计算机科学
COMPUTER SCIENCE

基于节点局部相似性的两阶段密度峰值重叠社区发现方法

段小虎, 曹付元

引用本文

段小虎, 曹付元. 基于节点局部相似性的两阶段密度峰值重叠社区发现方法[J]. 计算机科学, 2022, 49(12): 170-177.

DUAN Xiao-hu, CAO Fu-yuan. Node Local Similarity Based Two-stage Density Peaks Algorithm for Overlapping Community Detection [J]. Computer Science, 2022, 49(12): 170-177.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.211000025

基于节点局部相似性的两阶段密度峰值重叠社区发现方法

段小虎[1]　曹付元[1,2]
1 山西大学计算机与信息技术学院　太原 030006
2 山西大学计算智能与中文信息处理教育部重点实验室　太原 030006
(duan.xiaohu@foxmail.com)

摘　要　为了有效地发现复杂网络中的重叠社区结构,引入了密度峰值聚类算法,但将此算法应用于社区发现还存在如何度量节点间距离、如何产生重叠划分结果等问题。为此提出了一种基于节点局部相似性的两阶段密度峰值重叠社区发现方法(Node Local Similarity Based Two-stage Density Peaks Algorithm for Overlapping Community Detection,LSDPC)。该方法结合大度节点有利指标和连接贡献度定义了一种新的节点局部相似性指标,首先通过节点局部相似性度量节点距离;然后通过节点的局部密度和最小距离计算节点中心值,利用切比雪夫不等式筛选出社区中心节点;最后经过初次划分与重叠划分两阶段得到最终的重叠社区划分结果。在真实网络数据集与合成网络数据集上的实验结果表明,所提算法可以有效发现重叠社区结构,且结果优于其他对比算法。

关键词:重叠社区发现;密度峰值;节点相似性;k近邻;隶属度
中图法分类号　TP391

Node Local Similarity Based Two-stage Density Peaks Algorithm for Overlapping Community Detection

DUAN Xiao-hu[1] and CAO Fu-yuan[1,2]
1 School of Computer and Information Technology,Shanxi University,Taiyuan 030006,China
2 Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education,Shanxi University,Taiyuan 030006,China

Abstract　In order to detect overlapping community structures in complex networks,the idea of density peaks clustering algorithm is introduced. However,applying the density peaks clustering algorithm to community detection still has problems such as how to measure the distance between nodes and how to generate overlapping partition results. Therefore,a node local similarity based two-stage density peaks algorithm for overlapping community detection is proposed (LSDPC). By combining hub promoted index and connection contribution degree,a new node local similarity index is defined,and the node distance is measured with the node local similarity. Then the local density and minimum distance of nodes are used to calculate their center values and Chebyshev inequality is used to select communities' center nodes. The overlapping communities are obtained through initial assignment and overlapping assignment. Experimental results on real network datasets and synthetic network datasets show that the proposed algorithm can effectively detect overlapping community structure,and the results are better than that of other algorithms.

Keywords　Overlapping community detection,Density peaks,Node similarity,k-nearest neighbors,Degree of membership

1　引言

随着信息技术的发展,许多复杂网络不断被发现,如蛋白质网络、社交网络[1]等,通过研究其结构和特点,人们可以更加深入地了解复杂网络系统。在复杂网络中,节点会表现出一种社区化的结构,整个网络可以被看作是由多个社区组成,同一社区内部的节点联系紧密,而不同社区间的节点连接稀疏[2]。社区发现旨在研究并识别复杂网络中的社区组织,从而更深入地了解复杂网络中节点间的关系和群体特征。在传统的社区发现算法中,网络通常会被划分为若干个互斥的社区,每个节点仅属于一个社区。重叠社区发现算法可以有效识别社区中的重叠结构,一个节点可以同时被划分到多个社区中,例如一个学生可以参加多个兴趣社团,一个科学家可能涉足多个研究领域等,因此,对重叠社区发现

到稿日期:2021-10-08　返修日期:2021-11-04
基金项目:国家自然科学基金(61976128);山西省应用基础研究计划项目(201901D111035)
This work was supported by the National Natural Science Foundation of China(61976128) and Applied Basic Research Program of Shanxi Province(201901D111035).
通信作者:曹付元(cfy@sxu.edu.cn)

的研究具有重要现实意义[3]。

密度峰值聚类算法[4](Density Peaks Clustering Algorithm,DPC)是由 Rodriguez 等在 2014 年提出,DPC 算法快速、高效,能够识别任意形状的簇。该算法基于一个假设:簇中心节点的密度大于周围邻居节点的密度,同时,簇中心节点之间的距离相对较远。在复杂网络中,社区内部连接紧密,社区间连接稀疏,社区结构不规则,因此 DPC 算法的思想很适合用于复杂网络的社区发现,但 DPC 算法需要节点间的距离矩阵作为输入,无法直接应用于复杂网络数据,同时该算法需要通过决策图来人工选择社区中心且不能发现重叠社区。

对此,本文提出了一种基于节点局部相似性的两阶段密度峰值重叠社区发现算法。首先结合大度节点有利指标[5]与连接贡献度提出了一种新的节点局部相似性度量指标,将原始网络数据转化为节点距离矩阵作为密度峰值算法的输入,所提出的节点局部相似性指标能够更好地衡量节点间的联系;然后利用节点的 k 近邻信息计算节点的局部密度,通过节点的局部距离与最小距离计算节点中心值,利用切比雪夫不等式[6]选择社区中心点,避免了人工决策图的繁琐;最后对网络依次进行初次划分和重叠划分,得到节点对各个社区的隶属度向量,通过转化得到重叠社区划分结果。在真实网络数据集与合成数据集上与已有算法进行了对比实验,证明了本文算法的有效性。

2 相关工作

2.1 重叠社区发现

在重叠社区结构中,一个节点可能隶属于多个社区,社区之间存在重叠现象。传统的重叠社区发现算法可以分为基于派系过滤[7]的算法、基于标签传播[8-10]的算法、基于局部扩展[11-13]的算法和基于边划分[14-16]的算法等。

Palla 等[7]提出的 CPM 算法是一种派系过滤算法,该算法通过合并相邻 k 完全子图来发现重叠社区,不适用于稀疏网络,且时间复杂度较高。Gregory[8]提出的 COPRA 算法将标签传播算法扩展到重叠社区发现中,节点根据其所有邻居节点的社区分布对自身标签进行更新。Xie 等[9]提出的 SLPA 算法选择一个节点作为 listener 节点,选择其邻居节点作为 speaker 节点进行标签传播,该算法记录了节点在每一次迭代的标签记录。Liu 等[10]针对标签传播结果不稳定的问题提出了 LPPB 算法,该算法改进了网络的传播特性,根据节点影响力来确定更新顺序,同时还综合计算了节点的属性特征和历史标签记录对传播的影响。Lancichinetti 等[11]提出了一种局部扩展算法,该算法的思想是:随机选择初始种子节点,通过社区自适应函数对种子社区进行进一步扩展。Yu 等[12]提出的 i-SEOCD 算法利用节点影响力来选取种子节点,对自适应函数进行了优化,加强了社区扩展的稳定性。Coscia 等[13]提出的 DEMON 算法对每个节点抽取 EgoMinusEgo 子图,在子图上利用标签传播算法进行局部划分,再对社区进行合并操作。Ahn 等[14]提出了一种边划分算法,其思想是利用 Jaccard 相似度来衡量边之间的联系,再对所有边进行凝聚层次聚类,对得到的层次树进行最大化密度分割,从而得到最优边社区划分结果。Pan 等[15]提出了一种基于局部边社区的挖掘算法,该算法利用边聚类系数对网络中的边进行排序,通过边的自适应函数对种子边进行局部扩展,直到所有边都被分配到社区中为止。

2.2 密度峰值聚类算法

密度峰值聚类算法是一种基于密度的聚类算法,其核心思想是簇中心节点的局部密度大于普通节点,且簇中心之间的距离相距较远。密度峰值聚类算法定义了节点的局部密度 ρ_i 与最小距离 δ_i。

定义 1(节点局部密度[4]) 节点局部密度 ρ_i 反映了节点与其周围节点的紧密程度,ρ_i 越大,表明节点更靠近簇中心区域。通常采用以下两种方法对节点的局部密度 ρ_i 进行定义。

(1)截断核密度方法,其定义如式(1)所示:

$$\rho_i=\sum_j \chi(d_{ij},d_c) \tag{1}$$

其中,d_{ij} 表示节点 v_i 与 v_j 之间的距离;d_c 表示全局截断距离;$\chi(d_{ij},d_c)$ 是一个指示函数,当 $d_{ij}<d_c$ 时,其值为 1,否则为 0。在这种方法中,节点 v_i 的局部密度为 v_i 在全局截断距离范围内的邻近节点个数。

(2)高斯核密度方法,该方法考虑了节点 v_i 与其余所有节点的距离,其定义如式(2)所示:

$$\rho_i=\sum_j \mathrm{e}^{-\left(\frac{d_{ij}}{d_c}\right)^2} \tag{2}$$

定义 2(节点最小距离[4]) 节点最小距离 δ_i 是针对簇中心点之间的距离较远这一假设而提出的,节点 v_i 的最小距离为局部密度高于 v_i 的节点中与 v_i 的最近距离,若 v_i 是局部密度最大的节点,则其最小距离为到所有节点距离的最大值。最小距离 δ_i 的具体定义如式(3)所示:

$$\delta_i=\begin{cases}\min\limits_{j:\rho_j>\rho_i}(d_{ij}), & \{j\mid\rho_j>\rho_i\}\neq\varnothing\\ \max\limits_{j}(d_{ij}), & \text{otherwise}\end{cases} \tag{3}$$

定义 3(追随节点) 所有比节点 v_i 局部密度更大的节点中与节点 v_i 距离最近的节点被称为节点 v_i 的追随节点,用 $leader_i$ 来表示。

由上面的定义可以看出,对于非局部密度最大节点,节点 v_i 与其追随节点之间的距离即节点 v_i 的最小距离 δ_i,局部密度最大的节点不存在追随节点。

在得到节点的局部密度 ρ_i 与最小距离 δ_i 后,画出二维决策图,然后人工选择局部密度与最小距离均较大的节点作为簇中心节点,最后将节点分配给其追随节点所在的簇。

一些学者通过改进密度峰值聚类算法将其应用于社区发现[16-21]。Huang 等[17]结合了 Jaccard 距离与节点间的最短路径来度量节点距离,再利用密度峰值聚类算法进行聚类,该算法计算距离的时间复杂度较高且只能发现非重叠社区。Bai 等[19]提出的 OCDDP 算法通过一个节点在有限步内到达另一个节点的路径数和节点的度来度量节点距离,当步数较大时,计算距离的时间复杂度非常高,同时,该算法仍需要通过决策图人工地选取社区中心。Xu 等[20]提出的 EADP 算法利用线性拟合的方法自动选择社区中心点,但当网络结构较为复杂时,线性拟合方法无法正确识别社区中心节点,从而直接影响重叠社区的划分结果。

2.3 相关概念与定义

设 $G=(V,E)$ 是一个无向无权图表示的复杂网络，$V=\{v_1,v_2,\cdots,v_n\}$ 表示网络中的所有节点集合，$E=\{e_1,e_2,\cdots,e_m\}$ 表示网络中的所有边集合，$\boldsymbol{A}_{n\times n}$ 是网络的邻接矩阵，邻接矩阵元素 $a_{ij}=1$ 表示节点 v_i 与 v_j 之间存在连边，$a_{ij}=0$ 表示节点 v_i 与 v_j 之间不存在连边。

定义 4(邻居节点[22]) 对于图 G 中的节点 v_i，与其直接相连的节点为节点 v_i 的邻居，节点 v_i 的邻居集合可表示为：

$$nei\,b_i=\{v_j\mid\langle v_i,v_j\rangle\in E\} \tag{4}$$

定义 5(节点的度[22]) 对于图 G 中的节点 v_i，v_i 的度是与节点 v_i 相连的节点个数，因此，节点 v_i 的度可表示为：

$$d_i=|nei\,b_i| \tag{5}$$

定义 6(共同邻居[23]) 对于图 G 中的节点 v_i 与节点 v_j，两节点之间的共同邻居集合可表示为：

$$com_{ij}=\{v_c\mid v_c\in neib_i\cap nei\,b_j\} \tag{6}$$

定义 7(大度节点有利指标[5]) 对于图 G 中的节点 v_i 与节点 v_j，两节点间的相似性取决于两个节点的共同邻居数量及度较小的节点，其定义如式(7)所示：

$$hpi(v_i,v_j)=\frac{|com_{ij}|}{\min(d_i,d_j)} \tag{7}$$

大度节点有利指标中，度较大的节点更容易与其他节点有高相似性，大度节点有利指标在链路预测中得到了广泛的应用[24]。

定义 8(修正大度节点有利指标) 社区发现任务中，节点之间可能存在直接连边，本文对式(7)进行了修正，修正后的大度节点有利指标可以表示为：

$$hpi'(v_i,v_j)=\frac{|com_{ij}|+a_{ij}}{\min(d_i,d_j)} \tag{8}$$

3 LSDPC 算法

LSDPC 算法主要包括 3 个部分：1)节点间距离计算：利用定义的局部相似性指标来衡量节点间的相似性，将复杂网络数据转化为距离矩阵形式；2)社区中心点选择：通过节点的局部密度与最小距离来计算节点的中心值，再利用切比雪夫不等式选择社区中心节点；3)社区划分：经过初次划分与重叠划分两个阶段，计算节点对各社区的隶属度，经过转化得到最终的重叠社区划分。

3.1 节点间距离计算

对于图中的任意两个节点，可以将它们之间的连接关系划分为以下 4 种情况：1)无直接连边且无共同邻居；2)无直接连边但有共同邻居；3)有直接连边但无共同邻居；4)有直接连边且有共同邻居。考虑到直接连边和共同邻居对节点间的相似性贡献存在差异，本文提出了连接贡献度，融入了节点间直接连边和共同邻居对两个节点相似性的贡献程度。

定义 9(连接贡献度) 对于图 G 中的节点 v_i 与节点 v_j，定义两节点之间的连接贡献度为：

$$cr(v_i,v_j)=a_{ij}+\frac{1}{2}\sum_{v_c\in com_{ij}}\frac{2+\sum_{v_p\in com_{ij}}a_{cp}}{d_c} \tag{9}$$

在式(9)中，第一项表示直接连接贡献度，当节点 v_i 与 v_j 之间存在直接连边时，直接连接贡献度为 1，否则为 0；第二项为间接连接贡献度，分子中的 2 表示节点 v_i 和节点 v_j 与它们的共同邻居节点 v_c 的连边数量，$\sum_{v_p\in com_{ij}}a_{cp}$ 表示节点 v_c 与其余共同邻居节点之间的连边数目(网络中不存在自环，即 $a_{cc}=0$)。本文将共同邻居节点 v_c 的连边中与节点 v_i 和 v_j 相连的两条边，以及与其他共同邻居节点之间的连边称为有效边，节点 v_c 的其余连边称为无效边。从信息传递的角度来看，节点 v_i 与 v_j 的所有共同邻居节点的有效边构成的边集合可以使节点 $v_i(v_j)$ 的信息经由一个或多个共同邻居节点传递到 $v_j(v_i)$。我们注意到，当节点 v_i 与 v_j 的共同邻居节点 v_c 的无效边过多时，两个节点之间的信息传递会被削弱，因此需要对共同邻居节点的无效边施加惩罚，于是将节点 v_c 对节点 v_i 与 v_j 的间接连接贡献度定义为节点 v_c 的有效边与 v_c 的度的比值。考虑到直接连接贡献度大于间接连接贡献度，需要对间接连接贡献度设置衰减系数，本文将衰减系数设置为 $\frac{1}{2}$。

定义 10(节点局部相似性) 对于图 G 中的节点 v_i 与 v_j，两节点基于修正大度节点有利指标与连接贡献度的局部相似性可表示为：

$$sim_{ij}=hpi'(v_i,v_j)\times cr(v_i,v_j) \tag{10}$$

最后，将节点 v_i 与 v_j 之间的距离 d_{ij} 定义为它们局部相似性的倒数，当节点间相似性越小时，它们之间的距离越大。为了避免两节点局部相似性为零时距离无穷大的情况，为分母增加一个小的正数 φ，通常 φ 取值为 1，此时节点间距离的取值范围为(0,1]。

$$d_{ij}=\frac{1}{sim_{ij}+\varphi} \tag{11}$$

计算节点间距离的方法如算法 1 所示。

算法 1 距离矩阵计算

输入：复杂网络 G=(V,E)

输出：相似度矩阵 $\mathbf{Sim}_{n\times n}$，距离矩阵 $\mathbf{D}_{n\times n}$

1. 初始化：$\mathbf{Sim}_{n\times n}\leftarrow O$，$\mathbf{D}_{n\times n}\leftarrow I$；
2. for each node v_i in V do
3. NI$\leftarrow$neib$_i\cup\{$neib$_j\mid v_j\in$ neib$_i\}$；/* 与节点 v_i 直接相连或有共同邻居的节点集合 */
4. for each node v_l in NI do
5. if i<l then
6. 利用式(8)－式(10)计算节点 v_i 与 v_l 的局部相似性 sim_{il}；
7. $sim_{li}\leftarrow sim_{il}$；
8. 利用式(11)计算节点 v_i 与 v_l 的距离 d_{il}；
9. $d_{li}\leftarrow d_{il}$；
10. end if
11. end for
12. end for
13. return D

3.2 社区中心点选择

本文采用了基于 k 近邻的局部密度计算方法，使节点的局部密度仅与近邻节点的距离相关，同时避免了截断距离 d_c 的选取不当对社区中心选择结果造成影响。局部密度[25]的计算式为：

$$\rho_i=\sum_{v_j\in knn_i}e^{-d_{ij}} \tag{12}$$

其中，knn_i表示距离节点v_i最近的 k 个近邻节点，当节点v_i与其 k 近邻节点距离更近时，其局部密度会更大。

计算节点的局部密度后，通过式(3)对节点的最小距离和追随节点进行计算。为了保证局部密度与最小距离的同等重要性，需要对其进行归一化操作，使值域控制在[0，1]。用 $P=\{\rho_1,\rho_2,\cdots,p_n\}$ 表示所有节点的局部密度集合，$\Delta=\{\delta_1,\delta_2,\cdots,\delta_n\}$表示所有节点的最小距离集合，局部密度和最小距离的归一化计算可以分别用式(13)、式(14)表示：

$$\rho_i^s=\frac{\rho_i-\min(P)}{\max(P)-\min(P)} \tag{13}$$

$$\delta_i^s=\frac{\delta_i-\min(\Delta)}{\max(\Delta)-\min(\Delta)} \tag{14}$$

社区中心节点通常拥有较大的局部密度值和最小距离值，将节点的中心值定义为局部密度与最小距离的乘积，定义如式(15)所示。节点的中心值越大，节点就越有可能成为社区中心节点，社区中心节点的中心值远大于普通节点。

$$\gamma_i=\rho_i^s\times\delta_i^s \tag{15}$$

定义 11(切比雪夫不等式[6]) 假设随机变量 X 的期望 $E(X)$与方差 $D(X)$已知，对于任意正数 ε，切比雪夫不等式可以表示为：

$$P(|X-E(X)<\varepsilon|)\geqslant 1-\frac{D(X)}{\varepsilon^2} \tag{16}$$

通过令 $\varepsilon=\varepsilon\sqrt{D(X)}$，可以推导得到以下变形公式：

$$P(|X-E(X)|<\varepsilon\sqrt{D(X)})\geqslant 1-\frac{1}{\varepsilon^2} \tag{17}$$

由式(17)可以得出，对于任意一个数据集，其期望值的 ε 个标准差范围内的数据比例至少大于 $1-\frac{1}{\varepsilon^2}$。例如，当 $\varepsilon=2$ 时，至少有 75%的数据位于期望值的 2 个标准差范围内；当 $\varepsilon=3$ 时，至少有 88.9%的数据位于期望值的 3 个标准差范围内，随着 ε 的增大，只有极少数节点位于该范围外。切比雪夫不等式仅依靠随机变量 X 的期望 $E(X)$与方差 $D(X)$便可以对事件$|X-E(X)|<\varepsilon$ 做出概率估计，且不需要事先知晓随机变量 X 的分布，具有普遍应用意义。

对于节点中心值而言，社区中心节点的中心值远大于普通节点。假设所有节点的中心值的平均值为 μ，标准差为 σ，将节点的中心值阈值定义为 $\mu+\varepsilon\sigma$，当节点v_i的中心值$\gamma_i>\mu+\varepsilon\sigma$ 时，可将节点v_i视为社区中心点，其中标准差系数 ε 需人为给定。

社区中心点选择方法如算法 2 所示。

算法 2 社区中心点选择

输入：复杂网络 G=(V,E)，节点距离矩阵 **D**，近邻数 k，标准差系数 ε

输出：社区中心点集合 C

1. 初始化：C←∅；
2. for each node $v_i\in$ V do
3. 根据节点距离矩阵 **D** 得到节点v_i的 k 近邻节点集合 knn_i；
4. 利用式(12)计算节点v_i的局部密度ρ_i；
5. end for
6. for each node $v_i\in$ V do
7. 利用式(3)计算节点的最小距离δ_i和追随节点 $leader_i$；
8. end for
9. for each node$v_i\in$ V do
10. 利用式(13)、式(14)计算归一化后的局部密度ρ_i^s和最小距离δ_i^s；
11. 利用式(15)计算节点的中心值γ_i；
12. end for
13. 计算节点中心值的平均值 μ 与标准差 σ；
14. for each node $v_i\in$ V do
15. if $\gamma_i>\mu+\varepsilon\sigma$ then
16. C←C∪{v_i}；
17. end if
18. end for
19. return C

3.3 社区划分

算法采用两阶段划分方法发现重叠社区：在初次划分阶段，将非社区中心节点划分到其追随节点所在的社区；在重叠划分阶段，将 k 近邻同属一个社区的节点与社区中心点定义为核心域节点，其余节点定义为非核心域节点，最后从所有非核心域节点中筛选出重叠节点。

本文算法采用节点对各个社区的隶属度来衡量节点的社区归属。在初次划分后，得到的 K 个社区可以表示为 $Comm=\{c_1,c_2,\cdots,c_K\}$，节点$v_i$对各个社区的隶属度向量可以表示为 $\boldsymbol{Mem}_i=(\boldsymbol{mem}_{i1},\boldsymbol{mem}_{i2},\cdots,\boldsymbol{mem}_{iK})$。核心域节点对其所在社区的隶属度为 1，对其余社区的隶属度为 0。核心域节点v_i对社区c_l的隶属度可以表示为：

$$\boldsymbol{mem}_{il}=\begin{cases}1, & v_i\in c_l\\ 0, & v_i\notin c_l\end{cases} \tag{18}$$

对于非核心域节点，可以采用以下方法计算对各个社区的隶属度。将所有节点按照局部密度降序排列，然后依次对队列中的节点进行遍历：若当前遍历节点是核心域节点，则将节点加入到已划分节点集合 LS 中；若当前遍历节点是非核心域节点，则通过考虑该节点与 LS 集合中的节点的相似性，以及 LS 集合中节点对各社区的隶属度来计算当前节点的隶属度向量，结束后将当前节点加入 LS 集合中。非核心域节点v_i的隶属度向量的计算式为：

$$\boldsymbol{Mem}_i=\frac{\sum\limits_{j\in LS}sim_{ij}\boldsymbol{Mem}_j}{\sum\limits_{j\in LS}sim_{ij}} \tag{19}$$

得到所有非核心域节点的隶属度向量后，遍历其隶属度向量中的元素，当节点v_i对社区c_l的隶属度与节点v_i对所有社区的最大隶属度的比值大于 α 时，将节点v_i划分到社区c_l中，计算式如式(20)所示：

$$\boldsymbol{mem}_{il}=\begin{cases}1, & \dfrac{\boldsymbol{mem}_{il}}{\max(\boldsymbol{Mem}_i)}>\alpha\\ 0, & \text{otherwise}\end{cases} \tag{20}$$

社区划分方法如算法 3 所示。

算法 3 社区划分

输入：社区中心节点集合 C，节点局部相似性矩阵 $\mathbf{Sim}_{n\times n}$，节点局部密度降序队列 PD，追随节点字典 leader

输出：社区隶属矩阵 $\mathbf{Mem}_{n\times K}$

1. 初始化：$\mathbf{Mem}_{n\times K}$←O，LS←∅，NC←∅；/* LS 为已划分节点集合，NC 为非核心域节点集合 */
2. for each node v_i in PD do

```
3.    if 节点v_i是社区中心点 then
4.      利用式(18)计算节点 v_i 对各社区的隶属度;
5.        else
6.      Mem_i ← Mem_leader_i ; /* 将节点 v_i 划分到其追随节点所在社区 */
7.    end if
8. end for
9. for each node v_i in PD do
10.   if 节点v_i是非核心域节点 then
11.     NC←NC∪{v_i};
12.     利用式(19)计算节点v_i的隶属度向量;
13.   end if
14.   LS←LS∪{v_i};
15. end for
16. for each node v_i in NC do
17.   利用式(20)更新节点v_i对社区 l 的隶属度;
18. end for
19. return Mem_n×K
```

3.4 时间复杂度分析

对于一个复杂网络 $G=(V,E)$，节点总数为 n，节点连边总数为 m，经由 LSDPC 算法划分后，得到 K 个社区。在计算节点间距离时，首先要计算与节点直接相连或有共同邻居的节点集合 NI，避免与网络中所有的节点计算距离。假设节点平均度为 $\bar{d}$，计算 NI 的时间复杂度为 $O(\bar{d}^2)$；假设两节点共同邻居个数为 c，则计算两个节点相似性的时间复杂度为 $O(c^2)$，因此计算距离矩阵的时间复杂度约为 $O(\bar{d}^2c^2n)$。在社区中心点选择过程中，计算所有节点的局部密度的时间复杂度为 $O(n^2)$，计算所有节点的最小距离时，要先对节点的局部密度进行降序排列，排序时间复杂度为 $O(n\log n)$，计算节点最小距离的时间复杂度为 $O\left(\frac{n^2-n}{2}\right)$，局部密度和最小距离归一化、节点中心值计算、中心点选择的时间复杂度均为 $O(n)$，因此社区中心点选择过程的总时间复杂度约为 $O(n^2)$；在社区划分过程中，第一阶段划分的时间复杂度为 $O(Kn)$，第二阶段划分的时间复杂度约为 $O(n^2)$，因此社区划分过程的总时间复杂度约为 $O(n^2)$。综上，LSDPC 算法的总时间复杂度约为 $O(\bar{d}^2c^2n+2n^2)$，其中 $c<\bar{d}\ll n$，$k\ll n$。

4 实验结果与分析

为了验证 LSDPC 算法的有效性，本文选取了 SLPA[9]，DCN[18]，OCCDP[19] 和 EADP[20] 这 4 种社区发现算法作为对比算法，分别在多个真实网络数据集以及合成网络数据集上进行了对比实验。算法采用 Python 编程语言编写，实验环境如下：处理器为 Intel Core i7-4790 3.60 GHz，内存大小 8 GB，操作系统为 64 位 Windows10。

4.1 实验数据集

本文选取了真实网络数据集与合成网络数据集进行实验，真实网络数据集是从 Newman[26] 和 Kunegis[27] 的网站上选取的 9 个经典的网络数据集，分别为空手道俱乐部网络(Karate)、海豚社会网络(Dolphins)、足球联赛网络(Football)、政治书籍网络(Polbooks)、《悲惨世界》人物共现网络(LesMis)、电子邮件通信网络(Email)、《冰与火之歌》角色网络(Asoiaf)、政治博客网络(Polblog)和科学家合作网络(Netscience)，具体如表 1 所列。

表 1 真实网络数据集

Table 1 Real network datasets

Dataset	n	m	$\bar{d}$
Karate	34	78	4.6
Dolphins	62	159	5.1
Football	115	613	10.7
Polbooks	105	441	8.4
LesMis	77	254	6.6
Email	1133	5451	9.6
Asoiaf	796	2823	7.1
Polblog	1224	19087	27.3
Netscience	1461	2742	4.8

表 1 中，n 表示节点数，m 表示边数，$\bar{d}$ 表示节点平均度。

合成数据集由 LFR Benchmark 基准网络生成工具[28-29]生成，生成工具可以根据不同需求，通过调整参数得到相应的复杂网络，同时会得到复杂网络的真实社区划分结果。LFR Benchmark 提供了 10 个参数，参数 N 表示网络中节点的数目，参数 k 表示节点的平均度，$maxk$ 表示节点的最大度，mu 为混合系数，mu 值越大，网络越复杂，社区结构越难挖掘，$t1$ 为节点度分布参数，$t2$ 为社区尺寸分布参数，$minc$ 表示最小社区中的节点数目，$maxc$ 表示最大社区中的节点数目，om 用于控制重叠节点最多所属社区的个数，on 用于控制重叠节点的数目。

为了测试这些参数对算法的影响，依次调整混合系数 $mu\in\{0.1,0.2,0.3,0.4\}$，重叠节点最多所属社区个数 $om\in\{2,3,4,5,6,7,8\}$，重叠节点数目 $on\in\{50,100\}$，网络中节点数目 $N\in\{1000,2000,3000\}$，最小社区尺寸和最大社区尺寸 $(minc,maxc)\in\{(50,100),(20,50)\}$，节点平均度和节点最大度 $(k,maxk)\in\{(20,50),(10,30)\}$，共生成 12 组合成数据集，每组合成数据集由 7 个子数据集构成，具体如表 2 所列。

表 2 LFR 网络数据集

Table 2 LFR network datasets

Network	N	k	$maxk$	$minc$	$maxc$	mu	om	on
LFR1	1000	20	50	50	100	0.1	2~8	50
LFR2	1000	20	50	50	100	0.2	2~8	50
LFR3	1000	20	50	50	100	0.3	2~8	50
LFR4	1000	20	50	50	100	0.4	2~8	50
LFR5	1000	20	50	50	100	0.1	2~8	100
LFR6	1000	20	50	50	100	0.2	2~8	100
LFR7	1000	20	50	50	100	0.3	2~8	100
LFR8	1000	20	50	50	100	0.4	2~8	100
LFR9	2000	20	50	50	100	0.1	2~8	100
LFR10	3000	20	50	50	100	0.1	2~8	100
LFR11	1000	20	50	20	50	0.1	2~8	100
LFR12	1000	10	30	50	100	0.1	2~8	100

4.2 评价指标

考虑到某些真实网络数据集并无真实社区结构作为对比，本文采用了重叠模块度 Q_{ov} 对各种算法在真实数据集上的划分进行评价；对于合成网络数据集，本文采用了重叠标准化互信息 ONMI 对实验结果进行评价。

(1)重叠模块度 Q_{ov}

传统的模块度评价指标对重叠社区发现并不适用，

Nicosia 等[30]提出了一种重叠模块度Q_{ov}，该指标考虑了节点对社区的归属系数，其定义如下：

$$Q_{ov}=\frac{1}{m}\sum_{c\in C}\sum_{i,j\in V}\left[F(\alpha_{ic},\alpha_{jc})a_{ij}-\frac{\beta^{\text{out}}_{l(i,j),c}d_i^{\text{out}}\beta^{\text{in}}_{l(i,j),c}d_j^{\text{in}}}{m}\right] \quad (21)$$

其中，$\beta^{\text{out}}_{l(i,j),c}=\sum_{v_j\in V}\frac{F(\alpha_{ic},\alpha_{jc})}{|V|}$，$\beta^{\text{in}}_{l(i,j),c}=\sum_{v_i\in V}\frac{F(\alpha_{ic},\alpha_{jc})}{|V|}$；$\alpha_{ic}$表示节点$v_i$对社区$c$的归属系数；$F(\alpha_{ic},\alpha_{jc})$为关于相关节点归属系数的函数，用于度量节点$v_i$到节点$v_j$的边对社区$c$的归属系数；$d_i^{\text{out}}$表示节点$v_i$的出度；$d_j^{\text{in}}$表示节点$v_j$的入度；$l(i,j)$表示节点$v_i$到$v_j$的连边；$Q_{ov}$模块度的值范围为$[0,1]$，$Q_{ov}$值越大，表示重叠社区划分的结果越好。

(2)重叠标准化互信息 *ONMI*[11]

假设对节点规模为n的复杂网络进行重叠社区划分的结果为$C_1=\{C_{11},C_{12},\cdots,C_{1|C_1|}\}$，其中$|C_1|$表示社区的数量。考虑到一个节点可能不只属于一个社区，节点v_i的隶属信息不能再使用某一个社区编号来表示，而需要使用一个长度为$|C_1|$的隶属向量来表示节点i对各个社区的隶属情况。若节点v_i属于第k个社区，隶属向量对应的第k个分量为1，否则为0。将隶属向量的第k个分量看作随机变量X_k，则有$P(X_k=1)=n_k/n$，$P(X_k=0)=1-n_k/n$，其中n_k表示第k个社区的节点数量。同样地，对于真实社区划分C_2，社区数量为$|C_2|$，节点v_i被划分到第l个社区可用随机变量Y_l表示。在给定Y_l的条件下，X_k的条件熵可以表示为：

$$H(X_k|Y_l)=H(X_k,Y_l)-H(Y_l) \quad (22)$$

X_k在Y上的条件熵可定义为：

$$H(X_k|Y)=\min_{l\in 1,2,\cdots,|C_2|}H(X_k|Y_l) \quad (23)$$

X在Y上的归一化条件熵可以表示为：

$$H(X|Y)=\frac{1}{|C_1|}\sum_{k=1}^{|C_1|}\frac{H(X_k|Y)}{H(X_k)} \quad (24)$$

利用同样的方法，可以得到Y在X上的归一化条件熵，最终得到的*ONMI*值为：

$$ONMI(X|Y)=1-\frac{H(X|Y)+H(Y|X)}{2} \quad (25)$$

*ONMI*的取值范围为0～1，*ONMI*越大，算法的划分结果与真实社区划分越接近。

4.3 实验与分析

(1)真实数据集

将本文算法与其余4种社区发现算法在真实数据集上进行实验，用重叠模块度Q_{ov}对社区划分结果进行评价，实验结果如表3所列。

表3 各算法在真实网络上的实验结果

Table 3 Experiment results of different algorithms on real networks

Dataset	LSDPC	SLPA	DCN	OCDDP	EADP
Karate	**0.754**	0.698	0.753	0.703	0.748
Dolphins	**0.780**	0.761	0.714	0.773	0.741
Football	0.747	0.699	0.704	0.720	**0.767**
Polbooks	**0.842**	0.829	0.823	0.839	0.834
LesMis	**0.781**	0.777	0.660	0.737	0.000
Email	**0.698**	0.631	0.415	0.642	0.461
Asoiaf	**0.716**	0.714	0.568	0.688	0.675
Polblog	0.798	0.799	0.531	**0.800**	0.794
Netscience	**0.979**	0.917	0.754	0.973	0.977

从真实数据集上的划分结果可以看出，DCN算法在规模较大的网络上的Q_{ov}值与其他算法相差较大，这是因为DCN算法在计算局部密度时仅考虑了节点的度，而未进一步挖掘节点的局部信息。在LesMis数据集上，由于EADP算法仅识别出了一个社区中心节点，故Q_{ov}值为0。本文提出的LSDPC算法在Karate，Dolphins，Polbooks，LesMis，Email，Asoiaf和Netscience这7个真实数据集上的Q_{ov}值均优于其他4种算法。在Football数据集上，LSDPC算法的Q_{ov}值小于EADP算法，但远大于其他3种算法。在Polblog数据集上，LSDPC算法的结果与SLPA，OCDDP算法较为接近，但大于EADP算法和DCN算法。从整体上来看，LSDPC算法的实验结果最优，OCDDP算法次之，而DCN算法的实验结果最差。

图1给出了LSDPC算法在Karate数据集上的可视化结果，网络被划分为以节点0和节点33为核心的两个社区，左边的红色节点表示第一个社区，右边的绿色节点表示第二个社区，蓝色节点9是重叠节点，该节点同时属于这两个社区。

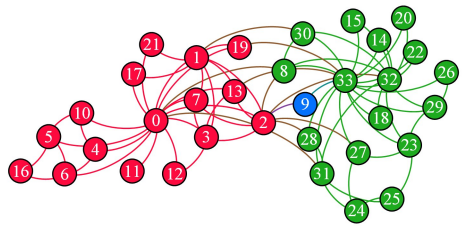

图1 LSDPC算法在Karate网络上的社区划分结果(电子版为彩色)

Fig. 1 Community partition result of LSDPC algorithm on Karate network

(2)合成数据集

在合成数据集上，将本文算法与其余4种算法进行了对比，用重叠标准化互信息ONMI对实验结果进行了评价。本文算法需要3个参数，分别为最近邻参数k、标准差系数ε和隶属度比值参数α。多次实验后的结果表明，最近邻参数k的取值在所有节点平均度$\bar{d}$附近时会得到更好的结果，故选取参数$k\in\{10,20,30\}$，标准差系数$\varepsilon\in\{2,3\}$，隶属度比值参数α取值范围为$(0,1)$。实验结果如图2所示，横坐标表示重叠节点最多归属社区数目*om*，纵坐标表示重叠标准化互信息*ONMI*值。需要注意的是，由于DCN算法不能发现重叠社区且实验结果与其他算法相差较大，不计入统计。

对图2进行分析可以看出，随着重叠节点所属的社区数目*om*的增加，几种算法的*ONMI*值出现了不同程度的下降。对比图2(a)－图2(d)或图2(e)－图2(h)，随着混合参数*mu*值的增加，社区结构趋于复杂，社区更难被发现，几种算法的*ONMI*值均减小；对比图2(a)和图2(e)、图2(b)和图2(f)、图2(c)和图2(g)或图2(d)和图2(h)，随着重叠节点数目*on*的增加，几种算法的*ONMI*值也均有下降；对比图2(e)、图2(i)和图2(j)发现，随着节点数目N的增加，所有算法的*ONMI*值均有增加；对比图2(e)和图2(k)发现，社区尺寸减小后，会得到更多的社区划分，除EADP算法的*ONMI*值下降外，其余算法的*ONMI*值均有提升；对比图2(e)和图2(l)

发现，降低节点的平均度与最大度后，社区结构更难被发现，所有算法的 *ONMI* 值均出现不同程度的下降。

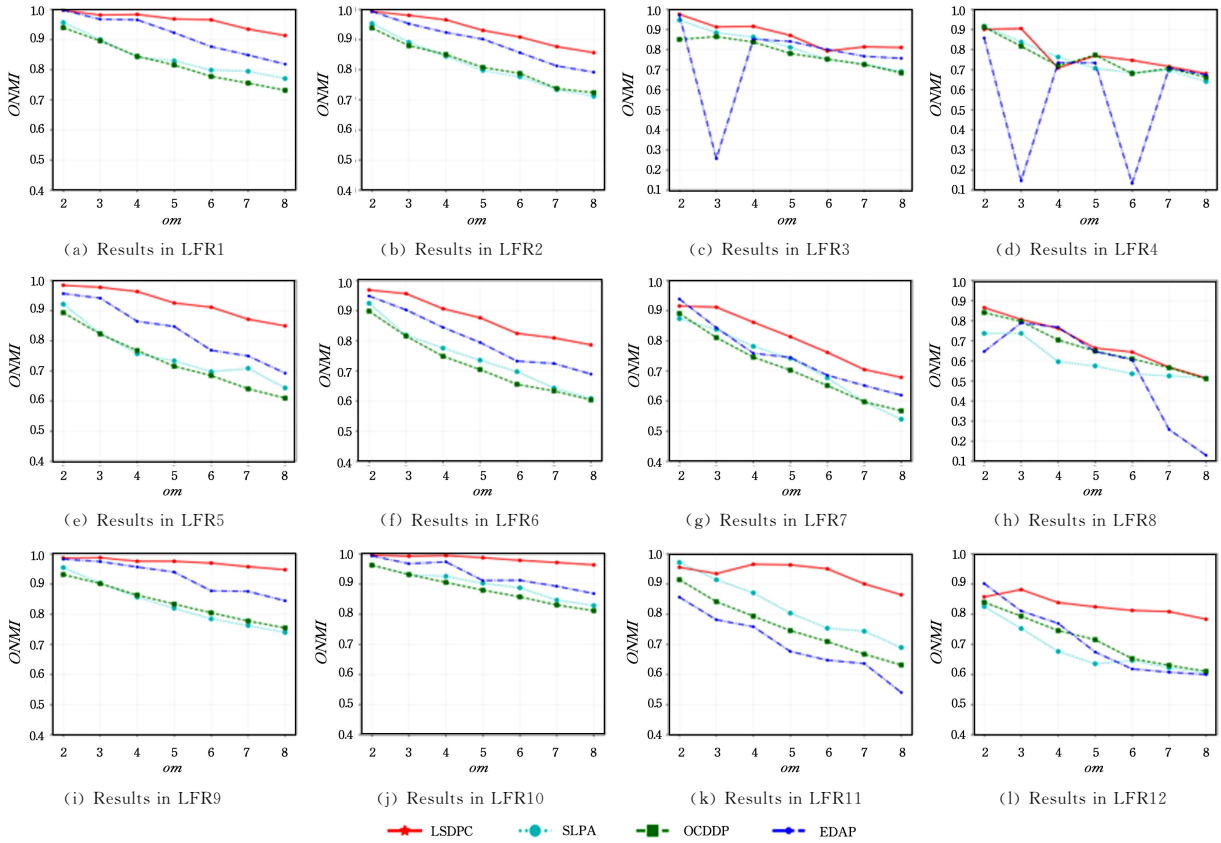

图 2 4 种算法在合成数据集上的结果

Fig. 2 Results of four algorithms on synthesis networks

与 OCDDP 算法相比，LSDPC 算法仅在个别混合参数 *mu* 较大的数据集中劣于 OCDDP 算法，如在 LFR4 数据集中，当 $om=2,4,5$ 时，OCDDP 算法的结果更优。整体来看，LSDPC 算法在绝大多数情况下的 *ONMI* 值都较大。这是由于 OCDDP 算法在度量节点间距离时只考虑了链接权重，同时社区中心点的选择依赖于人工判断。

与 EADP 算法相比，LSDPC 算法在大多数据集上能够得到更大的 *ONMI* 值。在其中 7 个数据集上，当重叠节点所属的社区数目 $om=2$ 时，EADP 算法的 *ONMI* 值与 LSDPC 算法较为接近甚至优于 LSDPC 算法，但随着 *om* 的增加，EADP 算法的 *ONMI* 值明显下降。同时，EADP 算法在 LFR3，LFR4 和 LFR8 合成数据集中出现了 *ONMI* 值震荡的现象，如在 LFR4 数据集中，混合参数 $mu=0.4$，当 *om* 为 3 和 6 时，实验结果远低于预期值，这是因为混合参数较大时，网络结构相对复杂，EADP 算法采用的最小二乘拟合的方法此时无法正确地识别社区中心节点，从而导致 *ONMI* 值远小于其他算法。

与 SLPA 算法相比，LSDPC 算法在绝大多数数据集上的 *ONMI* 值均优于 SLPA 算法。仅在 LFR11 数据集上，当 $om=2$ 时，SLPA 算法优于 LSDPC 算法。这是因为 SLPA 算法虽然时间复杂度低，但在标签更新顺序以及标签选择上存在随机性，导致算法结果不稳定。

因此，在不同参数生成的合成数据集上，相比对比算法，本文提出的 LSDPC 算法得到的重叠划分结果更好。

结束语 本文提出了一种基于局部相似性的两阶段密度峰值重叠社区发现算法 LSDPC，该算法设计了一种节点局部相似性指标，该指标不仅考虑了大度节点有利指标，还考虑了节点间的连接贡献度，可以更好地度量节点之间的局部联系。在此基础上，利用密度峰值的思想与切比雪夫不等式选取社区中心点，避免了利用人工先验知识确定社区个数的繁琐性与主观性，同时给出了两阶段的社区划分策略，以发现重叠社区。在仿真实验中，选取了多组真实网络数据集与合成数据集进行了对比实验，结果表明，LSDPC 算法发现的社区质量优于已有的许多算法，可以得到更加合理的重叠社区划分结果。另外，本文算法中存在 3 个人工调节参数，下一步工作会对算法进行优化，使其可以自动发现参数，降低调参难度，同时进一步优化算法的时间复杂度。

参 考 文 献

[1] HAN N, QIAO S J, YUAN C A, et al. A Fast Parallel Community Detection Algorithm for Mobile Social Networks[J]. Journal of Chongqing University of Technology (Natural Science), 2020, 34(1): 94-102.

[2] FORTUNATO S, HRIC D. Community detection in networks: a user guide[J]. Physics Reports, 2016, 659: 1-44.

[3] ZHAO W J, ZHANG F B, LIU J L. Review on Community Detection in Complex Networks [J]. Computer Science, 2020, 47(2): 10-20.

[4] RODRIGUEZ A, LAIO A. Clustering by fast search and find of

density peaks[J]. Science,2014,344(6191):1492-1496.
[5] RAVASE E,SOMERA A L,MONGRU D A,et al. Hierarchical organization of modularity in metabolic networks[J]. Science,2002,297(5586):1553-1555.
[6] DING J J,CHEN Z T,HE X X,et al. Clustering by finding density peaks based on Chebyshev's inequality[C]// Proceedings of the 2016 35th Chinese Control Conference. IEEE,2016:7169-7172.
[7] PALLA G,DERENYI I,FARKAS I,et al. Uncovering the overlapping community structure of complex networks in nature and society[J]. Nature,2005,435(7043):814-818.
[8] GREGORY S. Finding overlapping communities in networks by label propagation[J]. New Journal of Physics,2010,12(10):2011-2024.
[9] XIE J,SZYMANSKI B K,LIU X. SLPA:Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process[C]// Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops. IEEE,2011:344-349.
[10] LIU S C,ZHU F X,GAN L. A label-propagation-probability-based algorithm for overlapping community detection[J]. Chinese Journal of Computers,2016,39(4):717-729.
[11] LANCICHINETTI A,FORTUNATO S,KERTÉSZ J. Detecting the overlapping and hierarchical community structure in complex networks[J]. New Journal of Physics,2009,11(3):1-18.
[12] YU Z Y,CHEN J J,GUO K,et al. Overlapping community detection based on influence and seeds extension[J]. Chinese Journal of Electronics,2019,47(1):153-160.
[13] COSCIA M,ROSSETTI G,GIANNOTTI F,et al. DEMON:a local-first discovery method for overlapping communities[C]// Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2012:615-623.
[14] AHN Y Y,BAGROW J,LEHMANN S. Link communities reveal multiscale complexity in networks[J]. Nature,2010,466(7307):761-764.
[15] PAN L,JIN J,WANG C J,et al. Detecting Link Communities Based on Local Information in Social Networks[J]. Chinese Journal of Electronics,2012,40(11):2255-2263.
[16] HUANG L,WANG G,WANG Y,et al. A link density clustering algorithm based on automatically selecting density peaks for overlapping community detection[J]. International Journal of Modern Physics B,2016,30(24):1650167.
[17] HUANG L,LI Y,WANG G S,et al. Community detection method based on vertex distance and clustering of density peaks [J]. Journal of Jilin University (Engineering and Technology Edition),2016,46(6):2042-2051.
[18] DING J J,HE X X,YUAN J Q,et al. Community detection by propagating the label of center[J]. Physica A:Statistical Mechanics and Its Applications,2018,503:675-686.
[19] BAI X Y,YANG P L,SHI X H. An overlapping community detection algorithm based on density peaks[J]. Neurocomputing,2017,226:7-15.
[20] XU M L,LI Y H,LI R X,et al. EADP:An extended adaptive density peaks clustering for overlapping community detection in social networks[J]. Neurocomputing,2019,337(2019):287-302.
[21] FENG Y F,CHEN H M. Topological structure based density peak algorithm for overlapping community detection[J]. Computer Science,2019,46(10):39-48.
[22] WANG X F,LI X,CHEN G R. Network science:an introduction [M]. Beijing:Higher Education Press,2012.
[23] SHAO J M,HAN Z C,YANG Q L,et al. Community detection based on distance dynamics[C]// Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2015:1075-1084.
[24] LÜ L Y. Link prediction on complex networks [J]. Journal of University of Electronic Science and Technology of China,2010,39(5):651-661.
[25] XIE J Y,GAO H C,XIE W X,et al. Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors[J]. Information Sciences,2016,354:19-40.
[26] NEWMAN M. Network data [EB/OL]. http://www.personal.umich.edu/~mejn/netdata/.
[27] KUNEGIS J. KONECT:the Koblenz network collection[C]// Proceedings of the 22nd International Conference on World Wide Web. ACM,2013:1343-1350.
[28] LANCICHINETTI A,FORTUNATO S,RADICCHI F. Benchmark graphs for testing community detection algorithms [J]. Physical Review E,2008,78(4):046110.
[29] FORTUNATO S. LFR benchmark graphs [EB/OL]. https://www.santofortunato.net/resources.
[30] NICOSIA V,MANGIONI G,CARCHIOLO V,et al. Extending the definition of modularity to directed graphs with overlapping communities[J]. Journal of Statistical Mechanics:Theory and Experiment,2009,2009(3):3024-3046.

DUAN Xiao-hu, born in 1996, postgraduate, is a member of China Computer Federation. His main research interests include cluster analysis and so on.

CAO Fu-yuan, born in 1974, professor, Ph.D supervisor, is a member of China Computer Federation. His main research interests include Data mining and machine learning.

(责任编辑:李亚辉)