一种新的基于标签传播的复杂网络重叠社区识别算法

邓 琨[1,2†], 李文平[1], 陈 丽[1], 刘星妍[1]

(1. 嘉兴学院 数理与信息工程学院，浙江 嘉兴 314001；
2. 蒂赛德大学 计算、媒体与艺术学院，米德尔斯伯勒 TS1 3BX)

摘 要: 针对现有基于标签传播的复杂网络重叠社区识别方法所存在的社区识别精度不稳定,以及随机性较强等缺陷,提出一种新的基于标签传播的复杂网络重叠社区识别算法NOCDLP(a novel algorithm for overlapping community detection based on label propagation). 该算法首先搜索网络中若干以度较高节点为中心的完全子图,并以这些完全子图为起点进行标签传播;其次通过分析节点与社区连接强度以及社区接纳某节点后的社区内部连接紧密度情况给出节点归属社区强度函数,以此作为标签传播的依据提高社区的识别精度;再次,在标签传播过程中,NOCDLP算法设置标签传播控制标记,以避免标签传播算法随机性较强的缺陷;最后,在已形成的社区中通过整理重叠节点获得更准确的重叠社区结构. 算法在人工网络与真实网络中完成测试,同时与多个经典算法进行对比分析,实验结果验证了NOCDLP算法是有效的、可行的.

关键词: 复杂网络；社区结构；社区识别；标签传播；重叠节点

中图分类号: TP391 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2019.0176 **开放科学(资源服务)标识码(OSID):**

引用格式: 邓琨,李文平,陈丽,等. 一种新的基于标签传播的复杂网络重叠社区识别算法[J]. 控制与决策, 2020, 35(11): 2733-2742.

A novel algorithm for overlapping community detection based on label propagation in complex networks

DENG Kun[1,2†], *LI Wen-ping*[1], *CHEN Li*[1], *LIU Xing-yan*[1]

(1. College of Mathematics Physics and Information Engineering，Jiaxing University，Jiaxing 314001，China；2. School of Computing，Media and the Arts，Teesside University，Middlesbrough TS1 3BX，UK)

Abstract: Existing label propagation based overlapping community detection algorithms are limited, in terms of lacking accuracy, exhibiting high randomness, etc., when applied to complex networks. To overcome these limitations, this paper proposes a novel algorithm for overlapping community detection based on label propagation (NOCDLP). In the algorithm, we first search for a number of complete subgraphs centered on nodes with higher degrees in a network and initiate the label propagation starting from these subgraphs. Then, a function to specify the bonds between nodes and communities is generated, by analyzing the strength of connections between nodes and communities, and the internal closeness of a particular community after a certain node is adopted. By introducing this function, the accuracy of community detection is increased significantly. Subsequently, in the process of label propagation, NOCDLP sets control marks to alleviate the high randomness in community detection. Finally, the algorithm cleans up overlapping nodes to improve the accuracy of the overlapping community structures generated. This algorithm is tested in both artificial and real-world networks. The experimental results show that the proposed algorithm is practical and more efficient in comparison with multiple classical algorithms.

Keywords: complex networks；community structures；community detection；label propagation；overlapping nodes

0 引 言

在现实世界中诸多复杂系统可由复杂网络的形式表示出来,如人与人之间的关系网络、计算机之间相互连接的因特网、科学家之间的合作网络等. 通过

收稿日期: 2019-02-18；**修回日期:** 2019-06-04.

基金项目: 教育部人文社会科学研究青年基金项目(17YJCZH033, 15YJCZH088)；国家自然科学基金项目(61672179, 61370083)；浙江省自然科学基金项目(LY15F020040)；浙江省教育厅科研基金项目(Y201636127)；浙江省教育科学规划课题项目(2020SCG046).

责任编委: 阳春华.

†通讯作者. E-mail: dengkun@hrbeu.edu.cn.

对这些复杂网络的研究与分析,发现其中存在着无标度特性、小世界特性,以及社区结构特性等复杂网络基本统计特性. 本文所涉及的“社区结构特性”普遍具有社区内部节点连接紧密、社区之间节点连接松散的特点. 复杂网络社区识别旨在探索到网络中真实拥有的社区结构,已应用于复杂网络的行为预测和拓扑结构分析等众多领域中,因此具有重要的理论意义与实用价值.

近年来,出现了许多优秀的社区识别算法,如FN算法[1]、LPA算法[2]等非重叠社区识别算法,虽然这些算法非常优秀,但其仅能将网络中的节点强硬地归属于单一社区. 在现实网络中,某些节点并不仅存在于单一社区,例如在社会网络中,人们通常根据家庭、朋友、职业和爱好等分类同时归属于多个社区. 鉴于此,相较于早期出现的非重叠社区识别算法,重叠社区识别更具实际意义. 至今,重叠社区识别方法层出不穷: Palla等[3]提出的基于派系过滤的CPM算法认为社区由若干个相邻的极大完全子图组成,而一个节点可以属于不同的极大完全子图,由此该算法通过派系过滤的方式识别网络中的重叠社区结构,相关算法见文献[4]; 基于种子扩展的LFM算法[5]首先定义了适应度函数,然后从不同的种子节点出发通过优化适应度函数扩展社区,由于每个节点有可能划分在不同社区中,因此LFM算法能够识别出网络中的重叠社区结构; 基于链接社区的LC算法[6]认为边通常拥有唯一角色,归属单一社区,因此该算法首先以边为研究对象,将网络中的每条边识别到不同社区中,当网络中的每条边都划分到不同社区时,重叠节点将自然呈现,以此完成重叠社区识别任务,相关算法见文献[7-8].

由于标签传播算法具有简单而高效的特点,已得到普遍关注,如: Gregory[9]提出的COPRA算法首先为网络中任意节点初始化社区标签和归属强度系数,然后,依据待更新节点的邻居节点标签及归属强度系数更新节点标签,为了防止重叠节点过多,COPRA算法采用节点归属强度系数与$1/p$(p为算法参数)比较的方式控制相同标签数量,最后在COPRA算法执行完成后标签相同的节点为一个社区,对应多个社区标签的节点将成为重叠节点; Xie等[10]提出的SLPA算法首先为网络中任意节点初始化标签,在标签传播的过程中,每个节点在接收其邻居节点标签的同时也向邻居节点发出标签,每个节点的存储空间可以保存之前迭代所接到的全部标签,为避免出现节点所对应社区标签过多的情况,SLPA算法使用相同标签所占比例大于参数x的方式确定哪些标签将保存下来,最终完成社区识别任务; Wu等[11]提出的BMLPA算法首先将网络中的任意节点标签标记为其邻居节点标签,然后通过计算社区标签归属强度之和找到节点所对应社区标签最大归属强度系数$b_{\max}$,最终某一标签是否保留取决于其是否满足该标签归属强度系数与$b_{\max}$的比值大于参数pu,该算法通过以上方式更新标签以完成社区识别任务,相关算法见文献[12]. 通过上述分析可以看出,传统的基于标签传播的重叠社区识别方法虽然具有简单、高效的特点,但依然存在如下缺陷: 1)算法随机性较强,识别结果不稳定; 2)算法需要预先设置相关参数,以辅助识别重叠节点.

本文针对已有基于标签传播的重叠社区识别算法存在的缺陷,提出一种新的基于标签传播的复杂网络重叠社区识别算法NOCDLP (a novel algorithm for overlapping community detection based on label propagation). 该算法首先搜索网络中若干以度较高节点为中心的完全连接子图; 然后通过定义节点归属社区强度函数作为标签传播的依据,在标签传播过程中,NOCDLP算法设置标签传播控制标记U,若某节点接收的邻居节点标签一致,则设置U为无需修改标记,该节点的标签在传播过程中将不再进行更新操作; 最后在已形成的社区中,重新整理重叠节点以获得更准确的重叠社区结构. 本文具体贡献主要包含以下几点: 1) 通过分析节点与社区连接强度及社区内部连接紧密度情况,提出节点归属社区强度函数,以提高算法识别社区的精度; 2)提出在标签传播过程中设置标签传播控制标记,以避免算法产生振荡效应; 3)搜索以度较高节点为中心的较小完全子图作为标签传播的初始社区,以提高算法的收敛效率; 4)提出无需设置参数的重叠节点整理方法,以提高算法的普适性.

1 算法NOCDLP

1.1 标签传播初始化

鉴于传统标签传播算法普遍具有较强随机性,造成算法识别精度不稳定的缺陷,NOCDLP算法提出采用以网络中若干个连接紧密的完全子图作为标签传播的原始社区. 由于在标签传播初期每个节点接收的邻居节点标签比较发散,在局部范围内若存在已有连接紧密的社区,则同时发出相同标签,使节点在收敛过程中更具指向性,从而降低算法的随机性.

定义1 (完全子图) 若网络N中有图G是一个极大完全子图,又有图G_1中存在g_1个节点,每个节点

对之间都有一条边相连,图G_1的节点集v_1和边集e_1均属于图G的节点集V和边集E,则称图G_1为图G的完全子图.

研究表明[11],在标签传播过程中以较小完全子图为中心进行社区扩展,往往能够取得较好的效果. 其原因是标签传播算法通常具有较强的传染性,以极大完全子图作为核心进行社区扩展会导致结果中产生巨型社区,影响社区识别质量. 鉴于此,本文在标签传播初始阶段首先搜索网络中若干以度较高节点为中心的较小完全子图作为标签传播的起始点,具体描述见算法1.

算法1　搜索网络中以度较高节点为中心的完全子图.

输入:复杂网络$N=(V,E)$//V为网络N的节点集合,E为网络N的边集合;

输出:完全子图集合GS.

begin

1) 网络N中每个节点初始化标签为节点编号;

2) while存在尚未搜索过的节点

3)　$i\leftarrow$随机选取一个节点;

4)　while i未搜索过

5)　　将i标记为已搜索;

6)　　$i\leftarrow$搜索节点i的邻居节点中度大于等于i的节点集合,从中选择度最大的节点,若度最大节点不唯一,则随机选取一个节点;

7)　end while;

8)　$j\leftarrow$获取节点i的邻居节点中度最大的节点,若度最大节点不唯一,则随机选取一个节点;

9)　iWQT $\leftarrow$ iWQT$\bigcup i$; // iWQT为完全子图节点集合

10)　while j与集合iWQT中所有节点均有关联边时

11)　　iWQT $\leftarrow$ iWQT$\bigcup j$;

12)　　$j\leftarrow$获取节点j的邻居节点中度最大节点,若度最大节点多于一个,则随机选取一个节点;

13)　end while

14)　GS $\leftarrow$ GS$\bigcup$iWQT;

15)　更新iWQT集合内所有节点的标签为相同标签

16) end while

end

由算法1可知,算法为网络N中全部节点初始化标签后,在未搜索过的区域中沿着节点度增加的方向搜索区域中度最大的节点i,然后找到与节点i相连的度最大的节点j,搜索与节点i、j相连接的完全子图,反复执行以上操作,最终获得若干个以度较高节点为中心的完全子图. NOCDLP算法以这些完全子图作为标签传播的初始点开始标签传播.

1.2　标签传播过程

由于传统标签传播算法通常以邻居节点所在社区的数量判断某节点的标签,往往会使得一些较大的社区快速扩张,但这些社区内部节点间连接通常较为稀疏,由此降低了社区识别的精度. 本文基于以上考虑,给出节点归属社区强度函数作为NOCDLP算法标签传播依据,具体定义如下.

定义2 (节点-社区连接强度)　若节点i为网络中的一个节点,C为一个社区,则节点-社区连接强度$t_{i\to C}$表示为

$$t_{i\to C}=\frac{d_{i\to C}}{k_i}. \tag{1}$$

其中:$d_{i\to C}$为节点i与社区C的连接边数,k_i代表节点i的度.

定义3 (社区连接紧密度)　若节点i为网络中的一个节点,C为一个社区,则社区连接紧密度定义为

$$D(i)=\frac{m_{i\to C}}{|C\bigcup\{i\}|}. \tag{2}$$

其中:$m_{i\to C}$为节点i加入社区C后社区C中的边数;$|C\bigcup\{i\}|$为节点i加入社区C后社区C中的节点数.

定义4 (节点归属社区强度)　若$t_{i\to C}$为节点i与社区C的节点-社区连接强度,$D(i)$为节点i加入社区C后的社区连接紧密度,则节点i归属社区C的节点归属社区强度$b_{i\to C}$表示为

$$b_{i\to C}=\sqrt{t_{i\to C}\cdot D(i)}. \tag{3}$$

由节点归属社区强度函数可以看出,每个节点在选择标签和归属社区时,不仅考虑了邻居节点所在社区的数量(节点-社区连接强度),也考虑了节点加入社区后社区内部连接紧密度情况(社区连接紧密度),其主要思想是在考虑标签快速扩散的同时,也要保证社区内部连接紧密度较高.

在给出节点归属社区强度函数的基础上,提出NOCDLP算法的标签传播过程,详见算法2.

算法2　NOCDLP算法的标签传播策略.

step 1: 执行算法1后所产生的结果作为标签传播的初始状态. 其中:任意节点i的标签存储空间表示为$\{(l_{i\to c},b_{i\to c}),U_i\}$,$l_{i\to c}$为节点$i$归属于社区$C$的标签,$b_{i\to c}$为节点$i$归属于社区$C$的节点归属社区强度,$U_i$为节点$i$的标签控制标记.

step 2: 为每个节点加入标签控制标记,将完全子图中所有节点的标签控制标记设置为无需更新,其余节点的标签控制标记设置为可更新.

step 3: 标签控制标记为可更新的节点i将接收每个邻居节点j($j\in\mathrm{adj}(i)$,$\mathrm{adj}(i)$为节点i的邻居节

点集合)中最高节点归属社区强度$b_{j\to C}$对应的标签$l_{j\to C}$;计算节点i在本次迭代中接收到的不同标签的节点归属社区强度,并以本次迭代所接收的不同标签及相应的节点归属社区强度更新节点i的标签存储空间. 若节点i接收所有邻居节点标签均为相同社区标签,则节点i的标签控制标记U_i将修改为无需更新.

step 4: 采用异步更新方式反复执行step 3,直至标签控制标记为可更新的所有节点所接收到的标签在两次迭代过程中均未发生变化,此时停止标签传播过程. 最终标签相同的节点为一个社区,若某节点标签存储空间中所存在的不同标签数量多于1个,则该节点为重叠节点.

由算法2可见,NOCDLP的标签传播过程区别于传统的标签传播算法仅从单一节点角度分析节点所属社区,忽略了接受某节点后社区内部连接紧密度情况,因此造成所识别的社区较大、社区内部连接较为松散的问题.

需要指出,虽然同步更新策略较异步更新策略更加稳定,但迭代次数明显高于异步更新策略,因此在时间效率上更低,并不适合当今的大数据环境;此外,在面对二部网络及具有星型结构的网络时,异步更新策略拥有更小的振荡效应[13],因此本文选用异步更新方式作为NOCDLP算法的标签更新策略.

在NOCDLP的标签传播过程中,为每个节点加入标签控制标记,在算法1搜索到的完全子图中,全部节点的标签已设置为相同标签,其标签控制标记均设置为无需修改,目的是在网络中搜索到以度较高节点为中心的若干个完全子图作为标签传播的起始社区,避免算法的振荡效应,以便快速地将周围邻居节点吸引过来,而在标签传播初期,局部范围内有多数相同的标签发出,也能降低算法的随机性. 在标签传播过程中,若某节点在接收邻居节点标签时均属于同一社区标签,则表明邻居节点更倾向于归属同一社区,因此将该节点标签控制标记设置为无需更新,从而在剩余的标签传播过程中该节点标签不再进行更新,以便进一步避免算法的振荡效应,提高算法收敛效率.

1.3 整理重叠节点

由于传统基于标签传播的重叠社区识别算法,通常采用设置参数的方式去除重叠节点中的标签,但在较为复杂的网络中很难准确设置相关参数以保证算法的准确性,本文采用无需设置参数的方式整理重叠节点,力求提高算法的普适性以及识别重叠节点的准确性. 下面首先给出社区紧密度增量的定义.

定义5 (社区紧密度增量) 设节点i为网络中的一个节点,m_{c+i}、m_c分别为节点i加入到社区C和未加入到社区C的边数,n_{c+i}、n_c分别为节点i加入到社区C与未加入到社区C的节点数. 社区紧密度增量ΔC可表示为

$$\Delta C=\frac{m_{c+i}}{n_{c+i}}-\frac{m_c}{n_c}. \tag{4}$$

基于社区紧密度增量的定义,NOCDLP算法的整理重叠节点方法流程如图1所示. 图1中: s为重叠节点,T为重叠节点s所对应的标签存储空间中包含的标签集合.

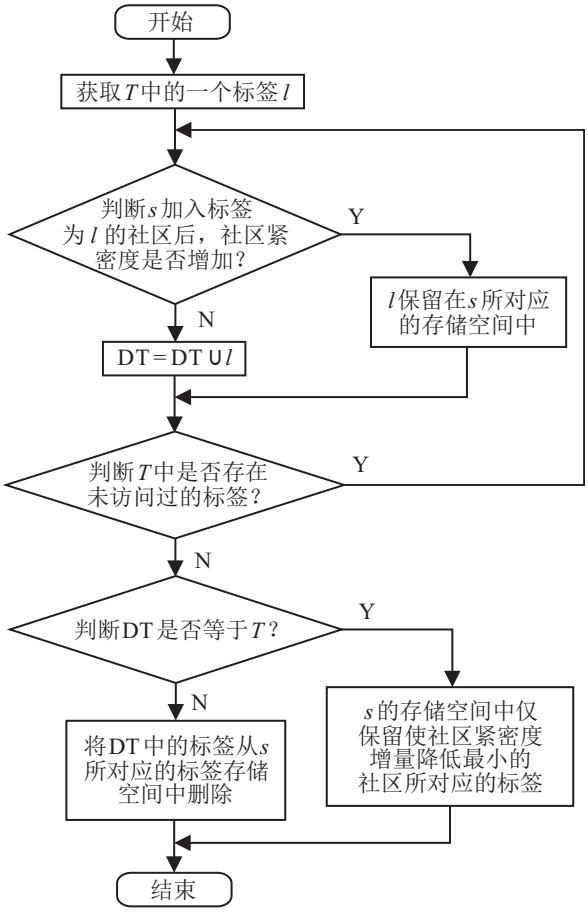

图1 整理重叠节点方法流程

由图1可见,若重叠节点s加入到标签为l的社区中可以增加社区紧密度增量,则标签l保留在重叠节点s对应的标签存储空间中,否则,将标签l从重叠节点s对应的标签存储空间中删除. 若重叠节点s加入其标签存储空间中任意标签所对应社区均无法提高社区紧密度增量,则重叠节点s的标签存储空间中仅保留使社区紧密度增量值降低最小的社区所对应的标签.

1.4 算法描述

经过上述操作,给出NOCDLP算法的基本执行过程,如图2所示. 由图2可见,NOCDLP算法首先通过标签传播初始化阶段为网络中所有节点初始化标

签,并搜索到网络中的若干完全子图作为标签传播的起始点;然后提出NOCDLP算法的标签传播策略,在该策略中使用节点归属社区强度函数作为标签传播的依据,该函数从节点与社区连接强度以及社区内部连接紧密度两方面考虑标签的扩散,从而避免了传统标签传播算法仅采用节点与社区连接强度作为标签扩散依据所带来的识别社区过大的现象. 此外,在标签传播策略中加入标签控制标记,使得算法能够进一步避免振荡现象和随机性,最终使用无需设置参数的整理重叠节点的方式准确辨别重叠节点,以提高算法的普适性及识别重叠节点的准确性. 经过以上操作,标签存储空间中标签相同的节点为一个社区,重叠节点的标签存储空间中拥有多个标签,最终完成对重叠社区的识别任务.

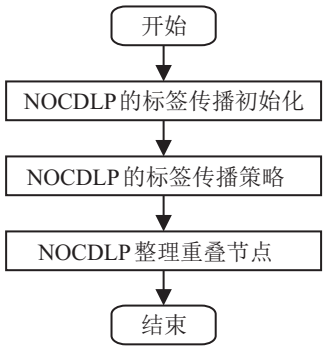

图 2　NOCDLP算法流程

1.5　NOCDLP算法的时间复杂度分析

设n为网络N中的节点数,k为节点平均度,r为标签传播初始化区域数,w为NOCDLP标签传播迭代数,o为执行标签传播策略后识别的重叠节点数. 下面给出NOCDLP的时间复杂度分析过程.

由标签传播初始化过程可以看出,NOCDLP算法初始化标签的时间复杂度为$O(n)$;搜索完全子图的时间复杂度不会超过$O(rnk)$;在NOCDLP标签传播策略中,每个节点计算节点归属社区强度的最坏时间复杂度为$O(k(k-1)/2)$,该策略的时间复杂度不会超过$O(wnk^2)$;NOCDLP算法整理重叠节点的时间复杂度为$O(ko)$. 因此,NOCDLP算法的时间复杂度约为$O(rnk+wnk^2+ko)$. 考虑到r、k、o、w远小于网络中的节点数n,从而NOCDLP算法的时间复杂度约为$O(qnk^2)$,其中q为常数.

2　实验分析

为了验证算法的有效性和可行性,本文在LFR Benchmark网络[14]和真实网络上对各算法进行测试. 实验在2台Intel Core i5-6200U CPU 2.30 GHz和8 GB内存笔记本上进行,操作系统为Windows 7,NOCDLP算法的编程工具为Matlab R2011b. 实验对比算法为基于派系渗透的CFINDER[3]、基于标签传播算法的COPRA[9]、SLPA[10]、OLLP[12]以及最近提出的基于多尺度社区识别MS算法[15]、基于核心节点集扩展的CoEuS算法[16]. 各算法开发环境及提出时间如表1所示. 算法参数设置如下:CFINDER[3]的派系规模$h=3\sim6$,间隔为1;COPRA[9]的节点隶属社区数量$p=2\sim6$,间隔为1;SLPA[10]的保留标签控制参数$x=0.2\sim0.6$,间隔为0.05,迭代次数$w=20$. 各算法在不同参数下选取评价指标最大值或评价指标最大平均值作为最终结果.

表 1　各算法开发环境及提出时间

序号	算法	开发环境	提出时间
1	CFINDER	Java	2005年
2	COPRA	Java	2010年
3	SLPA	C++	2011年
4	OLLP	Matlab	2015年
5	MS	Java	2016年
6	CoEuS	Java	2017年

社区识别算法按识别结果划分,可分为确定性和非确定性两类[17],例如:CFINDER、MS与CoEuS等针对同一网络,每次执行结果相同的算法,称为社区识别结果确定性算法. 由于MS与CoEuS无需设置参数,算法仅需要在不同网络中运行以获取运行结果的评价指标值作为最终结果,对于CFINDER,本文采取在其不同参数下,选取评价指标最大值作为最终对比结果. COPRA、SLPA、OLLP、NOCDLP等算法针对同一网络,每次执行结果不同,称为结果非确定性社区识别算法. 由于OLLP与NOCDLP无需设置参数,本文运行算法10次,取评价指标平均值作为最终对比结果. 对于COPRA与SLPA算法,本文在不同参数情况下于每个网络中分别运行算法10次,取评价指标的平均值,最终以评价指标的最大平均值作为最终对比分析结果.

2.1　评价指标

为更准确地评价各算法的性能,选用3个经典社区识别评价指标,分别从社区识别准确率、重叠节点识别精度以及识别社区连接紧密度3方面分析各算法.

1) 社区识别准确率评价指标为统一化互信息NMI (normalized mutual information)[5],NMI是评价真实社区结构A与运行算法所获得的社区结构B的相似度. NMI函数可定义为

$$\mathrm{NMI}(A,B)=$$

$$\frac{-2\sum_{y=1}^{\mathrm{CA}}\sum_{u=1}^{\mathrm{CB}}M_{yu}\log\left(\frac{M_{yu}M}{M_{y.}M_{.u}}\right)}{\sum_{y=1}^{\mathrm{CA}}M_{y.}\log\left(\frac{M_{y.}}{M}\right)+\sum_{j=1}^{\mathrm{CB}}M_{.u}\log\left(\frac{M_{.u}}{M}\right)}. \tag{5}$$

其中: M 为矩阵,行表示真实社区,列表示算法识别到的社区; M_{yu} 为真实社区 y 与算法识别到的社区 u 的重合节点数量, $M_{y.}$ 为第 y 行元素之和, $M_{.u}$ 为第 u 列元素之和; CA 为网络中真实存在社区个数, CB 为算法识别到的社区数. NMI 的取值范围在 0 ~ 1 之间,如果 NMI 取值为 1,则识别到的社区结构与真实社区结构完全一致;若 NMI 取值为 0,则识别到的社区结构与真实的社区结构截然不同. 算法识别社区结构越准确, NMI 取值越高,反之 NMI 取值越低.

2) 识别重叠节点精度评价指标为 F-score[17],可以表示为

$$F\text{-score}=\frac{2\cdot\mathrm{precision}\cdot\mathrm{recall}}{\mathrm{precision}+\mathrm{recall}}. \tag{6}$$

其中: precision 为算法正确识别到的重叠节点数量与算法识别到的重叠节点数量的比值; recall 为算法正确识别的重叠节点数量与社区中真实存在的重叠节点数量的比值. F-score 的取值在 0 ~ 1 之间,即算法识别重叠节点的精度越高, F-score 的取值越高.

3) 识别社区连接紧密度指标 EQ (extend Q)[4] 可表示为

$$\mathrm{EQ}=\frac{1}{2m}\sum_{z}\sum_{i,j\in C_z}\frac{1}{p_ip_j}\left[M_{ij}-\frac{k_ik_j}{2m}\right]. \tag{7}$$

其中: M_{ij} 为网络中的邻接矩阵元素,若节点 i、j 之间有边相连,则 $M_{ij}=1$,反之 $M_{ij}=0$; m 为网络中的边数; k_i 为节点的度; p_i 为节点 i 同时隶属于不同社区的数量. 需要说明的是,算法所识别到的社区内部节点连接越紧密,相应的 EQ 值越高,代表社区识别质量越高,反之亦然.

2.2 基准网络

为了客观反映各算法的性能,设计 4 组不同类型的网络如下: 1) 重叠节点较少且重叠节点同时归属社区较少,但社区结构逐渐模糊; 2) 重叠节点较多,重叠节点同时归属社区较少,但社区结构逐渐模糊; 3) 社区结构较为清晰,但重叠节点同时归属的社区逐渐增多; 4) 社区结构较为模糊,但重叠节点同时归属社区逐渐增多. LFR Benchmark 具体参数设置见表 2. 表 2 中: 参数 n 为网络规模; k 为网络中节点的平均度; $k_{\max}$ 为网络中节点的最大度; $C_{\min}$、$C_{\max}$ 为代表网络中最小的社区节点数及最大的社区节点数; O_n 为网络中重叠节点数; P_m 为重叠节点同时可归属的社区数; μ 为混合比例数, μ 值越小社区结构越清晰,反之亦然.

表 2 LFR Benchmark 网络参数设置

network	n	k	$k_{\max}$	$C_{\max}$	$C_{\min}$	O_n	P_m	μ
N_1	5 000	10	30	50	20	500	2	0.1 ~ 0.4
N_2	5 000	10	30	50	20	2 500	2	0.1 ~ 0.4
N_3	5 000	10	30	50	20	500	2 ~ 6	0.1
N_4	5 000	10	30	50	20	500	2 ~ 6	0.3

在 LFR Benchmark 网络上,各对比算法在 NMI 与 F-score 方面进行对比分析,在给出各算法运行结果前,先给出 CFINDER、COPRA 和 SLPA 算法在不同类型网络中所取得的 NMI 与 F-score 指标最大值所对应的参数值. 图 3 为 3 种算法在不同网络取得最

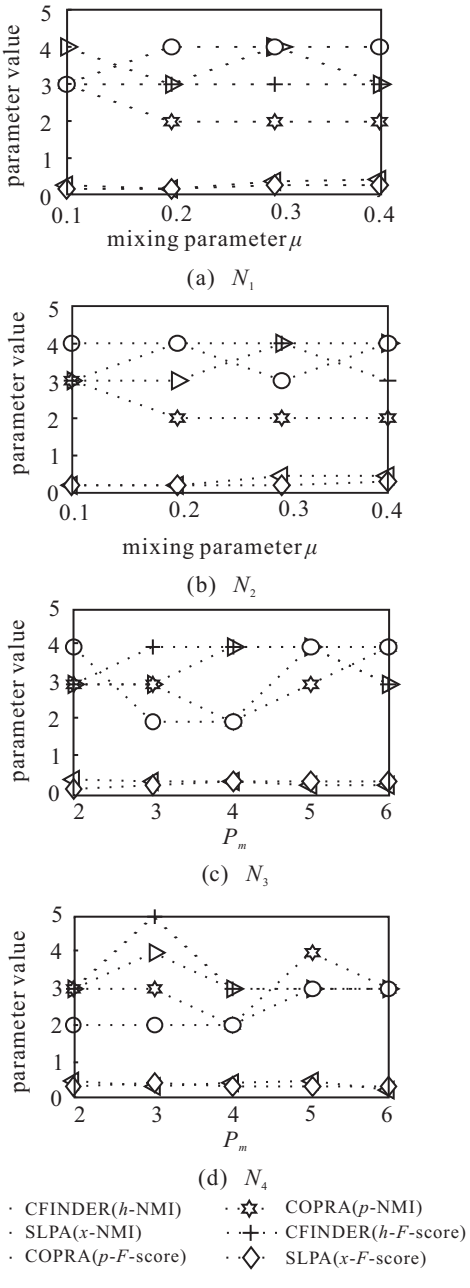

图 3 针对 $N_1 \sim N_4$ 网络获得最终值的参数值情况

大值所对应的相关参数设置情况. 其中: CFINDER(h-NMI)、 COPRA(p-NMI)、 SLPA(x-NMI) 分别代表 CFINDER、COPRA、SLPA 算法在 $N_1 \sim N_4$ 网络中取得最大NMI值所对应的参数值; CFINDER(h-F-score)、COPRA(p-F-score)、SLPA(x-F-score) 分别代表CFINDER、COPRA、SLPA算法在 $N_1 \sim N_4$ 网络中取得最大F-score值所对应的参数值.

2.2.1 算法识别精度分析

图4给出了针对 N_1 和 N_2 网络各算法NMI指标的对比结果. 在重叠节点较低的网络 N_1 中, 各算法随着 μ 值的增加, 社区识别精度均降低, 其根本原因是因为 μ 值的增加使社区结构逐渐变得模糊, 从而增加了社区识别的难度. 但是, 不难发现, 与其他算法相比较, NOCDLP 算法在 $\mu = 0.1 \sim 0.4$ 时, 均取得了最优的社区识别精度. 而在重叠节点较多的 N_2 网络中, COPRA 算法在 $\mu = 0.1$ 时、SLPA 算法在 $\mu = 0.2$ 时, 社区识别精度与NOCDLP算法相当, 但由于两种算法在社区识别过程中随机性较强, 使得算法出现了过强的振荡现象, 虽然这些算法在某些网络社区识别精度较高, 但在多数情况下 COPRA 与 SLPA 的社区识别精度均低于NOCDLP算法. 由图4可见, CFINDER 算法在 N_2 网络中, 当 $\mu = 0.4$ 时, 其社区识别精度高于NOCDLP算法, 但在其他网络中, CFINDER 的社区识别精度远低于NOCDLP算法. 这是因为CFINDER 算法在社区识别过程中, 采用搜索网络中极大完全子图并进行渗透的方式进行社区扩展, 所以该算法在具有较多极大完全子图的网络中通常会有较好的效果. 但是, 网络中通常并不存在较多、较大的完全子图, 因此该算法在网络较为稀疏的情况下社区识别精度不高, 也造成CFINDER 不能在多数网络中保持较高的识别精度.

图5给出了针对 N_3 和 N_4 网络各算法在NMI指标的对比结果. 由图5可见, 在社区结构较清晰的网络 N_3 中, NOCDLP 算法除在 $P_m = 3$ 时社区识别精度略低于SLPA算法外, 其他情况下NOCDLP均取得了最优的社区识别结果. 当 $P_m = 3$ 时, SLPA 的社区识别精度略高于NOCDLP算法, 这是由于在社区结构较为清晰以及重叠节点同时归属社区较少时, 即使算法选取起始点不当, 通常也不会对算法造成更大影响, 往往识别效果较好. 但随着网络结构逐渐变得模糊, 以及重叠节点逐渐增多, 当算法选择起始点不当时, 随机性较强的特点便会渐渐显现出来, 造成该算法无法在多数网络中均保持较好的稳定性, 使得SLPA 在多数网络中社区识别效果不佳. 在社区结构

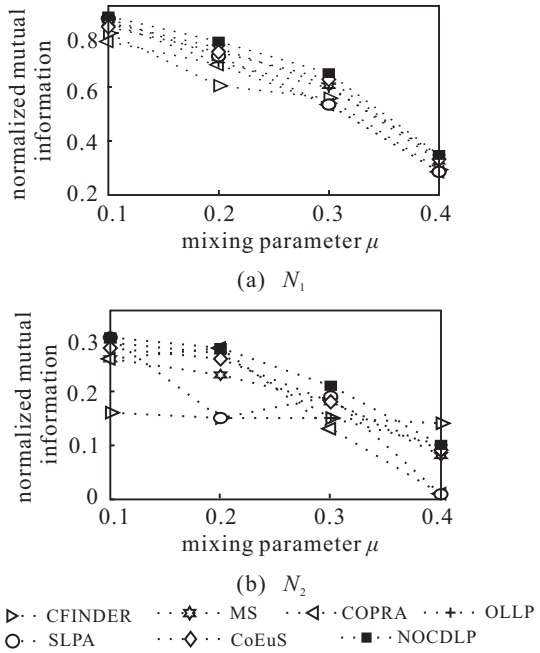

图 4　针对 N_1 和 N_2 网络社区识别精度对比结果

较模糊的 N_4 网络中, 当 $P_m = 3$ 时, COPRA 算法的NMI要高于NOCDLP算法, 这是因为COPRA 算法在识别社区时受网络结构与随机性较强特点的影响, 当算法随机选择到较为适合的起始点, 且网络中的节点与社区内连接和与社区外连接有较大差别时, 社区识别效果较好, 但在复杂网络中, 随机找到较为适合的起始点概率较小, 因此, 该算法的社区识别精度差异较大, 在多数情况下不能取得较优的社区识别结果.

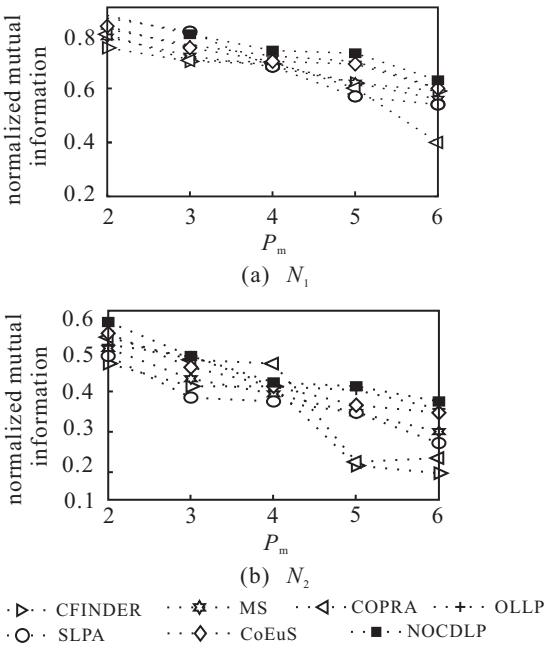

图 5　针对 N_3 和 N_4 网络社区识别精度对比结果

由图4和图5可见: CoEuS 与CFINDER 算法具有较为稳定的社区识别结果, 但其对网络结构要求较高, 因此这些算法均无法取得较高的社区识别精度; COPRA、SLPA 和OLLP 算法虽然在某些网络中社区

识别结果较好,但它们均存在随机性较强的缺陷,因此无法在多数网络中保持较优的社区识别结果;MS算法在社区识别过程中选择从不同粒度元素展开分析,但是不同粒度元素之间往往并不容易进行转换分析,因此影响了社区识别质量;NOCDLP算法在运行初期运用以搜索度较高的节点为中心的完全子图作为标签传播的初始社区进行社区识别,保证了算法能够找到适合的起点开始算法,且在识别社区的过程中分别考虑了节点与社区的连接强度以及节点加入社区后的社区内部连接情况,同时采用了稳定的标签传播策略,从而提高了算法的稳定性,算法的社区识别质量较高.

2.2.2 重叠节点识别精度分析

图6为各算法在$N_1 \sim N_4$网络中重叠节点识别精度F-score对比结果. 由图6可见,仅在N_1网络中,

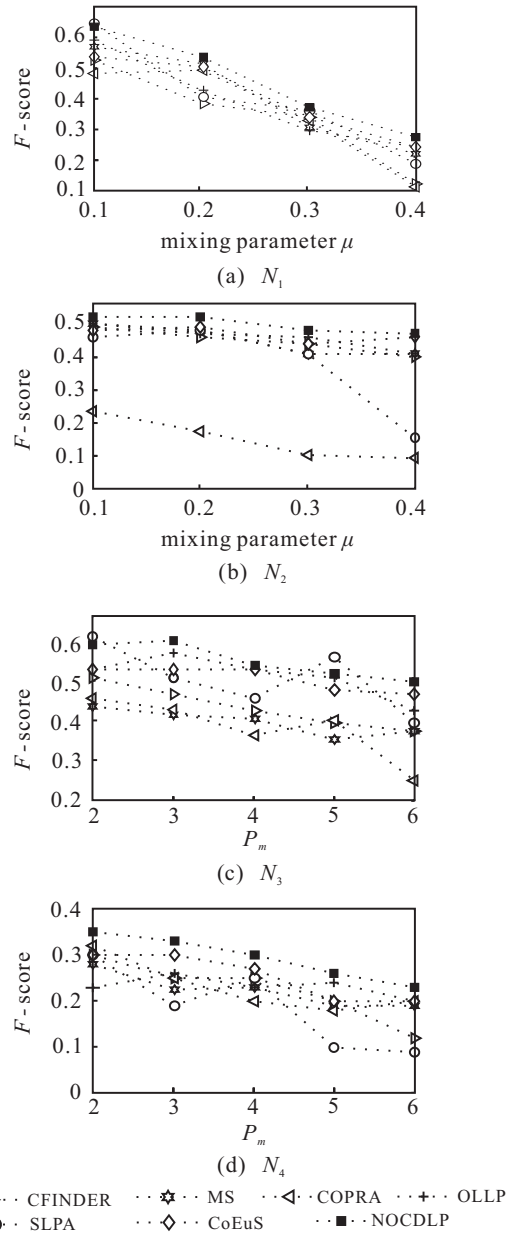

图6 针对$N_1 \sim N_4$网络重叠节点识别精度对比结果

当$\mu = 0.1$时,以及在N_3网络中,$P_m = 2$和$P_m = 5$时,NOCDLP算法的识别精度低于SLPA算法,而在其他情况下,NOCDLP算法的识别精度均高于其他对比算法. 这是因为,SLPA算法在网络结构较清晰时可以避免过强的随机性,使得算法重叠节点识别率较高,但是在纷繁复杂的网络中,网络结构往往并不清晰,因此SLPA算法并不稳定,从实验中也可看出,其振荡幅度过大. COPRA算法、OLLP算法与SLPA算法存在类似情况. 由图6可见:CoEuS和CFINDER算法在识别重叠节点过程中结果较为稳定,但是由于算法对重叠节点限制过于严格,往往将重叠节点识别为非重叠节点,从而降低了重叠节点识别精度;MS算法在不同粒度的元素下进行转换完成社区识别任务,在转换的过程中往往不会从节点角度分析重叠节点,因此使得识别重叠节点的准确率不高;NOCDLP算法在增加算法稳定性的同时从节点与社区结构的角度分析重叠节点,完成重叠节点整理,从实验结果看NOCDLP算法的重叠节点识别精度较高.

2.3 真实网络

社区识别的真正目的在于识别出真实网络中的社区结构,虽然已在基准网络上验证了算法的性能,但仍需要进一步在真实网络中测试算法的有效性.

由于NMI和F-score评价指标均为社区识别精度评价指标,而精度评价需要对算法识别的社区结构与网络真实存在的社区结构进行对比. 因为在真实网络数据集中已知真实社区结构的网络较少,且通常不具重叠性,所以无法进行较好的精度分析. 鉴于此,本节采用EQ评价指标评估各算法识别社区的连接紧密度情况,同时运用表3所列出的8种真实网络数据集验证各算法识别社区连接紧密程度. 表4给出了NOCDLP算法及其对比算法针对8个真实网络数据集得到的EQ值结果,其中"\\"表示算法识别社区失败或所得EQ值小于0.001. 表4中:对于MS与CoEuS算法,列出针对8种真实网络两种算法运行结果的EQ值;对于COPRA与SLPA算法,在不同参数情况下,对每个网络分别运行算法10次,然后取EQ指标的平均值,以指标最大平均值(avg)作为最终结果;对于OLLP与NOCDLP算法,采用运行算法10次取EQ指标平均值作为算法对比结果. 此外,表4针对非确定性算法(COPRA、SLPA、OLLP、NOCDLP)的运行结果值给出了对应结果的方差(var).

由表4可见:NOCDLP算法在karate、 dolphins、lesmis、email、netscience和internet六个真实网络中所取得的平均EQ值均高于CFINDER、MS、CoEuS

表 3　真实网络数据集

网络	节点	边	平均度	描述
Karate	34	78	4.59	空手道俱乐部网络[18]
Dolphins	62	159	5.13	海豚社会网络[18]
Lesmis	77	254	6.6	悲惨世界关系网络[18]
Email	1 133	5 451	9.62	电子邮件交往网络[19]
Netscience	1 588	2 742	3.45	作者合作网络[18]
PGP	10 680	24 316	4.55	信任网络[20]
Word	7 207	31 784	8.82	词汇语义网络[18]
Internet	22 963	48 436	4.22	互联网快照网络[18]

的 EQ 值和 COPRA、SLPA、OLLP 的平均 EQ 值; 在 PGP 网络中所取得的平均 EQ 值低于 SLPA、CoEuS 的 EQ 值; 在 word 网络中所获得的平均 EQ 值低于 OLLP 算法的平均 EQ 值, 之所以产生这种情况, 是由于 PGP 网络中某些区域非常稠密, 同时某些区域又较为松散, SLPA 与 CoEuS 算法往往将相对松散的区域识别产生较多无意义的小社区 (3 个节点以下的社区), 而 NOCDLP 算法不会产生这些小的社区, 它会将这些小社区吸收到以较小完全子图为中心的社区中, 从而损失了一些社区连接紧密度; OLLP 算法对于社区边缘的识别并不敏感, 因此其在较为稠密的 word 网络中识别的社区规模较大, 而 NOCDLP 算法对社区边缘的识别较为敏感, 其识别社区的规模小于 OLLP 算法, 在社区内部连接紧密度变化不大的情况下, 划分社区规模较小时将损失社区间的连接, 因此 NOCDLP 算法的 EQ 值略低于 OLLP 算法的 EQ 值; 由于 NOCDLP 算法在识别社区的过程中能够搜索到局部较为稠密的区域作为算法的起点, 并且通过分析节点与社区以及社区内部的情况帮助社区有针对性地扩展, 同时增加标签控制标记以减少算法的振荡性, 因此 NOCDLP 算法在多数的真实网络中取得了较好的社区识别效果.

表 4　各算法的 EQ 值对比结果

EQ		karate	dolphins	lesmis	email
NOCDLP	AVG	**0.396 6**	**0.450 3**	**0.531 1**	
	VAR	0.000 084 64	0.001 030 41	0.000 125	0.000 275 56
CFINDER		0.185 8($h=3$)	0.361 2($h=3$)	0.395 3($h=6$)	0.264 1($h=4$)
MS		0.259 9	0.327 0	0.386 6	0.151 3
CoEus		0.227 5	0.324 9	0.309 8	0.183 5
COPRA	AVG	0.317 7($p=3$)	0.415 2($p=4$)	0.476 2($p=2$)	0.351 1($p=3$)
	VAR	0.000 800 89	0.003 931 29	0.000 153 76	0.001 004 89
SLPA	AVG	0.342 6($x=0.3$)	0.386 2($x=0.4$)	0.321 3($x=0.45$)	0.179 0($x=0.45$)
	VAR	0.000 660 49	0.009 101 16	0.004 316 49	0.000 888 04
OLLP	AVG	0.356 6	0.420 1	0.323 7	0.273 1
	VAR	0.000 657 609	0.007 672 9	0.000 438 06	0.000 692 2
EQ		netscience	PGP	word	internet
NOCDLP	AVG	**0.886 5**	0.498 3	0.202 9	**0.408 1**
	VAR	0.000 948 64	0.000 338 56	0.000 007 29	0.000 176 89
CFINDER		0.761 1($h=3$)	\\	0.164 9($h=4$)	\\
MS		\\	\\	\\	\\
CoEus		0.150 6	0.507 7	0.012 8	0.257 9
COPRA	AVG	0.711 2($p=6$)	0.426 9($p=2$)	\\	0.035 7($p=3$)
	VAR	0.001 122 25	0.000 625	\\	0.004 9
SLPA	AVG	0.707 6($x=0.45$)	**0.695 7(x=0.45)**	0.201 7($x=0.45$)	0.112 4($x=0.45$)
	VAR	0.007 191 04	0.000 806 56	0.000 047 61	0.000 225
OLLP	AVG	0.761 3	0.312 3	**0.237 2**	0.198 9
	VAR	0.008 042 25	0.000 499 6	0.000 152 1	0.000 397 6

由表 4 还可见, NOCDLP 算法的结果方差远小于 COPRA、 SLPA、 OLLP 算法, 表明 NOCDLP 算法的稳定性优于几种对比算法. 在 8 个真实网络中, NOCDLP 算法不仅取得了较高的 EQ 值, 同时也取得了较小的结果方差, 验证了 NOCDLP 算法较为稳定, 并且应用范围更加广泛.

3　结　论

本文提出的 NOCDLP 算法为网络中所有节点初始化标签, 搜索到网络中的若干完全子图并赋予完全子图中的节点为相同标签, 以这些完全子图作为标签传播的起点使得算法在标签传播初期局部范围内有多数相同的标签发出, 在避免出现振荡效应的同时能

够较好地降低算法的随机性. 提出了NOCDLP算法的标签传播策略,使用节点归属社区强度函数作为标签传播的依据,该函数从节点与社区连接紧密度以及社区内部连接紧密度两方面考虑标签的扩散,从而避免了传统标签传播算法所具有的识别社区过大的现象. 此外,在标签传播策略中加入标签控制标记,使得算法能够进一步避免振荡现象和随机性. 最终采用无需设置参数的重叠节点整理方式进一步识别重叠节点,以提高算法对重叠节点的识别精度. 无论是在较为复杂的人工网络数据集,还是在具有上万个节点的大型真实网络数据集中,NOCDLP均表现出较好的运行效果. 此外,由于NOCDLP无需输入任何参数,算法更具普适性.

参考文献(References)

[1] Newman M E J. Fast algorithm for detecting community structure in networks[J]. Physical Review E, 2004, 69(6): 066133.

[2] Raghavan U N, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks[J]. Physical Review E, 2007, 76(3): 036106.

[3] Palla G, Derényi I, Farkas I, et al. Uncovering the overlapping community structure of complex networks in nature and society[J]. Nature, 2005, 435(7043): 814-818.

[4] Shen H W, Cheng X Q, Guo J F. Quantifying and identifying the overlapping community structure in networks[J]. Journal of Statistical Mechanics Theory and Experiment, 2009, 53(7): 07042.

[5] Lancichinetti A, Fortunato S, Kertesz J. Detecting the overlapping and hierarchical community structure in complex networks[J]. New Journal of Physics, 2009, 11(3): 033015.

[6] Ahn Y Y, Bagrow J P, Lehmann S. Link communities reveal multiscale complexity in networks[J]. Nature, 2010, 466(7307): 761-764.

[7] Liu W, Suzumura T, Ji H. Finding overlapping communities in multilayer networks[J]. Plos One, 2018, 13(4): e0188747.

[8] Mondragon R J, Iacovacci J, Bianconi G. Multilink communities of multiplex networks[J]. Plos One, 2018, 13(3): e0193821.

[9] Gregory S. Finding overlapping communities in networks by label propagation[J]. New Journal of Physics, 2010, 12(10): 103018.

[10] Xie J R, Szymanski B K, Liu X. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process[C]. Proceedings of IEEE ICDM Workshop on DMCCI. Vancouver: IEEE, 2011: 344-349.

[11] Wu Z H, Lin Y F, Gregory S, et al. Balanced multi-label propagation for overlapping community detection in social networks[J]. Journal of Computer Science and Technology, 2012, 27(3): 468-479.

[12] 张健沛,邓琨,杨静,等. 基于边标签传播的复杂网络社区识别方法[J]. 电子学报, 2015, 43(6): 1113-1118.
(Zhang J P, Deng K, Yang J, et al. Community detection in complex networks based on link label propagation[J]. Acta Electronica Sinica, 2015, 43(6): 1113-1118.)

[13] Leung I X Y, Hui P, Lio P, et al. Towards real-time community detection in large networks[J]. Physical Review E, 2009, 79(6): 066107.

[14] Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms[J]. Physical Review E, 2008, 78(4): 046110.

[15] Brutz M, Meyer F G. A flexible multiscale approach to overlapping community detection[J]. Social Network Analysis & Mining, 2016, 5(1): 1-17.

[16] Liakos P, Ntoulas A, Delis A. COEUS: Community detection via seed-set expansion on graph streams[C]. Proceedings of IEEE International Conference on Big Data. Boston: IEEE, 2017: 8257983.

[17] Xie J R, Kelley S, Szymanski B K. Overlapping community detection in networks: The state of the art and comparative study[J]. Acm Computing Surveys, 2011, 45(4): 115-123.

[18] Newman M E J. Network data from Mark Newman's home page[EB/OL]. (2013-04-19)[2018-05-10]. http://www-personal.umich.edu/∼mejn/netdata/.

[19] Guimera R, Danon L, Diaz-guilera A, et al. Self-similar community structure in a network of human interactions[J]. Physical Review E, 2003, 68(6): 065103.

[20] Boguna M, Pastor-satorras R, Diaz-guilera A, et al. Models of social networks based on social distance attachment[J]. Physical Review E, 2004, 70(5): 056122.

作者简介

邓琨(1980−), 男, 副教授, 博士, 从事复杂网络结构分析、数据挖掘等研究, E-mail: dengkun@hrbeu.edu.cn;

李文平(1979−), 男, 副教授, 博士, 从事数据挖掘、私隐保护等研究, E-mail: liwenping@hrbeu.edu.cn;

陈丽(1975−), 女, 副教授, 博士, 从事机器学习、数据挖掘等研究, E-mail: chenli0506@gmail.com;

刘星妍(1980−), 女, 高级工程师, 硕士, 从事数据挖掘、软件测试等研究, E-mail: liuxingyan1980@163.com.

(责任编辑: 郑晓蕾)