计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.201100148

结合密度参数与中心替换的改进 *K*-means 算法及新聚类有效性指标研究

张亚迪　孙　悦　刘　锋　朱二周
安徽大学计算机科学与技术学院　合肥 230601
(zhangyd0831@gmail.com)

摘　要　聚类是一种经典的数据挖掘技术,它在模式识别、机器学习、人工智能等多个领域得到了广泛的应用。通过聚类分析,目标数据集的深层次结构可以被有效地发掘出来。作为一种常用的划分聚类算法,*K*-means 具有实现简单、能够处理大型数据等优点。然而,受收敛规则的影响,*K*-means 算法仍然存在着对初始类簇中心的选取非常敏感、不能很好地处理非凸型分布和有离群值的数据集等问题。文中提出了一种基于密度参数和中心替换的改进 *K*-means 算法 DC-Kmeans。该算法采用数据对象的密度参数来逐步确定初始类簇中心,使用中心替换方法更新偏离实际位置的初始中心,因而比传统聚的类算法更加精确。为了获得最佳聚类效果,文中同时提出了一个能够对聚类结果进行有效评价的新聚类有效性指标 SCVI 和一个能够快速获得目标数据集最佳类簇数的新算法 OCNS。实验结果表明,所提聚类方法对各种类型的数据集都是有效的。

关键词:聚类算法;聚类有效性指标;最佳类簇数;类簇中心;数据挖掘
中图法分类号　TP181

Study on Density Parameter and Center-Replacement Combined *K*-means and New Clustering Validity Index

ZHANG Ya-di, SUN Yue, LIU Feng and ZHU Er-zhou
School of Computer Science and Technology, Anhui University, Hefei 230601, China

Abstract　As a classical data mining technique, clustering is widely used in fields as pattern recognition, machine learning, artificial intelligence, and so on. By effective clustering analysis, the underlying structures of datasets can be identified. As a commonly used partitional clustering algorithm, *K*-means is simple of implementation and efficient on classifying large scale datasets. However, due to the influence of the convergence rule, the traditional *K*-means is still suffering problems as sensitive to the initial clustering centers, cannot properly process non-convex distributed datasets and datasets with outliers. This paper proposes the DC-Kmeans (density parameter and center replacement *K*-means), an improved *K*-means algorithm based on the density parameter and center replacement. Due to the gradually selecting of initial clustering centers and continuously update imprecision old centers, the DC-Kmeans is more accurate than the traditional *K*-means. Two novel methods are also proposed for optimally clustering: 1) a novel clustering validity index (CVI), SCVI (Sum of the inner-cluster compactness and the inter-cluster separateness based CVI), is proposed to evaluate the results of the DC-Kmeans; 2) a new algorithm, OCNS (optimal clustering number determination based on SCVI), is designed to determine the optimal clustering numbers for different datasets. Experimental results demonstrate that the proposed clustering method is effective for many kinds of datasets.

Keywords　Clustering algorithm, Clustering validity index, Optimal clustering number, Cluster center, Data mining

1　引言

作为一种经典的数据挖掘方法,聚类分析在模式识别、机器学习、人工智能等领域得到了广泛的应用。在没有先验信息的情况下,聚类分析能够找到目标数据集的内在结构。通过该方法,数据对象被分为若干个类簇,并且各个类簇中的数据对象尽可能相似,而不同类簇之间的数据对象尽可能不同。

当前,多种聚类算法被提出并应用于不同领域的数据分析问题。一般而言,聚类算法可分为层次聚类、划分聚类、基于密度的聚类、基于网格的聚类和基于模型的聚类这 5 类[1]。在各种聚类算法中,*K*-means 因其简单、有效而成为现实中应用最为广泛的聚类算法。与此同时,该算法对大规模数据集也具备一定的处理能力[2]。*K*-means 算法由初始化阶段和迭代阶段两个阶段组成。初始化阶段随机分配初始类簇中心。迭代阶段有两个步骤,第一步(也称为分配步骤)在初始类簇中心可用的情况下,将所有数据对象放到离各个中心点最近

到稿日期:2020-11-23　返修日期:2021-04-19
基金项目:安徽省自然科学基金(面上项目)(2008085MF188)
This work was supported by the Natural Science Foundation of Anhui Province(General Project)(2008085MF188).
通信作者:朱二周(ezzhu@ahu.edu.cn)

的类簇中;第二步(也称为更新步骤)更新所有类簇的中心点。

在经典的 K-means 算法中,其初始类簇中心采用随机的方法产生。众所周知,整个 K-means 的性能在很大程度上依赖于算法中初始类簇中心的选取[3]。不合理的初始类簇中心将在很大程度上降低 K-means 算法的聚类质量。与此同时,初始化不当也会增加 K-means 算法的运行时间。另一方面,K-means 算法的更新步骤对 K-means 算法的性能提升也很重要。一般情况下,该算法在达到可接受的聚类结果之前,需要对类簇中心进行多次调整。该步骤通过计算各个类簇中所有数据对象的距离平均值来更新相应的中心点。然而,新类簇中心的形成过程容易受到离群值的干扰。不当的离群值处理方法会导致新中心与实际中心存在较大的偏差,进而使得相应的类簇被离群值"扭曲"。

综上所述,传统 K-means 算法的性能受到两个问题的影响,即类簇中心的初始化和新类簇中心的形成方法。第一个问题由初始类簇中心的随机选择所引起,第二个问题则由离群值导致。针对这两个问题,本文提出了一种基于密度参数和中心点替换的改进 K-means 算法。该算法通过引入密度参数来逐步选择初始类簇中心;通过采用中心替换的方法来形成新的类簇中心。通过这两个方面的改进,使得 DC-Kmeans 算法比传统的 K-means 算法更加准确和高效。

聚类算法为不同数据集的划分提供了有效的途径。然而,由于聚类算法的无监督学习特性,很难确定一个给定数据集的某一划分是有效的。事实上,不同的聚类算法甚至不同参数的同一聚类算法对相同的数据集都有可能产生不同的划分结果[4]。聚类有效性指标(Clustering Validity Index,CVI)通常被用来评价聚类算法所产生的结果。通过 CVI 可以确定给定数据集的最佳类簇数(K_{opt},optimal clustering number)。CVI 一般被定义为以目标数据集(经聚类算法划分)和类簇数 K 为参数的函数。通过反复执行不同 K 值的 CVI 函数,可以得到最优(通常是最大值或最小值)指标值。因此,最优指标值对应的 K 值为目标数据集的 K_{opt}。

当前,已有许多 CVI 被提出并应用于聚类算法的结果评价。然而,还没有一个 CVI 可以最优地处理所有类型的数据集[5]。例如,常用的 CH[6] 指标对凸型数据集的处理非常有效。但是,它不能有效地处理非凸型和不平衡型的数据集。SIL[7] 指标可应用于许多类型的数据集,但在处理重叠数据集时,性能较差。COP[8] 指标对凸型数据集和部分重叠型数据集具有良好的性能,但它不能很好地解决非凸数据集的问题。SMV[9] 指标能够处理有噪声的数据集,但在处理非凸型和重叠型数据集时存在困难。为了对多种数据集的聚类结果进行有效和稳定的评价,本文在簇内紧凑度和簇间分离度的加权组合的基础上提出了一个新的聚类有效性指标 SCVI(Sum of the inner-cluster compactness and the inter-cluster separateness based CVI)。

总体而言,本文的主要贡献可以归纳为以下几个方面:

(1)提出了一种改进的 K-means 算法 DC-Kmeans。DC-Kmeans 算法在计算目标数据集各数据对象的密度参数的基础上逐步确定初始类簇中心。由于不随机选择初始类簇中心,DC-Kmeans 比传统的 K-means 算法更加稳定。与此同时,为了避免数据集离群值的影响,DC-Kmeans 使用中心替换方法对传统 K-means 算法生成的类簇中心进行替换。而且,与以往的针对 K-means 确定初始类簇中心阶段的改进与提升算法相比,本文提出的新算法中针对每个数据对象的密度参数是唯一的。该方法首先解决了传统聚类方法中初始类簇中心随机选择的缺陷;其次,迭代删除每个已经确定的初始类簇中心邻域内的数据对象,解决了初始类簇中心选择过于集中在高密度区域的问题。通过该方法,可以针对不同的数据集动态地寻找最大值和最小值,从而求取动态平均值,基于此,有针对性地获取密度参数较大且位置分布较为均匀的点作为初始聚类中心点,因此 DC-Kmeans 算法比传统算法具有更高的聚类精度。

(2)提出了一个新的聚类有效性指标 SCVI。SCVI 用于评价聚类算法对多种类型数据集的聚类划分结果的质量。不同于传统的基于簇内紧凑度和簇间分离度的线性组合的指标形成方式,SCVI 指标由二者的加权线性组合来定义。SCVI 达到最小指标值时对应的目标数据集的划分为数据集的最佳划分。

(3)基于 DC-Kmeans 算法和新定义的 SCVI 指标,设计了一种确定目标数据集 K_{opt} 的新算法 OCNS。该算法对凸型/非凸型数据集、平衡/不平衡数据集、圆弧数据集以及带有离群值的数据集均能进行有效的处理。

2 相关工作

本节主要从 K-means 算法的改进研究和聚类有效性研究这两个方面展开。

2.1 改进 *K*-means 算法的研究

当前,许多改进的 K-means 算法被提出以应对传统 K-means 存在的问题。Redmond 等[3] 使用 kd-tree 来估计分布于各个位置上的数据对象的密度,并将估算密度最大的数据对象的位置作为初始类簇中心。该方法在确定初始类簇中心时具有较高的效率。但是,它不能保证所有找到的中心都是对应类簇的实际中心。GK-means[10] 通过将网格结构和自定义的空间指标与 K-means 算法相结合,来确定初始类簇中心。在有效的网格划分情况下,该算法能够准确确定初始类簇中心。为了避免传统 K-means 算法因收敛规则而存在的局部最优和对数据对象离群点敏感的问题,Islam 等[11] 提出了遗传搜索算法 GenClust。在 K-means 快速爬坡周期的干预下,GenClust 能够快速获得高质量的聚类结果。但是,由于计算复杂度高,该算法不适合处理大规模数据集。Yoder 等[12] 提出了半监督的 K-means++算法。它是一种增量式寻找类簇中心的方法。除第一个初始类簇中心外,其余的类簇中心不是随机选取的。通过事先标记数据对象,其他中心被指定为离第一个中心尽量远的位置。该方法基于标记数据对象的合理选择,取得了较好的效果。Hussain 等在 K-means 聚类算法的基础上提出了统一的共聚框架 KCC[13]。KCC 改进了 K-means(和 K-means++)算法的初始化策略以及处理高维数据集的能力。然而,该算法的复杂性远远高于许多其他 K-means 改进算法。Fadaei 等[14] 提出了重新聚类的方法来降低 K-means 算法在动态网络中的处理成本。

该方法减少了迭代过程中检查节点的数量和总消耗时间，但需要在聚类精度和数据处理速度之间进行谨慎的权衡。Grid-Kmeans[15]算法将网格聚类中的网格划分思想与 K-means 算法相结合，采用动态变化的网格操作代替数据对象操作，同时网格步长和阈值也随着 K 值的变化而动态变化。结合网格划分思想对 K-means 算法进行有效预处理的前提下，该算法能够避免传统 K-means 算法效率低、聚类精度差、对噪声点敏感等缺点。

上述研究主要针对 K-means 算法的第一个问题，即类簇中心的初始化问题。然而，迭代阶段对 K-means 算法性能的影响还需进一步研究。由于离群值的存在，初始阶段指定的一些类簇中心的位置可能偏离实际的位置。与上述工作不同的是，本文提出的新算法 DC-Kmeans 在两个阶段都对 K-means 进行了改进，利用密度参数逐步选择初始类簇中心，采用中心替换方法更新类簇中心。

2.2 聚类有效性指标(CVI)的研究

CVI 是寻找目标数据集最佳类簇数(K_{opt})的关键。当前，许多 CVI 被提出并应用于确定不同类型数据集的 K_{opt} 和评价聚类算法结果的质量。总体而言，现有的基于数据集几何结构的 CVI 可以分为 3 类：基于数据集几何结构的 CVI、基于数据对象隶属度的 CVI 和基于数据集几何结构和隶属度的 CVI。

Dunn 于 1974 年提出的 DI[16]指标是第一类 CVI 的代表，由类簇之间的最小距离与类簇内的最大距离之比计算得出。由于对异常值和噪声数据对象敏感，除凸数据集外，DI 不能很好地处理空间分布不规则的数据集。通过标准化簇内紧凑度，Hubert 等提出了 CI[17]指标。由于只考虑簇内紧凑度，该指标简单且易于计算，然而，在处理数据集时并不稳定。Maulik 等提出的 I[18]指标由 3 个分量组成：$1/K$，E_1/E_K 和 $\max\limits_{i,j\in[1,K]} d(v_i,v_j)$。其中，$K$ 为类簇数目；$d(v_i,v_j)$ 为类簇中心 v_i 和 v_j 之间的欧氏距离；E_K 定义为 $\sum\limits_{k\in[1,K]}\sum\limits_{j\in[1,n]} u_{kj}d(v_j,v_k)$。3 个部分相互制约，从而构成 I 指标。但由于参数的不确定性，该指标性能不稳定。Calinski 等提出的 CH[6]指标由簇内紧凑度和簇间分离度的比值决定。通过比较研究，在许多情况下，CH 指标在评价聚类算法的结果时优于大多数 CVI。Davies 等提出的 DBI[19]指标由簇内紧凑度和簇间可分离性的比值定义。簇内紧凑度是通过簇内数据对象与类簇中心点之间的距离来评估的。此外，其通过目标数据集各类簇中心之间的距离来评估聚类间的可分离性。该指标适用于处理非凸型数据集，但它不能很好地处理的重叠数据集。SIL[7]指标能够处理具有不同空间分布的数据集，但是它在处理重叠数据集时存在困难。同时，SIL 指标的计算复杂度比其他许多指标都高。Gurrutxaga 等提出的 COP[8]指标也是基于簇内紧凑度和簇间分离度的比值。对于单个类簇，簇内紧凑度是通过所有数据对象到其中心的平均距离来计算的。对于目标数据集的所有类簇，用不同中心之间的最远距离来衡量簇间分离度。该指标适用于具有“簇内紧凑、簇间分离”特征的数据集。然而，目标数据集重叠越大，COP 指标的性能就越差。SMV[9]使用新的双中心度量来表示簇间分离度。该方法精度高，但适用范围小。

基于数据集隶属度的 CVI 主要用于评价模糊聚类的划分。Bezdek 提出的 PC(分区系数)[20]指标和 PE(分区熵)[21]指标是该类别的经典。PC 指标和 PE 指标都随着类簇数量的变化呈现单调递减的趋势。由于实现简单，这两个指标对模糊聚类是有效的。然而，这两种指标在面对大规模数据集时性能较差。基于紧凑性和重叠测度，Zalik 提出了用于模糊聚类的 COr[22]指标。COr 只考虑具有足够隶属度的数据对象来计算紧凑性和两个簇间隶属度较小的数据对象来计算重叠度。在评估类簇的大小或密度差异很大的分区时，该指标稳定且有效。Kim 等提出了用于模糊聚类的 OS[23]指标，该指标根据重叠度与可分度的比值来构造。Chen 等将提出的 P[24]指标作为模糊参数。由于模糊参数的不确定性，该指标不像其他模糊指标那样稳定。

Tang 等基于数据集的几何结构和隶属度，提出了 VT[25]指标，以克服聚类数目趋于数据对象个数时的单调下降趋势和类簇数与模糊性之间的强交互作用。该指标避免了模糊加权指标增大时验证指标的数值不稳定性。Pakhira 等提出了 PBM[26]指标，这个指标中的 3 个因素相互竞争，使指标达到最优值。PCAES[27]指标由两项组成，为了度量紧凑性，第一项计算模糊隶属度的平方比及其最小值的和；为了度量可分性，第二项计算两个最近类簇的中心点之间距离的相对值。由于该指标结构复杂，相比只考虑数据集的几何结构信息或隶属度的指标，需要进行更多的计算。

综上所述，各种 CVI 都有各自的优缺点。没有一种 CVI 可以对所有类型的数据集进行最优处理。现有的 CVI 对于“簇内紧凑、簇间分离”的数据集具有良好的评价性能。但大多数算法都不能很好地处理非凸型分布的数据集和重叠程度较大的数据集。而本文提出的 SCVI 指标试图优化处理更多类型的数据集。

3 DC-Kmeans：一种改进的 K-means 算法

本文提出的 DC-Kmeans 首先通过计算数据集各数据对象的密度参数来确定初始类簇中心，以此来避免因随机选择初始类簇中心而造成的聚类结果不稳定的问题。其次，对传统 K-means 算法生成的具有偏差的类簇中心进行替换，从而避免数据集离群值对聚类结果的影响。

3.1 基于密度参数的初始类簇中心的选取

DC-Kmeans 采用了基于密度参数的增量选择类簇中心的策略。本节及后续讨论基于如下假设：在欧氏空间 R^m 中，数据集 $D=\{x_1,x_2,\cdots,x_n\}$ 包含 n 个数据对象，每个对象 $x_i=\{x_{i1},x_{i2},\cdots,x_{im}\}$ 均有 m 个属性。与此同时，数据集 D 被某聚类算法划分为 K 个类簇 $C=\{C_1,C_2,\cdots,C_K\}$。其中 $|C_k|$ 为类簇 C_k 中所包含的数据对象的数量；类簇集合 C 中各个类簇对应的中心点为 $V=\{v_1,v_2,\cdots,v_K\}$。在数据集 D 中，任意两个数据对象 x_i 与 $x_j(i,j=1,2,\cdots,n)$ 之间的欧氏距离 $d(x_i,x_j)$ 定义为：

$$d(x_i,x_j)=\sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\cdots+(x_{im}-x_{jm})^2} \quad (1)$$

基于欧氏距离，所有数据对象之间的最大距离($LaDist$)

和最小距离(*SmDist*)分别定义如下:

$$LaDist=\sum_{i=1}^{n-1}\max_{1\leqslant i<j\leqslant n} d(x_i,x_j)^2 \tag{2}$$

$$SmDist=\sum_{i=1}^{n-1}\min_{1\leqslant i<j\leqslant n} d(x_i,x_j)^2 \tag{3}$$

上文虽然假设数据集 D 被分成 K 个类簇,但是对于不同的聚类算法,其生成的各个类簇中数据对象的个数可能是不同的。随着数据对象数目的变化,每个数据对象对之间的距离也会发生变化。为此,定义基于所有数据对象之间的最大距离和最小距离的动态平均距离(*DyAveDist*):

$$DyAveDist=(LaDist+SmDist)/(2*K) \tag{4}$$

其中,K 为数据集 D 被划分的类簇的数目。根据动态平均距离,密度参数可以进行如下定义。

定义 1(密度参数(density parameter,ρ)) 在数据集 D 中,以 $x_i(i=1,2,\cdots,n)$ 为中心、以 *DyAveDist* 为半径的圆形区域内的数据对象个数称为数据对象 x_i 的密度参数,即:

$$\rho(x_i,DyAveDist)=\sum_{i=1,j\neq i}^{n}u(DyAveDist-d(x_i,x_j)) \tag{5}$$

其中,$u(x)$ 为跳跃函数。当 $x\geqslant 0$ 时,$u(x)=1$;否则,$u(x)=0$。

在寻找初始类簇中心的过程中,多数基于密度的聚类算法在邻域半径[31]的选择上更加依赖于外部参数,也就是说,参数选择不当会大大影响算法的性能。针对这个问题,我们首先定义了基于所有数据对象之间的最大距离和最小距离的动态平均距离(*DyAveDist*),该距离随着每次迭代过程动态变化,能够及时获取不同划分阶段的邻域半径,以便更加高效稳定地确定密度参数,减小聚类结果受外部参数的影响。

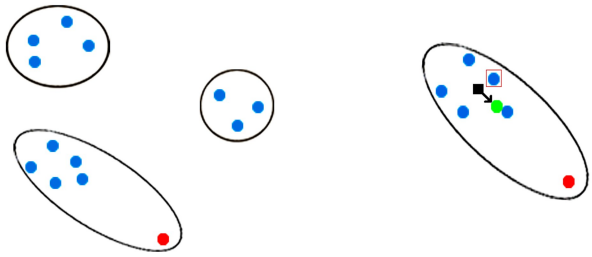
将密度参数最高的数据对象作为第一个初始类簇中心并从原始数据集 D 中删除。同时,将以第一个初始类簇中心为中心、以 *DyAveDist* 为半径的圆形区域内的数据对象从 D 中删除,第二个初始类簇中心是数据集 D 中剩余密度参数最高的数据对象。如此进行下去,直到找出指定的 K 个初始类簇中心为止。

3.2 类簇中心的替换

传统 K-means 算法的另一个缺陷为它对数据集的离群值非常敏感。实际上,传统 K-means 算法生成的某些初始类簇中心并不是目标数据集中的真实类簇中心(本文称这些点为"伪中心")。此外,由于离群值的影响,生成的类簇中心的位置会与实际的类簇中心的位置产生偏差[32]。而这一问题将严重降低传统 K-means 算法的精度。

K-medoids 聚类算法生成的中心始终是目标数据集的真实数据点。本文受 K-medoids 算法的启发,提出采用中心替换方法对传统 K-means 算法生成的伪中心进行更新。具体地,一旦 K-means 算法为一个类簇生成了一个伪中心,它就会被该类簇中最邻近的点所取代。同时,该邻近点应尽量远离该类簇的离群值。在聚类的过程中,依次更新伪类簇中心,直到指定了所有的真实类簇中心为止。

图 1 给出了包含离群点的类簇及其对应的伪中心的替换过程。图 1 中,蓝点代表正常的数据对象,红点代表离群点。图 1(a)给出了由 MATLAB 软件随机生成的由 3 个类簇组成的数据集。图 1(a)左下方的类簇包含一个由红色点表示的离群点。图 1(b)为该类簇的中心点的替换过程。在没有离群点干扰的情况下,传统的 K-means 算法以黑色矩形所表示的对象为类簇中心。但是,如果考虑类簇中存在的离群点,那么得到的类簇中心就会偏离"实际"的类簇中心。如图 1(b)所示,沿着箭头方向,类簇中心从黑色矩形的位置移动到了绿色圆点的位置。这种偏差会导致聚类算法的性能下降。实际上,随着类簇中心的偏离,大量不属于该类簇的数据对象将会被包含在下一次聚类算法的迭代中。在图 1(b)中,我们采用改进的方法将红色矩形框中的蓝点作为最终的类簇中心。该中心是数据集中实际存在的离黑色矩形最近、离红色离群点尽可能远的数据对象。



(a)Dataset contains the outlier (b)Fake center replacement

图 1 离群点及伪中心的替换(电子版为彩色)

Fig. 1 Example of outlier and fake center replacement

3.3 DC-Kmeans 的流程及时间分析

DC-Kmeans 算法的流程如算法 1 所示。DC-Kmeans 算法不仅能够稳定地找到初始类簇中心,而且具有处理离群值的能力。在算法 1 中,步骤(1)计算数据集 D 中所有数据对象之间的动态平均距离。步骤(2)计算所有数据对象的密度参数。步骤(3)根据密度参数,找到数据集 D 的 K 个初始类簇中心并放入集合 V 中。步骤(4)—(8)实现对数据集 D 的最终划分。具体来说,步骤(5)对每个类簇进行初始化;步骤(6)将数据对象放入相应的类簇中;步骤(7)采用中心替换方法更新类簇中心。

算法 1 DC-Kmeans 算法

输入:数据集 $D=\{x_1,x_2,\cdots,x_n\}$;类簇数 K

输出:数据集 $C=\{C_1,C_2,\cdots,C_K\}$

1. 计算数据集 D 中任意一对数据对象(x_i,x_j)之间的动态平均距离(DyAveDist);
2. for i=1,2,…,n do
   计算数据对象 x_i 的密度参数 $\rho(x_i,DyAveDist)$;
3. for k=1,2,…,K do //寻找 K 个初始类簇中心并将它们放入初始类簇中心集合 V 中.
   从数据集 D 中选取密度参数最高的数据对象 x,并且从数据集 D 中删除 x 邻域内的所有数据对象;
   将 x 设置为第 k 个初始类簇中心,设为 v_k;
   V←v_k; //将 vk 放入初始类簇中心的集合 V.
4. Repeat
5. Let $C_k=\emptyset(1\leqslant k\leqslant K)$; // 初始化各个类簇 C_k.
6. For i=1,2,…,n do //将 D 中的各个数据对象放入相应的类簇中.
   计算数据对象 x_i 和 V 中各个类簇中心之间的距离;
   依据最近原则将 x_i 放入相应的类簇中;
7. for k=1,2,…,K do //更新各个类簇的中心.

计算类簇 C_k 的中心 v_k 与该类簇中其余各个数据对象之间的距离；

寻找距离 v_k 最近的数据对象(v_k')，与此同时，(v_k')应与 C_k 中的离群点尽可能远；

if $v_k \neq v_k' v_k \leftarrow v_k'$ //v_k' is 被更新为类簇 C_k 的新中心.

8. Until $\sum_{i=1}^{K}\sum_{X\in C_i} d(v_i,x)^2$ converged. //直到标准函数 $\sum_{i=1}^{K}\sum_{X\in C_i} d(v_i,x)^2$ 收敛为一个常数，此时 v_k 是类簇 C_k 的新中心.

假设在数据集 $D=\{x_1,x_2,\cdots,x_n\}$ 中包含 n 个数据对象，每个数据对象 $x_i=(x_{i1},x_{i2},\cdots,x_{im})$ 是一个 m 维的向量。DC-Kmeans 算法将通过 p 次迭代将数据集 D 划分成 K 个类簇 $C=\{C_1,C_2,\cdots,C_K\}$。中心替换的数量是 t。DC-Kmeans 的时间复杂度如下：

(1)所有数据对象之间的动态平均距离的计算代价为 $T_1(n)=m\times n^2+n^2/2+n^2/2+1$；

(2)寻找 K 个最大密度对象并将其作为初始类簇中心的计算代价为 $T_2(n)=n^2+n+K\times n$；

(3)将剩余数据对象放入对应类簇的计算代价为 $T_3(n)=K\times m\times n$；

(4)采用中心替换策略得到新类簇中心的计算代价为 $T_4(n)=K\times m\times n+t\times K\times m\times n$。

(5)由于 DC-Kmeans 算法重复 p 次执行步骤(5)－(7)，因此算法的全部计算代价为：

$$T(n)=T_1(n)+T_2(n)+p\times(T_3(n)+T_4(n)+K\times|C_i|^2)$$

一般情况下，K,m,t,p 的值远小于 n 的值($K,m,t,p\ll n$)，可以视为常数，因此，DC-Kmeans 算法的计算复杂度可以表示为 $O(n^2+n^2+p\times K\times m\times n\times|C_i|^2)=O(n^2)$。

4 SCVI：新的聚类有效性指标

由于不同的聚类算法甚至同一聚类算法在不同的参数配置下都可能产生不同的聚类划分结果，为了评价聚类算法产生的结果，本节基于加权簇内紧凑度和簇间分离度的线性组合，提出了一个新的聚类有效性指标 SCVI。

4.1 SCVI 的定义

本节定义同样基于 3.1 节所作出的假设。

定义 2(类簇的簇内距离) 对于数据集 D 中给定的类簇 C_i，该类簇中所有数据对象之间的加权均方欧氏距离(简称类簇 C_i 的簇内距离，记为 T_i)定义为：

$$T_i=\frac{2K}{|C_i|(|C_i|-1)}\sum_{j,k=1}^{|C_i|}\frac{|C_i|}{n}(d(x_j,x_k))^2=\frac{2K}{n(|C_i|-1)}\sum_{j,k=1}^{|C_i|}(d(x_j,x_k))^2 \tag{6}$$

其中，$|C_k|$ 为类簇 C_i 中数据对象的个数；$|C_i|/n$ 是数据集 D 中类簇 C_i 的权重。簇内距离是构建簇内紧凑度度量的主要组成部分。该定义不涉及类簇的中心点，故它对重叠数据集的处理具有较好的效果。

定义 3(簇内紧凑度) 数据集 D 的簇内紧凑度(记为 T)定义为：

$$T=\sum_{i=1}^{K}T_i=\frac{2K}{n}\sum_{i=1}^{K}\frac{1}{(|C_i|-1)}\sum_{j,k=1}^{|C_i|}(d(x_j,x_k))^2 \tag{7}$$

许多现有的 CVI，如 CH[6]，COP[8] 和 DB[19]，都是通过计算数据集中所有类簇的簇内距离的平均值来定义簇内紧凑度的。由式(7)可以看出，本文簇内紧凑度 T 定义为数据集 D 中所有类簇的簇内距离加权之和。通过这样的定义，单个类簇对接下来定义的 SCVI 值的影响不会因求均值而被削弱。

定义 4(全局中心) 数据集 D 的全局中心(记为 V_0)定义为：

$$V_0=\frac{1}{n}(\sum_{i=1}^{n}x_{i1},\sum_{i=1}^{n}x_{i2},\cdots,\sum_{i=1}^{n}x_{im}) \tag{8}$$

定义 5(簇间距离) 数据集类簇中心集 V 中所有的类簇中心点 v_i 与全局中心 V_0 之间的加权均方欧氏距离，即簇间距离(记为 S_0)，定义为：

$$S_0=\frac{1}{K}\sum_{i=1}^{K}\frac{|C_i|}{n}(d(v_i,V_0))^2=\frac{1}{nK}\sum_{i=1}^{K}|C_i|(d(v_i,V_0))^2 \tag{9}$$

其中，$|C_i|$ 为簇 C_i 中数据对象的个数；$|C_i|/n$ 是数据集 D 中类簇 C_i 的权重。本文以簇间距离作为一个分量来衡量某一类簇对簇间分离度的影响。

定义 6(簇间分离度) 数据集 D 的簇间分离度(记为 S)定义为：

$$S=2\times K\times S_0=\frac{2}{n}\sum_{i=1}^{K}|C_i|(d(v_i,V_0))^2 \tag{10}$$

由式(7)可以看出，数据集 D 的簇内紧凑度 T 是由所有类簇的簇内距离之和而不是平均值定义的。因此，T 值普遍比 S 值大一些。为了平衡 T 和 S 对 SCVI 指标的影响程度，将 S_0 乘以 $2K$ 加以扩大。通过这种方法，T 和 S 这两部分对 SCVI 指标的影响程度大致均衡。

定义 7(SCVI 指标) 新的 SCVI 指标定义为：

$$SCVI(K)=T+S=\frac{2K}{n}\sum_{i=1}^{K}\frac{1}{(|C_i|-1)}\sum_{j,k=1}^{|C_i|}(d(x_j,x_k))^2+\frac{2}{n}\sum_{i=1}^{K}|C_i|(d(v_i,V_0))^2 \tag{11}$$

通过寻找 $SCVI(K)$ 函数的最小值，可以得到数据集 D 的最佳类簇数 K_{opt}：

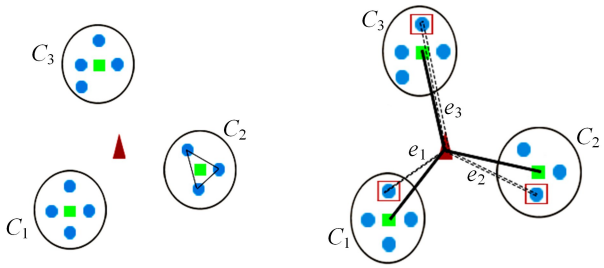
$$K_{opt}=\{K|\min_{2\leqslant K\leqslant\sqrt{n}}SCVI(K)\} \tag{12}$$

由式(11)可以看出，$SCVI(K)$ 的计算时间(记为 $T(n)$)由两部分组成：计算簇内紧凑度的时间(记为 $T_1(n)$)和计算簇间分离度的时间(记为 $T_2(n)$)。由式(6)和式(7)可得出 $T_1(n)$ 的计算时间为 $T_1(n)=K\times m\times(C_{max})^2$。其中，$K$ 为待划分目标数据集 D 的类簇数；m 为数据集 D 中数据对象的属性数；$|C_{max}|$ 是 D 中最大类簇的数据对象的个数。由于不同的类簇可能有不同的数据对象个数，因此用最大的类簇来估算簇内紧凑度的计算时间，即 $C_{max}\leftarrow\max\{|C_1|,|C_2|,\cdots,|C_K|\}$。根据式(8)和式(10)，$T_2(n)$ 的计算时间为 $T_2(n)=m\times n+K\times m$。其中，$m\times n$ 为全局中心 V_0 的计算时间。根据式(11)，可得出 $SCVI(K)$ 的计算时间为 $T(n)=T_1(n)+T_2(n)=K\times m\times(C_{max})^2+m\times n+K\times m=m\times(K\times(C_{max})^2+n+K)$。在一般情况下，$K$ 和 m 远小于 n，它们可以被当作常数。因此，$T(n)$ 的计算时间大致可以表示为 $\max\{O(n),O((C_{max})^2)\}$。

4.2 SCVI 的合理性分析

在 SCVI 指标的定义中，用 S(簇间分离度)来评价目标数据集 D 的不同类簇之间的差异。举例来说，图 2 给出了一个

包含 3 个类簇的数据集。该数据集是由 MATLAB 软件随机生成的。如图 2(a)所示，D 中的数据对象(蓝点)被分成 3 个类簇，绿色的方块是传统 K-means 算法指定的每个类簇对应的类簇中心(它们不是目标数据集的真实数据对象)，红框(见图 2(b))中的蓝点(它们是目标数据集的真实数据对象)是由 DC-Kmeans 算法更新的新的类簇中心。通过式(8)指定该数据集的全局中心(红色三角形)。从图 2(b)可以看出，全局中心和 3 个类簇中心是由直线连接的。原始的类簇中心(绿色方块)由实线连接，被替换的中心(红框中的蓝点)由虚线连接。每条虚线代表连接全局中心与每个类簇的类簇中心的距离。由式(9)计算簇间距离 S_0 ($S_0=(|C_1|e_1^2+|C_2|e_2^2+|C_3|e_3^2)/n$)，此时，簇间分离度 S 的计算复杂度可降为 $K\times m$。



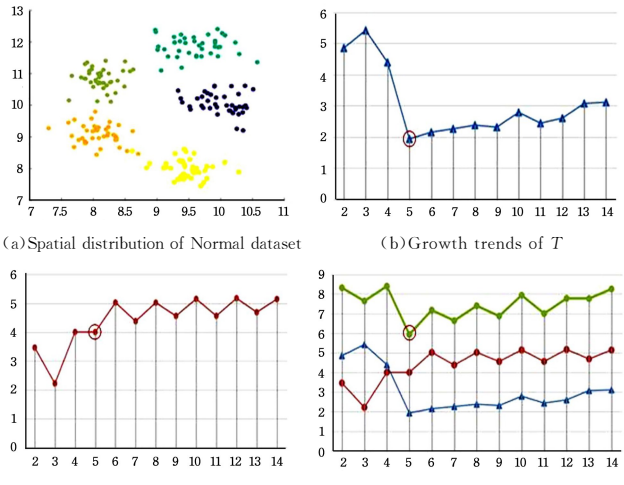
(a) Inter-cluster separateness (b) Inner-cluster compactness

图 2 SCVI 中簇内紧凑度和簇间分离度的度量(电子版为彩色)

Fig. 2 Calculating the inter-cluster separability and the inner-cluster compactness of SCVI

利用簇内紧凑度 T 来评价单个类簇内部各数据对象之间的相似程度。根据式(6)计算每个类簇的簇内距离。在图 2(b)中，类簇 C_2 的簇内距离为 $T_2=2K(e_1^2+e_2^2+e_3^2)/n(|C_2|-1)$。该方法避免了不同类簇的类簇中心之间的直接连接。因此，SCVI 指标能够处理非凸型和重叠型数据集。

图 3(a)为经过 DC-Kmeans 聚类算法处理的模拟数据集 Normal 的二维空间分布。从图中可以看出该数据集由 5 个类簇组成。图 3(b)、图 3(c)和图 3(d)分别给出了簇内紧凑度(T)、簇间分离度(S)和 SCVI 指标的增长趋势。在 3 个子图中，横坐标和纵坐标分别表示类簇数 K 和对应的指标值。



(a) Spatial distribution of Normal dataset (b) Growth trends of T

(c) Growth trends of S (d) Growth trends of SCVI

图 3 数据集 Normal 的空间分布和 SCVI 指标值的增长趋势(电子版为彩色)

Fig. 3 Spatial distribution of the Normal dataset and the growth trends of SCVI

由图 3(b)可以看出，除 $K=2$ 外，当 K 达到 5 时，T 值一般都是递减的。之后，随着 K 的增长，T 值略有增加。值得注意的是，从 $K=4$ 到 $K=5$，T 值急剧下降，之后平稳增长，所以可以将 $K=5$ 看成拐点，因此，式(7)对于计算每个类簇的内簇紧性是可行的。从图 3(c)可以看出，当 $K<5$ 时，S 值波动较大，而当 $K>5$ 时，S 波动逐渐趋于平缓。也就是说，S 值在 $K\geqslant 5$ 时达到稳定状态，S 值随着 K 的增长有轻微的变化。因此，计算不同类簇间的簇间分离度用式(10)是合理的。图 3(d)中的绿色线显示了 SCVI 指标值的增长趋势。其中，蓝色和红色曲线分别为图 3(b)和图 3(c)中 T 和 S 的增长趋势。由图 3(d)可以看出，当 K 达到 5 时，T 达到最小值，S 进入稳定状态，在拐点处得到 SCVI 指标的最小值。此时，$K=5$ 为 Normal 数据集的最佳类簇数。

5 最佳类簇数确定算法:OCNS

聚类有效性指标的性能一般是通过其是否可以准确找出目标数据集的类簇数目来判断的，因此本文将 DC-Kmeans 聚类算法与 SCVI 指标相结合，提出了一种确定目标数据集最佳类簇数的算法 OCNS，以此来验证 SCVI 指标的优势。第 6 节的实验中，SCVI 指标的实验结果侧面反映出 OCNS 算法的性能，该算法能够确定圆弧型、凸型、非凸型、重叠型和不平衡型等不同类型数据集的最佳类簇数。算法 2 给出了该算法的主要步骤。在该算法中，对于不同的 K 值，步骤 1－3 将数据集 D 划分为 K 个不同的类簇。根据经验法则，K 值在[2，$\sqrt{n}$]区间内。步骤 4 计算每个类簇的簇内紧凑度。步骤 5 计算所有类簇的簇间的分离度。步骤 6 使用式(11)和式(12)来获得数据集 D 的 K_{opt}。

算法 2 OCNS 算法

输入：数据集 D={x_1，x_2，…，x_n}；

输出：数据集 D 的最佳类簇数 K_{opt} 和 SCVI(K)的值.

1. 确定类簇数 K 的搜索范围[K_{min}，K_{max}]；

2. For K=K_{min} to K_{max} do //根据经验规则，K 的值处于区间[2，$\sqrt{n}$].

3. 利用 DC-Kmeans 算法对数据集 D 进行聚类处理；

4. 利用式(7)来每个类簇的簇内紧凑度；

5. 利用式(10)来计算各个不同类簇之间的簇间分离度；

6. 令 min←SCVI(2)；// 利用式(11)来计算 SCVI(K)的值；

For K=3，4，…，$\sqrt{n}$ do // 利用式(12) 来获得数据集 D 的最佳类簇数 K_{opt}；

if min>SCVI(K)

then min←SCVI(K)；

K_{opt}←K；

else Keep min unchanged.

6 实验结果

本节分别给出了 DC-Kmeans 算法和 SCVI 指标的实验测试结果。如前文所述，SCVI 指标的实验结果就是 OCNS 算法准确性的表现；同样，其他 8 个聚类有效性指标的实验结果也反映了 DC-Kmeans 聚类算法在相应指标上的实验结果。

本文实验的代码采用 Java 语言编写，具体的软硬件配置环境如表 1 所列。在实验中，首先使用经验规则 $K\leqslant\sqrt{n}$ 来得到不同测试数据集 K 的范围。其次，针对不同的数据集生成的聚类结果进行评价，并将 SCVI 指标的性能与现有的 8 个

聚类有效性指标（CH^{+}[6]，I^{+}[18]，STR^{+}[28]，DBI^{-}[19]，COP^{-}[8]，SMV^{-}[9]，$BCVI^{-}$[15]，$DCVI^{-}$[29]）的性能进行比较。由于在指标值取最大时得到最佳类簇数，因此将 CH，I 和 STR 分别标记为 CH^{+}，I^{+} 和 STR^{+}。相反地，DBI，COP，SMV，BCVI，DCVI 和 SCVI 被标记为 DBI^{-}，COP^{-}，SMV^{-}，$BCVI^{-}$，$DCVI^{-}$ 和 $SCVI^{-}$。

表 1 实验的软硬件配置环境

Table 1 Software and hardware configuration environment of experiment

CPU	Inter(R) Core (TM)i7-8565U CPU @ 1.80 GHz
RAM	LPDDR3 2133 MHz(8 GB)
Hard disk	NVMe PCIe 高速固态硬盘
OS	Microsoft Windows 10 Enterprise (64 bit)

6.1 测试数据集描述

如表 2 所列，实验中用于测试的数据集由 8 个模拟数据集[1)]（表 2 中的前 8 个数据集）和 8 个 UCI 真实机器学习数据集[2)]（表 2 中的后 8 个数据集）组成。

表 2 8 个模拟数据集（前 8 个）和 8 个 UCI 真实机器学习数据集（后 8 个）的描述

Table 2 Description of 8 (the first 8) simulated datasets and 8 (the last 8) UCI real machine learning datasets

DataSets	Points Number	Cluster Number	Dimension	K_{opt} Range
Normal	200	5	2	[2,14]
D900	900	9	2	[2,30]
R15	600	15	2	[2,24]
N7	28000	7	2	[2,167]
K3	102000	3	2	[2,319]
Curve	180	3	2	[2,13]
Pathbased	300	3	2	[2,17]
Semicircle	300	3	2	[2,17]
Iris	150	3	4	[2,12]
Seeds	210	3	7	[2,14]
Haberman	306	2	3	[2,17]
Column	310	3	6	[2,17]
Hayes-Roth	132	3	5	[2,11]
Ionosphere	351	2	34	[2,18]
PageBlocks	5473	5	10	[2,73]
Magic	19020	2	10	[2,137]

图 4 给出了表 1 中 8 个模拟数据集的空间分布情况。如图 4(a)所示，Normal 数据集包含 5 个类簇，其中每个类簇都呈球形分布，并且该数据集中存在一些离群点。如图4(b)—图 4(e)所示，数据集 D900，R15，N7 和 K3 分别包含 9，15，7 和 3 个类簇。N7 和 K4 是规模较大的数据集。这 4 个数据集的球形分布具有“簇内紧凑性”和“簇间分离性”的特点，同时，在这些数据集中也存在一些离群点。图 4(f)—图4(h)所示的 Curve，Pathbased 和 Semicircle 为非球形分布的数据集。在图 4(f)中，Curve 数据集的 3 个类簇为 3 个同心圆弧。图 4(g)显示了 Pathbased 数据集的空间分布。在这个数据集中，内部的两个球状类簇被一个半圆型类簇包围。如图 4(h)所示，Semicircle 数据集呈弧形分布。

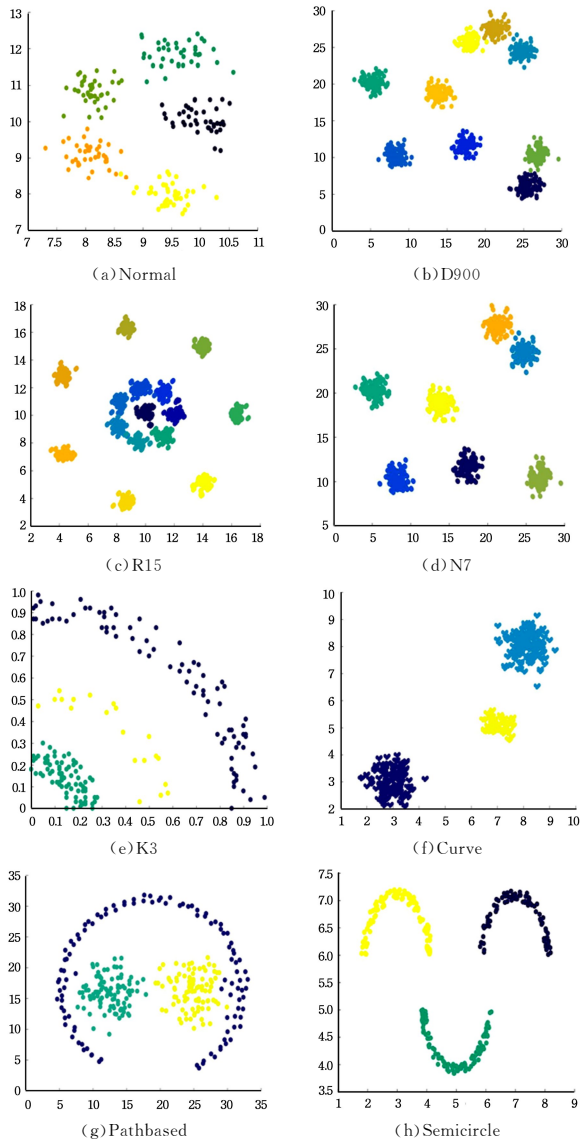


图 4 8 个模拟数据集的空间分布

Fig. 4 Spatial distributions of the eight simulated datasets

图 5 给出了 8 个 UCI 真实数据集的空间分布。如表 1 所列，大多数数据集都是高维的。在低维度空间中显示它们之前需要先对其降维。目前，降维工具可分为线性降维工具和非线性降维工具两大类。本文采用了广泛应用的非线性降维工具 T-SNE[30] 对所有高维数据集进行处理。如图 5(a)所示，Iris 数据集的 150 个数据对象被划分为 3 个类簇，每个类簇包含 50 个数据对象。在 3 个类簇中，两个类簇略有重叠。如图 5(b)、图 5(d)和图 5(e)所示，Seeds，Column 和 Hayes-Roth 数据集的各自类簇之间有很多重叠的数据对象。Haberman 数据集由两个类簇组成，如图 5(c)所示，它的两个类簇几乎完全重叠。如图 5(f)所示，Ionosphere 由两个类簇组成且线性可分。如图 5(g)和图 5(h)所示，PageBlocks 和 Magic 分别包含 5 个和 2 个类簇，它们比其他数据集更加复杂并且包含更多的数据对象；同时，这两个数据集的不同类簇之间也有大部分重叠区域。

1) http://cs.joensuu.fi/sipu/datasets/

2) https://archive.ics.uci.edu/ml/datasets.php

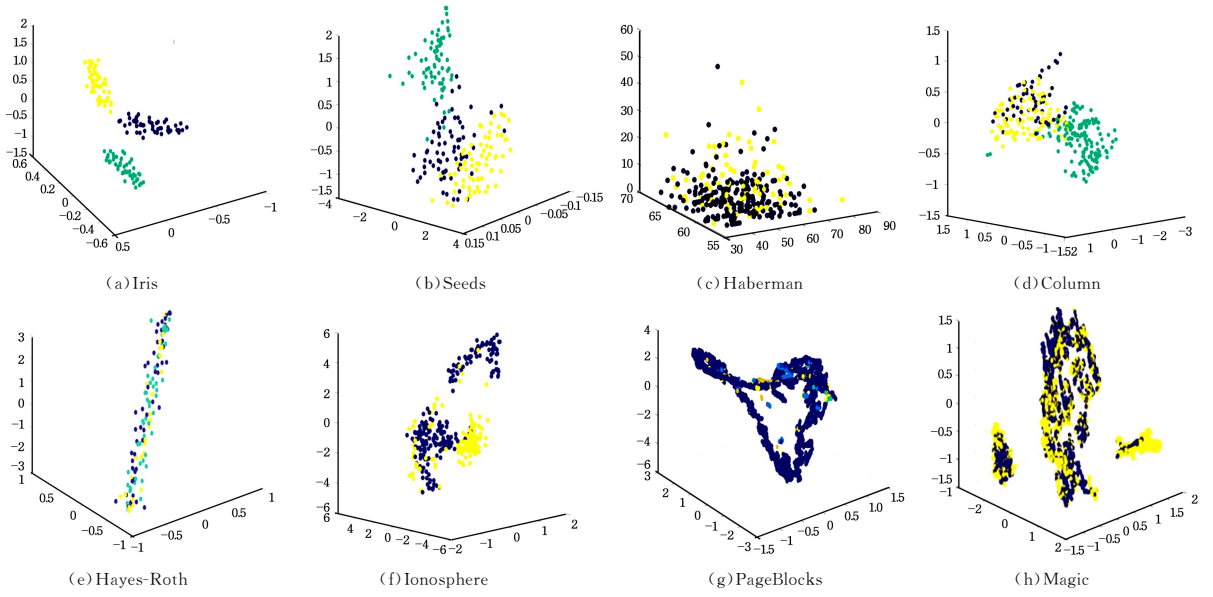


图 5 8 个 UCI 数据集的三维空间分布

Fig. 5 Three-dimensional spatial distributions of the eight UCI real machinelearning datasets

表 3 列出了本节讨论的 16 个数据集的所有空间分布特征。根据数据集的名称、样本点数、组成和结构，将表 3 分为 8 列。对于给定的数据集，“√”表示它具有表中第一行列出的相应特征，“×”代表其没有相应的特征。例如，在表 3 的第 3 行，D900 数据集的所有 9 个类簇都有 100 个样本点，它是凸型和平衡分布的数据集且在一些类簇中有离群点。图 4 和图 5 中的数据空间分布表明，DC-Kmeans 算法能够对多种数据集进行优化处理，同时还可以对规模较大的数据集进行处理。

表 3 16 个测试数据集的空间特征

Table 3 Spatial distribution characteristics of sixteen datasets

Data Sets	Points Number	Cluster Formation	Overlapping	Camber	Arc	Balanced	Outlier
Normal	200	5 * 40	×	×	√	√	√
D900	900	9 * 100	×	×	√	√	√
R15	600	15 * 40	×	×	√	√	√
N7	28000	7 * 4000	×	×	√	√	√
K3	102000	14400+43500+44100	×	×	√	×	√
Curve	180	20+80+80	×	√	×	×	×
Pathbased	300	93+97+110	×	√	×	×	×
Semicircle	300	90+99+111	×	√	×	×	×
Iris	150	3 * 50	√	×	√	√	×
Seeds	210	3 * 70	√	×	√	√	√
Haberman	306	82+225	√	×	×	×	√
Column	310	60+100+150	√	×	×	×	√
Hayes-Roth	132	30+51+51	√	√	×	×	×
Ionosphere	351	126+225	√	×	×	×	×
PageBlocks	5473	4913+329+28+88+115	√	×	×	×	√
Magic	19020	6688+12332	√	×	×	×	√

6.2 DC-Kmeans 的性能评价

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。本文使用 Purity 指标来评价聚类结果的准确性，其定义为：

$$purity=\sum_{i=1}^{K}\frac{|C_i|}{n}\max\left(\frac{m_{ij}}{|C_i|}\right) \tag{21}$$

其中，$|C_i|$ 为类簇 C_i 中所有数据对象的个数；m_{ij} 为类簇 C_i 的成员中属于类簇 C_j 的个数；K 是目标数据集中类簇的数目；n 是目标数据集中所包含的数据对象个数。本文将 Purity 指标的值转化为百分数来进行比较。

表 4 列出了 K-medoids，K-means++，DC-Kmeans 3 种算法对表 2 所列的 16 个数据集的处理精度。

表 4 不同算法的精度 Purity 比较

Table 4 Purity comparisons among different algorithms

（单位：%）

Data Sets	K-medoids	Kmeans++	DC-Kmeans
Normal	67.50	81.55	99.50
D900	74.27	80.11	99.78
R15	72.20	91.22	99.67
N7	81.28	87.14	100.00
K3	85.88	85.88	100.00
Curve	66.48	88.89	99.22
Pathbased	60.40	74.25	75.25
Semicircle	87.03	100.00	100.00
Iris	76.07	82.20	92.67
Seeds	68.62	75.53	88.10
Haberman	71.90	72.29	85.95
Column	69.39	71.23	71.29
Hayes-Roth	42.35	46.97	46.97
Ionosphere	66.10	70.31	71.51
PageBlocks	90.01	89.97	91.78
Magic	64.87	65.07	65.46

DC-Kmeans 算法每次划分的结果都能保持一致，因此只需要运行一次即可。相反地，其他两种算法的精度则取 10 次重复实验的平均值。由表 4 可以看出，由于初始类簇中心是随机选取的，K-medoids 算法的精度在 3 种算法中是最差的。在 K-means＋＋算法中，除第一个初始类簇中心外，其他类簇中心不再随机选择。因此 K-means＋＋算法的准确率优于 K-medoids 算法。DC-Kmeans 引入密度参数来选择初始类簇中心并且在更新阶段采用中心替换策略，因此 DC-Kmeans 的聚类精度在 3 种聚类算法中是最好的。

6.3 SCVI 指标的性能评价

本节首先通过测试列于表 2 的 8 个模拟数据集和 8 个 UCI 真实机器学习数据集来评估 SCVI$^-$ 指标的性能。然后，将 SCVI$^-$ 指标与现有的 8 个聚类有效性指标（CH$^+$，I$^+$，STR$^+$，DBI$^-$，COP$^-$，SMV$^-$，BCVI$^-$ 和 DCVI$^-$）的性能进行对比。表 5 和表 6 分别列出了 8 个模拟数据集和 8 个 UCI 真实机器学习数据集的 SCVI$^-$ 指标值。在两个表中，不同数据集的 K 值范围均受到经验规则 $2\leqslant K\leqslant\sqrt{n}$ 的限制。例如，数据集 Normal 中有 200 个数据对象，故只需要计算当 K 值落在区间[2,14]时的 SCVI(K)函数值。下划线数字所对应的 K 值是 SCVI$^-$ 指标在不同数据集上获得的最佳类簇数。由于 N7，K3，PageBlocks 和 Magic 数据集中有 28 000 个、102 000 个、5 473 个和 19 020 个数据对象，K 的范围被限制为[2,167]，[2,319]，[2,73]和[2,137]的区间。为了节省空间，表 5 和表 6 只显示了两个数据集的部分指标值，但其余 6 个数据集的 SCVI$^-$ 指标值没有省略。

表 5　8 个模拟数据集的 SCVI$^-$ 指标

Table 5　Values of SCVI$^-$ index on eight simulated datasets

K_{opt}	Datasets							
	Normal	D900	R15	N7	K3	Curve	Pathbased	Semicircle
2	7.706	352.269	71.788	221.436	19.698	0.551	296.377	14.034
3	7.600	248.224	75.927	217.321	15.485	0.402	277.532	9.734
4	8.409	335.580	88.303	280.850	23.458	0.536	354.893	12.720
5	5.839	286.497	78.697	266.379	19.396	0.457	340.109	10.878
6	7.062	324.622	84.770	274.336	24.203	0.513	371.280	12.843
7	6.492	266.804	75.834	185.860	21.129	0.454	346.348	12.130
8	7.339	242.516	76.091	215.441	24.698	0.473	364.556	13.369
9	6.750	210.299	67.738	194.897	22.260	0.502	353.227	10.676
10	7.714	235.219	72.735	218.020	25.009	0.549	379.988	11.814
11	7.010	217.662	51.248	201.323	22.987	0.536	361.503	10.240
12	7.647	237.373	53.348	219.131	25.174	0.543	397.172	10.918
13	7.708	222.525	48.373	205.241	23.502	0.569	393.946	10.082
14	8.198	240.047	49.638	221.386	25.326	—	360.570	10.820
15	—	227.457	44.971	208.670	23.862	—	351.862	10.250
16	—	242.069	48.027	221.933	25.392	—	351.194	10.630
17	—	230.148	45.747	209.569	24.076	—	337.427	10.234
18	—	242.569	48.520	223.152	25.413	—	—	—
19	—	231.378	46.372	211.625	24.235	—	—	—
20	—	243.585	48.953	223.615	25.493	—	—	—
21	—	234.065	46.967	214.027	24.468	—	—	—
22	—	245.055	49.117	222.73	25.616	—	—	—
23	—	234.314	47.220	212.502	24.540	—	—	—
24	—	243.111	49.211	221.842	25.573	—	—	—
25	—	235.931	—	214.477	24.759	—	—	—
26	—	244.033	—	222.720	25.759	—	—	—
27	—	235.537	—	214.251	25.098	—	—	—
28	—	245.668	—	222.131	25.985	—	—	—
29	—	237.613	—	215.897	25.142	—	—	—
30	—	243.882	—	222.046	26.059	—	—	—
…	—	—	—	…	…	—	—	—
167	—	—	—	217.996	26.680	—	—	—
…	—	—	—	—	…	—	—	—
319	—	—	—	—	25.795	—	—	—

表 6　8 个 UCI 真实数据集的 SCVI$^-$ 指标

Table 6　CVI values of SCVI$^-$ index on eight UCI real machine learning datasets

K_{opt}	Datasets							
	Iris	Seeds	Haberman	Column	Hayes-Roth	Ionosphere	PageBlocks	Magic
2	11.158	35.725	559.488	8964	3665.456	32.429	79145808	41769
3	8.573	30.565	609.887	8649	2731.935	44.430	67842680	57198
4	11.458	39.778	665.974	11307	3497.953	55.256	85415552	69990
5	9.979	37.656	673.770	9424	2872.127	66.784	67395928	75415
6	12.390	45.933	761.896	11156	3381.695	72.524	78670032	88847
7	11.806	41.575	754.686	11477	2926.371	78.519	72720576	93746
8	12.692	43.832	834.138	11847	3305.951	88.988	80033016	99568
9	11.369	40.969	795.408	12042	2965.773	100.360	76201808	101766
10	14.370	50.636	896.901	13163	3291.707	111.029	77091800	114227
11	13.150	43.134	819.847	14756	3008.819	115.171	75975976	114162

（续表）

K_{opt}	Datasets							
	Iris	Seeds	Haberman	Column	Hayes-Roth	Ionosphere	PageBlocks	Magic
12	14.348	50.684	865.571	16152	—	122.271	82999048	125291
13	—	44.230	963.312	14514	—	133.382	80579744	128236
14	—	47.075	998.450	15408	—	141.325	86503864	126858
15	—	—	913.581	15836	—	144.766	84464936	140490
16	—	—	1008.586	16497	—	155.004	92610352	138928
17	—	—	979.979	17430	—	158.369	84908464	139542
18	—	—	—	—	—	164.458	92475552	142773
19	—	—	—	—	—	—	92072912	148516
20	—	—	—	—	—	—	96898816	153827
21	—	—	—	—	—	—	95548784	152908
22	—	—	—	—	—	—	99750712	158989
23	—	—	—	—	—	—	86759768	162261
24	—	—	—	—	—	—	101636712	169392
25	—	—	—	—	—	—	100918320	172210
26	—	—	—	—	—	—	80638400	177547
27	—	—	—	—	—	—	78092016	176560
28	—	—	—	—	—	—	82433072	181244
29	—	—	—	—	—	—	81010624	182514
30	—	—	—	—	—	—	83138976	186500
…	—	—	—	—	—	—	…	…
73	—	—	—	—	—	—	90426528	311667
…	—	—	—	—	—	—	—	…
137	—	—	—	—	—	—	—	461489

表 7 列出了 9 个聚类有效性指标分别与 DC-Kmeans 结合所确定出的 16 个数据集的 K_{opt}。在表 7 中，第一列数据集名称后面括号中的数字为对应数据集的真实类簇数。用“√”表示第一行的聚类有效性指标相对应的第一列数据集的真实类簇数；用“○”表示第一行的聚类有效性指标相对应的数据集的近类簇数；符号“×”表示不能得到对应数据集的正确的类簇数。在每个符号后面，括号中的数字对表示“最佳类簇数”和相应的指标值。例如，第二行和第二列交叉处的数字对(5,630.84)表示当 Normal 数据集上 K 值为 5 时，CH^+ 指标将获得最大值(630.84)。从表 7 可以看出，$SCVI^-$ 可以得到所有 16 个测试数据集的真实类簇数。也就是说，在寻找目标数据集的 K_{opt} 方面，$SCVI^-$ 是所有对比指标中最优的。

表 7 用不同 CVIs 对 16 个数据集的聚类效果评价

Table 7 Clustering effects evaluated by different CVIs for sixteen datasets

DataSets(K_{opt})	CH^+	I^+	STR^+	DBI^-	COP^-	SMV^-	$BCVI^-$	$DCVI^-$	$SCVI^-$
Normal(5)	√(5,630.84)	×(2,0.410)	√(5,6.110)	√(5,0.436)	√(5,0.210)	√(5,0.401)	√(5,0.844)	√(5,0.960)	√(5,5.539)
D900 (9)	√(9,9332.95)	×(2,4.582)	√(9,21.122)	√(9,0.349)	√(9,0.166)	○(8,0.350)	○(8,13.943)	○(8,19.158)	√(9,210.299)
R15 (15)	√(15,4871.94)	×(2,0.933)	√(15,24.530)	√(15,0.315)	√(15,0.156)	√(15,0.253)	×(5,7.003)	×(5,8.189)	√(15,44.971)
N7 (7)	√(7,380485)	×(5,1.798)	×(158,71.111)	√(7,0.274)	√(7,0.133)	√(7,0.286)	√(7,1.614)	√(7,1.612)	√(7,185.860)
K3 (3)	√(3,1667182)	○(2,0.789)	×(209,453.44)	√(3,0.269)	√(3,0.129)	√(3,0.309)	×(12,0.140)	×(12,0.136)	√(3,15.485)
Curve(3)	×(8,476.35)	○(4,0.084)	×(8,4.152)	√(3,0.538)	×(8,0.291)	√(3,0.531)	○(4,0.049)	√(3,0.059)	√(3,0.402)
Pathbased (3)	×(17,407.76)	○(2,3.912)	×(17,0.776)	√(3,0.686)	○(4,0.316)	×(14,0.594)	×(5,31.789)	√(3,37.546)	√(3,277.532)
Semicircle (3)	×(16,2312.15)	○(2,0.734)	×(13,2.727)	×(13,0.507)	×(13,0.244)	×(11,0.516)	○(4,0.854)	○(4,1.087)	√(3,9.734)
Iris(3)	√(3,560.37)	√(3,0.806)	○(2,2.272)	○(2,0.405)	○(2,0.205)	○(2,0.484)	√(3,1.088)	√(3,1.323)	√(3,8.573)
Seeds(3)	√(3,375.81)	○(2,1.722)	×(13,1.033)	○(2,0.691)	√(3,0.311)	√(3,0.619)	√(3,3.305)	√(3,4.186)	√(3,30.565)
Haberman (2)	×(4,256.30)	√(2,5.196)	×(4,0.359)	×(4,0.847)	√(2,0.255)	×(13,0.702)	×(4,62.003)	×(4,73.877)	√(2,599.488)
Column(3)	×(5,224.38)	○(2,203.77)	√(3,4.351)	○(2,0.099)	○(2,0.088)	○(2,0.131)	√(3,1170.50)	√(3,1282.97)	√(3,8649.06)
Hayes-Roth (3)	×(11,1057.89)	○(2,16.511)	×(9,1.751)	○(2,0.506)	○(2,0.253)	×(5,0.572)	○(4,343.05)	√(3,432.56)	√(3,2731.94)
Ionosphere (2)	√(2,118.82)	√(2,0.453)	○(3,0.067)	×(11,1.362)	×(12,0.344)	×(11,0.686)	×(5,5.998)	×(5,6.349)	√(2,32.429)
PageBlocks (5)	×(47,17915.6)	×(3,2650.91)	×(42,15.278)	×(2,0.365)	×(2,0.027)	×(68,0.361)	×(3,12394817)	×(3,16343546)	√(5,67395928)

表 8 列出了 9 个聚类有效性指标的时间复杂度，可以看出 COP^- 的时间复杂度最高，达到了 $O(n^2)$；其余 8 个指标的时间复杂度均小于 $O(n^2)$。从表 8 中也可以看出，$SCVI^-$ 指标的时间性能依赖于目标数据集各个类簇所包含的数据对象的个数。如果各个类簇所包含数据对象的数目大致均等，则 $SCVI^-$ 指标的时间性能接近线性时间；否则，$SCVI^-$ 指标的性能取决于数据集中最大类簇所包含的数据对象的数目。表 9 列出了 9 个聚类有效性指标在评估各个数据集聚类划分结果时所消耗的时间。从表 9 可以看出，在 8 个聚类有效性指标中，COP^- 由于时间复杂度为 $O(n^2)$，消耗的时间最多。与此同时，由于 Normal，D900，R15，N7，Pathbased，Semicircle，Iris，Seeds，Column，Hayes-Roth，Ionosphere 等 11 个数据集的各个类簇中的数据对象的个数要么相等，要么大致相等，故 $SCVI^-$ 的时间开销和其余 7 个线性时间的聚类有效性指标

的时间开销相差不大。然而，当 $SCVI^-$ 面对不均衡分布的数据集（即 K3，Curve，Haberman，PageBlocks，Magic 5 个数据集）时，其时间开销比其余 7 个线性时间的聚类有效性指标更高。

表 8 8 个指标的时间复杂度

Table 8 Time complexity of 9 CVIs

Indicator name	CH^+	I^+	STR^+	DBI^-	COP^-	SMV^-	$BCVI^-$	$DCVI^-$	$SCVI^-$
Time complexity	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n^2)$	$O(n)$	$O(n)$	$O(n)$	$\max\{O(n),O((C_{\max})^2)\}$

表 9 9 个指标在评估 16 个数据集划分结果上的时间消耗

Table 9 Time costs of 9 CVIs on 16 datasets

（单位：ms）

Index	Data Sets								
	CH^+	I^+	STR^+	DBI^-	COP^-	SMV^-	$BCVI^-$	$DCVI^-$	$SCVI^-$
Normal	0.306178	0.371982	0.419426	0.1156	12.51277	0.336827	0.1222	0.085333	0.31277
D900	0.354855	1.388471	1.6679682	3.0822	34.575664	0.450705	0.3019	0.083531	1.22500
R15	0.237672	0.188996	0.9045675	3.3243	19.81088	0.317596	0.2028	0.152038	0.48184
N7	6.991033	2.183214	28.9679679	8.3897	16994.542	6.921624	6.6962	0.740658	9.60321
K3	11.745666	3.019723	67.5859734	9.9608	173695.93	20.760064	12.291	2.76913	50280.2
Curve	0.118986	0.087436	0.2904336	0.1409	5.93458	0.374086	0.3374	0.142122	0.71183
Pathbased	0.148132	0.122892	0.4827033	3.1128	8.38431	0.447399	0.2737	0.15985	0.52954
Semicircle	0.213634	0.150836	0.5330016	0.2223	9.172443	0.279136	0.1129	0.077221	0.69813
Iris	0.128	0.11508	0.3602043	0.1556	4.314747	0.231061	0.0970	0.063399	0.52646
Seeds	0.325408	0.231361	0.8940186	0.3933	8.074826	0.436883	0.3308	0.120788	0.92733
Haberman	0.563081	0.182686	0.6108858	0.1886	7.952835	0.282442	0.1340	0.097652	2.71636
Column	0.37709	0.311287	1.518696	0.6132	15.076676	0.359662	0.2310	0.056488	4.32414
Hayes-Roth	0.615061	0.149635	0.4754025	0.2172	5.403951	0.286949	0.1189	0.08293	0.94283
Ionosphere	1.29202	3.246577	22.4185941	0.2544	12.046136	0.342235	0.8653	0.170367	8.65894
PageBlocks	4.914185	0.408939	23.3637237	1.7718	221.54844	1.792903	4.3634	0.252695	199.267
Magic	10.252633	1.562143	25.8032088	2.8445	4457.2765	4.270277	5.5316	5.53165	2075.63

结束语 本文首先提出 DC-Kmeans 算法来解决传统 K-means 算法的不足。在初始阶段，DC-Kmeans 使用密度参数来避免随机选择初始类簇中心的问题。在迭代阶段，DC-Kmeans 在类簇中心被离群值扭曲时，使用中心替换方法对类簇中心进行更新。通过 DC-Kmeans 算法对目标数据集进行划分后，利用簇内凑紧度和簇间分离度的加权组合定义新的 SCVI 指标，以此来评价聚类结果的质量。最后，在 DC-Kmeans 算法和 SCVI 指标的基础上，设计了一种新的算法（OCNS）来确定不同数据集的最佳类簇数。对多种类型数据集的测试结果表明，本文提出的方法是有效的，具有广泛的应用前景。但是，由于 $SCVI^-$ 指标的时间复杂度为 $\max\{O(n),O((C_{\max})^2)\}$，其性能取决于数据集最大类簇所包含的数据对象的数目。故 $SCVI^-$ 指标在处理数据对象数目分布不均衡的数据集时，其时间开销高于对比实验中的其余 7 个线性时间的聚类有效性指标。因此，在今后的工作中，需要进一步的研究来克服这一缺点。

参 考 文 献

[1] XU R，WUNSCH D. Survey of clustering algorithm [J]. IEEE Transactions on Neural Networks，2005，16(3)：645-678.

[2] LIANG B，LIANG J Y，CHAO S，et al. Fast global Kmeans clustering based on local geometrical information[J]. Information Sciences，2013，245：168-180.

[3] REDMONDS J，HENEGHAN C. A method for initialising the Kmeans clustering algorithm usingkd-trees[J]. Pattern Recognition Letters，2007，28(8)：965-973.

[4] ZHOU S B，XUZ Y. A novel internal validity index based on the cluster centre and the nearest neighbour cluster[J]. Applied Soft Computing，2018，71：78-88.

[5] ZHU E Z，MA R H. An effective partitional clustering algorithm based on new clustering validity index[J]. Applied Soft Computing，2018，71：608-621.

[6] CALINSKI T，HARABASZ J. A dendrite method for cluster analysis[J]. Communications in Statistics，1974，3(1)：1-27.

[7] ROUSSEEUW P J. Silhouettes：A graphical aid to the interpretation and validation of cluster analysis[J]. Journal of Computational and Applied Mathematics，1987，22：53-65.

[8] GURRUTXAGA I，ALBISUA I，ARBELAITZ O，et al. SEP/COP：An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index[J]. Pattern Recognition，2010，43：3364-3373.

[9] YUE S H，WANG J P，WANG J，et al. A new validity index for evaluating the clustering results by partitional clustering algorithm[J]. Soft Computing，2016，20(3)：1127-1138.

[10] CHEN X Y，SU Y L，CHEN Y，et al. GKmeans：an Efficient Kmeans Clustering Algorithm Based on Grid[C]// Proceedings of the 1st International Symposium on Computer Network and Multimedia Technology (CNMT 2009). Wuhan，China，2009：18-20.

[11] ISLAM M Z，ESTIVILL-CASTRO V，RAHMAN M A，et al. Combining Kmeans and a genetic algorithm through a novel arrangement of genetic operators for high quality clustering[J]. Expert Systems with Applications，2018，91：402-417.

[12] YODER J，PRIEBE C E. SEMI-SUPERVISED Kmeans++[J]. Journal of Statistical Computation and Simulation，2017，87(13)：2597-2608.

[13] HUSSAIN S F，HARIS M. A Kmeans based co-clustering

(kCC) algorithm for sparse, high dimensional data[J]. Expert Systems with Applications, 2019, 118: 20-34.
[14] FADAEI A H, KHASTEH S H. Enhanced Kmeans re-clustering over dynamic networks[J]. Expert Systems with Applications, 2019, 132: 126-140.
[15] ZHU E Z, ZHANG Y X, WEN P, et al. Fast and Stable Clustering Analysis based on Grid-mapping Kmeans Algorithm and New Clustering Validity Index[J]. Neurocomputing, 2019, 363: 149-170.
[16] DUNN J C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters[J]. Journal of Cybernetics, 1974, 3: 32-57.
[17] HUBERT L, SCHULTZ J. Quadratic assignment as a general data analysis strategy[J]. British Journal of Mathematical and Statistical Psychology, 1976, 29(2): 190-241.
[18] MAULIK U, BANDYOPADHYAY S. Performance evaluation of some clustering algorithms and validity indices[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2002, 24(12): 1650-1654.
[19] DAVIES D L, BOULDIN D W. A cluster separation measure [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1979, 1(2): 224-227.
[20] BEZDEK J C. Numerical taxonomy with fuzzy sets[J]. Journal of Mathematical Biology, 1974, 7(1): 57-71.
[21] BEZDEK J C. Cluster validity with fuzzy sets[J]. Journal of Cybernetics, 1974, 3(3): 58-74.
[22] ZALIK K R. Cluster validity index for estimation of fuzzy clusters of different sizes and densities[J]. Pattern Recognition, 2010, 43(10): 3374-3390.
[23] KIM D W, LEE K H, LEE D. On cluster validity index for estimation of the optimal number of fuzzy clusters[J]. Pattern Recognition, 2004, 37(10): 2009-2025.
[24] CHEN M Y, LINKENS D A. Rule-base self-generation and simplification for data-driven fuzzy models[J]. Fuzzy Sets and Systems, 2004, 142(1): 243-265.
[25] TANG Y G, SUN F C, SUN Z Q. Improved validation index for fuzzy clustering[C]// Proceedigs of the 2005 American Control Conference (ACC 2005). 2005: 1120-1125.
[26] PAKHIRA M K, BANDYOPADHYAY S, MAULIK U. Validity index for crisp and fuzzy clusters[J]. Pattern Recognition, 2004, 37(3): 487-501.
[27] WU K L, YANG M S. A cluster validity index for fuzzy clustering[J]. Pattern Recognition Letters, 2005, 26(9): 1275-1291.
[28] STARCZEWSKI A. A new validity index for crisp clusters[J]. Pattern Analysis and Applications, 2017, 20: 687-700.
[29] ZHU E Z, ZHU B B, WEN P, et al. Effective Clustering Analysis based on New Designed CVI and Improved Clustering Algorithms[C]// Proceedings of the 16th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA 2018). 2018: 766-772.
[30] MAATEN L V D. t-SNE[OL]. https://lvdmaaten. github. io/tsne.
[31] WANG Z Y, LIU J L. Kernel Subspace Clustering Based on Second-order Neighbors [J]. Computer Science, 2021, 48(6): 86-95.
[32] PENG C C, CHEN Y L, XUN Y M. k-modes Clustering Guaranteeing Local Differential Privacy [J]. Computer Science, 2021, 48(2): 105-113.

ZHANG Ya-di, born in 1996, postgraduate. Her main research interests include cluster analysis and machine learning.

ZHU Er-zhou, born in 1981, Ph.D, associate professor, postgraduate supervisor. His main research interests include virtualization, program analysis, data mining, and information security.

(责任编辑:柯颖)