文章编号:1671-6833(2024)05-0086-09

基于边界剥离思想的全局中心聚类算法

程明畅[1], 敖 兰[1,2], 刘 浏[1,3,4]

(1. 四川师范大学 可视化计算与虚拟现实四川省重点实验室,四川 成都 610066;2. 四川师范大学 数学科学学院,四川 成都 610066;3. 成都理工大学 数学地质四川省重点实验室,四川 成都 610059;4. 成都理工大学 数理学院,四川 成都 610059)

摘 要: 全局中心聚类算法如 k-means、谱聚类在类簇分布出现重叠粘连现象时往往容易陷入局部最优且参数难以设定,极大地限制了全局中心聚类算法在实际应用中的效果。为解决此问题,提出了一种基于边界剥离思想的全局中心聚类算法。首先,设计了一步边界剥离法,根据样本点间的反向 k 近邻关系定义了一种局部距离加权密度,并利用密度经验分布函数一阶差分最大处的密度值作为阈值将数据集分为边界集与核心集。其次,嵌入传统的全局中心聚类算法对核心集进行聚类,得益于核心集的簇间重叠问题已明显改善,嵌入算法将更容易收敛到真实的簇中心。最后,提出一种边界吸引算法,从已被归类的核心集样本点出发,借助已有的反向 k 近邻关系迭代融合边界集中的样本点以完成对整个数据集的聚类。相较于目前以迭代方式进行的边界剥离算法,所提算法在计算效率上具有明显优势,不需要额外设定复杂的终止条件而直接通过阈值进行边界划分,并且全局性方法在数据局部密度存在差异的情形下具备更强的鲁棒性。在实验阶段,采用 3 个合成数据集以及 6 个真实数据集从算法性能、参数敏感性、时间消耗多个方面进行评估,实验结果进一步验证了此算法的有效性与实用性。

关键词:全局中心聚类算法; 边界剥离; 簇重叠; 反向 k 近邻; 经验分布

中图分类号:TP311.13; TP391 **文献标志码**:A **doi**:10.13705/j.issn.1671-6833.2024.02.002

全局中心聚类算法如 k-means、谱聚类等凭借其算法的高效、易移植、易推广等特点仍流行于目前绝大多数应用领域。此类算法旨在寻找全局上的簇中心以发现真实的类簇,因此将该类算法统称为全局中心聚类算法。相较基于密度带噪声应用的空间聚类算法(density-based spatial clustering of applications with noise, DBSCAN)[1]、均值漂移算法(mean-shift, MS)[2]、密度峰值聚类算法(density peaks clustering, DPC)[3-4]等,全局中心聚类算法的一大优势在于不易引发链式效应(chain effect)而导致簇的错误合并。但此类算法对样本空间中簇的凸型与良好分离假设具有较强的依赖性,并且对所选初始簇中心的位置较为敏感,容易陷入局部最优解。特别在实际问题中,数据多数特征不具有明显的区分度,这导致簇间重叠问题难以避免,其严重影响了算法对真实簇中心的识别准确率,并加剧了算法的局部收敛问题。

近年来,许多学者对全局中心聚类算法做出了一系列的改进工作,包括引入分裂融合技术提高算法自适应性[5-6],定义新的距离度量方式以改善算法局部收敛问题[7-10]。虽然目前的各种改进算法从理论方法上对传统算法进行了优化,但其并没有放宽全局中心聚类算法对良好分离假设的依赖,当数据集内簇的分布情况欠佳时,算法的聚类效果仍然不理想。边界剥离算法(border peeling, BP)[11]通过在算法过程中排除簇边缘的样本点或离群点,使得聚类结果更加稳健。BP 算法将边界点定义为局部密度较低的点,其迭代剥离低密度样本点以发现簇的核心区域,进一步利用 DBSCAN 算法对高密度点进行聚类,最终将边界点按剥离路径逆向合并到每个簇中。作为一种针对密度型算法的改进,BP 算法能够有效改善簇间重叠问题从而克服链式效应的

收稿日期:2023-11-03;**修订日期**:2023-12-25

基金项目:国家自然科学基金资助项目(12075162);数学地质四川省重点实验室开放基金资助(scsxdz2023-4);四川师范大学学科建设专项(XKZX2021-04)

通信作者:刘浏(1981—),男,四川威远人,成都理工大学教授,博士,主要从事大数据分析、统计过程控制研究,E-mail:liuliums@cdut.edu.cn。

引用本文:程明畅,敖兰,刘浏. 基于边界剥离思想的全局中心聚类算法[J]. 郑州大学学报(工学版),2024,45(5):86-94. (CHENG M C, AO L, LIU L. Border-peeling inspired globally central clustering algorithm[J]. Journal of Zhengzhou University(Engineering Science), 2024,45(5):86-94.)

不良影响,但迭代剥离过程受局部密度差异性的影响,迭代终止条件较难确定且计算复杂。ROBP[12]作为 BP 算法的一种改进,其利用 Cauchy 核函数进行密度估计中的局部放缩处理,并设计了一种基于共享邻域信息的链接准则,能够从一定程度上缓解局部密度差异性带来的影响并提高算法稳健性,但仍然使用迭代方式进行边界剥离,计算效率与终止条件的设定问题难以解决。

为克服簇间重叠所导致的局部收敛问题,本文提出一种基于边界剥离思想的全局中心聚类算法(border-peeling inspired globally central clustering algorithm, BPCC)。该算法的核心过程主要为边界剥离和边界吸引两个步骤。不同于上述的 BP 算法,此处的边界剥离方式不再是通过逐层迭代来识别边界点,而是通过样本点间的反向 k 近邻关系定义样本点密度,并利用密度的经验分布函数自动确定分割阈值并一次性地剥离所有边界点。较 BP 算法而言,设计的一步边界剥离法对数据的局部密度差异不再敏感,且无须额外设定烦琐的终止条件,能够快速地剥除边界点以提高簇间分离度,从而显著降低被保留核心点所在区域内的簇间重叠程度,有效改善全局中心聚类算法的收敛状况。在边界吸引过程中利用已有的反向 k 近邻关系,从已分配标签的核心点出发,通过迭代将边界点重新划归到各个类簇,其过程中不产生额外的距离计算。

1　全局中心聚类算法

1.1　k-means 聚类算法

k-means 聚类算法是最具代表性的全局中心聚类算法,其核心思想为将每个样本点分配到距离最近的中心点所在的簇中,因此类标签与簇中心实则为聚类目标的一体两面。换而言之,如何准确地找到簇中心是最为关键的问题。对于一组数据 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]\in\mathbf{R}^{n\times p}$,其中 n 为数据量,p 为数据维数,给定一个整数 K,存在一组划分为 $C=\{C_1,C_2,\cdots,C_K\}$ 的集合,则 k-means 问题[13]一般被描述为如下的最小化目标函数问题:

$$J(C)=\sum_{j=1}^{k}\sum_{\boldsymbol{x}_i\in C_j}\|\boldsymbol{x}_i-\boldsymbol{\mu}_j\|^2。\tag{1}$$

式中:$\boldsymbol{\mu}_j=\dfrac{1}{|C_j|}\sum_{\boldsymbol{x}_i\in C_j}\boldsymbol{x}_i$ 代表第 j 个簇中心。

由于类标签的 $\{0,1\}$ 属性,上述最小化目标为 NP-hard 问题,无法通过目标函数直接求解,因此诞生了目前被广泛使用的 k-means 算法(lloyd's algorithm)[14]。该算法通过在分配样本点与更新簇中心两个步骤间的迭代得到上述问题的局部最优解,其算法复杂度仅为 $O(n)$,但却对初始条件十分敏感,容易陷入不理想的局部最优解。

1.2　谱聚类算法

谱聚类算法(spectral clustering, SC)[15-16]依据图割(graph cut)的思想发展而来,其将原始样本空间中的点看作图的顶点,通过全连接、k 近邻等方式创建图的边,并由邻接矩阵 $\boldsymbol{W}=\{w_{ij}\}\in\mathbf{R}^{n\times n}$ 表示图中两两点间的相似关系。根据图割理论,以 Ncut[15]为例,谱聚类的目标函数为

$$\begin{cases}\min\limits_{\boldsymbol{H}\in\mathbf{R}^{n\times k}}\operatorname{tr}(\boldsymbol{H}'\boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}\boldsymbol{H}')\ ;\\ \text{s. t. }\ \boldsymbol{H}'\boldsymbol{H}=\boldsymbol{I}。\end{cases}\tag{2}$$

式中:$\boldsymbol{H}$ 为标签矩阵;$\boldsymbol{D}=\begin{cases}d_{ij}=\sum\limits_{z=1}^{n}w_{iz},i=j;\\0,i\neq j。\end{cases}$ 为顶点的度矩阵;$\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W}$ 为图拉普拉斯矩阵。

与 k-means 问题类似,由于问题(2)中的标签矩阵 $\boldsymbol{H}$ 是离散的,目标函数无法直接求解。而根据 Ky Fan 定理[17],式(2)的解存在于由拉普拉斯矩阵 $\boldsymbol{L}$ 最小的 k 个特征值所对应的特征向量所张成的空间中,因此谱聚类算法实际是对 $\boldsymbol{L}$ 进行特征值分解,并将前 k 个最小特征值所对应的特征向量作为聚类对象,最终利用 k-means 算法求得近似解。容易发现谱聚类算法的结果受邻接矩阵 $\boldsymbol{W}$ 的影响较大,因此在相同的数据集上根据不同的规则构造出的 $\boldsymbol{W}$ 可能带来截然不同的聚类结果。

2　BPCC 算法

2.1　边界剥离

对于全局中心聚类算法,通常考虑簇的分布为正态分布或近似正态分布,即越靠近均值处(簇中心)样本点分布越集中,越远离均值处样本点分布越稀疏。为了刻画样本的聚集程度,通过样本点间的反向 k 近邻(reverse k-nearest neighbors, RNN)关系定义样本点的密度。对于数据集内的样本点而言,其 RNN 集合的大小能够从一定程度上代表该样本点局部区域内的密度,适用于发现边界点或离群点,因此借助样本点间的 RNN 关系,能够有效地识别数据集的边界区域。具体地,给出 RNN 的数学定义如下。

定义 1　RNN。对于任意样本点 $\boldsymbol{x}_j$,其 k 近邻集合为 $N_k(\boldsymbol{x}_j)$,则 $\boldsymbol{x}_i$ 的反向 k 近邻集合为

$$RN_k(\boldsymbol{x}_i)=\{\boldsymbol{x}_j\mid\boldsymbol{x}_i\in N_k(\boldsymbol{x}_j)\}。\tag{3}$$

从定义 1 可以看出,RNN 是一种建立在 k 近邻基础上的非对称关系,即 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 不一定互为对方的近邻点。相较于 k 近邻,RNN 能够更加准确地反映出样本点处的密度大小,即样本点 $\boldsymbol{x}_i$ 的 $RN_k(\boldsymbol{x}_i)$ 集合越大,其局部密度也越大。同时,RNN 是一种离散关系,无法反映出 $RN_k(\boldsymbol{x}_i)$ 集合内的样本点在空间中的覆盖范围。为了更精确地定义样本点的密度,利用样本点与其 $N_k(\boldsymbol{x}_i)$ 集合中点的平均距离作为权重,从而可以在 RNN 的基础上进一步定义样本点的密度。

定义 2 样本点密度。对于任意样本点 $\boldsymbol{x}_i$,其密度为

$$\rho(\boldsymbol{x}_i) = q_i \left| RN_k(\boldsymbol{x}_i) \right| 。 \tag{4}$$

式中:$|\cdot|$ 表示集合内元素的个数;q_i 为 $\boldsymbol{x}_i$ 处密度的局部距离权重,

$$q_i = \exp\left(-\frac{1}{k} \sum_{\boldsymbol{x}_j \in N_k(\boldsymbol{x}_i)} dis(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 \right) ; \tag{5}$$

$dis(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 为样本点 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 的欧氏距离,

$$dis(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{l=1}^{p} (\boldsymbol{x}_{il} - \boldsymbol{x}_{jl})^2} 。 \tag{6}$$

容易发现 q_i 越小表示在 $\boldsymbol{x}_i$ 处的 k 近邻范围内样本点的平均距离越大,即局部分布越稀疏,则应赋予该处较小的密度权重。

在得到每个样本点的密度后,进一步利用样本点密度的分布信息设计一种快速简洁的全局一步剥离法。

定义 3 经验分布函数。设 $Y = \{Y_i \mid i = 1,2,\cdots,n\}$ 为来自总体 F 的一组随机样本,则经验分布函数(empirical distribution function, EDF)可定义为

$$F_n(y) = \frac{1}{n} \sum_{i=1}^{n} I\{Y_i \leqslant y\} 。 \tag{7}$$

其中,$I\{\cdot\}$ 为示性函数,

$$I\{Y_i \leqslant y\} = \begin{cases} 1, & Y_i \leqslant y; \\ 0, & Y_i > y。 \end{cases} \tag{8}$$

样本点密度的 EDF 能够从全局上直观地反映出密度值分布的变化情况,并且 EDF 为离散函数,因此一种简单有效的边界剥离方式就是取 EDF 数列一阶差分最大处所对应的密度值作为划分阈值。为此将样本点密度区间平均划分为 S 段,分段区间端点为 $\{\rho_{(0)}, \rho_{(1)}, \cdots, \rho_{(S)}\}$,则划分边界点与核心点的密度阈值为

$$T = \underset{\{\rho_{(t)} \mid t=0,1,\cdots,S-1\}}{\arg\max} \left(F_n(\rho_{(t+1)}) - F_n(\rho_{(t)}) \right) 。 \tag{9}$$

定义 4 边界点与核心点。记被剥离的样本点为 $\boldsymbol{b}_i$,称为边界点,其构成的集合记为 B,称为边界集;记保留的样本点为 $\boldsymbol{m}_i$,称为核心点,其构成的集合记为 M,称为核心集;$B \cap M = \phi$,$B \cup M$ 为全数据集。

最后结合式(10)与定义 4 给出数据集 X 的全局一步剥离条件为

$$\begin{cases} \boldsymbol{x}_i \in B, & \rho(\boldsymbol{x}_i) \leqslant T; \\ \boldsymbol{x}_i \in M, & \rho(\boldsymbol{x}_i) > T。 \end{cases} \tag{10}$$

如图 1 所示,EDF 为取值在 [0,1] 上的单调递增函数,EDF 一阶差分最大处的横坐标值即为分割阈值 T。

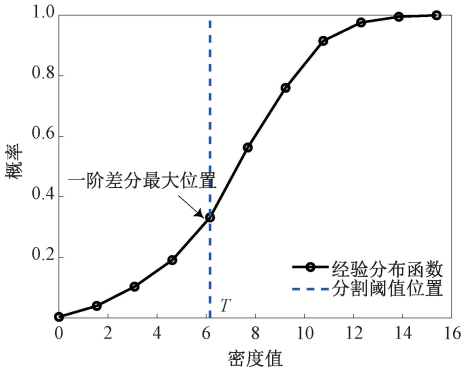

图 1 由经验分布函数确定边界阈值

Figure 1 Determine the boundary threshold using empirical distribution function

2.2 聚类过程

2.2.1 嵌入全局中心聚类算法

在完成边界剥离后,嵌入 k-means、谱聚类等全局中心聚类算法对核心集 M 进行聚类,得到所有核心点的类标签。在簇的凸型假设下,全局中心聚类算法的目标是找到真实的簇中心位置。由于 M 是原始数据集 X 的子集,并且相较于 X,在 M 上各簇的分离结构更加清晰,能够有效避免簇间重叠问题,进而使得嵌入算法能够更容易地收敛到准确的簇中心。

2.2.2 边界吸引

在嵌入算法完成对核心集 M 的聚类后,各个簇中心位置已被确定且所有核心点已被归类,最后的任务则是为边界点分配类标签。从已被归类的样本点 $\boldsymbol{m}_i \in M$ 出发,采用迭代将 $RN_k(\boldsymbol{m}_i)$ 集合内距 $\boldsymbol{m}_i$ 最近的边界点 $\boldsymbol{b}_j$ 划归到 $\boldsymbol{m}_i$ 所属的类中,即 $\boldsymbol{I}(\boldsymbol{b}_j) = \boldsymbol{I}(\boldsymbol{m}_i)$,其中 $\boldsymbol{I} \in \mathbf{R}^n$ 为类标签向量。

$$\boldsymbol{b}_j = \underset{\boldsymbol{b}_j \in RN_k(\boldsymbol{m}_i)}{\arg\min} dis(\boldsymbol{b}_j, \boldsymbol{m}_i) 。 \tag{11}$$

进一步更新核心集 M 和 $\boldsymbol{m}_i$ 的 RNN 集合:

$$M = M \cup \{\boldsymbol{b}_j\} 。 \tag{12}$$

$$RN_k(\boldsymbol{m}_i) = RN_k(\boldsymbol{m}_i) - \{\boldsymbol{b}_j\} 。 \tag{13}$$

如图 2 所示,在通过 RNN 关系建立的图中,核

心点 $\boldsymbol{m}_i$ 与边界点 $\boldsymbol{b}_j$ 间存在边的连接,表示 $\boldsymbol{b}_j$ 属于集合 $RN_k(\boldsymbol{m}_i)$。边界吸引过程将持续迭代至所有核心点的 $RN_k(\boldsymbol{m}_i)$ 集合均为空。在一次迭代中可能出现同一边界点被两个以上核心点吸引的情况,处理方式为将该边界点归类到最后将其吸引的核心点所在的簇中。

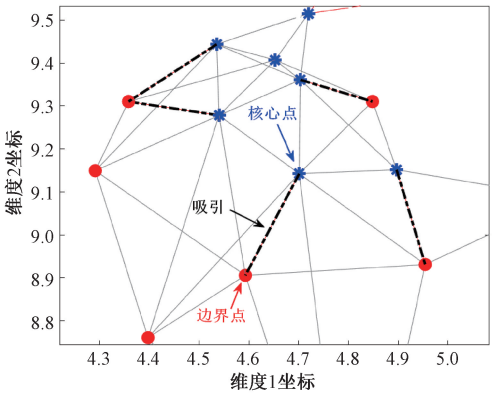

图 2　D31 数据集局部区域上边界吸引过程一次迭代示例

Figure 2　Example of one iteration of the boundary attraction process on local regions of D31 dataset

2.3　算法描述

综上,本文所提的 BPCC 算法流程如图 3 所示,伪代码如下。

算法 1　BPCC 算法。

输入:数据集 $\boldsymbol{X}$, 近邻数 k,簇数目 K;

输出:聚类标签向量 $\boldsymbol{I}$。

#初始化

① $\boldsymbol{Dis}_{n\times n} \leftarrow$ 两两样本点欧式距离矩阵,其中 $\boldsymbol{Dis}(i,j)$ 由式(6)计算得到;

② $\boldsymbol{I}_{n\times 1} \leftarrow$ 全 0 向量;

#边界剥离

③ $[M,\ RN_k] \leftarrow BP(\boldsymbol{X},\ k,\ \boldsymbol{Dis})$;

#嵌入算法聚类

④ $\boldsymbol{I}(M) \leftarrow$ 根据设定类簇数目 K,利用全局中心聚类算法(如 k-means、SC)对核心集 M 中的样本点进行聚类并更新标签,其余样本点标签保持为 0;

#边界吸引

⑤ $\boldsymbol{I} \leftarrow BA(\boldsymbol{I},\ M,\ RN_k)$;

⑥ Return $\boldsymbol{I}$。

算法 2　边界剥离(BP)算法。

输入:$\boldsymbol{X}$, k, $\boldsymbol{Dis}$;

输出:核心集 M, 样本点的 RNN 集合 RN_k。

① $k\boldsymbol{N}_{n\times k} \leftarrow$ 全零矩阵,表示 k 近邻关系矩阵;

② For $i = 1:n$ do:

③　　$k\boldsymbol{N}(i,:) \leftarrow \boldsymbol{Dis}(i,:)$ 前 k 个最小距离对应的列标签,表示样本点 $\boldsymbol{x}_i$ 的 $N_k(\boldsymbol{x}_j)$ 集合;

④ End for

⑤ For　$j = 1:n$ do:

⑥　　$RN_k(\boldsymbol{x}_j) \leftarrow k\boldsymbol{N}$ 中所有包含 j 的行标签集合;

⑦　　$\rho(\boldsymbol{x}_j) \leftarrow$ 根据式(5)计算 $\boldsymbol{x}_j$ 的密度;

⑧ End for

⑨ $T \leftarrow$ 密度区间等分为 S 段(默认 $S=10$),根据式(10)计算分割阈值;

⑩ $M \leftarrow$ 根据式(11)找到 $\boldsymbol{X}$ 中的核心点集合;

⑪ Return M, RN_k。

算法 3　边界吸引(BA)算法。

输入:$\boldsymbol{I}, M, RN_k$;

输出:聚类标签向量 $\boldsymbol{I}$。

① $\boldsymbol{F}_{n\times 1} \leftarrow$ 全 1 向量,1 表示对应样本点还未被查询;

② For $i = 1:n$ do:

③　　将 $RN_k(\boldsymbol{x}_i)$ 中属于核心集 M 的样本点剔除;

④　　IF $RN_k(\boldsymbol{x}_i) = \varnothing$ do:

⑤　　$\boldsymbol{F}(\boldsymbol{x}_i) \leftarrow 0$,表示 $\boldsymbol{x}_i$ 已被查询并为离群点;

⑥　　End if

⑦ End for

⑧ While $\boldsymbol{F}$ 不为全 0 向量　do:

⑨　　$E \leftarrow M \cap \{\boldsymbol{x}_i \mid \boldsymbol{F}(\boldsymbol{x}_i) = 1\}$ 为还未被查询的核心点;

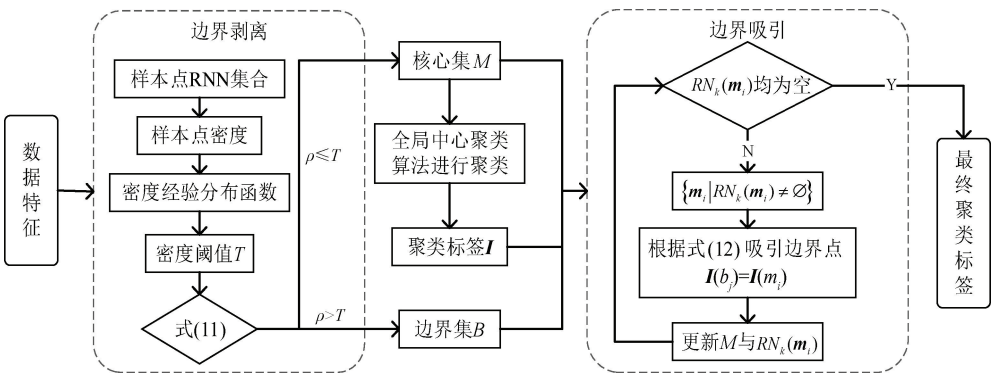

图 3　BPCC 算法流程

Figure 3　Flow chart of BPCC

⑩ IF $E=\varnothing$ do:
⑪ $\boldsymbol{I}(\boldsymbol{F}==1)\leftarrow 0$;
⑫ Break
⑬ End if
⑭ For $\boldsymbol{m}$ in E do:
⑮ $tmp\leftarrow RN_k(\boldsymbol{x}_i)$ 中第 1 个样本点;
⑯ $M\leftarrow M\cup\{tmp\}$;
⑰ $\boldsymbol{I}(tmp)\leftarrow\boldsymbol{I}(\boldsymbol{m})$;
⑱ $RN_k(\boldsymbol{m}_i)\leftarrow RN_k(\boldsymbol{m})-\{tmp\}$;
⑲ IF $RN_k(\boldsymbol{m})=\varnothing$ do:
⑳ $\boldsymbol{F}(\boldsymbol{m})\leftarrow 0$;
㉑ End if
㉒ End for
㉓ End while
㉔ Return $\boldsymbol{I}$。

3 算法实验

为了验证所提 BPCC 算法对全局中心聚类算法的实际提升效果,本文分别在 9 个数据集(见表 1)上进行算法的对比实验,并选择 k-means 算法(KM),分别以全连接、k 近邻方式建立相似度矩阵的谱聚类算法 SC-full、SC-KNN 以及 Power k-means(PKM)作为对比算法。相应地,将 KM、SC-full、SC-KNN 算法嵌入到 BPCC 中,分别称为 BP-KM、BP-SC-full、BP-SC-KNN。本文所有实验的运行环境为 Inter i9-13900K (5.8 GHz),32 GB RAM,Windows11 64 bit,MATLAB R2022b。由于实验数据集具有真实类标签,本文利用 3 个外部评价指标:纯度 *Purity*[18]、归一化互信息 *NMI*[19]、调整的兰德指数 *ARI*[20] 评价聚类算法的效果,其中 *Purity* 和 *NMI* 取值为 [0,1],*ARI* 取值为 [-1,1],各指标得分值越大,说明聚类效果越理想。

表 1 实验数据集

Table 1 Experimental datasets

数据集	来源	样本数量	维数	类别数
D31[21]	合成	3 100	2	31
R15[21]	合成	600	2	15
Aggregation[22]	合成	788	2	7
Iris[23]	UCI	150	4	3
Seeds[23]	UCI	210	7	3
Wine[23]	UCI	178	13	3
Texture[24]	Keel	5 500	40	11
Segment[24]	Keel	2 310	19	7
Optdigits[24]	Keel	5 620	65	10

3.1 合成数据集实验分析

对比实验中,合成数据集选择了 D31、R15、Aggregation,所有算法的簇数目参数均指定为各个数据集的真实簇数目,SC-KNN 以及 BPCC 算法中需要指定的近邻参数 k 的取值见表 2,下文中真实数据集上的参数设置相同,故后不赘述。图 4 展示了近邻参数 $k=8$ 时所提方法的边界剥离效果,可以看到分布在各个簇边界的样本点基本被识别为边界点,保留的核心点所在区域内的簇间重叠粘连现象得到明显改善。

表 2 最优聚类结果中近邻参数 k 的取值

Table 2 Parameter setting of k in optimal clustering results

数据集	近邻参数 k			
	SC-KNN	BP-SC-KNN	BP-SC-full	BP-KM
D31[21]	13	11	4	8
R15[21]	9	9	4	4
Aggregation[22]	14	14	4	3
Iris[23]	10	15	10	10
Seeds[23]	18	19	19	19
Wine[23]	9	20	20	20
Texture[24]	8	8	9	13
Segment[24]	28	27	6	8
Optdigits[24]	8	15	13	17

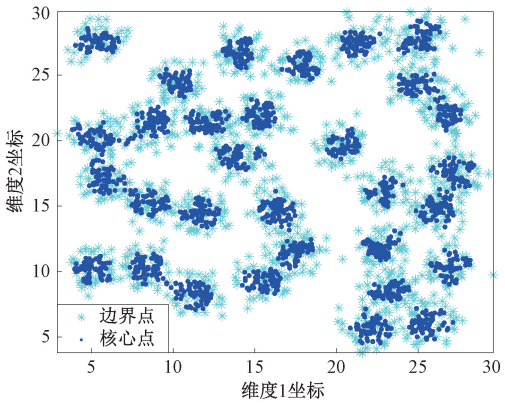

图 4 D31 数据集上全局一步剥离法效果

Figure 4 The performance of the global one-step peeling method on the D31 dataset

实验结果如表 3 所示,在合成数据集的理想环境下,BPCC 的聚类结果并不一定优于传统全局中心聚类算法,但也十分接近。需要说明的是,出现此种情况是符合预期的,因为合成数据本身具有良好的簇分布,簇间重叠粘连问题本不严重,传统全局中心聚类算法也能收敛到准确的簇中心,这便导致边界剥离的作用十分有限。另一方面,由于边界吸引过程省略了样本点到中心点的距离计算,反而造成在部分数据集上 BPCC 的结果略逊于直接使用原始算法的结果。

3.2 真实数据集实验结果

真实数据集选择了 Iris、Seeds、Wine、Texture、

Segment、Optdigits,与合成数据集显著的差别在于更高的维数会导致簇间重叠问题加剧。考虑到其中部分数据集不同维度的量纲差异较大,因此利用 Z-score 标准化方法对 Wine、Segment、Optdigits 3 个数据集进行了标准化处理。实验结果如表 4、表 5 所示。

3.3　参数敏感性分析

此外,BPCC 依赖于样本点间的 RNN 关系以确定样本点的密度,因此近邻参数 k 的选取会直接影响边界剥离的效果,并且由此参数所确定的 RNN 集合还将进一步决定边界吸引过程中边界点的类标签分配,因此有必要对参数 k 进行敏感性分析,以验证所提算法的稳健性。具体而言,本文取 k 在 $[3,30]$,k 为整数,分别在 9 个数据集上以 BP-KM 进行实验。实验结果如图 5 所示,在较大的参数取值范围中,所提算法在大部分数据集上的表现并没有出现明显的波动,且呈现出样本量越大,参数表现越稳定的趋势。由此可见,BPCC 对近邻参数 k 并不敏感,其在实际使用中的效果不易受到参数选择的制约。

3.4　时间消耗分析

3.4.1　时间复杂度分析

BPCC 的时间消耗主要由 3 部分构成:①边界剥离阶段需要借助 RNN 关系计算每个样本点的密度,为此需要计算样本点间的两两距离,其计算复杂度为 $O(n^2)$;②需要利用全局中心聚类算法对核心

表 3　合成数据集聚类结果比较

Table 3　Comparison of clustering results for synthetic datasets

方法	D31 数据集			R15 数据集			Aggregation 数据集		
	Purity	*NMI*	*ARI*	*Purity*	*NMI*	*ARI*	*Purity*	*NMI*	*ARI*
KM	0.890 0	0.935 2	0.860 4	**0.996 7**	**0.994 2**	**0.992 8**	0.810 9	0.845 3	0.737 1
BP-KM	0.949 4	0.937 7	0.899 6	0.993 3	0.989 3	0.985 9	0.847 7	0.891 1	0.797 7
SC-full	**0.976 5**	**0.9670**	**0.952 2**	**0.996 7**	**0.994 2**	**0.992 8**	**0.993 7**	**0.982 4**	**0.986 9**
BP-SC-full	0.960 3	0.948 6	0.920 6	0.993 3	0.989 3	0.985 9	0.991 1	0.980 9	0.979 9
SC-KNN	0.695 2	0.828 2	0.605 6	0.985 0	0.977 7	0.967 8	0.748 7	0.791 2	0.653 8
BP-SC-KNN	0.928 7	0.920 2	0.861 8	0.991 7	0.986 4	0.982 1	0.988 6	0.970 3	0.976 9
PKM	0.822 3	0.886 9	0.754 9	0.851 7	0.856 2	0.746 2	0.888 3	0.860 1	0.854 0

表 4　UCI 数据集聚类结果比较

Table 4　Comparison of clustering results for UCI datasets

方法	Iris 数据集			Seeds 数据集			Wine 数据集		
	Purity	*NMI*	*ARI*	*Purity*	*NMI*	*ARI*	*Purity*	*NMI*	*ARI*
KM	0.886 7	0.741 9	0.716 3	0.890 5	0.710 1	0.710 3	0.971 9	0.892 6	0.914 9
BP-KM	0.966 7	0.880 1	0.903 7	**0.914 3**	**0.719 9**	**0.761 9**	**0.977 5**	**0.911 9**	**0.932 6**
SC-full	0.900 0	0.766 1	0.743 7	0.890 5	0.686 9	0.705 4	0.623 6	0.562 0	0.428 8
BP-SC-full	0.966 7	0.880 1	0.903 7	0.900 0	0.698 5	0.725 6	**0.977 5**	**0.911 9**	**0.932 6**
SC-KNN	0.840 0	0.722 4	0.642 3	0.742 9	0.595 1	0.517 9	0.797 8	0.476 5	0.476 9
BP-SC-KNN	**0.973 3**	**0.901 1**	**0.922 2**	0.904 8	0.712 8	0.739 0	0.971 9	0.897 4	0.916 8
PKM	0.926 7	0.811 8	0.800 8	0.895 2	0.694 9	0.716 6	0.966 3	0.875 9	0.897 5

表 5　Keel 数据集聚类结果比较

Table 5　Comparison of clustering results for Keel datasets

方法	Texture 数据集			Segment 数据集			Optdigits 数据集		
	Purity	*NMI*	*ARI*	*Purity*	*NMI*	*ARI*	*Purity*	*NMI*	*ARI*
KM	0.433 5	0.587 2	0.412 2	0.561 9	0.587 2	0.443 5	0.587 9	0.648 1	0.531 3
BP-KM	0.489 5	0.691 3	0.569 5	0.732 5	0.646 3	**0.613 9**	0.658 5	0.726 1	0.646 6
SC-full	0.444 9	0.647 2	0.519 1	0.145 9	0.038 3	/	0.352 1	0.507 2	0.355 5
BP-SC-full	0.462 2	0.723 5	0.598 8	**0.735 9**	**0.651 8**	0.568 7	0.637 7	0.704 2	0.618 0
SC-KNN	0.582 7	0.846 4	0.730 9	0.518 6	0.586 9	0.440 9	0.631 9	0.706 6	0.599 0
BP-SC-KNN	**0.587 1**	**0.903 8**	**0.833 6**	0.520 3	0.578 5	0.431 2	**0.706 8**	**0.825 2**	**0.757 2**
PKM	0.471 8	0.657 3	0.537 4	0.432 9	0.522 9	0.231 9	0.630 6	0.678 9	0.582 4

注:“/”代表数值小于 5×10^{-4}。

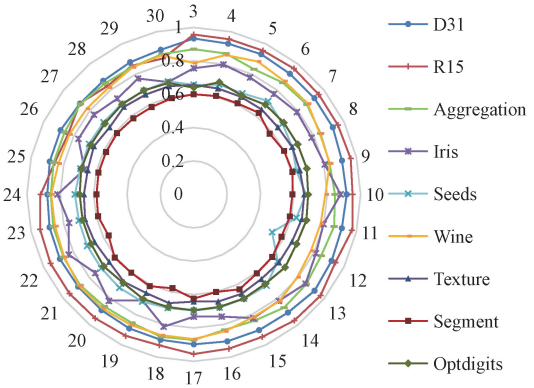

图 5 BP-KM 分别取 k 在[3,30]聚类结果的 *NMI* 值

Figure 5 *NMI* values of clustering results with parameter k in[3, 30] for BP-KM respectively

集进行聚类,这部分的时间复杂度取决于嵌入算法,如利用 k-means 则计算复杂度为 $O(|R|)$,而谱聚类由于会进行 SVD 分解,计算复杂度通常达到 $O(|R|^3)$,其中 $|R|$ 为核心点个数;③在边界吸引过程中需要使用双层循环合并核心点的 RNN 集合内还未被归类的边界点,由于迭代过程中 R 会反复更新,导致迭代次数无法确定,但一定小于 n,故该部分的计算复杂度将小于 $O(n^2)$。综上,BPCC 的全局计算效率在很大程度上取决于嵌入算法的时间复杂度。

3.4.2 模拟数据实验分析

为进一步验证算法效率,随机生成 5 组包含 5 个类簇且样本数量分别为{500, 1 000, 5 000, 10 000, 20 000}的二维模拟数据集进行时间消耗对比,实验结果如图 6 所示。当嵌入算法为 KM 时,时间消耗由边界剥离步骤主导,会慢于直接使用 KM 算法;当嵌入算法为 SC-full 时,时间消耗则由嵌入算法主导,得益于边界点不参与嵌入算法聚类,BP-SC-full 的运行时间显著短于 SC-full。因此对于计算复杂度较高的全局中心聚类算法,BPCC 能够从聚

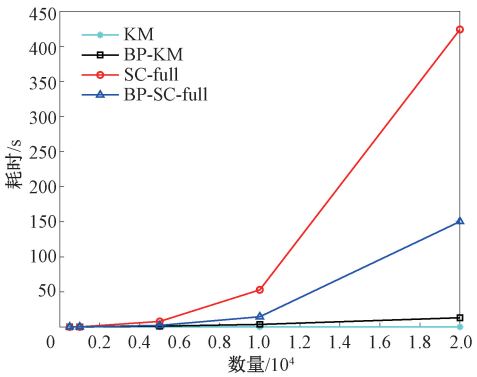

图 6 不同算法时间消耗对比

Figure 6 Time consumption comparison of different algorithms

类效果和计算效率两方面带来提升。

4 结论

本文提出一种基于边界剥离思想的全局中心聚类算法 BPCC。所提算法根据样本点密度的经验分布将样本点快速划分为边界点与核心点。得益于核心集上簇的优良分布,利用全局中心聚类算法能够准确地对核心集进行划分,进一步利用已有的 RNN 关系将边界点分配至各簇,完成最终聚类。实验结果表明,BPCC 能够有效解决簇边界区域重叠所造成的局部收敛问题,提升全局中心聚类算法的聚类表现,特别在实际数据集上优势明显。此外,该算法不会产生高昂的计算成本,在嵌入谱聚类等计算复杂度较高的算法时,还能额外缩减计算成本,提高算法效率。另一方面,虽然 BPCC 对近邻参数 k 不敏感,有利于实际使用,但其仍然需要人为指定簇数目 K,因此该算法在参数自适应性上仍存在提升空间,在未来工作中可考虑引入基于分裂融合策略的聚类算法,以放宽需要指定簇数目作为先验条件的限制,进一步提升算法的适用性。

参考文献:

[1] ESTER M, KRIEGEL H P, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]//Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. Palo Alto: AAAI, 1996: 226-231.

[2] COMANICIU D, MEER P. Mean shift: a robust approach toward feature space analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(5): 603-619.

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

[4] GUO W J, WANG W H, ZHAO S P, et al. Density peak clustering with connectivity estimation[J]. Knowledge-Based Systems, 2022, 243: 108501.

[5] CHENG M C, MA T F, LIU Y B. A projection-based split-and-merge clustering algorithm[J]. Expert Systems with Applications, 2019, 116: 121-130.

[6] SIERANOJA S, FRÄNTI P. Adapting k-means for graph clustering[J]. Knowledge and Information Systems, 2022, 64(1): 115-142.

[7] HUANG J Z, NG M K, RONG H Q, et al. Automated variable weighting in k-means type clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(5): 657-668.

[8] ZHANG Y Q, CHEUNG Y M. A new distance metric exploiting heterogeneous interattribute relationship for ordinal-and-nominal-attribute data clustering [J]. IEEE Transactions on Cybernetics, 2022, 52(2): 758-771.

[9] 周成龙, 陈玉明, 朱益冬. 粒 K 均值聚类算法[J]. 计算机工程与应用, 2023, 59(13): 317-324.
ZHOU C L, CHEN Y M, ZHU Y D. Granular K-means clustering algorithm[J]. Computer Engineering and Applications, 2023, 59(13): 317-324.

[10] 邓秀勤, 郑丽苹, 张逸群, 等. 基于新的距离度量的异构属性数据子空间聚类[J]. 郑州大学学报(工学版), 2023, 44(2):53-60.
DENG X Q, ZHENG L P, ZHANG Y Q, et al. Subspace clustering of heterogeneous-attribute data based on a new distance metric [J]. Journal of Zhengzhou University (Engineering Science), 2023, 44(2):53-60.

[11] AVERBUCH-ELOR H, BAR N, COHEN-OR D. Border-peeling clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(7): 1791-1797.

[12] DU M J, WANG R, JI R, et al. ROBP a robust border-peeling clustering using Cauchy kernel[J]. Information Sciences, 2021, 571: 375-400.

[13] CAPÓ M, PÉREZ A, LOZANO J A. An efficient split-merge re-start for the K-means algorithm [J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(4): 1618-1627.

[14] LLOYD S. Least squares quantization in PCM[J]. IEEE Transactions on Information Theory, 1982, 28(2): 129-137.

[15] SHI J B, MALIK J. Normalized cuts and image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888-905.

[16] VON LUXBURG U. A tutorial on spectral clustering[J]. Statistics and Computing, 2007, 17(4): 395-416.

[17] ZHA H Y, HE X F, DING C, et al. Spectral relaxation for K-means clustering[C]//Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic. Cambridge: MIT, 2001: 1057-1064.

[18] ZHANG X L, WANG W, NØRVÅG K, et al. K-AP: generating specified K clusters by efficient affinity propagation[C]//Proceedings of the 2010 IEEE International Conference on Data Mining. Piscataway: IEEE, 2010: 1187-1192.

[19] MEILĂ M. Comparing clusterings—an information based distance[J]. Journal of Multivariate Analysis, 2007, 98(5):873-895.

[20] HUBERT L, ARABIE P. Comparing partitions[J]. Journal of Classification, 1985, 2:193-218.

[21] 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.

[22] GIONIS A, MANNILA H, TSAPARAS P. Clustering aggregation[C]// Proceeding of the 21st International Conference on Data Engineering (ICDE'05). Piscataway: IEEE, 2005:341-352.

[23] KELLY M, LONGJOHN R, NOTTINGHAM K. The UCI machine learning repository[DB/OL]. [2023-06-29] https://archive.ics.uci.edu/datasets.

[24] ALCALA-FDEZ J, FERNANDEZ A, LUENGO J, et al. KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework [J]. Journal of Multiple-Valued Logic and Soft Computing, 2011, 17(2/3): 255-287.

Border-peeling Inspired Globally Central Clustering Algorithm

CHENG Mingchang[1], AO Lan[1,2], LIU Liu[1,3,4]

(1. V. C. & V. R. Key Lab of Sichuan Provence, Sichuan Normal University, Chengdu 610066, China; 2. School of Mathematical Sciences, Sichuan Normal University, Chengdu 610066, China; 3. Geomathematics Key Laboratory of Sichuan Province, Chengdu University of Technology, Chengdu 610059, China; 4. College of Mathematics and Physics, Chengdu University of Technology, Chengdu 610059, China)

Abstract: The globally central clustering algorithms, such as k-means and spectral clustering, often suffer from the problem of local optima and difficulty in parameter setting with overlapping and adhesive clusters in the data distribution, which might greatly limits the effectiveness of globally central clustering algorithms in practical applications. To address this issue, a border-peeling inspired globally central clustering algorithm was proposed. Firstly, a one-step border peeling method was designed, which defines a locally distance-weighted density according

to the reverse k-nearest neighbor relationships between sample points. The density value at the maximal point of the first-order difference of the density empirical distribution function was utilized as the threshold to divide the dataset into boundary and core sets. Then, the traditional globally central clustering algorithms were embedded to cluster the core set. Benefiting from the significant improvement in the overlapping of the core set, the embedding algorithms could converge to the true cluster centers easily. Finally, a boundary attraction algorithm was proposed, which could progressively amalgamate sample points from the boundary set, utilizing existing reverse k-nearest neighbor relationships, and commencing from the already categorized core set sample points. Compared with the currently iterative border peeling algorithms, the proposed algorithm had significant advantages in computational efficiency. There was no additional complex termination conditions but only direct performs boundary partitioning using a threshold. Furthermore, the global approach also exhibited stronger robustness local data densities were different. In the experimental phase, three synthetic datasets and six real-world datasets were used to evaluate the algorithm′s performance, parameter sensitivity, and time consumption, further validating the efficacy and practicality of this algorithm.

Keywords: globally central clustering algorithm; border peeling; overlapping; reverse k-nearest neighbors; empirical distribution

(上接第60页)

Network Intrusion Detection Method Based on Improved Multi-factorial Optimization Bat Algorithm

ZHANG Zhen, ZHANG Siyuan, TIAN Hongpeng

(School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: In addressing the challenge of diminished intrusion detection accuracy resulting from the abundance of redundant and irrelevant features in high-dimensional network data, an improved multi-factorial optimization bat algorithm (IMFBA) was introduced for precise data feature selection, with the ultimate goal of improving network intrusion detection accuracy. Within the multi-factorial optimization framework, global and local feature selection tasks were formulated. Information exchange between these tasks was facilitated by selection and vertical cultural transmission operators, strategically designed based on the bat algorithm. The global feature selection task was accelerated in identifying optimal solution spaces, thereby enhancing the algorithm′s convergence speed and stability. By incorporating the reverse learning strategy and differential evolution into the bat algorithm, the initial solution selection stage and individual updating process were refined to address the absence of a mutation mechanism, fostering solution diversity and aiding the algorithm in escaping local optima. An adaptive parameter adjustment strategy was introduced, determining weightings for guiding individual updates based on potential optimal solution quality. This could mitigate the risk of knowledge negative transfer during multi-task feature selection, achieving a balance between global exploration and local exploitation. The feature subsets selected by IMFBA demonstrate classification accuracy of 95.37% and 85.14% on the KDD CUP 99 and NSL-KDD intrusion detection datasets, respectively. This reflected increased by 3.01 percentage points and 9.78 percentage points compared to the complete dataset. Experiment results confirm the efficacy of EMFBA in selecting higher-quality feature subsets and, consequently, enhancing network intrusion detection accuracy.

Keywords: intrusion detection; cyber security; feature selection; bat algorithm; multi-factorial optimization