成,一个稀疏,另一个密集. 对于Jain数据集,DPC-RD-MCM和IDPC-FA算法可以获得准确的聚类结果;FKNN-DPC和FNDPC算法找到了正确的类簇中心,但在样本分配过程中,稀疏类簇中的一些样本被错误地分配给了密集类簇;DPCSA和DPC算法没有找到正确的类簇中心,导致聚类效果较差.

图4为6种算法对Cth数据集的聚类结果. Cth数据集由3个块状类簇与一个环形类簇组成,外部的环状类簇样本分布较稀疏. 从图4可以看出: DPC-RD-MCM和FKNN-DPC算法能准确地把样本分配给相应类簇. DPCSA算法错误地将中心的两个块状类簇合并为一个;IDPC-FA、DPC和FNDPC算法均将部分稀疏的环状类簇样本分配给附近的密集块状类簇,导致聚类结果较差.

3.3 复杂形态数据集的实验结果分析

复杂形态数据集是有着多尺度、交叉环绕或簇类形状各异等结构的数据集. 本文选用6个复杂形态数据集,其基本特征由表3给出. 表4显示了6种聚类算法在6个复杂形态数据集上的AMI、ARI和FMI值.

表3 复杂形态数据集的基本特征

数据集	数据来源	样本规模	数据维数	类簇个数
Aggregation	文献[29]	788	2	7
Flame	文献[30]	240	2	2
Pathbased	文献[31]	300	2	3
Spiral	文献[31]	312	2	3
R15	文献[32]	600	2	15
D31	文献[32]	3 100	2	31

表4 6种聚类算法在复杂形态数据集上的聚类性能

数据集	聚类算法	AMI	ARI	FMI	Arg-	数据集	聚类算法	AMI	ARI	FMI	Arg-
Aggregation	DPC-RD-MCM	0.992 2	0.995 6	0.996 6	20	Flame	DPC-RD-MCM	**1**	**1**	**1**	14
	IDPC-FA	**1**	**1**	**1**	–		IDPC-FA	**1**	**1**	**1**	–
	FKNN-DPC	0.990 5	0.994 9	0.996 0	20		FKNN-DPC	0.926 7	0.966 6	0.984 5	5
	DPCSA	0.953 7	0.958 1	0.967 3	–		DPCSA	**1**	**1**	**1**	–
	FNDPC	0.986 4	0.991 3	0.993 2	0.02		FNDPC	**1**	**1**	**1**	0.13
	DPC	0.992 2	0.995 6	0.996 6	4.00		DPC	**1**	**1**	**1**	2.8
Pathbased	DPC-RD-MCM	**0.940 1**	**0.959 0**	**0.972 7**	24	Sprial	DPC-RD-MCM	**1**	**1**	**1**	3
	IDPC-FA	0.844 2	0.859 3	0.906 7	–		IDPC-FA	**1**	**1**	**1**	–
	FKNN-DPC	0.930 5	0.949 9	0.966 5	9		FKNN-DPC	**1**	**1**	**1**	6
	DPCSA	0.707 3	0.613 3	0.751 1	–		DPCSA	**1**	**1**	**1**	–
	FNDPC	0.575 1	0.506 7	0.706 5	0.01		FNDPC	**1**	**1**	**1**	0.07
	DPC	0.521 2	0.471 7	0.666 4	3.8		DPC	**1**	**1**	**1**	1.8
R15	DPC-RD-MCM	**0.993 8**	**0.992 8**	**0.993 3**	16	D31	DPC-RD-MCM	0.958 2	0.940 7	0.942 6	25
	IDPC-FA	**0.993 8**	**0.992 8**	**0.993 3**	–		IDPC-FA	0.957 5	0.940 2	0.942 1	–
	FKNN-DPC	**0.993 8**	**0.992 8**	**0.993 3**	25		FKNN-DPC	**0.965 4**	**0.952 3**	**0.953 8**	28
	DPCSA	0.988 5	0.985 7	0.986 6	–		DPCSA	0.955 2	0.935 3	0.937 4	–
	FNDPC	**0.993 8**	**0.992 8**	**0.993 3**	0.03		FNDPC	0.955 5	0.936 4	0.938 5	0.04
	DPC	**0.993 8**	**0.992 8**	**0.993 3**	0.7		DPC	0.955 4	0.936 5	0.938 5	0.6

从表4可以看出: Aggregation数据集上, IDPC-FA算法取得了最佳的聚类效果; Flame数据集上, FKNN-DPC算法聚类效果欠佳, 其余算法的聚类效果很好; Pathbased数据集上, DPC-RD-MCM算法聚类性能最优; Spiral数据集上, 6种算法均能准确聚类; R15和D31数据集上, 各算法聚类结果差别不大, DPCSA算法在R15数据集上聚类效果较差, FKNN-DPC在D31数据集上聚类效果最优. 综合比较表4的聚类结果可知, DPC-RD-MCM算法在复杂形态数据集上的聚类性能较为优秀.

3.4　UCI数据集的实验结果分析

UCI数据集能有效测试各算法在不同数据集上识别类簇的能力. 本文选用10个UCI数据集测试算法性能. 选用的UCI数据集的基本特征如表5所示. 表6给出了6种算法在UCI数据集上的聚类性能.

表5　UCI数据集的基本特征

数据集	数据来源	样本规模	数据维数	类簇个数
Iris	文献[31]	150	4	3
Wine	文献[33]	178	13	3
Seeds	文献[34]	210	7	3
Ecoli	文献[33]	336	8	8
Ionosphere	文献[35]	351	34	2
Libras	文献[29]	360	90	15
Dermatology	文献[33]	366	33	6
Wdbc	文献[36]	569	30	2
Parkinsons	文献[36]	197	23	2
Waveform(noise)	文献[36]	5 000	40	3

表6　6种聚类算法在UCI数据集上的聚类性能

数据集	聚类算法	AMI	ARI	FMI	Arg-	数据集	聚类算法	AMI	ARI	FMI	Arg-
Iris	DPC-RD-MCM	**0.896 8**	**0.922 2**	**0.947 8**	7	Wine	DPC-RD-MCM	0.792 8	0.800 1	0.867 2	24
	IDPC-FA	0.862 3	0.885 7	0.923 3	–		IDPC-FA	0.767 5	0.771 3	0.847 8	–
	FKNN-DPC	0.883 1	0.903 8	0.935 5	22		FKNN-DPC	**0.848 1**	**0.883 9**	**0.922 9**	8
	DPCSA	0.883 1	0.903 8	0.935 5	–		DPCSA	0.748	0.741 4	0.828 3	–
	FNDPC	0.883 1	0.903 8	0.935 5	0.11		FNDPC	0.789 8	0.802 5	0.868 6	0.26
	DPC	0.724 7	0.703 7	0.803 2	0.2		DPC	0.706 5	0.672 4	0.783 5	2
Seeds	DPC-RD-MCM	0.743 0	0.788 6	0.858 4	26	Ecoli	DPC-RD-MCM	0.643 0	0.755 4	**0.831 9**	36
	IDPC-FA	0.729 9	0.767 0	0.844 4	–		IDPC-FA	**0.663 8**	**0.756 1**	0.828 4	–
	FKNN-DPC	**0.775 7**	**0.802 4**	**0.868 2**	9		FKNN-DPC	0.587 8	0.589 4	0.702 7	2
	DPCSA	0.660 9	0.687 3	0.791 8	–		DPCSA	0.440 6	0.459 3	0.646 7	–
	FNDPC	0.713 6	0.754 5	0.836 1	0.07		FNDPC	0.483 3	0.561 8	0.717 8	0.35
	DPC	0.729 8	0.767 0	0.844 4	0.7		DPC	0.497 8	0.446 5	0.577 5	0.4
Dermatology	DPC-RD-MCM	0.858 1	0.846 8	0.884 5	4	Wdbc	DPC-RD-MCM	0.636 0	0.748 4	0.885 8	30
	IDPC-FA	**0.863 8**	**0.877 2**	**0.901 8**	–		IDPC-FA	0.623 7	0.742 3	0.882 9	–
	FKNN-DPC	0.806 6	0.836 1	0.870 9	35		FKNN-DPC	**0.642 3**	**0.761 3**	**0.889 4**	2
	DPCSA	0.745 1	0.606 2	0.689 6	–		DPCSA	0.336 1	0.377 1	0.759 5	–
	FNDPC	0.789 8	0.799 5	0.841 8	0.17		FNDPC	0.607 6	0.730 5	0.875 8	0.05
	DPC	0.608 6	0.611 0	0.705 6	1.5		DPC	0.000 7	0.002 8	0.725 7	1.2
Inonsphere	DPC-RD-MCM	0.334 2	0.442 0	**0.772 3**	15	Libras	DPC-RD-MCM	**0.618 7**	**0.424 8**	**0.478 2**	14
	IDPC-FA	0.135 5	0.218 3	0.643 2	–		IDPC-FA	0.573 3	0.381 6	0.424 7	–
	FKNN-DPC	**0.348 5**	**0.479 0**	0.771 6	8		FKNN-DPC	0.555 4	0.345 9	0.404 4	10
	DPCSA	0.133 5	0.213 5	0.639 0	–		DPCSA	0.538 8	0.309 5	0.379 1	–
	FNDPC	0.163 0	0.248 3	0.651 3	0.06		FNDPC	0.549 4	0.329 0	0.386 9	0.17
	DPC	0.135 5	0.218 3	0.643 2	0.5		DPC	0.535 8	0.319 3	0.371 7	0.3
Parkinsons	DPC-RD-MCM	**0.272 8**	**0.391 0**	**0.832 2**	8	Waveform (noise)	DPC-RD-MCM	0.218 8	**0.221 0**	**0.543 6**	4
	IDPC-FA	0.215 1	0.363 2	0.819 0	–		IDPC-FA	**0.254 6**	0.219 8	0.498 5	–
	FKNN-DPC	0.072 8	0.160 1	0.658 2	7		FKNN-DPC	0.071 1	0.012 2	0.502 5	6
	DPCSA	0.177 2	0.268 6	0.814 0	–		DPCSA	0.152 4	0.134 9	0.462 3	–
	FNDPC	0.215 1	0.363 2	0.819 0	0.04		FNDPC	0.159 6	0.164 1	0.489 1	0.07
	DPC	0.247 8	0.125 6	0.618 7	1.2		DPC	0.089 6	0.069 5	0.458 0	2.1

表6中: DPC-RD-MCM算法在Iris、Libras、Parkinsons和Waveform (noise)数据集上取得了最优的聚类效果; FKNN-DPC算法在Wine、Seeds、Wdbc和Ionosphere数据集上聚类性能最好; IDPC-FA算法

在Ecoli和Dermatology数据集上的聚类性能最优. 综合比较表6的聚类结果可知,DPC-RD-MCM算法在UCI数据集上的聚类性能较为优秀.

上述3组实验的结果表明: DPC-RD-MCM算法对密度分布不均数据的聚类效果十分优异;对多尺度、交叉环绕的复杂形态数据集和真实数据集也有不错的聚类效果,是一种对各类数据均具有较好普适性的聚类算法.

4 结 论

DPC算法对密度分布不均数据集的聚类效果不理想,分配策略容错性差. 针对以上问题,本文提出了一种基于相对密度估计和多簇合并的密度峰值聚类算法. 该算法引入K近邻及相对密度思想,重新定义样本的局部密度和寻找密度峰值,找到密度峰值后,按照DPC的分配方式把样本分成多个微簇,计算微簇之间的关联度,据此合并生成的微簇. DPC-RD-MCM算法有效避免了DPC算法样本分配过程中存在的错误连带效应,提高了对密度分布不均数据集的聚类效果. 实验表明,DPC-RD-MCM算法对密度分布不均数据集的聚类准确度高,在复杂形态数据集和UCI真实数据集的聚类性能高于对比算法. 本算法用K取代d_c,使参数更容易确定,但并未实现无参聚类;本算法也未能针对高维数据进行改进,如何在保证聚类性能的条件下实现无参聚类并提高算法对高维数据集的聚类效果是下一步的研究重点.

参考文献(References)

[1] Sun L, Liu R N, Xu J C, et al. An adaptive density peaks clustering method with fisher linear discriminant[J]. IEEE Access, 2019, 7: 72936-72955.

[2] 张曦, 李璠, 付雪峰, 等. 随机学习萤火虫算法优化的模糊软子空间聚类算法[J]. 江西师范大学学报: 自然科学版, 2021, 45(2): 137-144.
(Zhang X, Li F, Fu X F, et al. The fuzzy soft subspace clustering algorithm optimized by random learning firefly algorithm[J]. Journal of Jiangxi Normal University: Natural Science Edition, 2021, 45(2): 137-144.)

[3] Morris K, Mcnicholas P D. Clustering, classification, discriminant analysis, and dimension reduction via generalized hyperbolic mixtures[J]. Computational Statistics & Data Analysis, 2016, 97: 133-150.

[4] Ducournau A, Bretto A, Rital S, et al. A reductive approach to hypergraph clustering: An application to image segmentation[J]. Pattern Recognition, 2012, 45(7): 2788-2803.

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

[6] Wu Q N, Zhang Q Q, Sun R Z, et al. Adaptive density peak clustering based on dimensional-free and reverse k-nearest neighbors[J]. Information Technology and Control, 2020, 49(3): 395-411.

[7] Zhang Z Y, Zhu Q S, Zhu F, et al. Density decay graph-based density peak clustering[J]. Knowledge-Based Systems, 2021, 224: 107075.

[8] 丁世飞, 徐晓, 王艳茹. 基于不相似性度量优化的密度峰值聚类算法[J]. 软件学报, 2020, 31(11): 3321-3333.
(Ding S F, Xu X, Wang Y R. Optimized density peaks clustering algorithm based on dissimilarity measure[J]. Journal of Software, 2020, 31(11): 3321-3333.)

[9] 金辉, 钱雪忠. 自然最近邻优化的密度峰值聚类算法[J]. 计算机科学与探索, 2019, 13(4): 711-720.
(Jin H, Qian X Z. Optimized density peak clustering algorithm by natural nearest neighbor[J]. Journal of Frontiers of Computer Science and Technology, 2019, 13(4): 711-720.)

[10] Fan T H, Yao Z F, Han L Z, et al. Density peaks clustering based on k-nearest neighbors sharing[J]. Concurrency and Computation: Practice and Experience, 2021, 33(5): e5993.

[11] Zhao J, Tang J, Fan T, et al. Density peaks clustering based on circular partition and grid similarity[J]. Concurrency and Computation: Practice and Experience, 2020, 32(7): e5567.

[12] Xie J Y, Gao H C, Xie W X, 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.

[13] 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.

[14] Bie R F, Mehmood R, Ruan S S, et al. Adaptive fuzzy clustering by fast search and find of density peaks[J]. Personal and Ubiquitous Computing, 2016, 20(5): 785-793.

[15] Zhuo L L, Li K L, Liao B, et al. HCFS: A density peak based clustering algorithm employing a hierarchical strategy[J]. IEEE Access, 2019, 7: 74612-74624.

[16] 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, Lyu L, et al. Density peaks clustering based on mutual neighbor degree[J]. Control and Decision, 2021, 36(3): 543-552.)

[17] Yuan X N, Yu H, Liang J, et al. A novel density peaks clustering algorithm based on K-nearest neighbors with adaptive merging strategy[J]. International Journal of Machine Learning and Cybernetics, 2021, 12(10):

2825-2841.

[18] Guan J Y, Sheng L, He X X, et al. A novel clustering algorithm by adaptively merging sub-clusters based on the normal-neighbor and merging force[J]. Pattern Analysis and Applications, 2021, 24(3): 1231-1248.

[19] Shi A Y, Zhao J, Tang J J, et al. Improved density peaks clustering based on firefly algorithm[J]. International Journal of Bio-Inspired Computation, 2020, 15(1): 24.

[20] Yu D H, Liu G J, Guo M Z, et al. Density peaks clustering based on weighted local density sequence and nearest neighbor assignment[J]. IEEE Access, 2019, 7: 34301-34317.

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

[22] Vinh N X, Epps J, Bailey J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance[J]. Journal of Machine Learning Research, 2010, 11: 2837-2854.

[23] 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.

[24] Jain A K, Law M H C. Data clustering: A user's dilemma[C]. Proceedings of the First International Conference on Pattern Recognition and Machine Intelligence. Kolkata, 2005: 1-10.

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

[26] Ren C H, Sun L F, Yu Y, et al. Effective density peaks clustering algorithm based on the layered K-nearest neighbors and subcluster merging[J]. IEEE Access, 2020, 8: 123449-123468.

[27] Yu H, Chen L Y, Yao J T. A three-way density peak clustering method based on evidence theory[J]. Knowledge-Based Systems, 2021, 211: 106532.

[28] Xu X, Ding S F, Wang L J, et al. A robust density peaks clustering algorithm with density-sensitive similarity[J]. Knowledge-Based Systems, 2020, 200: 106028.

[29] Gionis A, Mannila H, Tsaparas P. Clustering aggregation[J]. ACM Transactions on Knowledge Discovery from Data, 2007, 1(1): 4.

[30] Fu L M, Medico E. FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data[J]. BMC Bioinformatics, 2007, 8(1): 3.

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

[32] 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.

[33] Blake C L, Merz C J. UCI repository of machine learning database[EB/OL]. [2016-12-28]. http://archive.ics.uci.edu/ml/index.html.

[34] Charytanowicz M, Niewczas J, Kulczycki P, et al. Complete gradient clustering algorithm for features analysis of X-ray images[C]. Information Technologies in Biomedicine, 2010, 69: 15-24.

[35] Sigillito V G, Wing S P, Hutton L V, et al. Classification of radar returns from the ionosphere using neural networks[J]. Johns Hopkins APL Technical Digest, 1989, 10(3): 262-266.

[36] Street W N, Wolberg W H, Mangasarian O L. Nuclear feature extraction for breast tumor diagnosis[C]. Proceedings of the IS & T/SPIE International Symposium on Electronic Imaging: Science and Technology. San Jose, 1993: 861-870.

作者简介

吴润秀 (1971–), 女, 教授, 硕士, 从事群智能算法及应用、数据挖掘、计算智能等研究, E-mail: wrx@nit.edu.cn;

尹士豪 (1997–), 男, 硕士生, 从事数据挖掘的研究, E-mail: yinshihao97@163.com;

赵嘉 (1981–), 男, 教授, 博士, 从事机器学习与数据挖掘、计算智能等研究, E-mail: zhaojia925@163.com;

李沛武 (1963–), 男, 教授, 博士, 从事计算机安全与大数据分析、信息安全、数据挖掘等研究, E-mail: lpw@nit.edu.cn;

刘宝宏 (1975–), 男, 副教授, 博士, 从事数据分析及处理的研究, E-mail: bhliu2006@gmail.com.

(责任编辑: 闫 妍)