软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2021,32(10):3085−3103 [doi: 10.13328/j.cnki.jos.006016]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种采用新型聚类方法的最佳类簇数确定算法*

朱二周[1,2], 孙 悦[2], 张远翔[2], 高 新[2], 马汝辉[3], 李学俊[2]

[1](计算智能与信号处理教育部重点实验室(安徽大学),安徽 合肥 230601)

[2](安徽大学 计算机科学与技术学院,安徽 合肥 230601)

[3](上海交通大学 电子信息与电气工程学院,上海 200240)

通讯作者: 朱二周, E-mail: ezzhu@ahu.edu.cn

摘 要: 聚类分析是统计学、模式识别和机器学习等领域的研究热点.通过有效的聚类分析,数据集的内在结构与特征可以被很好地发掘出来.然而,无监督学习的特性使得当前已有的聚类方法依旧面临着聚类效果不稳定、无法对多种结构的数据集进行正确聚类等问题.针对这些问题,首先将 *K*-means 算法和层次聚类算法的聚类思想相结合,提出了一种混合聚类算法 *K*-means-AHC;其次,采用拐点检测的思想,提出了一个基于平均综合度的新聚类有效性指标 DAS(平均综合度之差,difference of average synthesis degree),以此来评估 *K*-means-AHC 算法聚类结果的质量;最后,将 *K*-means-AHC 算法和 DAS 指标相结合,设计了一种寻找数据集最佳类簇数和最优划分的有效方法.实验将 *K*-means-AHC 算法用于测试多种结构的数据集,结果表明:该算法在不过多增加时间开销的同时,提高了聚类分析的准确性.与此同时,新的 DAS 指标在聚类结果的评价上要优于当前已有的常用聚类有效性指标.

关键词: 聚类分析;聚类算法;聚类有效性指标;最佳类簇数;数据挖掘

中图法分类号: TP181

中文引用格式: 朱二周,孙悦,张远翔,高新,马汝辉,李学俊.一种采用新型聚类方法的最佳类簇数确定算法.软件学报,2021,32(10):3085−3103. http://www.jos.org.cn/1000-9825/6016.htm

英文引用格式: Zhu EZ, Sun Y, Zhang YX, Gao X, Ma RH, Li XJ. Optimal clustering number determining algorithm by the new clustering method. Ruan Jian Xue Bao/Journal of Software, 2021,32(10):3085−3103 (in Chinese). http://www.jos.org.cn/1000-9825/6016.htm

Optimal Clustering Number Determining Algorithm by the New Clustering Method

ZHU Er-Zhou[1,2], SUN Yue[2], ZHANG Yuan-Xiang[2], GAO Xin[2], MA Ru-Hui[3], LI Xue-Jun[2]

[1](Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Anhui University, Hefei 230601, China)

[2](School of Computer Science and Technology, Anhui University, Hefei 230601, China)

[3](School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: Clustering analysis is a hot research topic in the fields of statistics, pattern recognition, and machine learning. Through effective clustering analysis, the intrinsic structure and characteristics of datasets can be well discovered. However, due to the unsupervised learning feature, the existing clustering methods are still facing the problems of unstable and inaccurate on processing different types of datasets. In order to solve these problems, a hybrid clustering algorithm, *K*-means-AHC, is firstly proposed based on the combination of the *K*-means algorithm and the hierarchical clustering algorithm. Then, based on the inflexion point detection, a new clustering validity index, DAS (difference of average synthesis degree), is proposed to evaluate the results of the *K*-means-AHC clustering algorithm. Finally, through the combination of the *K*-means-AHC algorithm and the DAS index, an effective method of finding the optimal clustering numbers and optimal partitions of datasets is designed. The *K*-means-AHC algorithm is used to test many kinds of

* 基金项目: 安徽省自然科学基金(2008085MF188); 国家自然科学基金(61972001)

Foundation item: Natural Science Foundation of Anhui Province, China (2008085MF188); National Natural Science Foundation of China (61972001)

收稿时间: 2019-09-09; 修改时间: 2019-11-17, 2020-01-18; 采用时间: 2020-02-13

datasets. The experimental results have shown that the proposed algorithm improves the accuracy of clustering analysis while without too much time overhead. At the same time, the new DAS index is superior to the current commonly used clustering validity indexes in the evaluation of clustering results.

Key words: clustering analysis; clustering algorithm; clustering validity index; optimal clustering number; data mining

作为一种无监督学习方法,聚类分析在数据挖掘领域具有重要的作用.聚类分析是在缺少先验信息的情况下,根据一些相似性标准将样本点划分为多个类簇.它使得在同一个类簇中的样本点尽可能地相似,而不同类簇中的样本点最大程度的不同[1].目前,聚类分析被广泛用于数据分析、模式识别以及图像处理等多个领域.聚类分析过程中通常要解决两个问题,即如何划分一个给定的数据集并使得划分结果最优以及将数据集划分为多少个类簇最为合适.其中,第 1 个问题通常由聚类算法来解决,而第 2 个问题则由聚类有效性指标来评价[2].

聚类算法是聚类分析的基础.根据类簇的不同形成方式,当前常用的聚类算法主要有基于划分的聚类算法、基于层次的聚类算法、基于模糊理论的聚类算法、基于分布的聚类算法、基于密度的聚类算法、基于图论的聚类算法、基于网格的聚类算法、基于分形理论的聚类算法以及基于模型的聚类算法等[3].目前使用较多的有基于划分的聚类算法和基于层次的聚类算法.基于划分的聚类算法首先将数据集分为 K 个类簇,然后从这 K 个类簇开始,并通过将某个准则最优化以达到最终的聚类结果.作为一种经典的基于划分的聚类算法,*K*-means 具有实现简单、能够对大型数据集进行高效划分的特点.然而,由于受收敛规则的影响,*K*-means 算法对初始类簇中心点的选取非常敏感.不恰当的中心点的选取,会使得该算法非常容易陷入局部最优问题.与此同时,除了凸型数据集以外,*K*-means 算法不能对诸如条形、环形等类型的非凸型数据集进行很好的处理[4].

另一方面,层次聚类算法则是通过将数据组织成若干组,并将其形成一个相应的树状图来进行聚类[5].根据树状图从上到下和从下到上的处理方式,可以将层次聚类分为两类,即分裂法和凝聚法.分裂法首先将数据集中的所有样本点归为一个类簇,然后依据某种准则对初始类簇进行逐步的分裂,直至达到某种预设条件或预定的类簇数为止;相反地,凝聚法在初始时将数据集的每个样本点当作一个类簇,然后依据某种准则合并这些初始的类簇,直至达到某种预设条件或预定的类簇数为止[6].作为层次聚类算法的代表,凝聚层次聚类(agglomerative hierarchical clustering,简称 AHC)算法具有稳定性好、能够对多种类型的数据集进行较好处理等优点.然而,与 *K*-means 算法相比,AHC 算法的计算复杂度较高,故不适合直接用于大型数据集的聚类分析.

鉴于 *K*-means 算法和 AHC 算法的优缺点,本文将这两种算法的思想相结合,提出了一种新的混合聚类算法,即 *K*-means-AHC 算法.*K*-means-AHC 算法首先利用 *K*-means 算法的处理方式,将数据集划分为若干个初始类簇;其次,在形成的初始类簇的基础上,采用 AHC 算法的思想,将较小的初始类簇合并成符合要求的更大的类簇.新的 *K*-means-AHC 算法不仅能够有效避免传统 *K*-means 算法对初始类簇中心点选取敏感和不能很好地处理非凸型数据集的问题,而且可以有效缩短 AHC 算法的计算时间.

对聚类分析而言,不同算法甚至同一算法的不同参数配置都有可能产生同一数据集的不同划分[7].与此同时,很多聚类算法需要提前获得目标数据集的类簇数目.然而,这个数据通常是不能提前获取的.聚类分析通常的做法是:先反复运行聚类算法,直到满足预定的条件为止;然后,运用聚类有效性指标来对聚类算法的划分结果进行评价.作为聚类分析的一个重要组成部分,聚类有效性指标是寻找目标数据集最佳类簇数的关键[8].聚类有效性问题的研究一般是通过建立一个指标函数,即聚类有效性指标来完成的.而最佳类簇数的确定,是在不同 K 值的情况下分别运行聚类算法来获取的.在聚类算法运行的过程中,当指标函数取值达到最优(最大指标值或最小指标值)时,对应的 K 值即为最佳类簇数[9].

根据聚类有效性指标构成成分的不同,可以将其分为仅考虑数据集几何结构信息的聚类有效性指标、仅考虑隶属度的聚类有效性指标和同时考虑数据集几何结构信息和隶属度的聚类有效性指标[10].聚类有效性一直是聚类分析领域的研究热点,一些经典和常用的指标有 CH 指标[11]、COP 指标[12]、DB 指标[13]、Dunn 指标[14]和 I 指标[15]等.然而研究表明,没有哪一种聚类有效性指标可以最优地处理所有类型的数据集.许多现有的聚类有效性指标存在着一些缺点,比如在寻找最佳类簇数时不稳定、无法对多种形状的数据集进行正确评价等[16].为了能够稳定地处理多种类型的数据集,本文提出了一个基于平均综合度的新的聚类有效性指标,即 DAS.其最

大值表示对于数据集非重叠类簇的最佳划分.

本文研究了能够对多种类型的数据集信息进行快速精准划分的聚类算法以及能够对划分结果进行有效评价的聚类有效性指标.综合而言,本文的主要贡献有以下几个方面.

(1) 提出了一种新的 *K*-means-AHC 混合聚类算法.*K*-means-AHC 算法充分结合了 *K*-means 算法简单、高效和 AHC 算法稳定性好、能够对多种类型的数据集进行处理的优点.首先,采用 *K*-means 算法的处理方式将数据集划分为若干个初始类簇,这一做法避免了凝聚层次聚类从单个点开始处理的问题,可在很大程度上减少将若干较小类簇合并成较大类簇的计算工作量;在形成的初始类簇的基础上,采用 AHC 算法的思想,逐个地将较小类簇合并成较大的类簇,直到满足条件为止;

(2) 提出了一个新的聚类有效性指标 DAS.DAS 运用两个类簇之间的最小距离来衡量簇间分离度,用一个类簇内部所有样本点的最小生成树的平均权重来衡量簇内紧密度.将类簇内紧密度和类簇间分离度的线性组合纳入聚类效果的整体质量评估.最终通过应用拐点检测的方法找到目标数据集的最佳类簇数.通过这种定义方法,DAS 指标的最大值能够被非常清晰地显示出来;

(3) 设计了一种新的确定数据集最佳类簇数的算法.通过将 *K*-means-AHC 算法和 DAS 指标相结合,设计并实现了一种确定数据集最佳类簇数的有效方法.合理性分析与实验验证得出,本文提出的聚类方法是准确和有效的.

1 相关工作

近年来,研究人员在传统聚类算法的基础上提出了许多新的或改进的聚类算法.Zhu 等人[17]提出了一种低秩稀疏子空间聚类算法,该算法通过从低维空间中学习到的关联矩阵而形成最终的聚类结果.Chen 等人[18]提出一种局部邻域搜索技术,并将其应用于改进的 DBSCAN 算法中,以此来减少数据集中各样本之间不必要的距离计算.在该模糊 C 均值(FCM)[19]算法的基础上,Qian 等人[20]通过应用知识杠杆原型转移(KL-PT)和知识杠杆原型匹配(KL-PM)这两种知识转移机制,提出了一种新的基于知识利用的迁移模糊 C 均值算法(KL-TFCM).Arora 等人[21]提出了一种基于噪声自适应非局部信息的改进型 FCM 算法(MFCM),该算法可用于使用局部和非局部空间信息对 MRI 脑图像进行有效分割.在基于图论的聚类算法中,谱聚类是一种经典的算法[22].在该领域中,Li 等人[23]构造了一个新的基于密度的矩阵,以作为谱聚类中的相似性矩阵.在此基础上,提出了基于密度的 *K* 均值以实现收敛全局优化,使其能够找到复杂数据的空间分布特征.Airel 等人[24]提出了一种新的基于图论的重叠聚类算法,该算法解决了以往一些重叠算法的局限性,同时具有可接受的计算复杂度.

Rodriguez 等人[25]在 2014 年提出了一种基于密度峰值的聚类算法 DPC(clustering by fast search and find of density peaks).该算法进行如下的假设:(1) 类簇中心点的密度大于周围邻居点的密度;(2) 类簇中心点与更高密度点之间的距离相对较大.基于这两个假设,该算法可以实现对任意形状类簇的聚类并降低异常点的干扰.针对 DPC 算法的样本局部密度定义和样本分配策略的缺陷,谢娟英等人[26]提出一种基于 *K* 近邻的快速密度峰值搜索并高效分配样本的聚类算法.该算法利用样本点的 *K* 近邻信息定义样本局部密度,搜索和发现样本的密度峰值,并以峰值点样本作为初始类簇中心.在提出的两种基于 *K* 近邻样本分配策略的基础上,得到数据集样本的分布模式.纪霞等人[27]针对 DPC 算法及其改进算法效率不高的缺陷,提出了一种相对邻域和剪枝策略优化的密度峰值聚类算法 RP-DPC.针对 DPC 算法无法自行选择类簇中心点的问题,马春来等人[28]提出了 DPC 改进算法.该算法采用类簇中心点自动选择策略,根据类簇中心权值的变化趋势来搜索“拐点”,并以“拐点”之前的一组点作为各个类簇中心.这一策略有效避免了通过决策图判定类簇中心的方法所带来的误差.徐晓等人[29]提出了基于网格筛选的密度峰值聚类算法,该算法通过稀疏网格筛选去除一部分密度稀疏的点,即不可能成为类簇中心的点,而只保留稠密网格单元中的点作为候选集进行类簇中心的选取.该算法保持了密度峰值聚类算法寻找类簇中心的准确性,同时降低了时间复杂度和内存需求,提高了运行效率.褚睿鸿等人[30]通过一个基于密度峰值的聚类集成模型来提升聚类结果的准确性、稳定性和鲁棒性.

在各种基于划分的聚类算法中,最为经典和常用的是 *K*-means 算法.研究人员在该算法的基础上提出了许

多改进算法.Capo 等人[31]提出了针对海量数据的 *K*-means 问题的有效近似方法,通过利用近似次优距离代替真实距离以有效降低距离计算的工作量.Zhang 等人[32]提出了一种基于密度的改进 *K*-means 算法,该算法通过合适的类簇数和最佳初始种子选取,提高了 *K*-means 的准确性和稳定性.Jiang 等人[33]从异常值检测的角度考虑 *K*-modes 聚类的初始化,并提出了两种不同的 *K*-modes 聚类初始化算法.通过使用两种离群值检测技术来计算每个对象的离群程度,进而避开了所选择的初始类簇中心是异常值的可能.Ismkhan[34]通过在每次迭代中移除一个类簇,再划分另一个类簇,然后再重新聚类来提高 *K*-means 的聚类质量和精度.*K*-means++[35]通过增量方式选取类簇中心,在该算法中,第 1 个初始类簇中心是随机选取的.一旦第 1 个中心被确定下来,增量选取的其余类簇中心将距已选取的中心尽可能地远.Grid-*K*-means 算法[16]将网格划分的思想应用于改进 *K*-means 算法,该算法使用动态变化的网格操作代替 *K*-means 算法中的样本点的操作.得益于网格操作的快速性,Grid-*K*-means 具备快速且准确处理大型数据集的能力.关于 *K*-means 算法的最新改进算法还有基于密度参数的改进 *K*-means 算法(DPI-*K*-means)[36]、基于 Density Canopy 的改进 *K*-means 算法(DC-*K*-means)[32]和基于可压缩近邻表示的聚类方法 Bit *k*-means[37]等.以上这些改进算法都对传统的 *K*-means 算法的缺陷做出了某方面的改进,但是都没有充分考虑 *K*-means 对非球状分布数据集的不适应性.本文在 *K*-means 算法的基础上,结合 AHC 算法的特点,提出了 *K*-means-AHC 混合算法.该算法具备快速处理多种形状分布的数据集的能力.

聚类有效性指标用于判断目标数据集应被划分为多少类簇更为合适.近年来,除了一些经典的指标,如 CH 指标、COP 指标、DB 指标、Dunn 指标以及 I 指标等,研究人员提出了许多新的聚类有效性指标以应对传统聚类有效性指标的不足.Zhou 等人[7]根据类簇中对象的几何分布,提出了一个基于类簇中心和最邻近距离的新的内部聚类有效性指标.Ahmed 等人[38]提出了一个基于 Jeffrey divergence 的聚类有效性指标,其中,Jeffrey divergence 用来衡量类簇之间的分离性.Yue 等人[39]基于两种常用的分区聚类算法——*C*-means 和模糊 *C*-means 以及它们的变体,提出了一种新的度量来表示类簇之间的分离性.根据这个度量,他们提出了一种新的聚类有效性指标来评估分区算法的聚类性能.Lin 等人[40]基于分散度和重叠度提出了一种新的有效性指标,其中,离散度估计数据集中类簇的总体数据密度,而重叠度则估算所有类簇之间的隔离程度.Yang 等人[41]利用基于标准欧氏距离和 ReliefF 算法的优化形态相似度距离来创建新的有效性指标,该指标可以平衡类簇内部和类簇之间的一致性问题.基于层次聚类算法,Zhou 等人[5]提出的 CSP 指标可以有效评估多种结构的数据集的聚类结果.黄晓辉等人[42]根据每个类簇中心不但能代表本簇的数据对象,且可尽可能地远离不属于本簇的数据对象的思想,设计了一个优化聚类算法的目标函数(*P*).Starczewski[43]提出的 STR 指标使用了拐点检测的方法来衡量数据集的划分准确性.Zhao 等人[44]指出,拐点检测通常是必需的,因为大多数指标随着类簇数量的增加显示出单调性.因此,具有明确的最小值或最大值的指数是优选的.鉴于拐点检测的必要性,本文提出的 DAS 指标将类簇内紧密度和类簇间分离度的线性组合纳入聚类效果的整体质量评估.最终,通过应用拐点检测的方法找到目标数据集的最佳类簇数.在文献[45]中,张远翔对 DAS 指标的相关背景、设计与实现进行了更加详尽的阐述.

2 *K*-means-AHC:基于 *K*-means 和 AHC 的混合算法

为了充分利用 *K*-means 算法和 AHC 算法的优点,本文将这两种算法的思想相结合,并提出了一种改进的混合算法,即 *K*-means-AHC.总体来讲,*K*-means-AHC 首先运用 *K*-means 算法处理数据的方式形成数据集的初始划分.在形成的初始划分的基础上,运用 AHC 算法处理数据的方式得到数据集的最优划分.*K*-means-AHC 算法的设计基于如下假设,即:在欧氏空间 R^m 中,数据集 $D=\{x_1,x_2,\ldots,x_n\}$ 具有 n 个样本点.每个样本点 $x_i=\{x_{i1},x_{i2},\ldots,x_{im}\}$ 具有 m 个属性.在数据集 D 中,样本点 x_i 和 x_j 之间的欧氏距离 $dist(x_i,x_j)$定义为

$$dist(x_i,x_j)=((x_{i1}-x_{j1})^2+\ldots+(x_{im}-x_{jm})^2)^{1/2}.$$

在给定的数据集 D 和类簇数 K 的情况下,首先,通过运用 *K*-means 算法处理数据的方法,将数据集 D 划分成 K_1 个类簇.与其他算法不同的是,这一步骤生成的初始类簇的数量是一个较大预估值($K_1>\sqrt{n}$).故初始类簇的数量 K_1 要远远大于算法最终生成的类簇的数量 K.其次,在生成的 K_1 个初始类簇的基础上,运用 AHC 算法处理数据的方法将 K_1 个初始类簇逐步合并,直到生成的类簇的数量等于 K 为止.*K*-means-AHC 算法的具体流程如图 1

所示.

输入:数据集 $D=\{x_1,x_2,\ldots,x_n\}$,类簇数 K 和初始类簇数 K_1;
输出:数据集 D 被分成 K 个类簇,即 $C=\{C_1,C_2,\ldots,C_K\}$.
(1) 从数据集 D 中随机挑选 K_1 个点;
(2) 将选取的 K_1 个点作为数据集 D 的初始划分 $C=\{C_1,C_2,\ldots,C_{K_1}\}$的类簇中心,即 $V=\{v_1,v_2,\ldots,v_{K_1}\}$;
(3) K-means 算法运行过程:
repeat
for i=1,2,…,n **do**
for j=1,2,…,K_1 **do**
计算 $dist(x_i,v_j)$; //$dist(x_i,v_j)$表示 D 中的每个点 x_i 与类簇中心 v_j 的距离.
将 D 中的每个点 x_i 分配至与之最近的类簇中心点 v_j 所在的类簇 C_j 中;
end for;
//计算误差平方和准则函数,判断函数值是否改变.
for j=1,2,…,K_1 **do** //计算每个类簇的平均值,将其作为新的中心点 v_j.
$v_j \leftarrow \left(\sum_{l=1}^{|C_j|} x_l\right) \Big/ |C_j|$; //$x_l$ 表示类簇 C_j 中的样本点的值,$|C_j|$表示类簇 C_j 中样本点的数量.
end for;
until 误差准则函数不变,得到划分好的 K_1 个类簇.
(4) AHC 算法运行过程:
repeat
计算得到的 K_1 个类簇中每对类簇 C_i,C_j 之间的距离 $dist(v_i,v_j)$.
//$dist(v_i,v_j)$表示类簇与类簇之间的最小距离.
将距离最近的两个类簇(设为 C_p 和 C_q)合并为一个新的类簇 C_r,即 $C_r \leftarrow C_p \cup C_q$,并更新$|C| \leftarrow |C|-1$;
until $|C| \leq K$.

Fig.1 Workflow of the K-means-AHC algorithm

图 1 K-means-AHC 算法的工作流程

在图 1 所示的算法中,第(1)步、第(2)步为形成初始类簇的工作过程,而第(3)步和第(4)步则是主要的聚类划分过程.其中,第(1)步和第(2)步从数据集 D 中随机挑选 K_1 个点作为数据集 D 的 K_1 个类簇的初始类簇中心点.当 K_1 个类簇的初始类簇中心点确定以后,第(3)步开始运行 K-means 算法.首先,将数据集 D 中的样本点分配到离其最近的中心点所在的类簇中,该过程计算时间大致为 $n\times K_1\times m$,其中,m 表示样本点的维数;然后计算误差准则函数,该步骤的大致计算时间为 $n\times m$;再计算每个类簇的均值,将其作为新的中心点,该步骤的计算时间大致为 $n\times m$.重复这 3 个步骤,直到误差准则函数值不变为止.假设需要重复 l 次,则整个第(3)步的计算时间大致为 $l\times n\times K_1\times m$.第(4)步是 AHC 算法的运行过程,计算每对类簇之间的距离.然后将距离最近的两个类簇合并成一个,重复执行,直到 C 中类簇的数量满足要求的 K 值为止.第(4)步的计算时间大致为$(K_1-|C|)\times n^2$.综合以上分析,整个 K-means-AHC 算法的计算时间大致为 $l\times n\times K_1\times m+(K_1-|C|)\times n^2$.通常情况下,$K_1$、$m$、$l$ 和$|C|$的值要远远小于 n 的值,故 K-means-AHC 算法的计算时间复杂度为 $O(n^2)$.

3 DAS:新的聚类有效性指标

本节首先给出 DAS 指标的定义,然后通过实例来解释该指标的含义及其设计的合理性.

3.1 DAS指标的定义

DAS 指标的定义是在欧氏空间下进行的,关于欧氏空间 R^m 的描述与第 2 节第 2 段相同.与此同时,假设数据集 D 被 K-mean-AHC 算法划分成 K 个类簇,即 $C=\{C_1,C_2,...,C_K\}$,且其中第 $i(i=1,2,...,K)$个类簇 C_i 包含$|C_i|$个样本点.

定义 1(簇内紧密度,简称 *cd*). 本文将由类簇 C_i 中所有样本点构成的最小生成树的平均权重定义为类簇 C_i 的簇内紧密度,记为 $cd(i)$:

$$cd(i)=W(C_i)/(|C_i|-1) \tag{1}$$

其中,$W(C_i)$是类簇 C_i 中所有样本点的最小生成树的权重.

定义 2(簇间分离度,简称 *sd*). 本文将第 i 个类簇中的样本点与其他不同类簇中的样本点之间最小距离的最

小值定义为该类簇的簇间分离度,记为 $sd(i)$:

$$sd(i)=\min_{1\leqslant j\leqslant|C|,j\neq i}\{\min\{dist(x_i,x_j)\mid x_i\in C_j,x_j\in C_j\}\} \tag{2}$$

定义 3(聚类综合度,简称 *csd*). 本文将第 i 个类簇的簇间分离度和簇内紧密度之差与簇内紧密度和簇间分离度之和的比值定义为聚类综合度,记为 $csd(i)$:

$$csd(i)=\frac{sd(i)-cd(i)}{sd(i)+cd(i)} \tag{3}$$

定义 4(平均聚类综合度,简称 *E*). 本文将所有类簇的聚类综合度的平均值定义为平均聚类综合度,记为 $E(K)$:

$$E(K)=\sum_{i=1}^{K}csd(i)\Big/K \tag{4}$$

定义 5(聚类有效性指标,简称 DAS). 假设数据集 D 被 K-mean-AHC 算法分别划分成 K 和 K+1 个类簇,即 $\{C_1,C_2,...,C_K\}$和$\{C_1,C_2,...,C_{K+1}\}$.其中,$\{C_1,C_2,...,C_K\}$是 D 的最优划分,即 D 的聚类划分数量在 K 时形成拐点.本文将 D 被划分成 K 个类簇的平均聚类综合度 $E(K)$与将 D 被划分成 K+1 个类簇的平均聚类综合度 $E(K+1)$的差定义为衡量聚类效果的聚类有效性指标,记为 $DAS(K)$:

$$DAS(K)=E(K)-E(K+1) \tag{5}$$

3.2 对DAS的解释

聚类有效性指标主要用于评价聚类算法对数据集的划分结果,而聚类有效性指标大多是通过簇内紧凑性和簇间分离性的某种组合来构造的.从簇内紧凑性来说,基于距离度量,我们希望一个类簇的样本之间的距离越小越好.但是考虑到类簇内部的最小距离和最大距离都不具有代表性,而平均距离对一些非凸结构的数据集也无法适用,因此,本文通过在单个类簇内部构造最小生成树,用最小生成树的平均权重作为这个类簇的簇内紧凑性的度量.另一方面,从簇间分离性来说,我们希望类簇与类簇之间的距离越大越好.因此,本文将不同类簇中的样本点之间的最小距离的最小值作为簇间分离性的度量.

以图 2 为例来解释 DAS 的含义及其相关概念.在图 2 中,根据欧氏距离的度量,数据集的所有样本点被划分成 4 个类簇(A,B,C,D).在每个类簇内部,样本点之间的连线代表着类簇内部的样本点的最小生成树.以类簇 A 为例,根据定义 1,类簇 A 的簇内紧密度为 $cd(A)=(e_1+e_2+e_3+e_4+e_5+e_6)/6$.在各个类簇之间,$h_1$、$h_2$ 和 h_3 分别是类簇 A 到类簇 C、类簇 D 和类簇 B 之间的最小单链距离.根据定义 2,类簇 A 的簇间分离度为 $sd(A)=\min\{h_1,h_2,h_3\}$.

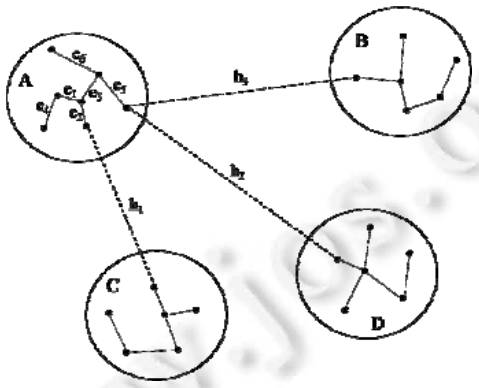


Fig.2 Distribution of clustering structure for the DAS

图 2 DAS 指标的类簇结构分布图

为了同时考虑簇内紧密度和簇间分离度,本文用 sd 与 cd 的差比上 sd 与 cd 的和作为聚类综合度 csd.在使 sd 和 cd 形成线性组合的同时,csd 也实现无量纲化.由于凝聚层次聚类(AHC)在每一次的迭代过程中总是将相邻最近的类簇合并到一起,因此对于层次聚类的结果来说,类簇间的距离不会小于类簇内最小生成树的平均值,即

$sd \geqslant cd$.所以,csd 函数的范围处于[0,1]之间,这样可以避免过度影响到 $E(K)$的极端情况.

设 C^*为数据集 D 中实际存在的类簇的数目.图 3 所示为由 4 个类簇(C^*=4)组成的数据集在给定不同 K 值的条件下各自的聚类结果.同种颜色的点代表它们在同一个类簇中.其中,图 3(a)中 K 的取值为 3,图 3(b)中 K 的取值为 4,图 3(c)中 K 的取值为 5.

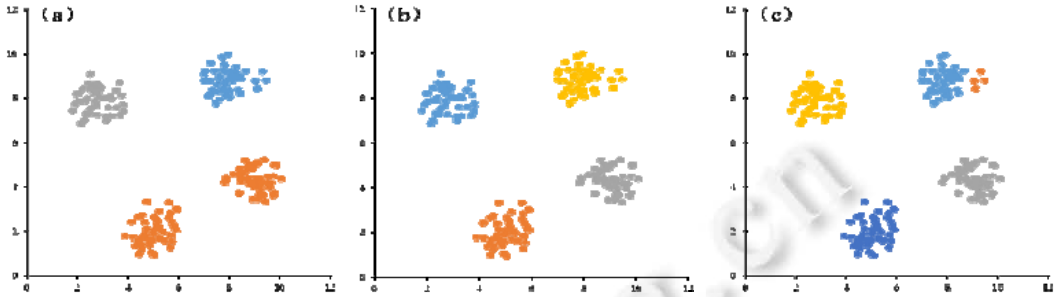


Fig.3 Clustering results of the tested datasets with different K values

图 3 不同 K 值下测试数据集的聚类结果

从图 3 可以看出:在 K 从 K=3 变化到 K=4 时,原本松散的一个类簇变成两个内部紧凑而且外部分离良好的类簇.但是类簇间的距离变化并不大,说明在这个过程中内部紧凑性发生了明显的变化.但是,由于本文使用类簇中所有样本的最小生成树的平均权重来定义簇内紧密度,所以造成 cd 的值在这个过程中没有产生较大的变化,进而 csd 的值也没有发生明显变化.然而在 K 从 K=4 变化到 K=5 的过程中,一个原本分离良好且内部紧凑的类簇被划分成两个更小的类簇.这两个类簇之间的距离变得非常小,而类簇内的紧密度变化不大,所以在这个过程中,类簇间分离度发生了很大的变化.根据定义 2,sd 的值明显变小,进而使得 csd 的值变小.因此,可以使用拐点检测的方法,即用 $E(K)$与 $E(K+1)$的差值作为本文的聚类有效性指标 DAS.在这个问题中,当 K 为 C^*时,差值最大,所以,当 $DAS(K)$取得最大值时的 K 值,即为最佳类簇数(K_{opt}):

$$K_{opt} = \{K \mid \max_{2 \leqslant K \leqslant \sqrt{n}} \{DAS(K)\}\} \tag{6}$$

以图 3 中的数据集作为例子,使用 K-means-AHC 算法对这个数据集进行聚类.当类簇的数量从 2 变化到 15 时,$E(K)$和 $DAS(K)$的值的变化分别如图 4(a)和图 4(b)所示.从图 4(a)可以看出:$E(K)$的值在 $K=C^*$变化到 $K=C^*+1$ 时,$E(K)$的值呈现显著的下降趋势,即在 C^*=4 时形成拐点.而从图 4(b)中可以看出,$DAS(K)$在 $K=C^*$时取到最大值.这个结果证明了本文所提出的 DAS 指标的有效性和拐点检测的合理性.

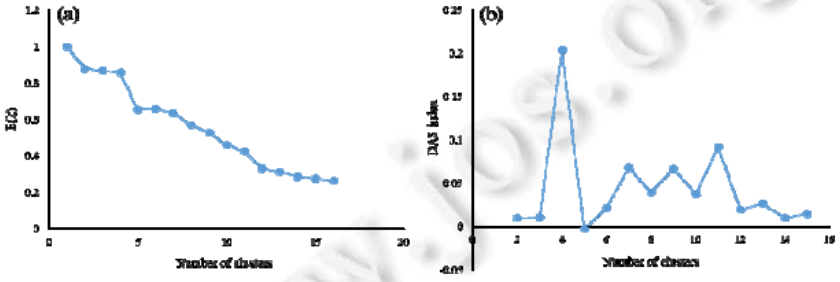


Fig.4 Change trends of $E(K)$ and $DAS(K)$ under different cluster numbers

图 4 不同类簇数下的 $E(K)$和 $DAS(K)$值的变化

3.3 最佳类簇数和最优划分的确定算法

基于 K-means-AHC 算法和 DAS 聚类有效性指标,本文设计了确定最佳类簇数的算法(如图 5 所示).通常,类簇数的搜索范围是[2,K_{max}].根据通行的经验规则 $K \in [2, \sqrt{n}]$,本文将 K_{max} 的上限设定为 $\sqrt{n}$ 的下取整.与此同

时,由于*K*-means-AHC算法在生成初始类簇时不必指定一个准确的*K*值,只需给出一个较大的初始值即可.即由*K*-means-AHC 算法生成的初始类簇的数量要比目标数据集 *D* 的真实划分数量要多.本文中,*K* 的初始值定为 $2\sqrt{n}$.相应地,*K*-means-AHC 算法生成的初始类簇的数量|*C*|也为 $2\sqrt{n}$.其中,*C* 为生成的目标数据集 *D* 的初始划分.具体而言,算法的第(1)步确定数据集 *D* 的初始类簇数量 $2\sqrt{n}$.第(2)步根据设定的初始类簇数量并利用*K*-means-AHC 算法的第(1)步~第(4)步形成数据集 *D* 的初始划分.在第(3)步,利用 *K*-means-AHC 算法的第(5)步逐步合并距离较近的相邻的类簇.与此同时,该步骤还在经验规则($2\leqslant K\leqslant \sqrt{n}$)规定的 *K* 值范围内计算平均聚类综合度 *E*(*K*).第(3)步的循环每进行 1 次,*C* 中类簇就减少 1 个.第(4)步利用 DAS 的定义计算不同 *K* 值下的指标值.第(5)步利用公式(6)来寻找最佳类簇数 K_{opt}.在 K_{opt} 确定以后,数据集 *D* 的最优划分也相应地被确定下来,即 $C=\{C_1,C_2,...,C_{Kopt}\}$.

输入:数据集 $D=\{x_1,x_2,\ldots,x_n\}$;
输出:最佳类簇数 K_{opt} 和数据集 *D* 的最优划分 $C=\{C_1,C_2,...,C_{Kopt}\}$.
(1) 根据数据集 *D* 的样本点的个数确定 *K* 的初始值,即 $K=|C|=2\sqrt{n}$;
(2) 利用 *K*-means-AHC 算法的第(1)步~第(4)步对数据集 *D* 进行划分,得到 *D* 的初始划分,即 $C=\{C_1,C_2,...,C_{2\sqrt{n}}\}$;
(3) **for** $K=\sqrt{n}$ down to 2 **do**
    输入类簇数 *K*,对第(2)步得到的|*C*|个类簇利用 *K*-means-AHC 算法的第(5)步继续聚类;
    根据公式(4),计算类簇数为 *K* 时的平均综合度 *E*(*K*);
    $|C|\leftarrow|C|-1$;
(4) **for** *K*=2 to $\sqrt{n}$ **do**
    根据第(3)步得到的 *E*(*K*),并利用公式(5)计算不同类簇数下的 *DAS*(*K*)指标值;
(5) 根据公式(6),得到最佳类簇数 K_{opt}.

Fig.5 Algorithm of determining the optimal clustering number and optimal clustering partitioning based on *K*-means-AHC and DAS

图 5 基于 *K*-means-AHC 和 DAS 的最佳类簇数和最优划分的确定算法

4 实验结果

表 1 列出了运行本文实验的计算机的配置环境.

Table 1 Experimental configuration details

表 1 实验配置具体环境

CPU	Inter(R)Core(TM) i5-8265U@1.60GHz
RAM	Samsung LPDDR3 2133MHz(8GB)
Hard Disk	WDC PC SN720 SDAPNTW-512G-1127
操作系统	Microsoft Windows 10 家庭版
编程环境	Java8/J2EE

如表 2 和表 3 所示,本文使用 20 个合成数据集(下载自 http://cs.joensuu.fi/sipu/datasets/)和 6 个真实数据集(下载自 https://archive.ics.uci.edu/ml/datasets.php)来验证 *K*-means-AHC 算法和 DAS 指标的性能.在这两个表中,“*K* 的初始值”是 *K*-means-AHC 算法中目标数据集初始划分中的类簇的数目.

与此同时,*K*-means-AHC 算法的性能(运行时间,准确性和均方差)将和 5 种已有算法进行比.这 5 种算法分别为经典的 *K*-means 算法、经典的凝聚型层次聚类算法(AHC)、基于密度参数的改进 *K*-means 算法(DPI-*K*-means)[36]、基于 Density Canopy 的改进 *K*-means 算法(DC-*K*-means)[32]和基于密度峰值的聚类算法(DPC)[25].DAS 指标的性能(寻找最佳聚类数)将与当前已有的 5 个经典的聚类有效性指标(CH^+[11]、COP^-[12]、DB^-[13]、$Dunn^+$[14]和 I^+[15])和 2 个最新提出的聚类有效性指标(CSP^+[5]和 STR^+[43])进行对比.在这些指标中,用“+”表示相应的指标取最大值时得到的最佳类簇数,用“–”表示相应的指标取最小值时得到的最佳类簇数.由于本文提出的 DAS 指标在指标函数取得最大值时获得最佳类簇数,故它被标记为 DAS^+.

Table 2 Descriptions of the 20 synthetic datasets

表 2 20 个人工合成数据集的描述

数据集	样本数	聚类数	K 的范围	K 的初始值
Circle2	400	2	$2\leqslant K\leqslant 20$	40
Circle3	2 000	3	$2\leqslant K\leqslant 45$	90
Circle4	1 000	4	$2\leqslant K\leqslant 32$	64
Circle5	1 500	5	$2\leqslant K\leqslant 39$	78
Parallel3	300	3	$2\leqslant K\leqslant 18$	36
Parallel4	400	4	$2\leqslant K\leqslant 20$	40
Parallel4-2	2 000	4	$2\leqslant K\leqslant 45$	90
Parallel5	600	5	$2\leqslant K\leqslant 25$	50
Parallel6	900	6	$2\leqslant K\leqslant 30$	60
Ring2	500	2	$2\leqslant K\leqslant 23$	46
Ring3	400	3	$2\leqslant K\leqslant 20$	40
Ring4	500	4	$2\leqslant K\leqslant 23$	46
Semicircle2	200	2	$2\leqslant K\leqslant 15$	30
Semicircle3	300	3	$2\leqslant K\leqslant 18$	36
Semicircle3-2	500	3	$2\leqslant K\leqslant 23$	46
Semicircle4	900	4	$2\leqslant K\leqslant 30$	60
Norm4	600	4	$2\leqslant K\leqslant 25$	50
Norm6	800	6	$2\leqslant K\leqslant 29$	58
Norm10	600	10	$2\leqslant K\leqslant 25$	50
Norm12	600	12	$2\leqslant K\leqslant 25$	50

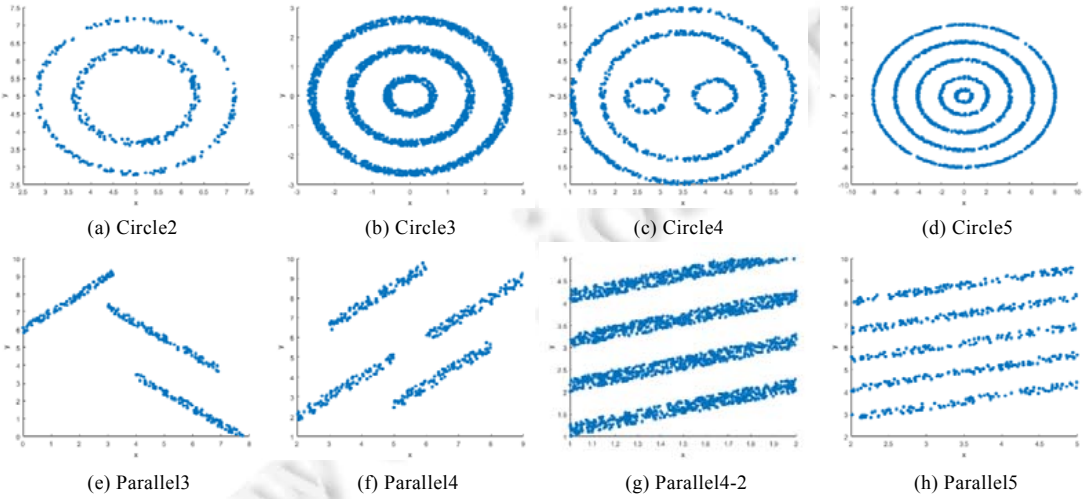
Table 3 Descriptions of the 6 real datasets

表 3 6 个真实数据集描述

数据集	样本数	维数	聚类数	K 的范围	K 的初始值
Column2	310	6	2	$2\leqslant K\leqslant 18$	36
Heart	270	13	2	$2\leqslant K\leqslant 17$	34
German	1 000	24	2	$2\leqslant K\leqslant 32$	64
Iris	150	4	3	$2\leqslant K\leqslant 13$	26
Haberman	306	3	2	$2\leqslant K\leqslant 18$	36
Tae	151	5	3	$2\leqslant K\leqslant 13$	26

4.1 *K*-means-AHC算法性能评测

图 6 给出了采用 *K*-means-AHC 算法对表 2 中 20 个合成数据集进行处理后的空间分布图.

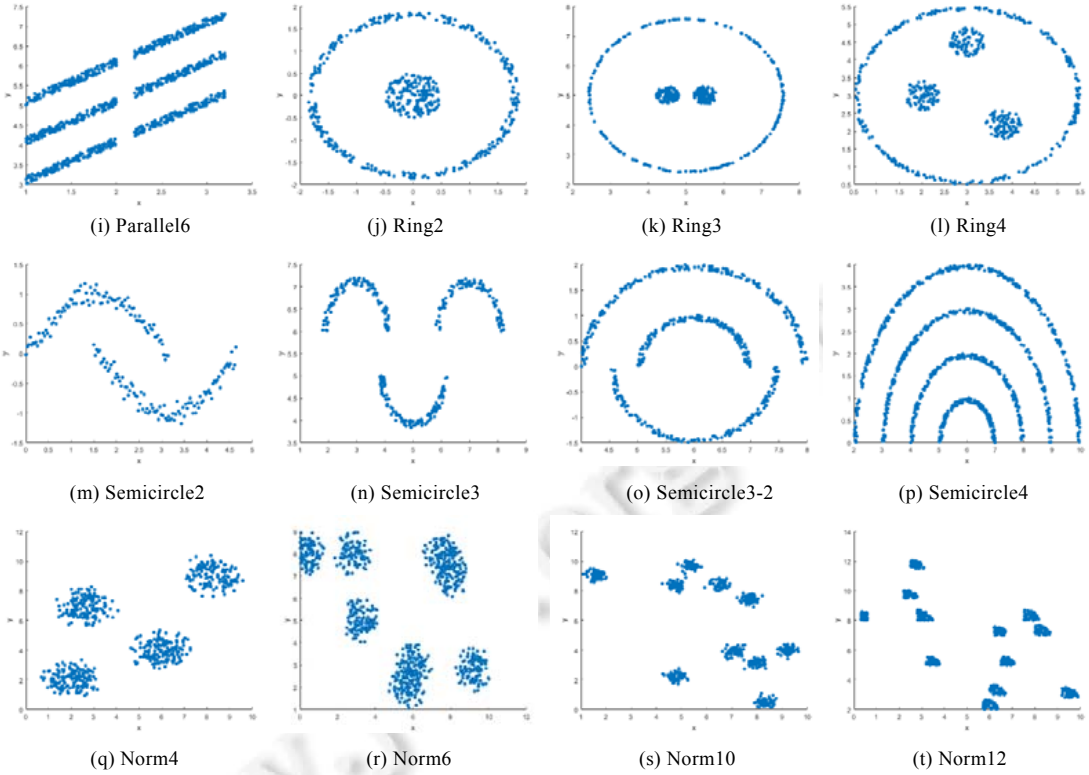


(a) Circle2 (b) Circle3 (c) Circle4 (d) Circle5

(e) Parallel3 (f) Parallel4 (g) Parallel4-2 (h) Parallel5

Fig.6 Spatial distributions of the 20 synthetic datasets

图 6 20 个合成数据集的结构分布图



(i) Parallel6 (j) Ring2 (k) Ring3 (l) Ring4

(m) Semicircle2 (n) Semicircle3 (o) Semicircle3-2 (p) Semicircle4

(q) Norm4 (r) Norm6 (s) Norm10 (t) Norm12

Fig.6 Spatial distributions of the 20 synthetic datasets (Continued)

图 6 20 个合成数据集的结构分布图(续)

从图 6 中可以看出,这些数据集包含了 5 种不同的结构形状.其中,Circle2、Circle3、Circle4、Circle5 是圆环状数据集;Parallel3、Parallel4、Parallel4-2、Parallel5、Parallel6 是直线型数据集;Ring2、Ring3、Ring4 是混合型数据集; Semicircle2、Semicircle3、Semicircle3-2、Semicircle4 是半圆环形数据集;Norm4、Norm6、Norm10、Norm12 是凸型数据集.图 7 所示为表 3 列出的 6 个真实数据集的三维空间分布图.这 6 个数据集均来自 UCI 机器学习真实数据集.由于除了 Hamberm 数据集外,其余 5 个真实数据集都是高维数据集,需要对它们进行降维处理,才能在低维空间中展示出来,本文选择目前主流的降维方法 T-SNE(https://lvdmaaten.github.io/tsne)来对高维数据进行降维处理.

K-means-AHC 算法的有效性评测将从运行时间、准确性和均方差这 3 个方面展开.与此同时,该算法的性能将和 5 种已有算法,即 *K*-mean、AHC、DPI-*K*-means、DC-*K*-means 和 DPC 的性能进行对比.对于 *K*-means 算法和 DPI-*K*-means 算法,由于在通常情况下无法预知目标数据集的类簇数,则这两种算法运行时间的统计方式为类簇数从 2 到 $\sqrt{n}$ 的总运行时间.对于 *K*-means-AHC 算法和 AHC 算法,其运行时间的统计方式为类簇数由 $\sqrt{n}$ 合并到 2 时的运行时间.对于 DC-*K*-means 和 DPC 算法,由于这两种算法不需要初始化类簇数,而是在运行结束时直接得到一个由算法计算出的类簇数,因此,对 DC-*K*-means 和 DPC 算法的对比是直接计算这两种算法的运行时长.

为了比较的准确性,我们对每种算法在特定数据集上运行 10 次,取 10 次运行结果的平均值和均方差.聚类结果的准确性通常可以用外部评价指标来衡量,常用的外部评价指标有 *F*-Measure、Entropy、Purity 等指标,本文使用 Purity 指标来评价聚类结果的准确性,其定义为

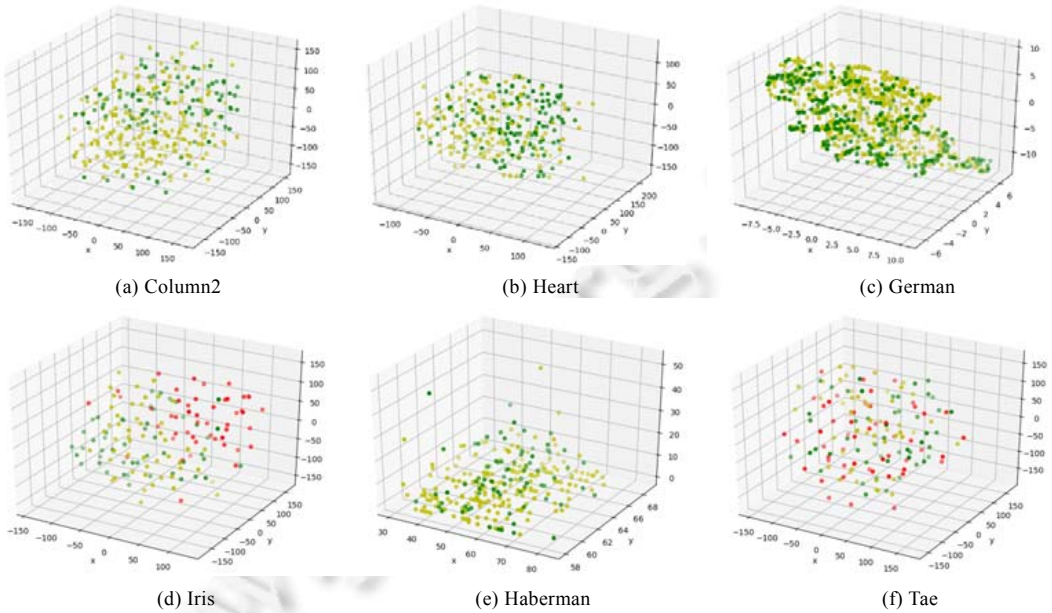
$$purity=\sum_{i=1}^{K}\frac{m_i}{m}\max\left(\frac{m_{ij}}{m_i}\right) \tag{7}$$

其中,m_i 为类簇 C_i 中所有成员的个数,m_{ij} 为类簇 C_i 中的成员属于类簇 C_j 的个数,K 是聚类的数目,m 是整个聚类划分所涉及到的成员个数.本文中,将 Purity 指标的值转化为百分数来进行比较.

均方差($\sigma(r)$)定义为

$$\sigma(r)=\sqrt{\sum_{i=1}^{N}(x_i-r)^2/N} \tag{8}$$

其中,N 代表算法运行的次数,x_i 表示第 i 次运行的结果,r 表示所有运行结果的平均值.



(a) Column2 (b) Heart (c) German

(d) Iris (e) Haberman (f) Tae

Fig.7 Spatial distributions of the 6 real datasets after dimensionality reduction

图 7 降维后的 6 个真实数据集的结构分布图

表 4 给出了 6 种算法对 20 个合成数据集的处理结果.从表 4 所列数据可以看出,*K*-means-AHC 算法能够完全正确地划分这些合成数据集,并在准确率、运行时间和稳定性等方面都有较好的表现.

在准确率方面,*K*-means-AHC 算法的平均准确率(均为 100%)是 6 种算法中最高的.AHC 算法的准确率与 *K*-means-AHC 接近.除了 Norm10 数据集以外,AHC 算法能够准确地得到其余 19 个数据集的聚类划分结果.DPC 算法可以得出 13 个数据集的准确聚类结果,对其余 7 个数据集也表现出较好的准确性.而其他 3 种算法均不能获得很好的聚类划分结果.

在准确率的均方差方面,由于 AHC 算法总是合并相邻最近的类簇,DPI-*K*-means 和 DC-*K*-means 算法在初始中心点的选择机制上都是寻找各自定义的密度最大的点,DPC 算法根据决策图确定中心点,因此这些算法在每次运行过程中样本点的划分都一致,使得每次运行结果的聚类准确率不变,即准确率的均方差都为 0.虽然 *K*-means-AHC 算法和 *K*-means 算法都是随机选取初始中心点,但是 *K*-means-AHC 算法的稳定性明显比传统的 *K*-means 算法要好,在大多数情况下运行结果都一致.具体而言,*K*-means-AHC 算法准确率的均方差除了 Normal10 数据集之外均为 0.

在运行时间方面,DPC 算法的平均耗时最小,但运行时间在稳定性上不如 *K*-means-AHC 算法.*K*-means-AHC 算法的平均耗时在小数据集上的表现稍大于 DPI-*K*-means 算法和 DC-*K*-means 算法,但在几个大数据集中,*K*-means-AHC 算法的平均耗时要稍小于 DPI-*K*-means 和 DC-*K*-means 算法.*K*-means-AHC 算法的平均耗时也要稍小于 *K*-means 算法的平均耗时,并远远小于 AHC 算法的平均耗时.在运行时间的均方差方面,*K*-means-AHC 算法在绝大多数情况下都表现出最好的稳定性,即耗时的均方差最小.

Table 4 Processing results of the 20 synthetic datasets by different algorithms

表 4 不同算法对 20 个合成数据集的处理结果

数据集	算法	*K*-means-AHC		AHC		*K*-means	
		Purity (%)	耗时(ms)	Purity (%)	耗时(ms)	Purity (%)	耗时(ms)
Circle2	均方差σ	0	7	0	92	1.51	47
	平均值	100	306	100	784	55.9	323
Circle3	均方差σ	0	33	0	5 574	0	105
	平均值	100	9 404	100	93 964	54.6	16 813
Circle4	均方差σ	0	14	0	663	0.82	99
	平均值	100	2 034	100	13 762	45.7	2 942
Circle5	均方差σ	0	17	0	2 580	10.83	231
	平均值	100	4 874	100	44 004	36.1	8 232
Parallel3	均方差σ	0	10	0	62	4	43
	平均值	100	172	100	500	86.9	175
Parallel4	均方差σ	0	10	0	80	16	47
	平均值	100	296	100	1 049	85.2	325
Parallel4-2	均方差σ	0	32	0	5 887	2.12	101
	平均值	100	9 043	100	99 493	92.8	16 509
Parallel5	均方差σ	0	10	0	182	12.82	67
	平均值	100	671	100	3 128	66.2	833
Parallel6	均方差σ	0	22	0	501	15	81
	平均值	100	1 580	100	9 898	86.2	2 234
Ring2	均方差σ	0	7	0	117	0	48
	平均值	100	471	100	1 906	66.6	574
Ring3	均方差σ	0	7	0	75	6.52	48
	平均值	100	296	100	1 036	61.8	324
Ring4	均方差σ	0	14	0	126	7.63	65
	平均值	100	485	100	1 875	64.7	542
Semicircle2	均方差σ	0	12	0	38	0.5	32
	平均值	100	86	100	189	82.6	88
Semicircle3	均方差σ	0	10	0	72	0	37
	平均值	100	176	100	494	100	177
Semicircle3-2	均方差σ	0	10	0	149	10	69
	平均值	100	482	100	1 957	58.4	567
Semicircle4	均方差σ	0	29	0	481	0	85
	平均值	100	1 631	100	9 952	50.3	2 208
Norm4	均方差σ	0	8	0	205	24.55	66
	平均值	100	687	100	3 080	95.2	807
Norm6	均方差σ	0	15	0	325	25.21	74
	平均值	100	1 241	100	6 757	90.2	1 648
Norm10	均方差σ	0.1	19	0	171	10.11	73
	平均值	100	689	89.8	3 030	84.9	818
Norm12	均方差σ	0	15	0	171	16.73	75
	平均值	100	687	100	3124	80.2	801

Table 4 Processing results of the 20 synthetic datasets by different algorithms (Continued 1)

表 4 不同算法对 20 个合成数据集的处理结果(续 1)

数据集	算法	DPI-*K*-means		DC-*K*-means		DPC	
		Purity (%)	耗时(ms)	Purity (%)	耗时(ms)	Purity (%)	耗时(ms)
Circle2	均方差σ	0	41	0	48	0	41
	平均值	55.8	227	53.7	207	80.7	105
Circle3	均方差σ	0	132	0	2 949	0	380
	平均值	54.5	10 195	54.3	9 830	100	2 033
Circle4	均方差σ	0	96	0	502	0	207
	平均值	46.0	1 921	44.6	1 787	50.9	421
Circle5	均方差σ	0	156	0	1 366	0	324
	平均值	47.2	5 165	34.4	5 179	100	887

Table 4 Processing results of the 20 synthetic datasets by different algorithms (Continued 2)

表 4 不同算法对 20 个合成数据集的处理结果(续 2)

数据集	算法	DPI-*K*-means		DC-*K*-means		DPC	
		Purity (%)	耗时(ms)	Purity (%)	耗时(ms)	Purity (%)	耗时(ms)
Parallel3	均方差σ	0	37	0	36	0	20
	平均值	88.5	132	88	131	100	50
Parallel4	均方差σ	0	46	0	45	0	28
	平均值	89.5	212	67.8	253	100	83
Parallel4-2	均方差σ	0	216	0	3 735	0	472
	平均值	91.1	10 167	64.1	13 307	74.3	2 180
Parallel5	均方差σ	0	70	0	52	0	50
	平均值	58.8	553	58	262	100	164
Parallel6	均方差σ	0	90	0	371	0	92
	平均值	87.3	1 443	49.8	1 662	69.4	353
Ring2	均方差σ	0	59	0	123	0	36
	平均值	66.6	390	78.8	372	100	125
Ring3	均方差σ	0	49	0	43	0	30
	平均值	63.8	219	77	213	77.2	87
Ring4	均方差σ	0	57	0	113	0	35
	平均值	66.6	372	66.6	380	100	126
Semicircle2	均方差σ	0	31	0	14	0	15
	平均值	82.8	64	81.7	54	100	28
Semicircle3	均方差σ	0	34	0	17	0	20
	平均值	100	127	62.6	111	100	53
Semicircle3-2	均方差σ	0	61	0	67	0	34
	平均值	51.6	385	67.1	430	93.4	124
Semicircle4	均方差σ	0	112	0	414	0	79
	平均值	50.3	1 493	34.2	1 563	100	331
Norm4	均方差σ	0	76	0	45	0	43
	平均值	100	560	75.6	354	100	159
Norm6	均方差σ	0	109	0	21	0	101
	平均值	88.0	1 083	76.1	416	100	274
Norm10	均方差σ	0	64	0	30	0	56
	平均值	100	533	20	455	99.8	174
Norm12	均方差σ	0	57	0	9	0	53
	平均值	83.4	539	16.7	398	100	167

与表 4 类似,表 5 是不同算法在运行 6 个真实数据集时的性能对比.

Table 5 Processing results of the 6 real datasets by different algorithms

表 5 不同算法对 6 个真实数据集的处理结果

数据集	算法	*K*-means-AHC		AHC		*K*-means	
		Purity (%)	耗时(ms)	Purity (%)	耗时(ms)	Purity (%)	耗时(ms)
Column2	均方差σ	0	12	0	72	0	35
	平均值	77.6	172	67.7	498	67.7	207
Heart	均方差σ	0.38	13	0	80	0.41	10
	平均值	65.8	139	55.7	427	59.2	161
German	均方差σ	0.28	49	0	738	0	117
	平均值	90.4	2 584	70.2	18 832	70.2	3 787
Iris	均方差σ	1.41	9	0	29	21.33	32
	平均值	88.1	45	72.3	102	84.6	49
Haberman	均方差σ	0.30	12	0	56	2.92	42
	平均值	88.6	164	86.5	442	86.3	173
Tae	均方差σ	2.31	9	0	30	4	20
	平均值	75.4	46	60.4	107	67.4	48

Table 5 Processing results of the 6 real datasets by different algorithms (Continued)

表 5 不同算法对 6 个真实数据集的处理结果(续)

数据集	算法	DPI-*K*-means		DC-*K*-means		DPC	
		Purity (%)	耗时(ms)	Purity (%)	耗时(ms)	Purity (%)	耗时(ms)
Column2	均方差σ	0	45	0	73	0	26
	平均值	67.7	175	67.7	162	87.7	134
Heart	均方差σ	0	38	0	195	0	23
	平均值	60.1	146	63.2	439	51.8	98
German	均方差σ	0	106	0	2 178	0	229
	平均值	70.2	3 967	70.2	31 169	93.7	692
Iris	均方差σ	0	28	0	25	0	9
	平均值	72.3	41	92	50	90.6	18
Haberman	均方差σ	0	43	0	71	0	21
	平均值	86	142	88.6	151	86	51
Tae	均方差σ	0	23	0	23	0	16
	平均值	66.2	45	78.9	106	80.1	22

从表中的数据可以发现,6 种算法都无法对数据集中的样本点进行完全正确的聚类划分.*K*-means-AHC 算法在准确性和运行时间上略低于 DPC 算法.得益于 AHC 算法的特性,*K*-means-AHC 算法在运行时间稳定性上要优于 DPC 算法.与其他算法相比,*K*-means-AHC 算法在运行真实数据集时同样保持较高的准确率及较低的运行时间开销.

4.2 DAS指标性能评测

为了比较的统一性,本节在对指标的性能进行对比时,先使用 *K*-means-AHC 算法对测试数据集进行统一划分,然后使用不同的聚类有效性指标对划分结果进行评价.表 6 列出了 DAS^+和其他 7 个指标对 20 个合成数据集划分效果的评测.其中,第 2 列(K_{opt})为各个数据集的真实划分类簇数,其他各列为各个指标得到的具体结果.在表 6 中,加粗的数字代表该指标可以得到对应数据集的真实最佳类簇数.括号里的数字代表各指标在得到其认为的最佳类簇数时的指标值,如表格中第 3 行最后一列的数字为 2(0.21811)可以解释为 *DAS*(2)=0.21811.

Table 6 Comparisons of evaluations on the clustering results of the 20 synthetic datasets by 8 indexes

表 6 不同指标对 20 个合成数据集聚类结果的评测效果对比

数据集	K_{opt}	各个指标得到的最佳类簇数			
		CH^+	COP^-	DB^-	$Dunn^+$
Circle2	2	6(110.721)	10(0.3754)	10(0.8010)	**2(0.16121)**
Circle3	3	31(1892.8)	31(0.3122)	31(2.9261)	2(0.14964)
Circle4	4	22(308.60)	22(0.4870)	19(1.9618)	2(0.11249)
Circle5	5	38(528.54)	38(0.4588)	38(1.3546)	2(0.11165)
Parallel3	3	18(1261.0)	18(0.2574)	18(0.5993)	2(0.06165)
Parallel4	4	15(529.68)	9(0.32543)	**4(0.69816)**	**4(0.29309)**
Parallel4-2	4	2(6028.50)	22(0.3316)	2(0.52095)	**4(0.30488)**
Parallel5	5	2(1143.50)	21(0.2748)	21(0.6100)	**5(0.26208)**
Parallel6	6	24(1965.3)	24(0.2583)	14(0.5764)	3(0.17743)
Ring2	2	11(519.35)	11(0.2459)	11(0.5458)	**2(0.32998)**
Ring3	3	17(835.15)	18(0.2072)	14(0.4590)	2(0.34110)
Ring4	4	12(361.63)	12(0.2769)	14(0.6410)	2(0.20477)
Semicircle2	2	**2(243.218)**	9(0.31655)	9(0.64037)	**2(0.21572)**
Semicircle3	3	14(724.31)	4(0.31008)	4(0.56286)	**3(0.43885)**
Semicircle3-2	3	9(368.639)	19(0.2674)	19(0.5375)	2(0.11467)
Semicircle4	4	27(290.50)	12(0.4843)	12(1.7437)	2(0.11394)
Norm4	4	5(3032.53)	5(0.20164)	5(0.47107)	**4(0.59827)**
Norm6	6	7(3489.40)	7(0.21160)	7(0.53865)	5(0.31767)
Norm10	10	13(9393.0)	13(0.2119)	3(0.46814)	4(0.42912)
Norm12	12	14(11762.0)	14(0.1740)	9(0.36240)	8(0.57024)

Table 6 Comparisons of evaluations on the clustering results of the 20 synthetic datasets by 8 indexes (Continued)

表 6 不同指标对 20 个合成数据集聚类结果的评测效果对比(续)

数据集	K_{opt}	各个指标得到的最佳类簇数			
		I^+	CSP^+	STR^+	DAS^+
Circle2	2	3(0.60361)	**2(0.81154)**	**2(0.72683)**	**2(0.21811)**
Circle3	3	**3(0.36740)**	2(0.90938)	2(0.51806)	**3(0.38327)**
Circle4	4	7(0.21295)	3(0.84612)	2(0.91930)	**4(0.26618)**
Circle5	5	**5(1.00553)**	4(0.88641)	2(0.23717)	**5(0.10756)**
Parallel3	3	2(1.68848)	2(0.90637)	6(1.27871)	**3(0.32581)**
Parallel4	4	2(2.14058)	2(0.90927)	12(0.3295)	**4(0.45678)**
Parallel4-2	4	2(0.63037)	2(0.90927)	6(1.39099)	**4(0.37306)**
Parallel5	5	3(0.80623)	**5(0.86936)**	14(2.0715)	**5(0.26653)**
Parallel6	6	3(0.34052)	3(0.90934)	2(0.50918)	**6(0.19673)**
Ring2	2	**2(0.97153)**	**2(0.92210)**	**2(1.36965)**	**2(0.25869)**
Ring3	3	5(0.56237)	2(0.93545)	2(0.74060)	**3(0.19157)**
Ring4	4	3(0.66239)	2(0.91570)	2(3.68903)	**4(0.16685)**
Semicircle2	2	**2(0.54996)**	**2(0.79341)**	**2(0.48896)**	**2(0.26031)**
Semicircle3	3	4(0.45531)	**3(0.89072)**	**3(0.66483)**	**3(0.28022)**
Semicircle3-2	3	**3(0.23147)**	2(0.82773)	12(4.0373)	**3(0.17311)**
Semicircle4	4	2(0.38169)	3(0.90490)	2(0.63387)	**4(0.13197)**
Norm4	4	3(2.43180)	2(0.90995)	**4(2.68012)**	**4(0.31976)**
Norm6	6	3(0.89551)	2(0.91557)	**6(1.09344)**	**6(0.20604)**
Norm10	10	2(1.42443)	2(0.93609)	9(4.76834)	**10(0.2663)**
Norm12	12	3(1.75474)	4(0.94771)	**12(7.8545)**	**12(0.3082)**

结合表 2 和图 6 可以看出:

- DAS^+指标可以得到所有的合成数据集的最佳类簇数和最优划分;
- CH^+指标仅可以得到数据集 Semicircle2 的最佳类簇数;
- COP^-指标不能得到任何数据集的最佳类簇数;
- DB^-指标只能得到 Parallel4 的最佳类簇数;
- $Dunn^+$指标的性能相对较好,它可以获得 Cricle2、Parallel4、Parallel4-2、Parallel5、Ring2、Semicircle2、Semicircle3 和 Norm4 这 8 个数据集的最佳类簇数;
- I^+指标可以得到 Circle3、Circle5、Ring2、Semicircle2 和 Semicircle3-2 这 5 个数据集的最佳类簇数;
- CSP^+指标可以获得 Circle2、Parallel5、Ring2、Semicircle2 和 Semicircle3 这 5 个数据集的最佳类簇数;
- STR^+指标可以获得 Circle2、Ring2、Semicircle2、Semicircle3、Norm4、Norm6 和 Norm12 这 7 个数据集的最佳类簇数.

由实验结果可知:其他 7 个指标对于非凸型数据集,如圆环状数据集、直线型数据集、半圆环形数据集和混合型数据集,都不能很好地给与处理.而本文提出的 DAS^+指标可以应对图 6 中所有类型的数据集.故本文的指标具有较为广泛的应用范围.

针对 6 个真实数据集,表 7 给出了不同指标之间的对比结果,各个表项的解释与表 6 相同.

Table 7 Comparisons of evaluationson the clustering results of the 6 real datasets by 8 indexes

表 7 不同指标对 6 个真实数据集聚类结果的评测效果对比

数据集	K_{opt}	各个指标得到的最佳类簇数			
		CH^+	COP^-	DB^-	$Dunn^+$
Column2	2	14(90.097)	**2(0.09266)**	3(0.72957)	**2(1.70658)**
Heart	2	3(39.3216)	3(0.20467)	**2(1.14957)**	**2(0.51145)**
German	2	4(271.778)	**2(0.19236)**	7(1.75141)	3(0.13252)
Iris	3	**3(221.799)**	5(0.28903)	**3(0.59034)**	2(0.33891)
Haberman	2	3(22.8471)	**2(0.20717)**	**2(1.78887)**	**2(0.18878)**
Tae	3	**3(57.1369)**	2(1.12424)	2(1.12424)	2(1.58212)

Table 7 Comparisons of evaluationson the clustering results of the 6 real datasets by 8 indexes (Continued)

表 7 不同指标对 6 个真实数据集聚类结果的评测效果对比(续)

数据集	K_{opt}	各个指标得到的最佳类簇数			
		I^+	CSP^+	STR^+	DAS^+
Column2	2	3(137.797)	**2(0.96067)**	12(0.1985)	**2(0.48024)**
Heart	2	7(29.03838	**2(0.19605)**	**2(0.04713)**	**2(0.38339)**
German	2	**2(29.1574)**	**2(0.73878)**	3(0.02475)	**2(0.30683)**
Iris	3	**3(1.28141)**	2(0.72154)	2(0.42977)	**3(0.15744)**
Haberman	2	**2(11.0046)**	**2(0.16917)**	17(0.0423)	**2(0.13945)**
Tae	3	2(12.3131)	**3(0.66232)**	12(0.1186)	**3(0.08332)**

由表 7 可以看出,DAS^+指标可以得到所有数据集的最佳类簇数.CH^+指标可以得到 Iris 和 Tae 两个数据集的最佳类簇数.COP^-指标可以得到 Column2、German 和 Haberman 这 3 个数据集的最佳类簇数.DB^-指标可以得到 Heart、Iris 和 Haberman 这 3 个数据集的最佳类簇数.$Dunn^+$指标可以得到 Column2、Heart 和 Haberman 这 3 个数据集的最佳类簇数.I^+指标可以得到 German、Iris 和 Haberman 这 3 个数据集的最佳类簇数.CSP^+指标的性能相对较好,可以得到除 Iris 数据集以外其他 5 个数据集的最佳类簇数.STR^+指标仅能得到 Heart 数据集的最佳类簇数.

综合对 20 个不同类型的合成数据集和 6 个真实数据集的实验结果来看,DAS^+指标比其他已有的 7 个指标更具稳定性,它能够得到不同结构数据集的最佳类簇数.多种类型的数据集的实验结果表明,本文提出的 DAS^+具有良好的稳定性和有效性.

5 总 结

层次聚类算法虽然可以对多种形状的数据集进行聚类,但其时间复杂度较高;而 *K*-means 算法虽然收敛快,但是对非凸型数据集的处理效果不好.本文将 *K*-means 算法和 AHC 算法处理数据集的思想相结合,提出了一种新的 *K*-means-AHC 混合聚类算法.新算法首先利用 *K*-means 算法的思想,快速形成数据集的初始类簇;在初始类簇的基础上,利用 AHC 算法的思想逐步合并初始类簇,直至形成数据集的最终划分.实验结果表明:*K*-means-AHC 算法在聚类精度、时间开销和稳定性等方面均有较大幅度的提升.在聚类结果评价方面,本文基于拐点的思想设计了一个新的 DAS 聚类有效性指标.针对不同类型数据集的实验结果表明,DAS 指标在稳定性上要优于当前已有的经典聚类有效性指标.但是,在数据集中存在大量噪声点时,本文算法在精度上有所降低.因此,未来的工作将集中在如何解决数据集噪声点的问题上.

References:

[1] Sun JG, Liu J, Zhao LY. Clustering algorithms research. Ruan Jian Xue Bao/Journal of Software, 2008,19(1):48−61 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/48.htm [doi: 10.3724/SP.J.1001.2008.00048]

[2] Mur A, Dormido R, Duro N, Dormido-Canto S, Vega J. Determination of the optimal number of clusters using a spectral clustering optimization. Expert Systems with Applications, 2016,65:304−314. [doi: 10.1016/j.eswa.2016.08.059]

[3] Xu DK, Tian YJ. A comprehensive survey of clustering algorithms. Annals of Data Science, 2015,2(2):165−193. [doi: 10.1007/s40745-015-0040-1]

[4] Jain AK. Data clustering: 50 years beyond *K*-means. Pattern Recognition Letters, 2010,31(8):651−666. [doi: 10.1016/j.patrec.2009.09.011]

[5] Zhou SB, Xu ZY, Liu F. Method for determining the optimal number of clusters based on agglomerative hierarchical clustering. IEEE Trans. on Neural Networks and Learning Systems, 2017,28(12):3007−3017. [doi: 10.1109/TNNLS.2016.2608001]

[6] Olson CF. Parallel algorithms for hierarchical clustering. Parallel Computing, 1995,21(8):1313−1325. [doi: 10.1016/0167-8191(95)00017-I]

[7] Zhou SB, ZY. A novel internal validity index based on the cluster center and the nearest neighbor cluster. Applied Soft Computing, 2018,71:78−88. [doi: 10.1016/j.asoc.2018.06.033]

[8] Rathore P, Ghafoori Z, C. Bezdek JC, Palaniswami M, Leckie C. Approximating Dunn's cluster validity indices for partitions of big data. IEEE Trans. on Cybernetics, 2019,49(5):1629−1641. [doi: 10.1109/TCYB.2018.2806886]

[9] Zhang YJ, Wang WN, Zhang XN, Li Y. A cluster validity index for fuzzy clustering. Information Sciences, 2008,178(4): 1205−1218. [doi: 10.1016/j.ins.2007.10.004]

[10] Yang Y, Jin F, Mohamed K. Survey of clustering validity evaluation. Application Research of Computers, 2008,25(6):1630−1632 (in Chinese with English abstract).

[11] Calinski T, Harabasz JA. A dendrite method for cluster analysis. Communications in Statistics, 1974,3(1):1−27. [doi: 10.1080/03610927408827101]

[12] Gurrutxag I, Albisua I, Arbelaitz O, Martin JI, Muguerza J, Perez JM, Perona I. SEP/COP: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index. Pattern Recognition, 2010,43(10):3364−3373. [doi: 10.1016/j.patcog.2010.04.021]

[13] Davies DL, Bouldin DW. A cluster separation measure. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1979, PAMI-1(2):224−227. [doi: 10.1109/TPAMI.1979.4766909]

[14] Dunn JC. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics, 1973,3(3):32−57. [doi: 10.1080/01969727308546046]

[15] Bandyopadhyay S, Maulik U. Nonparametric genetic clustering: comparison of validity indices. IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2001,31(1):120−125. [doi: 10.1109/5326.923275]

[16] Zhu EZ, Zhang YX, Wen P, Liu F. Fast and stable clustering analysis based on grid-mapping *K*-means algorithm and new clustering validity index. Neurocomputing, 2019,363:149−170. [doi: 10.1016/j.neucom.2019.07.048]

[17] Zhu XF, Zhang SC, Li YG, Zhang JL, Yang LF, Fang Y. Low-rank sparse subspace for spectral clustering. IEEE Trans. on Knowledge and Data Engineering, 2019,31(8):1532−1543. [doi: 10.1109/TKDE.2018.2858782]

[18] Chen YW, Tang SY, Bouguila N, Wang C, Du JX, Li HL. A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data. Pattern Recognition, 2018,83:375−387. [doi: 10.1016/j.patcog.2018.05.030]

[19] Bezdek JC, Ehrlich R, Full W. FCM: The fuzzy *C*-means clustering algorithm. Computers & Geosciences, 1984,10(2-3):191−203. [doi: 10.1016/0098-3004(84)90020-7]

[20] Qian PJ, Zhao KF, Jiang YZ, Su KH, Deng ZH, Wang ST, Jr RFM. Knowledge-leveraged transfer fuzzy *C*-means for texture image segmentation with self-adaptive cluster prototype matching. Knowledge-based Systems, 2017,130:33−50. [doi: 10.1016/j.knosys.2017.05.018]

[21] Arora N, Pandey R. Noise adaptive FCM algorithm for segmentation of MRI brain images using local and non-local spatial information. In: Mohamed BH, ed. Proc. of the 15th Int'l Conf. on Intelligent Systems Design and Applications. New York: IEEE, 2015. 610−617. [doi: 10.1109/ISDA.2015.7489187]

[22] Jia HJ, Ding SF, Xu XZ, Nie R. The latest research progress on spectral clustering. Neural Computing & Applications, 2014, 24(7-8):1477−1486. [doi: 10.1007/s00521-013-1439-2]

[23] Li Y, Liu XY. A modified spectral clustering algorithm based on density. In: Zu Q, Hu B, eds. Proc. of the 2nd Int'l Conf. on Human Centered Computing. Berlin: Springer-Verlag, 2016. 901−906. [doi: 10.1007/978-3-319-31854-7_97]

[24] Airel PS, José Fco. MT, Jesús A. CO, *et al*. A new overlapping clustering algorithm based on graph theory. In: Batyrshin I, González Mendoza M, eds. Proc. of the 11th Mexican Int'l Conf. on Artificial Intelligence. Berlin: Springer-Verlag, 2012. 61−72. [doi: 10.1007/978-3-642-37807-2_6]

[25] Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science, 2014,344(6191):1492−1496. [doi: 10.1126/science.1242072]

[26] Xie YJ, Gao HC, Xie WX. Fast peak density search clustering algorithm based on the optimization of *K*-nearest neighbor. SCIENTIA SINICA Informations, 2016,46(2):258−280 (in Chinese with English abstract). [doi: 10.1360/N112015-00135]

[27] Ji X, Yao S, Zhao P. Relative neighborhood and pruning strategy optimized density peaks clustering algorithm. Acta Automatica Sinica, 2020,46(3):1−14 (in Chinese with English abstract). [doi: 10.16383/j.aas.c170612]

[28] Ma CL, Shan H, Ma T. Improved density peaks based clustering algorithm with strategy choosing clustering center automatically. Computer Science, 2016,43(7):255−258 (in Chinese with English abstract). [doi: 10.11896/j.issn.1002-137X.2016.7.046]

[29] Xu X, Ding SF, Sun TF, Liao HM. Large-scale density peaks clustering algorithm based on grid screening. Journal of Computer Research and Development, 2018,55(11):2419−2429 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2018.20170227]

[30] Chu YH, Wang HJ, Yang Y, Li TR. Clustering ensemble based on density peaks. Acta Automatica Sinica, 2016,42(9):1401−1412 (in Chinese with English abstract). [doi: 10.16383/j.aas.2016.c150864]

[31] Capo M, Perez A, Lozano JA. An efficient approximation to the *K*-means clustering for massive data. Knowledge-based Systems, 2017,117:56−69. [doi: 10.1016/j.knosys.2016.06.031]

[32] Zhang G, Zhang CC, Zhang HY. Improved *K*-means algorithm based on density Canopy. Knowledge-based Systems, 2018,145:289−297. [doi: 10.1016/j.knosys.2018.01.031]

[33] Jiang F, Liu GZ, Du JW, Sui YF. Initialization of *K*-modes clustering using outlier detection techniques. Information Sciences, 2016,332:167−183. [doi: 10.1016/j.ins.2015.11.005]

[34] Ismkhan H. I-*k*-Means−+: An iterative clustering algorithm based on an enhanced version of the *k*-means. Pattern Recognition, 2018,79:402−413. [doi: 10.1016/j.patcog.2018.02.015]

[35] Yoder J, Priebe CE. Semi-supervised *K*-means++. Journal of Statistical Computation and Simulation, 2017,87(13):2597−2608. [doi: 10.1080/00949655.2017.1327588]

[36] Zhu EZ, Ma RH. An effective partitional clustering algorithm based on new clustering validity index. Applied Soft Computing, 2018,71:608−621. [doi: 10.1016/j.asoc.2018.07.026]

[37] Zhou GB, Wu JX, Zhou H. Clustering method based on nearest neighbours representation. Ruan Jian Xue Bao/Journal of Software, 2015,26(11):2847−2855 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4895.htm [doi: 10.13328/j.cnki.jos.004895]

[38] Ahmed BS, Rachid H, Sebti FF. Cluster validity index based on Jeffrey divergence. Pattern Analysis and Applications, 2017,20(1):21−31. [doi: 10.1007/s10044-015-0453-7]

[39] Yue SH, Wang JP, Wang J, Bao XJ. A new validity index for evaluating the clustering results by partitional clustering algorithms. Soft Computing, 2016,20(3):1127−1138. [doi: 10.1007/s00500-014-1577-1]

[40] Lin PL, Huang PW, Li CY. A validity index method for clusters with different degrees of dispersion and overlap. In: Proc. of the 8th Int'l Conf. on Advanced Computational Intelligence. New York: IEEE, 2016. 222−229. [doi: 10.1109/ICACI.2016.7449829]

[41] Yang SL, Li KS, Liang ZP, Li W, Xue Y. A novel cluster validity index for fuzzy *C*-means algorithm. Soft Computing, 2016,22(6):1921−1931. [doi: 10.1007/s00500-016-2453-y]

[42] Huang XH, Wang C, Xiong LY, Zeng H. A weighting *k*-means clustering approach by integrating intra-cluster and inter-cluster distances. Chinese Journal of Computers, 2019,42(12):2836−2848 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2019.02836]

[43] Starczewski A. A new validity index for crisp clusters. Pattern Analysis and Applications, 2017,20(3):687−700. [doi: 10.1007/s10044-015-0525-8]

[44] Zhao QP, FräNti P. WB-index: A sum-of-squares based index for cluster validity. Data and Knowledge Engineering, 2014,92:77−89. [doi: 10.1016/j.datak.2014.07.008]

[45] Zhang YX. Research on determining optimal number of clusters in cluster analysis [MS. Thesis]. Hefei: Anhui University, 2020 (in Chinese with English abstract).

附中文参考文献:

[1] 孙吉贵,刘杰,赵连宇.聚类算法研究.软件学报,2008,19(1):48−61. http://www.jos.org.cn/1000-9825/19/48.htm [doi: 10.3724/SP.J.1001.2008.00048]

[10] 杨燕,靳蕃,Mohamed K.聚类有效性评价综述.计算机应用研究,2008,25(6):1630−1632.

[26] 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.中国科学:信息科学,2016,46(2):258−280. [doi: 10.1360/N112015-00135]

[27] 纪霞,姚晟,赵鹏.相对邻域与剪枝策略优化的密度峰值聚类算法.自动化学报,2020,46(3):1−14. [doi: 10.16383/j.aas.c170612]

[28] 马春来,单洪,马涛.一种基于簇中心点自动选择策略的密度峰值聚类算法.计算机科学,2015,43(7):255−258. [doi: 10.11896/j.issn.1002-137X.2016.7.046]

[29] 徐晓,丁世飞,孙统风,廖红梅.基于网格筛选的大规模密度峰值聚类算法.计算机研究与发展,2018,55(11):2419−2429. [doi: 10.7544/issn1000-1239.2018.20170227]

[30] 褚睿鸿,王红军,杨燕,李天瑞.基于密度峰值的聚类集成.自动化学报,2016,42(9):1401−1412. [doi: 10.16383/j.aas.2016.c150864]

[37] 周国兵,吴建鑫,周嵩.一种基于近邻表示的聚类方法.软件学报,2015,26(11):2847−2855. http://www.jos.org.cn/1000-9825/4895.htm [doi: 10.13328/j.cnki.jos.004895]

[42] 黄晓辉,王成,熊李艳,曾辉.一种集成簇内和簇间距离的加权 *k*-means 聚类方法.计算机学报,2019,42(12):2836−2848. [doi: 10.11897/SP.J.1016.2019.02836]

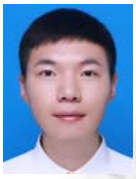
[45] 张远翔.聚类分析中的最佳聚类数确定方法研究[硕士学位论文].合肥:安徽大学,2020.

朱二周(1981－),男,博士,副教授,主要研究领域为数据挖掘,机器学习,程序安全.

高新(1993－),男,学士,主要研究领域为数据挖掘,机器学习.

孙悦(1995－),女,学士,主要研究领域为数据挖掘,机器学习.

马汝辉(1984－),男,博士,副教授,主要研究领域为云计算,大数据处理,虚拟化.



张远翔(1995－),男,学士,主要研究领域为数据挖掘,机器学习.

李学俊(1976－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为智能软件,工作流系统,边缘计算,服务计算.