计算机科学
COMPUTER SCIENCE

一种基于属性相似性和分布结构连通性的聚类算法

孙浩文，丁家满，李博文，贾连印

引用本文

孙浩文，丁家满，李博文，贾连印. 一种基于属性相似性和分布结构连通性的聚类算法[J]. 计算机科学, 2024, 51(7): 124-132.

SUN Haowen, DING Jiaman, LI Bowen, JIA Lianyin. Clustering Algorithm Based on Attribute Similarity and Distributed Structure Connectivity [J]. Computer Science, 2024, 51(7): 124-132.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231000125

一种基于属性相似性和分布结构连通性的聚类算法

孙浩文　丁家满　李博文　贾连印
昆明理工大学信息工程与自动化学院　昆明 650500
云南省人工智能重点实验室(昆明理工大学)　昆明 650500
(shw_haowen@163.com)

摘　要　聚类分析针对不同的数据特点采用不同的相似性度量,现实世界中数据分布复杂,存在分布无规律、密度不均匀等现象,单独考虑实例属性相似性或分布结构连通性会影响聚类效果。为此,提出了一种基于属性相似性和分布结构连通性的聚类算法(A Clustering Algorithm Based on Attribute Similarity and Distributed Structure Connectivity,ASDSC)。首先,利用待聚类数据集中的所有数据实例构建完全无向图,定义了一种兼顾属性相似和分布结构连通的新颖相似性度量方式,用于计算节点相似性,并构造邻接矩阵更新边的权重;其次,借助邻接矩阵执行递增步长的随机游走,依据顶点的连通中心性来识别簇中心并给定簇编号,同时获取其他顶点的连通性;然后,利用连通性计算顶点间的依赖关系,并据此进行簇编号的传播,直至完成聚类。最后,为了验证该方法的聚类性能,在16个合成数据集和10个真实数据集上与5种先进聚类算法进行了对比实验,ASDSC算法取得了优异性能。

关键词:聚类;相似性度量;属性相似性;分布结构连通性;簇编号传播
中图分类号　TP391

Clustering Algorithm Based on Attribute Similarity and Distributed Structure Connectivity

SUN Haowen,DING Jiaman,LI Bowen and JIA Lianyin
Faculty of Information Engineering and Automation,Kunming University of Science and Technology,Kunming 650500,China
Artificial Intelligence Key Laboratory of Yunnan Province,Kunming University of Science and Technology,Kunming 650500,China

Abstract　According to different data characteristics,clustering analysis adopts different similarity measures. However,the data distribution is complex in the real world,and there are various phenomena such as irregular distribution and uneven density. Considering attribute similarity or distribution structure connectivity alone will reduce clustering performance. Therefore,this paper proposes a clustering algorithm based on attribute similarity and distributed structure connectivity(ASDSC). Firstly,a completely undirected graph is constructed using all data instances,and a novel similarity measurement method is defined to calculate the node similarity by the topology structure and the attributes similarity,and the adjacency matrix are constructed to update the weights of edges. Secondly,based on the adjacency matrix,random walk with increasing step is performed. Subsequently,the cluster centers and their numbers are obtained according to the connected centrality of nodes,and the connectivity of other nodes is also acquired. Then,the connectivity is used to calculate the dependencies among nodes,and the propagation process of cluster number is carried out accordingly until the clustering process is completed. Finally,comparative experiments with 5 advanced clustering algorithms are conducted on 16 synthetic datasets and 10 real datasets,and the result show that the ASDSC algorithm has achieved excellent performance.

Keywords　Clustering,Similarity measure,Attribute similarity,Distributed structure connectivity,Cluster number propagation

聚类分析是数据挖掘和知识发现中一种重要且有效的方法,旨在挖掘数据潜在的信息,已经被广泛应用于文本挖掘[1]、欺诈检测[2]、社交网络分析[3-4]、基因检测[5]、数据压缩[6]等领域。作为一种重要的无监督学习技术,聚类分析试图根据某种相似性度量方式,将一组待聚类的实例划分为多个非重叠的簇,使得同一簇中的实例彼此之间高度相似,且不同簇中的实例相似性尽可能地小。相似性度量方式在聚类方法的设计中起着至关重要的作用,在相同的数据集上,不同的相似性度量方式可能会产生不同的聚类结果[7]。

在聚类分析过程中,常见的度量策略是利用实例之间的

到稿日期:2023-10-18　返修日期:2024-03-29
基金项目:国家自然科学基金(62262034,62262035);云南省科技揭榜项目(202204BW050001)
This work was supported by the National Natural Science Foundation of China(62262034,62262035) and Ranking the Top of the List for Science and Technology Project of Yunnan Province(202204BW050001).
通信作者:丁家满(tjoman@126.com)

空间距离来确定两个实例之间的属性相似性，该类距离通常使用欧氏距离或余弦相似度来表示。比如，经典的基于划分的聚类算法 K-Means 及其衍生算法[8]将所有实例看作空间中的点，将属性值看作坐标值，通过计算欧氏距离来衡量相似性，这类算法特别适合发现球状互斥的簇，但处理非球状数据效果不佳。而著名的密度聚类算法 DBSCAN 及其改进算法[9]使用一组关于邻域的参数来描述样本分布结构的连通性，并依据紧密程度来衡量相似性，它能识别任意形状的簇。但单一邻域半径不适合处理密度差异明显的数据，且不同的数据集也需要不同的邻域半径，这使得聚类效果受限。层次聚类算法[10]也是考虑实例属性相似性，但不同于基于划分的聚类算法，通过计算实例与簇中心距离进行聚类，层次聚类采用分步计算两两之间距离的方式进行聚类形成层次，但并未考虑结构连通性。同样地，基于图结构的谱聚类算法[11]通过计算实例之间属性相似性来更新边的权重，认为距离较远的两点之间边的权重较低，距离较近的两点间边的权重较高，然后对所有数据点组成的图进行切图以达到聚类的目的，该算法也仅仅考虑了实例之间的属性相似性，忽略了分布结构的连通性。随着大数据时代的到来，数据不仅属性特征不断丰富，诸如社区结构、蛋白质互作用等复杂网络数据也不断涌现。在进行聚类分析时，只考虑实例属性相似性或分布结构连通性将导致数据对象之间相关关系的丢失，影响聚类效果。

针对上述问题，本文定义了一种兼顾属性和分布结构的相似性度量方式，并据此提出了一种基于属性相似性和分布结构连通性的聚类算法 ASDSC。ASDSC 算法不仅可以有效地提高稀疏簇中遥远对象之间的相关性，也可以降低相邻但来自不同簇对象之间的相关性，进而捕获具有同质属性且结构高度凝聚的簇。该方法首先利用所有数据实例构建完全无向图，定义一种兼顾属性相似和分布结构连通的新颖相似性度量方式，用于计算节点相似性，并构造邻接矩阵更新边的权重；其次，借助邻接矩阵执行递增步长的随机游走，依据顶点的连通中心性来识别簇中心并给定簇编号，同时获取其他顶点的连通性；然后，利用连通性计算顶点间的依赖关系，并据此进行簇编号传播，直至完成聚类。综上所述，本文的主要贡献包括 4 个方面：

1）定义了一种新颖的相似性度量方法，通过分析数据的属性相似性和分布结构连通性来评估实例之间的相似性。

2）相似性度量方法被用于构造完全无向图的邻接矩阵，利用该邻接矩阵执行随机游走，可以识别具有强连通性的簇中心。

3）根据顶点连通性获取非中心顶点的高连通近邻，并计算顶点倾向性，以此获得顶点间的依赖关系，从而进行簇中心编号的传播，达到完成聚类的目的。

4）应用不同规模和不同维度的数据集进行实验，结果表明 ASDSC 在标准互信息和准确率方面性能良好。

1 相关工作

相似性度量在聚类算法中起着重要作用，直接影响聚类的质量。在传统聚类方法中，处理属性相似性的常见策略是利用各种空间距离度量方法。由于待聚类数据集中的数据对象通常是由数值属性来描述的，因此距离度量可以衡量数据对象之间的属性相似性。一般来说，具有相似属性的对象往往会被划分到同一个簇中，因此基于属性相似性的聚类可以检测到具有同质属性的簇。例如，CBKM[12]和 LLKM 算法[13]采用欧几里得距离衡量数据对象之间的属性相似性，通过迭代地计算簇内距离的误差平方和，使得在同一个簇中对象的属性是相似的，而不同簇中的对象是相异的。虽然此类方法在发现球状簇方面表现良好，但其忽略了并非所有属性都是同等重要的事实。因此，FKMAWCW[14]定义了一种新的距离度量方式，用于表示数据对象之间的属性相似性，它通过组合概率距离和非欧几里得距离，在聚类过程中可以降低噪声属性对聚类结果的影响。除此之外，新兴的图聚类方法也为属性相似性的衡量提供了一定的借鉴意义，其中 Co-Homo[15]定义了一种属性图框架，为每个簇定义并优化相关权重向量，以捕获数据对象之间的属性相似性，并通过同步或异步更新簇的相关权重，进而检测具有不同属性相关模式的簇。这类方法实现了簇内对象的属性同质化，但忽略了簇的分布结构连通性，这会导致数据对象之间相关关系的丢失。

另一方面，根据数据的分布结构连通性去衡量数据对象之间的相似性也是常见策略之一。Erich 等[16]利用簇的高密度连通性挖掘数据对象在结构上的相似性，将紧密相连的对象划分到一个类别，进而探索和发现不同形状的簇。DenMune[17]在数据分布上的相互最近邻一致性原则的指导下计算每个对象的密度，并选择高密度点作为种子，进一步决定簇的结构框架，并在不同形状的数据集上表现出鲁棒性，但单一参数不能很好地反映数据集密度的变化。考虑到复杂数据通常是高维的、异构的，图聚类算法 SPRG[18]随机森林构造鲁棒的数据亲和图，并根据它的特征结构，即特征值和特征向量，将数据样本划分为不相交子集，从而达到聚类的目的，但只考虑了实例之间的属性相似性，忽略了分布结构的连通性。为了获得更好的图切割，有学者使用近邻策略的 SMKNN 算法[19]，通过从 K 近邻图中移除枢纽顶点来获得子图，最后合并满足最大相似性的相邻簇，以达到聚类的目的。然而当前许多方法在使用 K 近邻图时要求用户指定参数 K，因此 AKNNG 算法[20]基于具有稀疏结构的自适应 K 近邻相似图来捕获更好的图形结构，并构建拉普拉斯矩阵和特征向量，进一步提升聚类精度，并降低最近邻数量的敏感性。上述基于近邻策略的聚类算法主要关注数据的分布结构连通性，聚类结果通常包括多个具有高度凝聚结构的簇，然而这种方法在很大程度上忽略了数据对象之间的属性相似性，这会使得属于同一个簇的对象的属性具有较大的随机性。

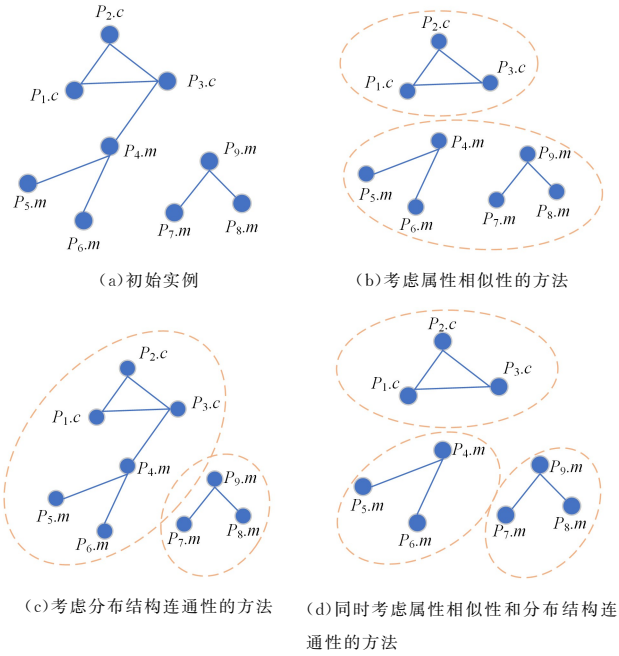
因此，在衡量数据对象之间的相似性时，同时考虑数据的分布结构连通性和属性相似性是聚类分析中的一个关键问题，也是捕获具有同质属性且结构高度凝聚的簇的目的。为此，本文提出了一种基于属性相似性和分布结构连通性的聚类算法 ASDSC。

2 问题描述

假设 $\boldsymbol{X}=\{x_1,x_2,x_3,\cdots,x_i,\cdots,x_N\}\in R^{d\times N}$ 是一组包含 N 个无标记对象的待聚类数据集，其中每个实例 $x_i=(x_{i1},$

$x_{i2},\cdots,x_{id})^{\mathrm{T}}\in R^d$ 都具有一组 d 维属性。为了便于对数据实例之间的关系进行建模，本文借助图结构来表示聚类问题，利用 $\boldsymbol{X}$ 中的所有数据对象构建完全无向图 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E},\boldsymbol{A})$。其中，$\boldsymbol{V}$ 代表顶点的集合，$v_i\in\boldsymbol{V}$ 与对象 $x_i\in\boldsymbol{X}$ 一一对应，顶点集合的大小为 $|\boldsymbol{V}|=N$，$\boldsymbol{E}$ 代表边的集合。$\boldsymbol{A}$ 代表图 $\boldsymbol{G}$ 的邻接矩阵，用于表示图 $\boldsymbol{G}$ 中的顶点之间的相似性，该矩阵具有对称性。

在聚类过程中，同一数据集根据不同的相似性度量方式会产生不同的聚类结果。通过空间距离度量属性相似性在识别具有同质属性的簇方面具有优势，但不能发现任意形状的簇。基于分布结构连通性可以利用簇的拓扑结构，从而挖掘任意形状的簇。但随着数据在属性特征剧增以及分布趋于复杂的背景下，只考虑实例属性相似性或分布结构连通性将导致数据对象之间相关关系的丢失，影响聚类效果。本文以一个简单的实例来说明同时考虑两者的重要性，如图 1(a)所示，其中文献关系图中的顶点表示文献，边表示两篇文献之间的相关性。此外，还包括每篇文献的 ID 与其所属的主要研究领域，其中主要研究领域用于描述顶点的属性，图中的属性值‘c’表示计算机，而属性值‘m’表示医学。



(a)初始实例 (b)考虑属性相似性的方法

(c)考虑分布结构连通性的方法 (d)同时考虑属性相似性和分布结构连通性的方法

图 1 一个简单的实例根据不同方法得到的聚类结果

Fig. 1 A simple example of clustering results according to different methods

1)考虑属性相似性的方法

若单独考虑属性的相似性，属于同一研究领域中的文献将被划分在一个簇中。图 1(b)中，文献 $P_1\sim P_3$ 同属于计算机领域的研究文献，医学领域的研究文献 $P_4\sim P_9$ 则属于另一个簇。但这样会忽略文献之间的结构，导致跨领域研究文献之间的相关性丢失。

2)考虑分布结构连通性的方法

在不考虑属性相似性时，根据文献之间的相关性，在分布结构上连通的文献会被划分到一个簇。因此，文献 $P_1\sim P_6$ 被划分到一个簇，如图 1(c)所示，但实际上它们具有不同的属性，一半属于计算机领域，一半属于医学领域。

3)考虑属性相似性和分布结构连通性的方法

图 1(d)给出了同时考虑顶点的属性相似性和分布结构连通性的聚类结果。文献被划分为 3 个簇，其中每个簇内的文献高度相关，且保证了它们在研究领域上的同质性。这在一定程度上表现了结合结构和属性相似性的方法的优势，也启发了本文算法的设计与实现。

3 ASDSC 聚类算法

3.1 算法思想

本文提出的 ASDSC 算法首先构建了完全无向图，确定实例之间的相似性并以此构造邻接矩阵；然后基于邻接矩阵执行随机游走，获得簇中心和顶点连通性，最后依据顶点连通性获取依赖关系来完成最终聚类。该算法的相关模型的框架如图 2 所示。

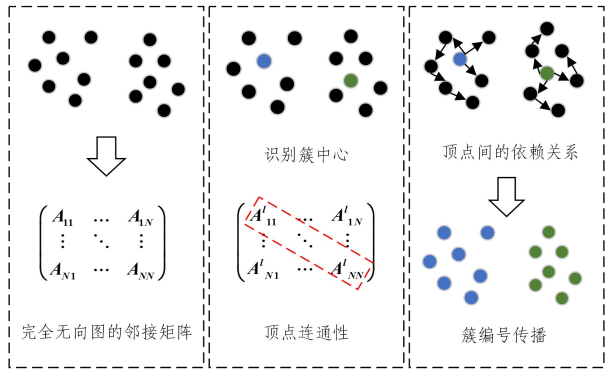


图 2 ASDSC 算法的模型图

Fig. 2 Diagram of ASDSC algorithm

整个算法主要包括 3 个阶段，首先是基于属性相似性和分布结构连通性构建邻接矩阵。具体来说，本文利用待聚类数据集中的所有实例构建完全无向图，用欧几里得距离衡量顶点之间的属性相似性，利用 K 距离密度表示顶点的分布结构连通性。结合两者定义了一种相似性度量方式来确定顶点之间的相似性，据此对完全无向图的边加权，同时构造邻接矩阵。其次是簇中心的识别和顶点连通性的获取。将上述邻接矩阵作为步长为 1 的随机游走，通过邻接矩阵的累乘来描述递增步长的随机游走的过程，当达到收敛条件时，停止随机游走，在截止邻接矩阵中获得收敛条件下顶点的连通性，并识别连通中心，即簇中心。最后挖掘顶点间的依赖关系并完成聚类。根据顶点连通性获取非中心顶点的高连通近邻，并计算顶点倾向性，以此来获得顶点间的依赖关系，从而进行簇中心编号的传播，达到完成聚类的目的。

3.2 构造邻接矩阵

通常，两个实例之间的距离越大，两者属性之间的相似性越低。同样地，密度在一定程度上反映了实例的紧密程度，表示实例的分布结构连通性。对于同一类别的实例而言，它们之间往往是紧密相连的。基于上述启发，本文定义了一种 K 距离密度来度量分布结构连通性，以实例之间的欧几里得距离作为属性相似性，并结合两者定义了一种相似性度量方式计算顶点之间的相似性。

假设 $\boldsymbol{D}(v_i,v_j)$ 度量的是顶点 v_i 和 v_j 之间的欧几里得

距离,顶点 v_i 离其第 K 个最近的邻居的距离表示为 $K\text{-}D(v_i)$。在顶点集合 V 中找出与 v_i 最邻近的 K 个顶点,则涵盖这 K 个顶点的集合为:

$$T_K(v_i)=\{v_j \mid v_j\in V \text{ and } i\neq j\ D(v_i,v_j)\leqslant K\text{-}D(v_i)\} \quad (1)$$

通常,在给出固定区域面积的情况下,密度是由给定区域面积内实例的数量衡量的,实例数量越多,密度越大。但本文对传统衡量密度的方法进行变形,在给出固定邻居数量的实例的情况下,某实例的密度是由邻居数量与每个离它最邻近点的距离总和的比值决定的。

定义 1(K 距离密度) 给定顶点 v_i 的 K 近邻集合为 $T_K(v_i)$,则顶点 v_i 的 K 距离密度为:

$$\rho(v_i)=\frac{K}{\sum_{j=1}^{K}{}_{v_j\in T_K(v_i)} D(v_i,v_j)} \quad (2)$$

当顶点 v_i 与其 K 近邻的距离之和越小,其密度越大,反之密度越小。这种方式能够有效地反映顶点之间的紧密程度。

结构连通性和属性相似性分别从不同角度描述顶点的特性,那么如何结合两者来衡量顶点相似性成为一个难题。为了同时考虑顶点属性的同质化和顶点结构的连通性,本文提出了一种新颖的相似性度量方式。

定义 2(结构和属性相似性) 本文采用欧几里得距离 $\boldsymbol{D}(v_i,v_j)$ 描述顶点 v_i 和 v_j 之间的属性相似性,用 $\rho(v_i)$ 和 $\rho(v_j)$ 分别描述顶点 v_i 和 v_j 在分布结构上的连通性,则顶点 v_i 和 v_j 之间的结构和属性相似性为:

$$A(v_i,v_j)=\exp(-\rho_i D^2(v_i,v_j)\rho_j) \quad (3)$$

上述相似性度量方式将定量衡量顶点 v_i 和 v_j 之间的近似性。若顶点 v_i 和 v_j 是较为稀疏的簇中的两个顶点,则顶点 v_i 和 v_j 较远,即 $\boldsymbol{D}(v_i,v_j)$ 较大;相应地,$\rho(v_i)$ 和 $\rho(v_j)$ 较小。在这种情况下,$\boldsymbol{A}(v_i,v_j)$ 很好地结合了顶点 v_i 和 v_j 的结构和属性相似性,提升了顶点 v_i 和 v_j 之间的相似性。若顶点 v_i 和 v_j 分别属于稀疏簇 $\boldsymbol{C}_{\text{Sparse}}$ 和稠密簇 $\boldsymbol{C}_{\text{Dense}}$,且 v_i 和 v_j 距离较近,则 $\rho(v_i)$ 较小,$\rho(v_j)$ 较大。此时 $\boldsymbol{A}(v_i,v_j)$ 可以在一定程度上降低 $\boldsymbol{D}(v_i,v_j)$ 的影响,从而降低了 v_i 和 v_j 的相似性。因此,这种方式可以很好地处理具有不同结构和属性的簇。

基于上述结构和属性相似性,我们为完全无向图 $\boldsymbol{G}$ 构造邻接矩阵 $\boldsymbol{A}$。作为完全无向图,$\boldsymbol{A}$ 是对称的,即 $\boldsymbol{A}(v_i,v_j)=\boldsymbol{A}(v_j,v_i)$。此外,对于任意的 v_i 和 v_j,总满足 $0\leqslant\boldsymbol{A}(v_i,v_j)\leqslant 1$。

3.3 基于随机游走识别簇中心

在图 $\boldsymbol{G}$ 中,不仅存在彼此接近且连通的顶点,也存在彼此接近但不连通的顶点,甚至相距很远却连通的顶点。如果图 $\boldsymbol{G}$ 中的顶点 v_i 和 v_j 之间存在连接两者的多条路径,则认为它们是连通的。相反地,若很少甚至没有路径连接顶点 v_i 和 v_j,则它们是不连通的。一方面,在不同的随机游走路径下,图 G 中的顶点在不同的连通图中表现出不同的重要性。另一方面,不同步长的随机游走,也会导致顶点之间的连通性发生改变。因此,本文基于一种递增步长的随机游走方法,去差异化不同步长和不同随机游走路径下的顶点的连通性,以捕获图 $\boldsymbol{G}$ 的连通中心,即簇中心。

定义 3(连通性) 给定邻接矩阵 $\boldsymbol{A}$,对于任意的顶点 v_i 和 v_j,当随机游走的步长为 l 时,两者之间的连通性为:

$$A^l(v_i,v_j)=\sum_{m=1}^{N}A^{l-1}(v_i,v_m)A(v_m,v_j) \quad (4)$$

上述过程是随着随机游走步长 l 的递增,通过矩阵 $\boldsymbol{A}$ 不断累乘实现的。其中,$\boldsymbol{A}(v_i,v_j)$ 表示随机游走的步长为 1 时顶点 v_i 和 v_j 之间的连通性,则 $\boldsymbol{A}$ 是随机游走步长为 1 时的邻接矩阵。相应地,当随机游走的起始顶点为 v_i,终止顶点为 v_j,执行步长为 l 的随机游走时,顶点 v_i 和 v_j 之间的连通性为 $\boldsymbol{A}^l(v_i,v_j)$,那么 $\boldsymbol{A}^l$ 表示执行步长为 l 的随机游走得到的邻接矩阵。

从另一个角度讲,$\boldsymbol{A}^l(v_i,v_j)$ 表示相距 l 个步长时,顶点 v_i 和 v_j 之间的可达性。若 $\boldsymbol{A}^l(v_i,v_j)>0$,则说明 v_i 和 v_j 是可达的,若 $\boldsymbol{A}^l(v_i,v_j)=0$,则说明 v_i 和 v_j 是不可达的,即 v_i 和 v_j 是不连通的。对于无向图 $\boldsymbol{G}$ 中的任意顶点 $v_i\in\boldsymbol{V}$,v_i 到其自身总是可达的。那么可以推理出,当随机游走步长大于 1 时,任意 $v_i\in\boldsymbol{V}$ 到自身的可达性总是不小于 v_i 到其他的顶点的可达性,即总满足条件 $\boldsymbol{A}^l(v_i,v_i)\geqslant\boldsymbol{A}^l(v_i,v_j)$。若 V_i 向自身随机游走的过程中存在断路,则顶点 V_i 的连通中心性 $\boldsymbol{A}^l(v_i,v_j)$ 就会减弱,进而出现 $\boldsymbol{A}^l(v_i,v_i)<\boldsymbol{A}^l(v_i,v_j)$ 的情况。类比于运输网络,将图中每个顶点视为运输网络中的一个站点,使用顶点的连通性来表示站点的繁忙程度。显然,连通性越强,站点越有可能成为交通枢纽。基于上述理论支持,我们定义了连通中心的识别方法。

定义 4(连通中心) 当随机游走步长为 l 时,顶点 $v_i\in\boldsymbol{V}$ 的可达性为 $\boldsymbol{A}^l(v_i,v_i)$。在矩阵 $\boldsymbol{A}^l$ 中,若对角元素 $\boldsymbol{A}^l(v_i,v_i)$ 为当前行或当前列的最大值,则 $\boldsymbol{A}^l(v_i,v_i)$ 矩阵为 $\boldsymbol{A}^l$ 的对角极大元。本文中称 V_i 为连通中心,并将其作为簇中心。即当随机游走步长为 l 时,顶点 $v_i\in\boldsymbol{V}$ 若要成为连通中心,则需要满足以下条件:

$$\forall\, i\neq j,\boldsymbol{A}^l(v_i,v_j)\leqslant A^l(v_i,v_i) \text{ 或 } \boldsymbol{A}^l(v_j,v_i)\leqslant A^l(v_i,v_i) \quad (5)$$

当随机游走步长为 1 时,若 $i=j$,则 $\boldsymbol{D}(v_i,v_j)=0$,根据式(4)可得邻接矩阵 $\boldsymbol{A}$ 的对角线均为 1。此时,邻接矩阵 $\boldsymbol{A}$ 中的所有对角元素 $\boldsymbol{A}^l(v_i,v_i)$ 均为极大对角元,即图 G 中的所有顶点 $v_i\in\boldsymbol{V}$ 均为连通中心,即每个点都为簇中心。随着随机游走步长 l 的递增,簇的数量逐渐收敛,直至达到收敛条件 k 停止该过程,识别出对应数量的簇中心。在本文提出的算法中,将簇中心的数量 k 作为该过程的收敛参数,由用户输入。算法 1 总结了基于随机游走识别簇中心的过程。

算法 1 簇中心识别算法

输入:邻接矩阵 **A**,簇数 K

输出:簇中心集合 C,截止邻接矩阵 $\mathbf{A}^l$

```
1. 初始化随机游走步长为 l=1;
2. 初始化簇中心集合 C_k^l=∅
3. while true
4.     A←A×A; /* 步长为 2 的随机游走 */
5.     if A^l(v_i,v_j)≤A^l(v_i,v_i)
6.         C←v_i; /* 识别簇中心 */
7.     end if
8.     l=l+1; /* 递增随机游走步长 */
9.     if len(C)=k
10.        break; /* 达到收敛条件 */
```

11. end if
12. end while

3.4 基于顶点倾向性的簇编号传播

初始状态下，图 $\boldsymbol{G}$ 中所有的顶点 $v_i \in \boldsymbol{V}$ 都是未标记的。假设簇的数量收敛到 k 时，随机游走步长递增至 l 停止，邻接矩阵 $\boldsymbol{A}$ 的收敛状态是 A^l，簇中心集合为 $C=\{c_1, c_2, \cdots, c_k\}$，且 $\boldsymbol{C} \subseteq \boldsymbol{V}$。经过该过程，簇中心被标记。聚类结果的质量不仅取决于簇中心的选择，非簇中心的分配过程也会影响聚类效果。在无向图 $\boldsymbol{G}$ 中，通常利用顶点度去衡量顶点的重要性，且认为度越大的顶点，其传播能力越强。换句话说，一个顶点会倾向于接受顶点度较高的邻居的编号。但本文提出的算法没有用顶点度去衡量顶点的重要性，而是用顶点自身的连通中心性去度量顶点重要性。类似地，一个顶点也会倾向于接受连通中心性较高的邻居顶点的编号。因此，本文基于顶点的连通中心性计算顶点的倾向性，从而获得顶点间的依赖关系，并执行簇编号的传播过程，直至完成聚类。

在截止矩阵 $\boldsymbol{A}^l$ 中，$Diag(\boldsymbol{A}^l)=\{\boldsymbol{A}^l(v_i, v_i) \mid_{i=1}^{N}\}$ 包含每个顶点的连通中心性。根据连通中心性 $\boldsymbol{A}^l(v_i, v_i)$ 的大小，对所有顶点 $v_i \in \boldsymbol{V}$ 进行降序排列，得到集合 $DCC^l = \{ranking, i, \boldsymbol{A}^l(v_i, v_i)\}_{\text{ranking}=1}^{N}$。

该集合包含的信息有顶点的连通中心性排名 ranking、顶点在原始数据中的索引 i，以及排名为 rankin 的顶点 v_i 的连通中心性 $\boldsymbol{A}^l(v_i, v_i)$。

定义 5（高连通近邻） 按照顶点的连通中心性排名 ranking，遍历所有图 $\boldsymbol{G}$ 中的所有顶点 $v_i \in \boldsymbol{V}$，可以找到比当前顶点 v_i 的连通中心性高的所有顶点集合 $Higher\_\boldsymbol{A}^l(v_i) \subseteq V$，称 $Higher\_\boldsymbol{A}^l(v_i)$ 集合中的所有顶点均为顶点 V_i 的高连通邻居。当顶点 $v_i \in \boldsymbol{C}$ 时，其高连通近邻集合为空，即当顶点 V_i 为簇中心时，$Higher\_\boldsymbol{A}^l(v_i)=\phi$。另一方面，对于非簇中心 $v_i \in V \backslash \boldsymbol{C}$，其高连通近邻为：

$$Higher\_\boldsymbol{A}^l(v_i)=\{v_j \mid v_j \in V \text{ and } i \neq j\ \boldsymbol{A}^l(v_j, v_j) > \boldsymbol{A}^l(v_i, v_i)\} \tag{6}$$

本文规定顶点执行单一簇编号的传播，即顶点只能选择其高连通近邻顶点的一个编号。因此，在已知簇中心、各顶点的连通中心性及各顶点高连通近邻的情况下，对于每一个非中心点 $v_i \in V \backslash \boldsymbol{C}$，需要计算它与所有的高连通邻居的相对倾向性。

定义 6（顶点倾向性） 非中心点 $v_i \in V \backslash \boldsymbol{C}$ 接受其高连通近邻 $v_j \in Higher\_\boldsymbol{A}^l(v_i)$ 编号的倾向性为：

$$RT^l(v_i, v_j)=\frac{\boldsymbol{A}^l(v_i, v_i)}{\boldsymbol{A}^l(v_j, v_j)} \tag{7}$$

通过比较非中心点 $v_i \in V \backslash \boldsymbol{C}$ 与其所有高连通近邻的倾向性，可以获得其最依赖的邻居顶点 v_{nearest}，具体定义为：

$$v_{\text{nearest}}=\{v_j \mid v_j \in Higher\_\boldsymbol{A}^l(v_i, v_i), RT^l(v_i, v_j)=\max \| RT^l(v_i, v_j) \| \} \tag{8}$$

则 $v_i \in V \backslash \boldsymbol{C}$ 最倾向于接受其邻居顶点 v_{nearest} 的编号，顶点的依赖关系为 $Label(v_i)=Label(v_{\text{nearest}})$。

按照 DCC^l 中的顶点顺序，可以获得所有非中心点之间的依赖关系。最后，根据顶点间的依赖关系，通过簇中心编号更新所有非中心顶点的编号，以达到聚类的目的。执行簇编号传播阶段的相关算法的流程如算法 2 所示。

算法 2 基于顶点倾向性的簇传播算法

输入：簇中心集合 C，截止邻接矩阵 $\mathbf{A}^l$

输出：预测标签 Label

1. $DCC^l \leftarrow$ sort(Diag($\mathbf{A}^l$))；/* 按降序排列顶点的连通中心性；
2. for ranking, i in DCC^l
3. if i in C
4. Higher_$\mathbf{A}^l(v_i)=\phi$；
5. continue
6. end if
7. Higher_$\mathbf{A}^l(v_i) \leftarrow DCC^l$[:ranking]；
/* 非中心顶点的高连通近邻 */
8. $v_{\text{nearest}} \leftarrow v_j \mid \max(\text{sort}(RT^l(v_i, v_j)))$；
/* 寻找依赖顶点 */
9. $\text{Label}(v_i)=\text{Label}(v_{\text{nearest}})$；
10. LD$\leftarrow$[$\text{Label}(v_{\text{nearest}})$]；/* 获得顶点依赖集合 */
11. end for

3.5 时间复杂度分析

对于数据集 $X \in R^{d \times N}$，假设簇中心数为 k，随机游走的步长为 l 时停止递增。本文提出的算法的时间复杂度主要包括以下部分：1）计算所有数据对象之间的距离，构造距离矩阵 $\boldsymbol{D}$，时间复杂度为 $O(N^2)$；2）计算所有顶点的 K 距离密度 $\rho(v_i)$，并构造邻接矩阵 $\boldsymbol{A}$ 耗时 $O(K \cdot N^2)$，其中 K 为最近邻居数；3）基于递增步长的随机游走过程是通过邻接矩阵 $\boldsymbol{A}$ 累乘 l 次获取截止矩阵 $\boldsymbol{A}^l$ 实现的，因此通过矩阵的快速幂运算需要 $O(\log N)$ 来计算所有顶点的连通中心性并识别簇中心；4）根据簇中心的连通中心性，计算顶点间的依赖关系的复杂度为 $O\left(\frac{N(N-1)}{2}\right)$；5）按照顶点间的依赖关系，基于簇中心编号更新非簇中心编号，此过程的时间复杂度为 $O(N^2)$。

4 实验分析

本章进行了大量的实验以测试本文提出算法的性能。首先，详细介绍了本文实验的实验设置，包括实验数据集、对比模型和用于评估聚类结果的指标。其次，给出了各对比算法在各数据集上的聚类结果并进行了讨论。然后，分析模型中的参数最近邻数量 K 对聚类结果质量的影响。最后分析了该算法的复杂度。本文中所有的实验都是采用 Python 3.8 实现的，运行在一台装有 8 核英特尔 i7 处理器 16 GB RAM 的电脑上，操作系统为 Windows 10。

4.1 实验设置

在进行实验之前，本文首先确定了相关实验设置，包括实验所涉及的数据集和用于评价聚类结果质量的指标。同时，为了测试本文提出算法的有效性，本文将所提方法与其他聚类方法加以对比。

4.1.1 实验数据

为了评估所提算法的聚类性能，我们使用两种不同类型的数据集进行实验，即真实数据集和合成数据集。本文实验共涉及 26 个不同规模和不同维度的数据集，包含 16 个合成数据集和 10 个真实世界的数据集。表 1 和表 2 分别列出了合成数据集和真实数据集的详细统计信息，包含数据类型、

数据规模、特征维度以及真实类别数量。

表1中的1－12号数据集为公开聚类数据集[1)]。其中，Jain数据集包括两个非高斯分布的簇，Aggregation数据集是由7类不同的簇组成，包含高斯簇和非高斯簇，R15具有15个类似的2维高斯分布的簇。4－7号数据集是具有相同规模和类别但不同维度的一系列数据集。相反地，A1，A2和A3是具有相同维度但规模和类别不同的一系列数据集。S1和S2在空间数据方面具有不同的复杂性，存在不同程度的重叠性。除此之外的13－16号是FCPS数据集[21]，Hepta包含了7个独立的簇且其中一个簇的密度比其他簇大得多，Chainlink是不可分超平面的规范数据集，TwoDiamonds内的两个菱形簇是紧密接触的，Atom是包含类似于原子内核和外壳的3维数据集。

表1 合成数据集

Table 1 Synthetic datasets

序号	数据集	规模	维度	类别
1	Jain	373	2	2
2	Aggregation	788	2	7
3	R15	600	2	15
4	Dim032	1024	32	16
5	Dim064	1024	64	16
6	Dim128	1024	128	16
7	Dim256	1024	256	16
8	A1	3000	2	20
9	A2	5250	2	35
10	A3	7500	2	50
11	S1	5000	2	15
12	S2	5000	2	15
13	Hepta	212	3	7
14	Chainlink	1000	3	2
15	TwoDiamonds	800	2	2
16	Atom	800	3	2

表2中，Colon_Cance和Prostate_Cancer数据集是基因学领域的两个高维数据集合[2)]，分别表示直肠癌和前列腺癌的基因组数据。除了上述两个数据集，其余10个实验数据集来源于UCI机器学习数据库。这些数据集具有不同的作用，其中包含3个类别的数据集Seeds，Iris，Wine分别用来判别小麦、葡萄酒和虹膜植物，Thyroid是由7个类构成的甲状腺数据集合，Penbased是一个数字数据集。

表2 真实数据集

Table 2 Real-world datasets

序号	数据集	规模	维度	类别
1	Colon_Cancer	62	2000	2
2	Prostate_Cancer	102	6033	2
3	Seeds	210	7	3
4	Thyroid	215	5	3
5	Wine	178	13	3
6	Iris	150	4	3
7	Dermatology	366	33	6
8	Libras	360	90	15
9	Ionosphere	351	34	2
10	Penbased	10992	16	10

4.1.2 评价指标和对比算法

评价指标用于评估聚类结果的优劣。根据是否需要借助外部信息来评估聚类结果，评价指标可以被分为两类：外部评价指标和内部评价指标。外部评估标准要求了解数据集的相关知识，特别是数据集中簇的基本结构和数量，被称为测试方法的黄金标准。外部评价的基本思想是将聚类的结果与数据集的预定义结果相匹配，利用外部数据来测试算法的有效性。常用的外部评价指标包含准确率(Accuracy，ACC)、标准互信息(Normalized Mutual Information，NMI)等。其中，ACC用于比较聚类的预测标签与数据提供的真实标签，其取值范围为[0，1]，值越接近1，表明聚类结果的质量越高。而NMI则是用于衡量聚类结果与真实标签分布的一致性。与ACC不同的是，NMI的值不会受到类别标签排列的影响。内部评估标准则是通过聚类算法划分数据集后的簇内紧凑性和簇间分离性，来测试算法的有效性。然而，基于内部评估指标，两种算法的评价指标对应的值不相等，不能绝对判断哪种算法更好。因此，本文选择外部评价指标中的两个常用指标ACC和NMI来评估聚类结果的质量。

另一方面，为了体现本文算法的性能，我们将本文算法分别与几种现有算法进行了比较，包括LLKM，CBKM，DBSCAN，SPRG，DenMune。其中，CBKM和SPRG算法需要输入簇的数量作为参数完成聚类，本文向它们提供簇数目的真实值。LLKM不需要事先给定簇的数量，而是通过LastLeap方法估计簇数目的改进K-Means方法。DBSCAN算法通过启发式的方法确定原始算法所涉及的两个参数$MinPts$和Eps。DenMune虽然不需要给定簇的数量，但仍需给定相互最近邻的数量K作为参数完成聚类，我们为所有数据集选择了一个固定的K值，即$K=37$。

4.2 实验结果及评价指标

为了验证ASDSC算法的性能，本文首先对比了ASDSC算法与LLKM、CNKM等5种对比算法在16个合成数据集上关于ACC和NMI的评价结果，并对本文算法在具有最佳聚类结果时的最近邻居数量K给出具体建议，如表3所列。其中，加粗字体表示各算法在对应评价指标下的最优结果，并通过图3展示了本文提出的ASDSC算法在Jain等数据集上最佳聚类结果的可视化。

上述实验验证了ASDSC算法在合成数据集上的适用性。为了进一步分析在真实世界中的聚类表现，我们在10个真实数据集上进行了测试，通过比较在ACC和NMI方面的聚类质量，分析各算法的聚类性能表现。其中，ASDSC算法在各数据集上ACC和NMI方面的平均聚类质量分别达到0.82和0.60，总体优于其他算法的平均性能。从表4可以观察到，相比其他算法，本文算法在Colon_Cancer和Prostate_Cancer这类超高维数据集上表现突出，原因是它不仅考虑了丰富的数据特征，还考虑了结构连通性。对于存在两个线性不可分簇的Iris数据集，本文算法结合其结构特征，可以识别紧密接触但来自不同簇的样本，因此本文方法在

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

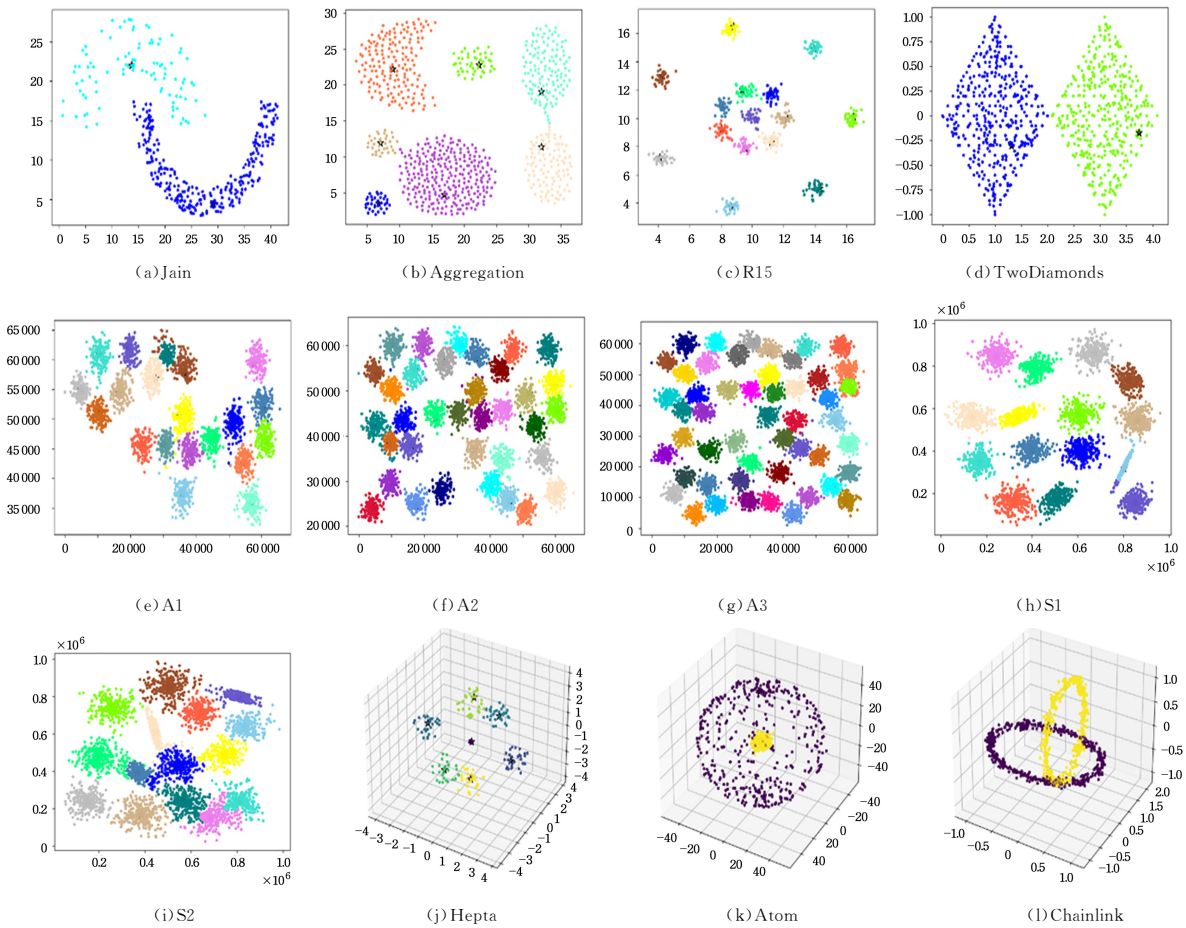
2) https://www.stat.cmu.edu/~jiashun/Resear-ch/software/GenomicsData/

ACC 和 NMI 方面均优于其他算法。但在数据集 Wine 上，由于它所有的属性变量都是连续的，这导致其在分布结构上的连通性变差，进一步影响本文算法的最终聚类性能。经过对比发现，在实验所涉及的 10 个真实数据集中，ASDSC 算法在 Colon_Cancer 等 7 个数据集上均获得了最佳 ACC，在其他数据集上也具有相对更好的聚类性能。这些实验结果验证了本文算法在不同类型的真实数据集上聚类性能的优越性。

表 3 6 种算法在真实数据集上的聚类结果

Table 3 Clustering results of 6 methods on real-world datasets

数据集	评价指标	LLKM	CNKM	DBSCAN	SPRG	DenMune	ASDSC(K)
Colon_Cancer	ACC	0.35	0.29	0.46	0.63	0.60	**0.69**(K=1)
	NMI	0.25	0.23	0.35	0.19	0.48	**0.68**(K=1)
Prostate_Cancer	ACC	0.24	0.25	0.55	0.51	0.57	**0.59**(K=4)
	NMI	0.08	0.13	0.28	0.34	0.13	**0.34**(K=4)
Seeds	ACC	**0.90**	0.89	0.81	0.88	0.88	0.89(K=25)
	NMI	**0.72**	0.69	0.7	0.68	0.68	0.70(K=25)
Thyroid	ACC	0.54	0.65	0.74	0.8	0.80	**0.82**(K=12)
	NMI	0.4	0.3	0.46	0.65	0.56	**0.53**(K=12)
Wine	ACC	0.7	0.73	0.67	**0.85**	0.71	0.81(K=1)
	NMI	0.88	0.5	0.45	**0.77**	0.57	0.52(K=1)
Iris	ACC	0.81	0.89	0.75	0.88	0.8	**0.91**(K=15)
	NMI	0.72	0.78	0.64	0.76	0.79	**0.80**(K=15)
Dermatology	ACC	0.86	0.87	0.61	0.82	**0.90**	0.87(K=21)
	NMI	0.76	0.8	0.57	0.78	**0.92**	0.81(K=21)
Libras	ACC	0.35	0.84	0.33	0.78	0.62	**0.86**(K=6)
	NMI	0.5	0.52	0.27	0.42	**0.57**	0.56(K=6)
Ionosphere	ACC	0.58	0.65	0.83	0.54	0.51	**0.92**(K=6)
	NMI	0.31	0.43	0.58	0.31	0.26	**0.59**(K=6)
Penbased	ACC	0.66	0.63	0.58	0.45	0.67	**0.92**(K=16)
	NMI	0.65	0.64	0.47	0.28	0.59	**0.77**(K=16)



(a) Jain (b) Aggregation (c) R15 (d) TwoDiamonds

(e) A1 (f) A2 (g) A3 (h) S1

(i) S2 (j) Hepta (k) Atom (l) Chainlink

图 3 ASDSC 算法在 12 个数据集上的可视化结果

Fig. 3 Visualization results of ASDSC on 12 datasets

4.3 模型参数分析

本文算法涉及两个参数，即最近邻数量 K 和真实类别数 k。真实类别数 k 作为所提算法的一个收敛条件，本文不再进行分析。另一方面，最近邻数量 K 决定着每个顶点的密度，即在一定程度上影响了顶点的结构相似性。相应地，后续构造邻居矩阵也会受到影响，而基于递增随机游走距离识别簇中心和簇编号传播的过程则是在上述邻接矩阵上完成的。因此，最近邻数量 K 的选取对聚类结果有着至关重要的影响。为了分析最近邻数量 K 对聚类结果的影响，针对表 1 中的每个合成数据集分别将 K 取值为 1 到 50 进行测试。用 ACC 来评估的关于不同值 K 下合成数据集的聚类结果如图 4 所示。

在图 4 中观察到，最近邻数量 K 对合成数据集 Dim256，Hepta，Chainlink 和 TwoDiamonds 的聚类结果几乎没有影响。对于数据集 Dim256，其属性的丰富性在一定程度上降低了结构对整体相似性的影响。而对于具有较少属性但均匀分布的 Hepta，Chainlink 和 TwoDiamonds，基本可以忽略 K 值对聚类结果的影响。除此之外，在 Aggregation，Dim032 和 Dim064 数据集上，聚类性能会随着 K 值产生一些较小的波动，ACC 始终保持在 0.8 及以上。对于一些失衡数据集，K 值对聚类结果的影响稍显复杂，过大或过小都会导致性能的损失。这里以 A1 数据集为例，随着 K 值的增加，聚类结果首先得到改善，然后经历一些波动，当 $K=21$ 时达到峰值。

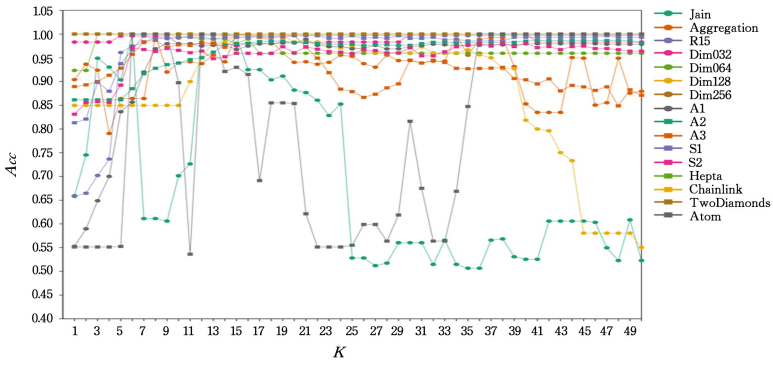


图 4 ASDSC 算法在合成数据集上的聚类质量随最近邻参数 K 变化的图

Fig. 4 Clustering quality of ASDSC on synthetic datasets varies with the nearest neighbor parameter K

结束语 本文提出了一种基于属性相似性和分布结构连通性的聚类算法，旨在生成多个非重叠的具有高度连通性且同质化属性的簇。该方法同时考虑到数据对象的属性和分布结构信息，有效地捕捉了稀疏簇中遥远对象之间的高相似性和来自不同簇的相邻对象之间的低相关性。通过随机游走差异化不同步长和不同路径下对象的连通性，来识别全局聚类中心，并为后续非中心的分配做铺垫。此外，根据数据的连通性获得顶点间的依赖关系，提升了簇内连通性，并降低了簇间的依赖。实验结果表明，本文方法获得了相对较好的聚类性能。但当数据集的规模较大时，该算法的时间效率有待提升。未来的工作包括解决参数自适应和提高计算性能这两个方面，以便进一步提高该算法的可扩展性。

参 考 文 献

[1] ZHANG Y，XIA Y Q，LIU Y，et al. Clustering sentences with density peaks for multi-document summarization[C]// Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics：Human Language Technologies. DC：Association for Computational Linguistics，2015：1262-1267.

[2] LAHIRI S，MISRA S，SANJOY KUMAR S，et al. Clustering-Based Semi-supervised Technique for Credit Card Fraud Detection[C]// Proceedings of Computational Intelligence in Communications and Business Analytics. Springer，2022：260-268.

[3] LI C W，CHEN H M，LI T R，et al. A stable community detection approach for complex network based on density peak clustering and label propagation[J]. Applied Intelligence，2022，52：1188-1208.

[4] INUWA-DUTSE I，LIPTROTT M，KORKONTZELOS I. A multilevel clustering technique for community detection[J]. Neurocomputing，2021，441：64-78.

[5] QU J L，CUI Y H. Gene set analysis with graph-embedded kernel association test[J]. Bioinformatics，2022，38(6)：1560-1567.

[6] GAO C M，ZHAO Y，WU R Z，et al. Semantic trajectory compression via multi-resolution synchronization-based clustering[J]. Knowledge-Based Systems，2019，174：177-193.

[7] ZHAO K，CHONG P，QIANG C. Twin learning for similarity and clustering：a unified kernel approach[C]// Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17). USA，2017：2080-2086.

[8] TIAN X Y，XU D C，DU D L，et al. The spherical k-means++ algorithm via local search scheme[J]. Journal of Combinatorial Optimization，2022，44：2375-2394.

[9] HANAFI N，SAADATFAR H. A fast DBSCAN algorithm for bigdata based on efficient density calculation[J]. Expert Systems with Applications，2022，203：117501.

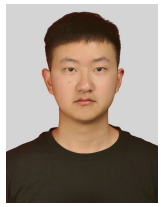
[10] KUMAR U，LEGENDRE C P，LEE J C，et al. On analyzing GNSS displacement field variability of Taiwan：Hierarchical Agglomerative Clustering based on Dynamic Time Warping technique[J]. Computers & Geosciences，2022，169：105243.

[11] LI H M，YE X C，IMAKURA A，et al. Divide-and-conquer based Large-Scale Spectral Clustering[J]. Neurocomputing，2022，501：664-678.

[12] FRÄNTI P，SIERANOJA S. How much can k-means be improved by using better initialization and repeats? [J]. Pattern Recognition，2019，93：95-112.

[13] GUPTA A，DATTA S，DAS S. Fast automatic estimation of the

number of clusters from the minimum inter-center distance for k-means clustering[J]. Pattern Recognition Letters, 2018, 116: 72-79.
[14] OSKOUEI A G, BALAFAR M A, MOTAMED C. FK-MAWCW: Categorical fuzzy *k*-modes clustering with automated attribute-weight and cluster-weight learning[J]. Chaos, Solitons & Fractals, 2021, 153(1): 111494.
[15] YANG Y M, LIU H M, GUAN Z Y, et al. CoHomo: A cluster-attribute correlation aware graph clustering framework[J]. Neurocomputing, 2020, 412: 327-338.
[16] SCHUBERT E, SANDER J, ESTER M, et al. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN [J]. ACM Transactions on Database Systems, 2017, 42(3): 19-21.
[17] ABBAS M, EL-ZOGHABI A, SHOUKRY A. DenMune: Density peak based clustering using mutual nearest neighbors[J]. Pattern Recognition, 2021, 109: 107589.
[18] ZHU X T, LOY C C, GONG S G. Constructing Robust Affinity Graphs for Spectral Clustering[C]// Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. OH: IEEE, 2014: 1450-1457.
[19] WANG Y, MA Y, HUANG H, et al. A split-merge clustering algorithm based on the k-nearest neighbor graph[J]. Information Systems, 2023, 111: 102124.
[20] CAI Y D, HUANG Z X, YIN J F. A new method to build the adaptive k-nearest neighbors similarity graph matrix for spectral clustering[J]. Neurocomputing, 2022, 493: 191-203.
[21] ULTSCH A, LÖTSCH J. The fundamental clustering and projection suite (fcps): a dataset collection to test the performance of clustering and data projection algorithms[J]. Data, 2020, 5(1): 13.



SUN Haowen, born in 1999, master, is a member of CCF (No. P9555G). His main research interests include machine learning and data mining.

DING Jiaman, born in 1974, professor, is a member of CCF (No. 77726M). His main research interests include data mining, cloud computing and machine learning.

（责任编辑：何杨）

CCF 福州换届选举，毛国君竞选连任主席

2024 年 6 月 8 日下午，CCF 福州会员活动中心（以下简称“CCF 福州”）换届选举大会在福州市闽侯县梅园酒店成功举行。CCF 副秘书长王新霞、CCF 福州委员及部分会员参加本次换届选举会议。会议由筹备组唐郑熠副教授主持。

CCF 福州主席毛国君教授首先回顾了 CCF 福州的发展历程，并详细介绍了过去两年的主要成绩和重要活动。他强调，CCF 福州在推动本地计算机技术创新、促进学术交流与合作方面取得了显著成绩，为广大会员提供了丰富的学术资源和实践机会。

随后，唐郑熠介绍了换届筹备组的组建背景和成员名单，详细讲述了 2024 年换届筹备的各项工作安排，并介绍了执委会候选人的具体情况和选举工作组成员。在换届选举过程中，唐郑熠详细说明了选举规则和流程，确保整个选举过程的公开、公正和透明。随后进行的主席、副主席、秘书长、执委、监督委员会主席和监委委员的选举，全部采用差额选举方式。各候选人依次上台，陈述自己对 CCF 文化的理解、过去工作的成果以及未来的工作设想。每位候选人的发言都充分展示了他们对 CCF 福州的热情和承诺及本人的工作设想。整个选举过程紧张有序，既体现了竞争的激烈性，也洋溢着和谐温馨的氛围。

最终，毛国君教授再次当选为 CCF 福州主席。他表示将在未来的工作中再接再厉，带领新一届执委会为会员们开展更加丰富多彩的活动，提供更优质、更全面的服务。他强调 CCF 福州将继续发挥好交流与合作的桥梁作用，为福州高校与企业的深度合作搭建平台，同时积极拓展福州与全国其他地区分部的合作交流，推动 CCF 福州的多渠道发展，力争使福州成为全国计算机技术创新的重要基地。

最后，CCF 副秘书长王新霞对 CCF 福州过去几年的工作给予了充分肯定，并对新一届领导班子寄予期望。她强调，各会员活动中心作为 CCF 服务机构是 CCF 服务本地会员、持续发展的关键，希望在新领导班子的带领下，CCF 福州能够继续发扬优良传统，不断开拓创新，再创辉煌，为全国的计算机科学与技术发展做出更大贡献。

据 CCF 微信公众号