基于子博弈完美均衡的启发式聚类算法

常璐瑶,牛新征*,罗 涛,钱早国
(电子科技大学计算机科学与工程学院,四川成都 611731)

摘 要: 聚类是一种典型且重要的数据挖掘方法,但现有聚类算法大多需要人为指定聚类的数量,并且聚类结果对参数敏感. 针对上述不足,本文提出一种基于子博弈完美均衡的启发式聚类算法(Heuristic Clustering algorithm based on Sub-game Perfect Equilibrium,HCSPE). 该算法充分挖掘数据点自身的分布特征信息,通过启发式方法得到自适应的参数值,从而使数据点局部密度属性值的得出具有客观性和普适性,降低了聚类结果对参数的敏感性. 基于博弈的思想,综合局部密度和相对距离两个属性形成数据点的竞争力,依靠竞争机制完成聚类数量的自动计算以及聚类中心的确定. 在多个规模和类型均不相同的数据集上的实验结果表明,本文所提出算法的性能指标整体优于其他算法,并且聚类结果更符合客观所需.

关键词: 博弈论;竞价机制;子博弈均衡;启发式算法;聚类
基金项目: 国家自然科学基金(No.62272087);四川省科技计划项目(No.2021YFS0391)
中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112(2024)03-0740-11
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20221206

Heuristic Clustering Algorithm Based on Sub-Game Perfect Equilibrium

CHANG Lu-yao, NIU Xin-zheng*, LUO Tao, QIAN Zao-guo
(*School of Computer Science and Engineering, University of Electronics Science and Technology of China, Chengdu, Sichuan* 611731, *China*)

Abstract: Clustering is a typical and important data mining method, but most of the existing clustering algorithms need to specify the number of clusters artificially, and the clustering results are sensitive to parameters. To address the above shortcomings, this paper proposes a heuristic clustering algorithm based on sub-game perfect equilibrium (HCSPE). The algorithm fully exploits the information of the distribution characteristics of data points themselves and obtains the adaptive parameter values by heuristic methods, so that the local density attribute values of data points are derived with objectivity and universality, and the sensitivity of clustering results to parameters is reduced. Based on the idea of game, the two attributes of local density and relative distance are integrated to form the competitiveness of data points, and the automatic calculation of the number of clusters and the determination of cluster centers are completed by relying on the competition mechanism. The experimental results on several data sets of different sizes and types show that the performance indexes of the proposed algorithm are better than other algorithms in general, and the clustering results are more in line with the objective requirements.

Key words: game theory; bidding mechanism; sub-game equilibrium; heuristic algorithm; clustering
Foundation Item(s): National Natural Science Foundation of China (No.62272087); Technology Planning Project of Sichuan Province (No.2021YFS0391)

1 引言

现如今从客观世界中获取到的信息数据量呈现指数增长趋势. 如何从这些数据中挖掘到有价值的结构和信息是一个极具研究和应用价值的问题. 当前数据挖掘方法主要分为监督学习和无监督学习两类[1],其中,监督学习算法需要大量带有标签的训练数据,具有较高的时间和人工成本. 因此,现有研究更倾向于无监督、弱监督学习[2]. 聚类作为无监督学习的一种重要方法,目的是按照某个标准把一个数据集分割成不同簇,使得同一个簇内数据对象的相似性尽可能大,同时不在同一个簇中的数据对象的差异性也尽可能地大[3].

收稿日期:2022-10-25;修回日期:2023-02-01;责任编辑:梅志强
*通讯作者:牛新征

聚类被广泛应用于医学[4~6]、道路交通[7~9]、商业[10~12]和经济学[13~15]等领域. 现阶段针对具有不同分布特征的数据,聚类大致分为基于密度的聚类算法、基于划分的聚类算法[16]、层次聚类算法[17]和模糊聚类算法[18]等. 上述算法大多需要预先给定聚类的数目,显然这与现实需求相违背. 此外,对参数的高敏感性使其在获得聚类结果的过程中具有较高的时间复杂度.

2014年发表在Science上的一种基于密度峰值的聚类算法(Clustering by Fast Search and Find of Density Peaks,DPC)重新定义了聚类中心[19],文章通过人工选择聚类中心的方式使聚类中心更具有倾向性和针对性,提高了聚类结果的目标满意率,对非聚类中心点的分配具有较低的时间复杂度. 但其关键参数截断距离d_c的得出不具有普适性,且通过手动框选来识别聚类中心的做法在处理密度较为接近的簇时,容易陷入局部最优,并最终影响聚类结果的准确性. 针对DPC算法存在的不足之处,文献[20]提出了一种基于热扩散的新方法,该方法基于无限域上的热扩散方程,通过核密度估计进行截断距离d_c的选择以及边界的修正,从而能够更加精准地创造聚类,但此算法复杂度较高,且没有很好的解决人工选择聚类中心的问题. 文献[21]提出了一种稳健的聚类算法,该算法根据K近邻思想进行数据点局部密度的计算,从而降低了对截断距离参数d_c的依赖. 并使用基于广度搜索与模糊加权K近邻两种技术来进行非聚类中心点的分配,但此算法仍没有很好的解决聚类中心的选定问题. 文献[22]创新性的将概率引入聚类算法中,所提出的算法使用从核函数和带宽计算的局部密度来初始化一个对象对其所选对象的吸引子概率,然后传播该概率直到该组吸引子变得稳定. 该方法开阔了聚类的思路,使得聚类算法在一定程度上可被视为一个优化问题,即通过获取目标函数的近似最优解来完成数据点的聚类[23]. 在DPC算法中,聚类中心的选择可视为一个在有数量限制的前提下,通过不断对比数据点决策函数值来尽可能获取最大和值,使得此使系统处于均衡状态的问题. 因此,DPC算法可转换成一个博弈论问题,即数据点以自身决策函数作为"竞争力",相互竞争成为数量受限的聚类中心,目标是获取效用收益的均衡最值,使得系统整体达到一个均衡稳定的状态.

本文充分考虑待聚类数据点自身所蕴含的信息,将数据点视为参与博弈的对象,提出了一种基于子博弈完美均衡的启发式聚类算法(Heuristic Clustering algorithm based on Sub-game Perfect Equilibrium,HCSPE). 采用启发式算法获取截断距离参数d_c,以此保证d_c的计算取值方式在不同的数据集中均具有普适性. 在聚类中心的选择阶段,定义数据点竞争力函数,通过分析数据集整体达到子博弈完美均衡时数据点的状态,最终确定聚类中心. 最后,使用最近优先分配原则完成非聚类中心数据点的分配工作. 通过与现有同类算法在不同数据集和多种指标上的对比,验证所提出算法聚的优越性.

2　相关工作

2.1　基于密度峰值的聚类算法

作为一种简明有效的聚类算法,DPC算法的核心思想在于对聚类中心的刻画. 作者认为要想成为聚类中心,数据点要满足以下两个条件:(1)自身密度足够大,即分布在它周围的数据点的密度均小于它;(2)与密度大于它的数据点的距离相对更大. 由此,对每个数据点定义了局部密度ρ和相对距离δ两个属性. 在数据集$X=\{x_i\}_{i=1}^n$中,对于每个数据点i,局部密度属性ρ_i定义如下:

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

其中,d_{ij}是数据点i和数据点j之间的欧几里得距离,d_c为截断距离,函数$\chi(x)$的定义见式(2).

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

在DPC算法中作者对d_c取值的要求为:需要满足每个数据点的平均密度的大小为数据点总数的1%~2%.

对于每个数据点i,相对距离参数δ_i定义如下:当i不具有最大局部密度时,

$$\delta_i=\min(d_{ij}),\ j:\rho_j\geqslant\rho_i \tag{3}$$

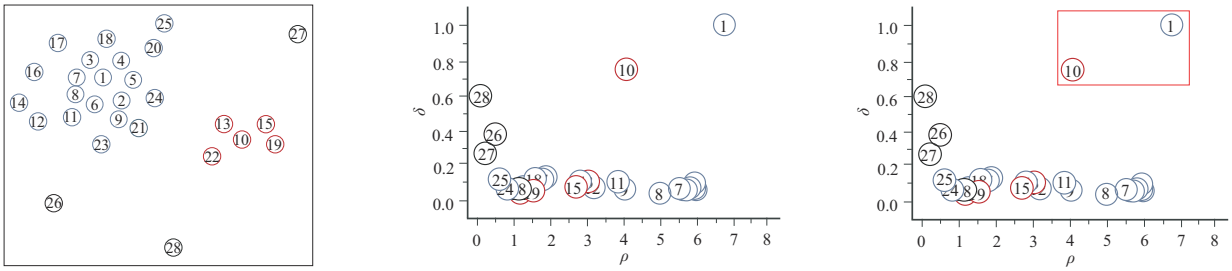
当i具有最大局部密度时,

$$\delta_i=\max(\delta_j),\ j:\rho_j<\rho_i \tag{4}$$

经计算得出数据集中每个数据点的局部密度ρ以及相对距离δ两个属性值. 由于DPC算法认为ρ值和δ值对聚类中心的选取起到决定性作用,所以作者创建了如图1(*b*)所示的由(ρ_i,δ_i)对应组成的决策图. 通过人工定性的选择出主观上符合聚类中心特征的数据点作为聚类中心.

2.2　子博弈完美均衡

博弈论将激励结构间的相互作用进行了公式化,主要研究在某种竞争背景下对策选择及均衡问题[24]. 形成一个博弈应当至少包含三个方面的内容:一是博弈参加者;二是参与博弈方可选择的全部策略的集合;三是博弈方的收益. 在这里,$p=\{p_i\}_{i=1}^n$用于表示处于同一个博弈中的所有参与者集合;$S=\{S_i\}_{i=1}^n$用于表示所有参与者的策略集合;$s_i=\{s_{ij}\}_{j=1}^k$用于表示在i时刻,对应某一参与者能够提供的所有策略集合;$U=\{U_i\}_{i=1}^n$用于表示所有参与者的收益效用集合. 定义贴现因子



(a) 原始数据点分布及其对应密度　(b) 决策图　(c) 人工选择聚类中心

图1　DPC算法聚类中心选择过程

$\alpha(t)$是随时间t不断变化变量,在数值上等于贴现率,体现的是参与者的耐心程度. 因此,一个非典型的博弈场景G的定义如下:

$$G=\{p;\ S;\ U;\ \alpha\} \tag{5}$$

当参与者n=2时,博弈G所对应的博弈树如图2所示,其中,节点代表参与者,枝分别代表分配策略$\{(v-k,k)\}_{k=0}^{v}$和当前节点的参与者是否同意相应分配策略(y:同意,n:不同意). 从图2可直观感受到,蓝色节点与根节点存在着对称关系,仅存在参与者决策次序的不同. 从绿色节点出发,可以看出参与者1所处于的状态与根节点完全相同. 在博弈树中,从单个决策节点出发,考虑余下的所有节点和枝构成的部分,若这些部分没有对初始博弈树造成割裂,那么这些节点和枝构成所的部分叫做当前博弈的子博弈. 对于有限博弈$G=\{p;\ S;\ U\}$,$S^*=\{S_i^*\}_{i=1}^n$是子博弈完美均衡,当且仅当:

$$U_i\big|_t(S_i^*\big|_t,S_{v-i}^*\big|_t)\geqslant U_i(S_i\big|_t,S_{v-i}^*\big|_t),\forall i\in G(v) \tag{6}$$

其中,t代表时刻,v是博弈争取的总资源. 当参与者的行动满足子博弈完美均衡时,即代表参与者在每一个子博弈上都选择了最优策略,不存在另一种策略的结果优于现有选择策略.

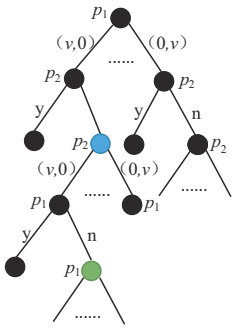


图2　两个参与者对资源v的博弈流程

3　算法描述

首先,DPC算法中数据点局部密度属性值的获取依赖于截断距离参数d_c,因此,d_c的选择在一定程度上决定着DPC算法的准确性. 但在DPC算法中,关于d_c值的获取方式并没有一个确定性描述,作者只是建议其取值应满足每个数据点平均密度的大小为数据点总数的1%~2%. 其次,DPC算法中聚类中心的选择是基于决策图人工手动框选出具有较大ρ值和δ值的点作为聚类中心. 这一选择方式受主观影响较大,并且在处理决策值较为接近的数据点时容易陷入局部最优,产生较大误差. 针对上述问题,本文提出一种基于子博弈完美均衡的启发式聚类算法:首先根据数据点自身分布特征,采用启发式方法得到自适应的参数d_c,基于d_c计算得到适应数据集自身的局部密度属性值和相对距离属性值. 然后以两个属性为核心形成竞争函数,基于博弈的思想完成聚类中心的确定,最终根据最近优先原则分配非聚类中心点.

3.1　基于启发式的自适应参数确定算法

如式(1)所示,DPC算法在计算数据点局部密度时,所使用的Cut-off核函数是离散的. 这种截断式的计算方法会造成部分数据点虽密度不同,但经过函数处理后会产生相同的密度值. 为避免上述问题,本文采用连续的高斯核函数进行数据点局部密度的计算[25]. 在含有n个数据点的数据集$X=\{x_i\}_{i=1}^n$中,使用高斯核定义数据点的局部密度如下:

$$\rho_i=\sum_{j\in X,j\neq i}\mathrm{e}^{-\left(\frac{d_{ij}}{d_c}\right)^2} \tag{7}$$

其中,d_{ij}表示数据点x_i与x_j之间的欧几里得距离.

基于信息熵理论基础,通常用熵来描述可能事件发生的不确定性. 当某一事件出现的概率越大,即不确定性越小,熵也就对应越小;反之,某一事件出现的概率越小,即不确定性越大,熵也就对应越大. 以二元信源为例做简要分析,设其中一个事件出现的概率分别为P,那么该信源熵H的计算公式如下:

$$H=-P\log_2P-(1-P)\log_2(1-P) \tag{8}$$

在二元信源中,熵H随概率P的变化如图3所示. 经分析可得,当二元信源中两个事件的出现概率相同时,信息熵达到最大.

推广到多元信源的情况,在n元信源中信息熵的计

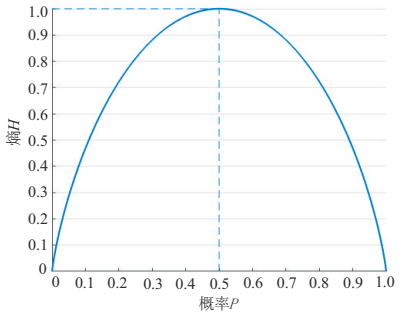


图3　二元信源中熵随概率变化图

算公式如下:

$$H=-\sum_{i\in X}P(i)\log_2 P(i) \tag{9}$$

$$P(i)=\frac{\left|N(i)\right|}{\sum_{i=1}^{n}\left|N(i)\right|} \tag{10}$$

其中,$N(i)$表示事件i发生的次数,n为事件总数.

同样的,在n元信源中,当n个事件出现的概率均相同时,此信源的熵达到最大.

在DPC算法中,截断距离参数d_c的值直接影响数据点局部密度的大小. 现在数据集X中,针对d_c的取值考虑两种极端情况:(1)当d_c值趋近于0时,由式(7)可得,属性ρ的值也趋近于0. 即若选取过小的d_c值,那么所有数据点的密度属性值都近似相等,均趋近于0;(2)当d_c值趋近于所有距离对中的最大值时,ρ趋近于$n-1$. 即若选取过大的d_c值,那么所有数据点的密度属性值都近似相等,均趋近于$n-1$. 在这两种极端情况中,各数据点局部密度值的出现概率都近似相等. 与上述n元信源的熵随事件概率而变化的分析相结合,此时系统整体熵值达到最大. 而对具有较好质量的聚类算法而言,不同的数据点应具有不同的密度,即各数据点ρ的取值都不尽相同,因此系统整体熵值达到最小. 如图4所示,启发式参数d_c值所对应系统熵最小.

在含有n个待聚类数据点的集合X中,系统密度熵$H(\rho)$的定义如下:

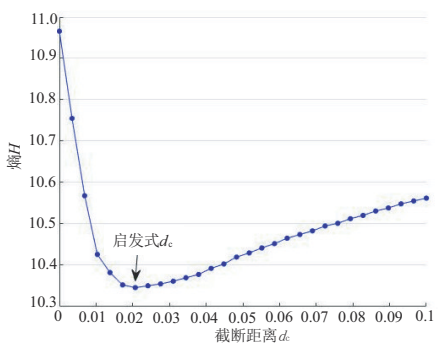


图4　不同截断距离值对应的系统熵值

$$H(\rho)=-\sum_{i\in X}P(\rho)\log_2 P(\rho) \tag{11}$$

$$P(\rho)=\frac{\left|\rho_{x_i}(d_c)\right|}{\sum_{i=1}^{n}\left|\rho_{x_i}(d_c)\right|} \tag{12}$$

对系统密度熵使用模拟退火算法,得到最小化目标函数$H(\rho)$的目标值,将此目标值作为计算数据点局部密度时所用到的参数d_c的值. 对于不同的数据集而言,截断距离参数d_c完全由数据集自适应其自身数据特征,通过启发式的方式的得出,所以本文中提出的参数确定方法具有普适性. 算法1为基于启发式的自适应参数确定算法的总体描述.

算法1　基于启发式的自适应参数确定算法

输入:数据集X,迭代参数T_{min}

输出:截断距离参数值d_c

```
1.计算数据点间的欧几里得距离,得到距离矩阵 dist[][]
2.FUNCTION getSA(dist):
3.  dc[]=random(min(dist[][]),max(dist[][])) //根据距离矩阵初始化 dc[]
4.  WHILE t > T_min
5.    FOR i = 0 : k-1
6.     target = get_target(dc[i],dist);
7.     dc_new = dc[i] + random(); //在邻域内产生新的解
8.      IF (dc_new in range(下界,上界))
9.       target_new = get_target(dc_new,dist);
10.      dc[i] = (target_new-target < 0) ? dc_new : dc[i];
11.     ELSE
12.      dc[i] = 1/(1+e^-(target_new-target)/T_min); //不在既定范围内进行概率替换
13.   END FOR
14.  END WHILE
15.  FOR i = 0 : k
16.     result = min(result,get_target(dc[i],dist));
17.  END FOR
18.END FUNCTION
19.FUNCTION get_target(dc,dist):
20.  FOR i = 0 : n-1
21.    由式(11)计算系统熵值 target;
22.  END
23.  RETURN target;
24.END FUNCTION
```

3.2　基于子博弈完美均衡的聚类中心选取算法

假定在数据集X中,共有$n+2$个数据点. 不失一般性,在这里对两个参与者p_1、p_2参与博弈的情况进行分析讨论. 在数据集X中,将剩余没有参与博弈的n个数据点博弈的目标值n,整个博弈过程如图5所示. 此博弈模型是一个无穷期的扩展型博弈,由若干子博弈组成,具有对称性和稳定性,每一层分别代表一个博弈决定时刻.

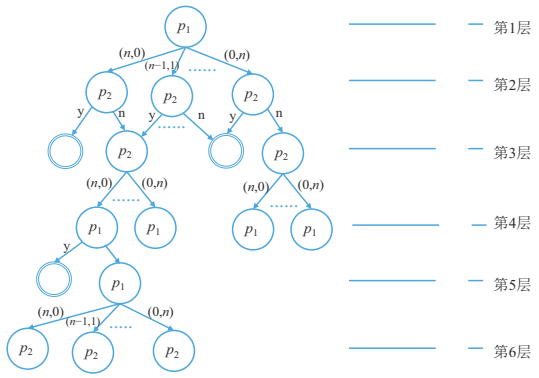


图5 两个参与者对n个数据点的博弈过程

在不考虑参与者自身的前提下,假定t时刻两个参与者对剩余数据点的分配策略达成共识,参与者1分得x个点,则参与者2分得$n-x$个点,此时二者的效用值分别为$\alpha(t)\times U(x)$和$\alpha(t-1)\times U(n-x)$. 设$\bar{n}_1$是参与者1在所有子博弈中能够达到的最大效用值,由于此博弈模型具有稳定性,所以参与者2在所有子博弈中能够达到的最大效用值为$\alpha(t)\bar{n}_1$,即参与者1提供给参与者2的不多于$\alpha(t)\bar{n}_1$. 由于待分配数据点个数n是恒定的,所以参与者1得到的不少于$n-\alpha(t)\bar{n}_1$,即参与者1在所有子博弈中能够达到的最小效用值为:

$$\underline{n_1}=n-\alpha(t)\bar{n}_1 \tag{13}$$

由稳定性可得,参与者2在所有子博弈中能够达到的最小效用值为:

$$\underline{n_2}=\alpha(t)\,\underline{n_1} \tag{14}$$

即参与者1提供给参与者2的不少于$\underline{n_2}$,在待分配数据点个数n保持不变的前提下,也就意味着参与者1得到的不多于$n-\underline{n_2}$,即参与者1能够达到的效用最大值为:

$$\bar{n}_1=n-\underline{n_2} \tag{15}$$

结合式(13)~(15)可得,$\bar{n}_1=\underline{n_1}$. 即在子博弈完美均衡中,参与者所能得到效用的最大值和最小值是一样的,此时整个博弈系统处于稳定状态.

基于上述对两个参与者博弈模型中子博弈完美均衡的讨论,将子博弈完美均衡的思想融入DPC算法对其进行改进. 首先明确在此应用场景中贴现因子α的定义. 基于信息论的基本原理,互信息可以看成是一个随机变量中包含的关于另一个随机变量的信息量,或者说是一个随机变量由于已知另一个随机变量而减少的不肯定性. 密度ρ和距离δ两个属性对聚类中心的选择起决定性作用,二者共同决定了某数据点成为聚类中心的不确定性. 数据点的ρ值和δ值越大,它就越有可能成为聚类中心,即成为聚类中心的不确定性越小. ρ和δ两个属性变量之间存在非独立关系,根据信息熵定义得到二者的联合分布熵如式(16)所示.

$$I(X;\,Y)=\sum_{x\in\chi}\sum_{y\in v}P(x,y)\log_2\frac{P(x,y)}{P(x)P(y)} \tag{16}$$

$$P(x_i)=\frac{\left|\rho_{x_i}(d_c)\right|}{\sum\limits_{i\in X/x}\left|\rho_{x_i}(d_c)\right|},P(y_i)=\frac{\left|\delta_{y_i}\right|}{\sum\limits_{i\in X/y}\left|\delta_{y_i}\right|},$$

$$P(x_i,y_i)=\frac{\left|(\rho_{x_i}(d_c),\delta_{y_i})\right|}{\sum\limits_{i\in X/\{x,y\}}\left|(\rho_{x_i}(d_c),\delta_{y_i})\right|} \tag{17}$$

数据点密度属性和距离属性综合代表了此数据点成为聚类中心的“竞争力”. 因此,在上述模型的基础上,将数据集中n+2个数据点两两组合,依次视为博弈的参与者,将非独立属性变量ρ、δ的互信息熵作为数据点间博弈的贴现系数α,效用函数为对剩余n个数据点进行博弈最终分得数据点的数量. 记录两个参与者达到子博弈完美均衡时的结果值,在每次博弈结束后,采用正反馈调节机制利用本次博弈结果对整个博弈系统进行调节,以此降低异常数据值对结果的影响. 算法2为基于子博弈完美均衡的聚类中心选取算法的总体描述.

算法2 基于动态子博弈完美均衡的聚类中心选取算法

输入:数据集X,参数d_c,调整参数δ

输出:聚类中心数组Cluster_center[]

1.FOR i = 1 : n

2. 使用式(7)计算出数据点i的密度rho[i];

3. 使用式(3)、式(4)计算数据点i的相对距离derta[i];

4.END FOR

5.FOR i = 1: n

6. FOR j = i : n

7. 将(i, j)视为博弈者,利用式(15)得到子博弈均衡效用值utility[k];

8. IF((utility[k]−mean(utility[1:k−1]))<δ×mean (utility[1:k−1])) //当本组均衡收益处于非异常状态

9. result.put(<utility[k],[i, j]>); //添加将本组博弈对象及均衡收益

10. ELSE

11. utility.delete(k);

12. END FOR

13.δ = sigmoid(δ×mean(utility[i : k−1])); //通过每个数据点所参与的所有博弈结果动态更新调节参数值

14.END FOR

15.number= mean(utility[]);

16.SORT(result.key); //通过效用函数值将博弈对象进行排序

17.FOR i =1: number

18. Cluster_center.add(result[i].getvalue());

19.END FOR

4 实验验证

本文选取了采样于真实世界且具有标签的UCI数

据集[26]和分布特殊且不具有标签的人工合成数据集作为实验对象,分别从内部、外部两个方面个多项指标对数据集展开实验评价[27]. 本节的实验环境设置为:Windows 10操作系统,Intel(R) Core(TM) i5-10400 CPU @ 2.90 GHz,16 GB内存. 实验基于Matlab和Java14语言编写实现.

4.1 数据集

本次实验共选用如表1所示的13个数据集,其中,7个是规模大小不同且具有标签的真实数据集,6个是规模及分布特征均不同但不具有标签的合成数据集. 7个真实有标签的数据集分别是玻璃品种glass、乳腺癌手术的患者存活情况haberman、鸢尾植物类Iris、小麦品种seeds、肿瘤的类别wdbc、葡萄酒品种Wine、葡萄酒品质Wine_Quality. 由于上述数据集具有多个属性,因此在实验部分,本文将其所有的属性均进行两两组合并分别进行实验,最终选择最优结果作为本数据集对应的实验结果.

表1 数据集

序号	数据集	实例数量	是否带标签	类别数量
data1	glass	214	是	6
data2	haberman	306	是	2
data3	Iris	150	是	3
data4	seeds	210	是	3
data5	wdbc	569	是	2
data6	Wine	178	是	3
data7	Wine_Quality	4 898	是	8
data8	dbmoon	2 000	否	—
data9	fish	1 200	否	—
data10	FuzzyX	1 000	否	—
data11	Parabolic	1 000	否	—
data12	Ring	1 000	否	—
data13	Zigzag	1 002	否	—

4.2 对比算法

本文选择具有相同机制的算法进行比较来保证HCSPE算法的有效性,并选择了具有不同机制的算法来保证实验的完整性. 5种对比算法分别为:K-Means(基于划分的K均值聚类算法)[28]、DBscan(基于密度的聚类算法)[29]、DPC(基于密度峰值的聚类算法)[19]、DPC-DCFN(基于密度峰值和模糊邻域的聚类算法)[30]、EC(基于密度极值点的聚类算法)[31].

4.3 评价指标

为了评估不同聚类算法的优劣,需要一些定量的指标对聚类结果进行评估. 根据数据集中是否提供真实标签信息,相关的指标可以分为内部评估指标和外部评估指标.

4.3.1 内部评价指标

内部评价指标是不借助于外部信息,仅根据聚类结果来进行评估,利用数据集的属性特征来评价聚类算法. 内部评价指标常用的有DBI(Davies-Bouldin)指数、SC(Silhouette Coefficient)和CHI(Calinski-Harabaz Index)等.

DBI:核心思想是计算每个簇与之最相似簇之间相似度,然后再通过求出所有相似度的平均值来衡量整个聚类结果的优劣. 对于聚类结果的DBI,簇与簇之间的相似度被定义为两个簇的簇内直径和与簇间距离的比值. DBI的取值范围是$[0,+\infty)$,值越小表示聚类结果越好.

SC:本质上衡量的是每个样本点到其簇内样本的距离与其最近簇结构之间距离的比值. 对于聚类结果的轮廓系数,它实际上是每个样本点SC的平均值. SC的取值范围是$[-1,1]$,取值越接近1,说明聚类性能越好.

CHI:本质是簇间距离与簇内距离的比值. 首先通过计算类内各点与类中心的距离平方和来度量类内的紧密度,然后再通过计算类间中心点与数据集中心点距离平方和来度量数据集的分离度. 最终CHI指标取值由分离度与紧密度的比值得到,取值范围是$(0,+\infty)$,值越大即簇与簇之间相距较远,表明聚类效果越好.

4.3.2 外部评价指标

外部评估方法是指在知道真实标签的情况下来评估聚类结果的优劣. 一般来说在只有少量的标注数据时,可以用外部评估法选择一个相对最优的聚类模型,然后再应用到其它未被标记的数据中.

纯度指标Purity:

$$\mathrm{Purity}(\Omega,C)=\frac{1}{N}\sum_{k}\max_{j}\left|w_k\cap c_j\right| \tag{18}$$

其中,N代表总样本个数,$\Omega=\{w_i\}_{i=1}^{k}$代表经聚类算法处理后簇的划分,$C=\{c_i\}_{i=1}^{k}$代表真实簇的划分. Purity取值范围是$[0,1]$,值越接近1表示聚类结果越好.

Normalized Mutual Information(NMI):

$$\mathrm{NMI}(\Omega,C)=\frac{I(\Omega;C)}{\left(H(\Omega)+H(C)\right)/2} \tag{19}$$

$$I(\Omega,C)=\sum_{k}\sum_{j}P\left(w_k\cap c_j\right)\log\frac{P\left(w_k\cap c_j\right)}{P\left(w_k\right)P\left(c_j\right)},$$

$$H(\Omega)=-\sum_{k}P\left(w_k\right)\log P\left(w_k\right) \tag{20}$$

其中,$P(w_k)$是样本属于计算得到簇w_k的概率,$P(c_j)$是样本属于真实簇c_j的概率. NMI的取值范围是$[0,1]$,反映出聚类算法处理后得到簇和真实簇之间的关联,二者关系越密切,NMI的值也就越大.

定义TP(True Positive)为真正例,表示两个同类样本在同一簇的情况;TN(True Negative)为真负例,表示两个非同类样本在不同簇的情况;FP(False Positive)为伪正例,表示两个非同类样本在同一簇的情况;FN(False Negative)为伪负例,表示两个同类样本在不同簇的情况. 基于以上,产生三个外部指标的定义.

精确率指标Precision:

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{21}$$

召回率Recall:

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{22}$$

F_β值:

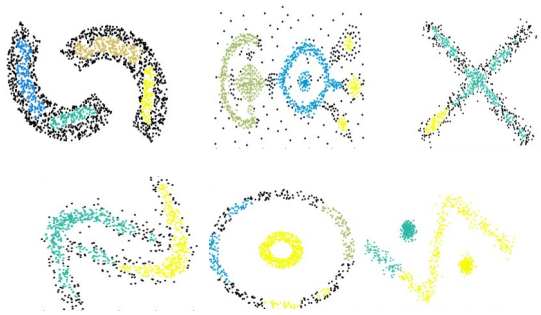
$$F_\beta=(1+\beta^2)\frac{\text{Precision}\cdot\text{Re call}}{\beta^2\cdot\text{Pr ecision}+\text{Re call}} \tag{23}$$

上述Purity、NMI、Precision、Recall、F_beta值五个指标广泛应用于聚类质量的评价上,它们可以很好的从簇的内部纯度、划分簇与真实簇之间的关联程度以及全局数据点划分的情况等方面对聚类结果进行全面综合的评价.

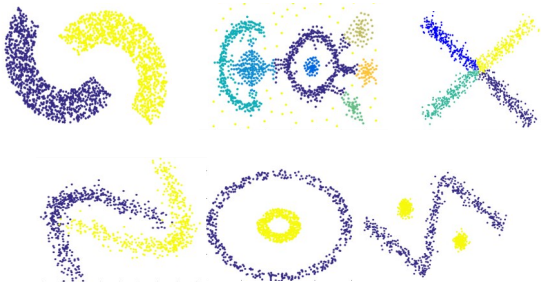
4.4 实验结果

4.4.1 主观评价

图6展示了在data8~data13共6个不带有标签的人工合成数据集上,经HCSPE算法和DPC算法分别处理后的聚类结果. 其中,dbmoon数据集的特征是类内较为密集但部分数据点的隶属类模糊;fish数据集的特征是类别较多,且部分类之间界限模糊;FuzzyX数据集的特征是类内相对紧凑但部分数据的距离跨度较大;Parabolic数据集的形状呈现半环形且类间界限模糊;Ring数据集是两个完整的嵌套圆;Zigzag数据集中,一部分类的内部紧凑,一部分类的内部呈现发散形态. 图6(a)、图6(b)分别为DPC算法和HCSPE算法对6个数据集的聚类结果,从图6可以直观感受出HCSPE算法的处理结果明显优于DPC算法,且更符合现实世界主观需要. 上述实验结果体现出HCSPE算法对不同分布特征的数据集具有有效性和鲁棒性.



(a) DPC算法对6个数据集的处理结果



(b) HCSPE算法对6个数据集的处理结果

图6 两种算法在不同数据集上的聚类表现

4.4.2 内部指标评价结果

将5种对比算法与HCSPE在13个数据集上进行对比实验,得到3个内部评价指标的结果值分别如表2、表3和表4所示.

表2 6种算法在13个数据集上获得DBI指标的实验结果

数据集	K-Means	DBscan	DPC	EC	DPC-DBFN	HCSPE
dbmoon	0.793 6	0.833 4	0.834 7	0.794 3	2.457 8	**0.701 2**
fish	0.471 9	0.250 0	3.552 6	**0**	0.250 0	0.015 6
FuzzyX	1.115 2	3.788 2	0.697 1	1.045 6	7.122 6	**0.548 2**
glass	0.898 3	2.003 4	0.363 5	0.736 6	4.288 0	**0.298 4**
haberman	0.965 5	2.375 2	3.187 3	1.231 3	0.420 8	**0.103 5**
Iris	0.407 7	0.652 5	0.391 7	0.391 7	0.391 7	**0.267 4**
Parabolic	0.770 3	0.968 4	0.809 0	0.808 2	3.257 5	**0.364 9**
Ring	1.348 7	1.148 7	1.839 4	0.743 5	3.132 5	**0.564 7**
seeds	0.717 5	0.776 4	1.919 5	0.724 2	0.687 1	**0.659 7**
wdbc	0.504 4	0.562 1	1.040 1	0.272 4	0.226 5	**0.158 7**
Wine	0.481 7	0.986 5	3.320 9	0.433 1	2.920 8	**0.401 1**
Wine_Quality	0.617 7	3.176 6	1.145 6	0.549 9	2.632 1	**0.512 6**
Zigzag	0.908 4	0.976 0	0.791 2	0.697 0	0.781 4	**0.609 8**

表3　6种算法在13个数据集上获得SC指标的实验结果

数据集	K-Means	DBscan	DPC	EC	DPC-DBFN	HCSPE
dbmoon	0.668 5	0.629 9	0.628 7	0.6481	0.113 4	**0.785 4**
fish	0.731 2	**0.948 9**	−0.022 1	−0.693 4	**0.948 9**	0.868 7
FuzzyX	0.539 5	0.106 6	0.453 6	0.384 6	0.300 7	**0.608 7**
glass	0.700 1	−0.082 8	0.720 6	0.662 7	−0.246 7	**0.765 4**
haberman	0.527 4	−0.517 9	0.012 9	0.317 1	0.829 4	**0.910 2**
Iris	0.844 2	0.747 8	0.843 1	0.843 1	0.843 1	**0.925 5**
Parabolic	0.701 2	0.550 9	0.643 6	0.643 9	0.004 3	**0.820 1**
Ring	0.528 0	0.523 3	−0.159 3	0.694 8	0.210 7	**0.598 0**
seeds	0.685 5	0.503 8	0.108 8	0.652 6	0.677 3	**0.895 2**
wdbc	0.834 3	0.632 6	0.741 3	0.690 6	0.456 2	**0.912 5**
Wine	0.819 3	0.138 6	−0.363 6	0.605 0	−0.245 8	**0.897 7**
Wine_Quality	0.772 2	−0.655 2	0.518 2	0.752 6	−0.752 3	**0.871 1**
Zigzag	0.642 2	0.347 9	0.465 8	0.608 3	−0.008 0	**0.658 9**

由表2可知,对于DBI指标,HCSPE在大部分数据集上取得了较好的结果,说明数据经此算法处理的结果中,任一簇与之最相似簇之间具有低相似度. 在fish数据集中,HCSPE的表现结果不如EC算法. EC算法主要针对具有较多类别数量的数据集开展研究,fish数据集经此算法得到的结果中,类内数据点十分紧凑,且类间距离远大于类内距离. DBI指标值为0,说明获得了高质量的聚类结果.

由表3可知,对于SC指标,HCSPE在大部分数据集上取得了较好的结果,即此算法能够使得在聚类结果中,任一样本点所在的簇结构与其最近簇结构之间具有较远距离. 但在fish数据集中,HCSPE的表现结果不如DPC-DBFN;在Ring数据集中,HCSPE的表现结果不如EC.

由表4可知,对于CHI指标,HCSPE在大部分数据集上取得了较好的结果,即此算法能够使得在聚类结果中,任一数据点和其所在类的类中心紧密连接的同时,类间中心点也与数据集中心点高度分离. 但在dbmoon数据集中,HCSPE的表现结果不如K-Means;在fish数据集中,HCSPE的表现结果不如DPC-DBFN.

表4　6种算法在13个数据集上获得CHI指标的实验结果

数据集	K-Means	DBscan	DPC	EC	DPC-DBFN	HCSPE
dbmoon	**2 808.15**	2 459.94	2 451.93	2 628.33	1.19	2 562.85
fish	1 747.71	25 468.35	315.78	2540.65	26 468.35	**25 641.21**
FuzzyX	717.44	52.86	378.16	407.23	0.64	**729.36**
glass	173.71	33.01	12.21	23.54	42.27	**256.98**
haberman	238.99	0.75	16.67	21.95	24.68	**541.33**
Iris	496.65	568.54	493.88	493.88	493.88	**658.12**
Parabolic	1 493.68	914.26	1 235.26	1 236.40	0.62	**1 659.97**
Ring	429.79	549.62	64.58	898.86	4.51	**1 320.01**
seeds	310.50	185.27	23.62	337.04	289.31	**541.60**
wdbc	1 300.21	965.36	292.58	4 309.15	659.23	**4 658.11**
Wine	505.42	4.65	5.21	2 167.10	5.21	**3 654.74**
Wine_Quality	2 816.87	0.32	19.13	1 816.50	35.55	**2 965.13**
Zigzag	1 117.94	391.29	736.71	951.15	1.48	**1 206.33**

将5种对比算法在每个数据集上的最好表现视为基准值,数值为1. HCSPE相较于基准值的优化提升情况如图7所示. 从中可以看出,HCSPE算法除了在data2(haberman)数据集以及data8(dbmoon)数据集的SC指标上所获得的结果低于基准值外,在其他数据集的三项内部指标上均取得了明显提升.

4.4.3　外部指标评价结果

将5种对比算法与HCSPE在13个数据集上进行对比实验,得到5个外部评价指标的结果值如表5所示. 其中,计算F_β值时参数β设定为1.

通过表5可以看出,HCSPE算法在Purity指标上相较于其他算法获得了绝对的优势,说明HCSPE算法使

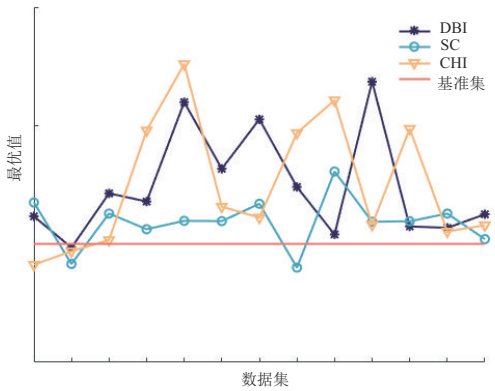


图7　HCSPE算法在内部指标上的对比实验结果

得每个簇内被正确分配的数据点占有绝对的比重. 除在haberman数据集外,HCSPE算法NMI指标上的结果均由于其他算法,说明HCSPE算法与真实簇具有较高的贴合度. 除在seeds数据集外,HCSPE算法在F_β值指标上取得了较为明显的优势. F_β值通过结合精确率和召回率表现出一个算法对正确样本的预测能力. 在医疗领域,把阳性实体错分的代价远高于把阴性实体错分的代价,因此需保证F_β值一定要取得最优. HCSPE算法在haberman和wdbc两个数据集的F_β值均高于对比算法,验证了此算法的可靠性.

将5种对比算法在每个数据集上的最好表现视为基准值,数值为1, HCSPE相较于基准值的优化提升情况如图8所示. 可以看出,HCSPE算法除了在haberman数据集的NMI指标上所获得的结果低于基准值外,在其他数据集的三项外部指标上均取得了提升,其中Purity指标的提升优化幅度最小,F_β指标的提升优化幅度最大.

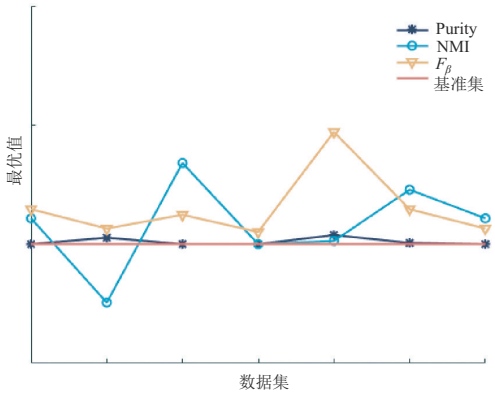


图8 HCSPE算法在外部指标上的对比实验结果

表5 6种算法在7个数据集上获得5种外部指标的实验结果

数据集	指标	glass	haberman	Iris	seeds	wdbc	Wine	Wine_Quality
K-Means	Purity	0.878 5	0.702 6	1.000 0	0.804 7	0.855 8	0.887 6	0.994 9
	NMI	0.034 4	0.015 2	0.733 6	0.115 2	0.405 4	0.574 4	0.001 9
	Precision	0.996 2	0.567 8	1.000 0	0.732 0	0.746 5	0.879 9	0.989 8
	Recall	0.263 7	0.595 9	0.595 1	0.353 2	0.779 3	0.522 7	0.356 8
	F_beta	0.417 0	0.581 5	0.746 1	0.476 5	0.762 5	0.655 8	0.524 6
DBscan	Purity	0.869 1	0.967 3	1.000 0	0.871 4	1.000 0	0.887 6	0.993 7
	NMI	0.704 2	0.027 5	0.587 6	0.791 4	0.000 0	0.574 4	0.009 0
	Precision	0.795 1	0.923 5	0.700 6	0.777 2	1.000 0	0.879 9	0.985 6
	Recall	0.832 9	0.601 3	0.658 2	0.975 9	0.337 9	0.522 7	0.356 2
	F_beta	0.813 6	0.728 3	0.603 1	0.865 3	0.505 1	0.655 8	0.523 3
DPC	Purity	0.990 6	0.702 6	1.000 0	0.804 7	0.855 8	0.887 6	0.994 9
	NMI	0.034 4	0.015 2	0.733 6	0.115 2	0.405 4	0.574 4	0.001 9
	Precision	0.996 2	0.567 8	1.000 0	0.732 0	0.746 5	0.879 9	0.989 8
	Recall	0.263 7	0.595 9	0.595 1	0.353 2	0.779 3	0.522 7	0.356 8
	F_beta	0.417 0	0.581 5	0.746 1	0.476 5	0.762 5	0.655 8	0.524 6
EC	Purity	0.878 5	0.918 3	1.000 0	0.952 9	0.042 1	0.151 6	0.839 9
	NMI	0.428 1	0.038 2	0.733 6	0.702 4	0.204 4	0.312 3	0.010 8
	Precision	0.842 7	0.880 1	1.000 0	0.990 4	0.007 6	0.049 2	0.723 0
	Recall	0.371 8	0.635 1	0.595 1	0.587 4	0.986 5	0.600 9	0.354 6
	F_beta	0.516 0	0.737 8	0.746 1	0.737 4	0.015 2	0.090 9	0.475 8
DPC-DBFN	Purity	0.6635	0.912 5	1.000 0	0.966 6	1.000 0	0.842 5	0.619 1
	NMI	0.297 3	0.002 6	0.733 6	0.576 4	0.000 0	0.412 9	0.012 7
	Precision	0.520 1	0.989 3	1.000 0	0.936 3	1.000 0	0.910 5	0.512 2
	Recall	0.370 5	0.610 8	0.5951	0.548 6	0.531 6	0.446 5	0.362 4
	F_beta	0.432 8	0.755 3	0.746 1	0.691 9	0.694 2	0.599 2	0.424 4
HCSPE	Purity	0.990 6	0.993 4	1.000 0	0.995 2	1.000 0	0.921 3	1.000 0
	NMI	0.780 4	0.023 1	0.983 4	0.809 1	0.405 4	0.582 1	0.015 6
	Precision	0.875 5	0.956 4	1.000 0	0.901 1	0.796 5	0.812 4	0.875 4
	Recall	0.896 5	0.791 2	0.659 8	0.896 5	0.803 3	0.598 6	0.689 9
	F_beta	0.885 9	0.865 9	0.795 0	0.798 7	0.799 8	0.689 3	0.771 6

5 结论

本文在DPC算法的基础上,提出了一种基于子博弈完美均衡的启发式聚类算法. 该算法首先根据数据点自身分布特征,采用启发式方法得到自适应的截断距离参数d_c,很好地解决了DPC算法中关键参数d_c值的获取方式没有定量描述的问题,使d_c值的获得具有普适性. 然后基于博弈的思想,根据数据点局部密度和相对距离两个属性定义了数据的竞争函数,通过数据点间的博弈自动完成聚类中心数量的计算以及数据点的确定,摒弃了DPC算法根据决策图手动框选聚类中心的做法,使得聚类中心的选取具有准确性以及客观性. 将所提出的算法与其他算法在多个数据集上进行了对比实验,通过内部、外部两个方面的多个指标对实验结果进行评价. 实验结果表明,HCSPE算法在真实数据集和人工合成数据集中均具有更优的聚类效果. 同时,HCSPE算法的提出也为聚类算法提供了一种新的思路.

参考文献

[1] WANG G T, SONG Q B. Automatic clustering via outward statistical testing on density metrics[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(8): 1971-1985.

[2] CHAO G Q, SUN S L, BI J B. A survey on multi-view clustering[J]. IEEE Transactions on Artificial Intelligence, 2021, 2(2): 146-168.

[3] FLORES-VIDAL P A, OLASO P, GÓMEZ D, et al. A new edge detection method based on global evaluation using fuzzy clustering[J]. Soft Computing, 2019, 23(6): 1809-1821.

[4] LIEW A W C, YAN H, YANG M S. Pattern recognition techniques for the emerging field of bioinformatics: A review[J]. Pattern Recognition, 2005, 38(11): 2055-2073.

[5] BRUSE J L, ZULUAGA M A, KHUSHNOOD A, et al. Detecting clinically meaningful shape clusters in medical image data: Metrics analysis for hierarchical clustering applied to healthy and pathological aortic Arches[J]. IEEE Transactions on Bio-Medical Engineering, 2017, 64(10): 2373-2383.

[6] GAO J, CHANG M T, JOHNSEN H C, et al. 3D clusters of somatic mutations in cancer reveal numerous rare mutations as functional targets[J]. Genome Med, 2017, 9(1): 4.

[7] KUMAR D, WU H Y, RAJASEGARAR S, et al. Fast and scalable big data trajectory clustering for understanding urban mobility[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(11): 3709-3722.

[8] COOPER C, FRANKLIN D, ROS M, et al. A comparative survey of VANET clustering techniques[J]. IEEE Communications Surveys & Tutorials, 2017, 19(1): 657-681.

[9] DUAN X Y, LIU Y N, WANG X B. SDN enabled 5G-VANET: Adaptive vehicle clustering and beamformed transmission for aggregated traffic[J]. IEEE Communications Magazine, 2017, 55(7): 120-127.

[10] JUNG J Y, BAE J, LIU L. Hierarchical clustering of business process models[J]. International Journal of Innovative Computing, Information and Control, 2009, 5(12): 4501-4511.

[11] PUNHANI R, ARORA V P S, SABITHA S, et al. Application of clustering algorithm for effective customer segmentation in E-commerce[C]//2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE). Piscataway: IEEE, 2021: 149-154.

[12] FU D Q, HE S J. New combination algorithms in commercial area data mining and clustering[C]//2016 IEEE International Conference on Big Data Analysis (ICBDA). Piscataway: IEEE, 2016: 1-5.

[13] MUSMECI N, ASTE T, MATTEO T D. Relation between financial market structure and the real economy: Comparison between clustering methods[J]. PLoS One, 2015, 10(3): e0116201.

[14] TETE T N, KAMLU S. Detection of plant disease using threshold, k-mean cluster and ann algorithm[C]//2017 2nd International Conference for Convergence in Technology (I2CT). Piscataway: IEEE, 2017: 523-526.

[15] BABKIN A, TASHENOVA L, SMIRNOVA O, et al. Analyzing the trends in the digital economy and the factors of industrial clustering[C]//Proceedings of the 2nd International Scientific Conference on Innovations in Digital Economy. New York: ACM, 2020: 1-10.

[16] BORDOGNA G, PASI G. A quality driven hierarchical data divisive soft clustering for information retrieval[J]. Knowledge-Based Systems, 2012, 26: 9-19.

[17] MURTAGH F, CONTRERAS P. Algorithms for hierarchical clustering: An overview[J]. WIREs Data Mining and Knowledge Discovery, 2012, 2(1): 86-97.

[18] RUSPINI E H, BEZDEK J C, KELLER J M. Fuzzy clustering: A historical perspective[J]. IEEE Computational Intelligence Magazine, 2019, 14(1): 45-55.

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

[20] MEHMOOD R, ZHANG G Z, BIE R F, et al. Clustering

by fast search and find of density peaks via heat diffusion [J]. Neurocomputing, 2016, 208: 210-217.

[21] 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.

[22] GAN G J, ZHANG Y P, DEY D K. Clustering by propagating probabilities between data points[J]. Applied Soft Computing, 2016, 41: 390-399.

[23] JAHANGOSHAI REZAEE M, ESHKEVARI M, SABERI M, et al. GBK-means clustering algorithm: An improvement to the K-means algorithm based on the bargaining game[J]. Knowledge-Based Systems, 2021, 213: 106672.

[24] 程乐峰. 电力市场多群体策略博弈的长期演化稳定均衡理论研究[D]. 广州: 华南理工大学, 2019.
CHENG Y F. Theoretical Investigation on the Long-Term Evolutionarily Stable Equilibrium of Multi-population Strategic Games in Electricity Market[D]. Guangzhou: South China University of Technology, 2019. (in Chinese)

[25] ALDERSHOF B, MARRON J S, PARK B U, et al. Facts about the Gaussian probability density function[J]. Applicable Analysis, 1995, 59(1/2/3/4): 289-306.

[26] CHANG S, SHIHONG Y, Qi L. Clustering characteristics of UCI dataset[C]//2020 39th Chinese Control Conference (CCC). IEEE, 2020: 6301-6306.

[27] AL-JABERY K, OBAFEMI-AJAYI T, OLBRICHT G, et al. Computational Learning Approaches to Data Analytics in Biomedical Applications[M]. San Diego: Academic Press, 2020.

[28] HARTIGAN J A, WONG M A. A K-means clustering algorithm[J]. Journal of the Royal Statistical Society Series C: Applied Statistics, 1979, 28(1): 100-108.

[29] Bäcklund H, Hedblom A, Neijman N. A density-based spatial clustering of application with noise[J]. Data Mining TNM033, 2011: 11-30.

[30] LOTFI A, MORADI P, BEIGY H. Density peaks clustering based on density backbone and fuzzy neighborhood [J]. Pattern Recognition, 2020, 107: 107449.

[31] WANG S L, LI Q, ZHAO C F, et al. Extreme clustering—A clustering method via density extreme points[J]. Information Sciences, 2021, 542: 24-39.

作者简介

常璐瑶 女, 1999 年 6 月出生于河南省平顶山市. 电子科技大学硕士研究生. 主要研究方向为数据挖掘与机器学习.
E-mail: chang_ly16@163.com

牛新征 男, 1978 年 5 月出生于贵州省贵阳市. 现为电子科技大学教授级高级工程师, 主要研究方向为数据挖掘和信息安全.
E-mail: xinzhengniu@uestc.edu.cn

罗 涛 男, 1999 年 6 月出生于四川省峨眉山市. 电子科技大学硕士研究生. 主要研究方向为图神经网络.
E-mail: llltt0603@163.com

钱早国 男, 1999 年 4 月出生于贵州省遵义市. 电子科技大学硕士研究生. 主要研究方向为分布式计算.
E-mail: mansurn@163.com