



研究与开发

融合节点分析与边分析的复杂网络社区识别算法

邓琨^{1,2}, 蒋庆丰³, 刘星妍¹

- (1. 嘉兴学院信息科学与工程学院, 浙江 嘉兴 314001;
2. 嘉兴学院浙江省医学电子与数字健康重点实验室, 浙江 嘉兴 314001;
3. 常熟理工学院计算机科学与工程学院, 江苏 常熟 225500)

摘要: 针对边社区识别与节点型社区识别两类算法在识别社区过程中均存在相应缺陷, 影响复杂网络社区识别质量的问题, 提出融合节点分析与边分析的复杂网络社区识别 (CDHNE) 算法。该算法首先运用边在网络中较为稳定的特点, 在算法执行初期通过边社区识别构建较为准确的社区结构; 然后利用节点较为灵活的特点, 在边社区形成后, 对边社区的边缘进行精确识别, 更准确地识别出复杂网络中的社区结构。在计算机生成网络实验中, 当网络的社区结构逐渐变得模糊、重叠节点数量与重叠节点归属社区数量不断增加时, CDHNE 算法的社区识别精度较传统算法平均提高 10%, 在重叠节点识别精度上较传统算法平均提高 15%; 在真实网络实验中, 算法识别的社区结构紧密度较好, 特别是面对拥有十几万个节点的大规模网络时, CDHNE 算法高质量地完成了识别任务, EQ 值达到 0.412 1。实验结果表明, CDHNE 算法在运行稳定性和处理大规模网络方面具有优势。

关键词: 复杂网络; 社区识别; 边社区; 节点分析; 边分析

中图分类号: TP391

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2023091

Community detection algorithm of hybrid node analysis and edge analysis in complex networks

DENG Kun^{1,2}, JIANG Qingfeng³, LIU Xingyan¹

1. College of Information Science and Engineering, Jiaxing University, Jiaxing 314001, China
2. Key Laboratory of Medical Electronics and Digital Health of Zhejiang Province, Jiaxing University, Jiaxing 314001, China
3. School of Computer Science and Engineering, Changshu Institute of Technology, Changshu 225500, China

Abstract: The community detection of hybrid node analysis and edge analysis in complex networks (CDHNE), a novel community detection algorithm, was proposed aiming at the problem that both edge community detection and node-based community detection algorithms had corresponding shortcomings in the process of detecting communities,

收稿日期: 2022-12-18; 修回日期: 2023-04-12

通信作者: 邓琨, dengkun@hrbeu.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61370083); 教育部人文社会科学研究专项任务项目 (No.22JDSZ3023); 浙江省教育科学规划课题项目 (No.2020SCG046); 教育部产学研合作协同育人项目 (No.220603372015422, No.220604029012441)

Foundation Items: The National Natural Science Foundation of China (No.61370083), The Humanity and Social Science Research Project of Ministry of Education of China (No.22JDSZ3023), Zhejiang Educational Science Planning Project (No.2020SCG046), The Ministry of Education's Industry-University Collaboration Education Project (No.220603372015422, No.220604029012441)



which affected the quality of complex network community detection. The relatively stable characteristics of the edge in the networks were firstly used by the algorithm to construct a more accurate community structure through edge community detection at the early stage of algorithm execution. Then, after the formation of the edge communities, the flexible characteristics of the node were used to accurately detect the boundary of edge communities, so as to more accurately detect the community structure in the complex networks. In the computer-generated network experiments, when the community structure of the network gradually became fuzzy, the number of overlapping nodes and the number of communities to which the overlapping nodes belonged kept increasing. Compared to traditional algorithms, the accuracy of community detection and overlapping nodes detection were improved by an average of 10% and 15%, respectively, by the CDHNE algorithm. In the real network experiments, the tightness of the community structure detected by the CDHNE algorithm was better. Especially when facing large-scale networks with more than 100 000 nodes, the detection task was completed by the CDHNE algorithm with high quality, and the EQ value reached 0.412 1. The experimental results show that the CDHNE algorithm has advantages in operational stability and handling large-scale networks.

Key words: complex network, community detection, edge community, node analysis, edge analysis

0 引言

现实世界中的诸多复杂系统均能抽象为复杂网络,系统中的个体可作为复杂网络中的节点,边代表个体之间的连接关系,如人与人之间交往的社会网络、通信终端之间相连的移动互联网络等。目前,针对复杂网络的研究吸引了各领域的学者展开探索,产生了大量学术研究成果,得出了复杂网络中的重要特性。本文研究的“社区结构特性”是复杂网络中一个重要特性。社区可以理解为一组拥有相似功能的节点集合,这些节点在社区内彼此紧密连接,社区间的连接较为疏松^[1]。复杂网络社区识别就是要揭示复杂网络中的真实社区结构,其对于发现复杂网络中的隐含规律和运行机理具有重要理论意义。

在现实世界中,与节点仅属于单一社区的传统识别方法相比,重叠社区识别算法将节点同时识别到多个社区中,例如,一名学生可以同时属于家庭社区及同学社区,一个教师可同时属于物理领域和数学领域等,因此针对重叠社区识别的算法研究通常更具现实意义^[2],相关研究成果也层出不穷。例如派系渗透方法(clique percolation method, CPM)^[3]认为社区内部的边有较大概率形成大的完全子图,而社区之间的

边很难形成完全子图,鉴于此,CPM通过派系渗透的方式完成社区识别任务。近期 Sathiyakumari 等^[4]提出 k 核与派系渗透结合的混合算法,该算法在随机图中执行最大派系枚举过程,在此过程中这些强相邻派系被融合成自然互连的簇,其主要目标是通过派系渗透中 k 核社区的最佳得分值发现有效互连的社区,在实验中发现通过上述操作,该算法的社区识别质量优于常规的派系渗透算法,其他相关算法还包括文献[5-6]所提算法等。基于种子扩展的兰奇基内蒂-福尔图纳托方法(Lancichinetti-Fortunato method, LFM)^[7]在定义适应度函数的基础上,使算法沿着适应度函数增加的方向扩展社区,直到网络中任意节点均归属相应社区,若任意节点在加入其他社区后,其适应度函数值均为负,则算法停止。因为每个社区在扩展过程中都较为独立,且节点能被识别到不同社区中,因此,LFM可以识别出重叠社区。最近,Asmi 等^[8]提出一种基于耦合种子扩展的重叠社区识别算法,该算法通过选择某一节点及其最相似的邻居构造一个耦合种子,然后使用适应度函数扩展此耦合种子,从而提高社区的识别质量,相关算法还包括文献[9-10]所提算法等。社区重叠传播算法(community overlap propagation algorithm, COPRA)^[11]首先为网络中全部节点初始化标签及社

区隶属系数,然后在标签传播过程中将每个节点的相邻节点社区隶属系数进行加权平均,以更新该节点的社区隶属系数,每个节点依据其社区隶属系数选择标签,当算法停止运行时,标签相同的节点为一个社区,最终拥有多个标签的节点确定为重叠节点。COPRA中设置了参数 ν 用于控制节点可以同时归属的最大社区数,基于标签传播思想的算法还包括文献[12-13]所提算法等。上述算法均以节点为研究对象考虑社区识别问题,其优势是节点在社区识别过程中较为灵活,更容易识别到每个节点的所属社区。然而此类算法的缺陷也同样明显,研究表明^[2],由于节点具有灵活的特点,节点型社区识别算法很难在算法执行初期形成初始社区结构,使得算法收敛效果欠佳,影响社区识别质量;另外,此类算法往往需要预设参数阈值以保证算法的收敛效果,但在纷繁复杂的网络中,预先掌握先验知识,以准确设置参数也是难以实现的。

鉴于以节点为研究对象的社区识别算法所存在的缺陷,Ahn等^[14]提出以边为研究对象的边聚类(link clustering, LC)算法,该算法的基本思想是,边在网络中通常具有单一角色而属于唯一社区,当网络中的边被识别到不同社区后,其相应的重叠节点也将自然显现。LC算法通过分析边的相似度将边进行聚类得到具有层次结构的边社区,并通过对边进行切割得到最终的社区结构,相关研究还包括文献[15-16]等。需要指出,在边社区识别过程中,一条边连接两个节点,同时受两个节点的共同牵制,若将任意一条边归属于一个社区,需要考虑的因素较多,因此边社区识别具有较为稳定的优势。然而在此类算法所识别的社区结构中,存在过多的重叠节点,产生“过度重叠”现象^[2],造成社区之间的边界不清,从而影响了社区识别的准确性。

不难看出,以节点为研究对象的社区识别方法和边社区识别方法均存在各自的缺陷,影响了社区识别的质量。因此,融合边与节点在社区识别中的各自优势,取长补短,以提升社区识别能力,将成

为一个较为合理的解决方案。鉴于此,本文提出一种融合节点分析与边分析的复杂网络社区识别(community detection of hybrid node analysis and edge analysis in complex networks, CDHNE)算法。CDHNE算法在社区识别初期发挥边较为稳定的优势,采用边社区识别的方式快速形成初始社区;然后发挥节点更为灵活的优势,以节点型社区识别方式处理边社区的边界区域,以提高识别重叠节点的准确率,最终完成社区识别任务。

1 定义与定理

在给出CDHNE算法的详细描述前,先给出如下定义与定理。设网络 $G=(V,E)$,其中 V 代表网络 G 的节点集合, E 为网络 G 的边集合。 $e_{v,v'}$ ($e_{v,v'} \in E$)代表连接节点 v 和 v' 的边,其中 $v, v' \in V$ 。

定义1 (邻居边) 边 $e_{v,v'}$ 的邻居边 $\text{adj}(e_{v,v'})$ 为分别与节点 v 与 v' 相连接,并不包含 $e_{v,v'}$ 的边集合,即 $\text{adj}(e_{v,v'}) = \{e_{v,i} \in E \mid i \in \text{adj}(v)\} \cup \{e_{j,v'} \in E \mid j \in \text{adj}(v')\} - \{e_{v,v'}\}$,其中, $\text{adj}(v)$ 代表节点 v 的邻居节点集合。

定义2 (紧密连接边) 边 $e_{v,v'}$ 的紧密连接边 $\text{TA}(e_{v,v'})$ 是与两个节点 v 、 v' 的共同邻居点 u 相连接的边集合,即 $\text{TA}(e_{v,v'}) = \{e_{v,u}, e_{v',u} \in E \mid u \in \text{adj}(v) \cap \text{adj}(v')\}$ 。

定义3 (紧密连接组) 设 $\text{TA}(e_{v,v'})$ 是边 $e_{v,v'}$ 与其共同邻居节点 u 的紧密连接边,则紧密连接组 $\text{TG}(e_{v,v'})$ 可表示为:

$$\text{TG}(e_{v,v'}) = \text{TA}(e_{v,v'}) \cup \{e_{v,v'}\} \quad (1)$$

定义4 (社区密度)^[14] 设社区 C_i 有 m 条边以及 n 个节点,其 C_i 的社区密度可定义为:

$$D_{C_i} = \frac{m - (n - 1)}{n(n - 1) / 2 - (n - 1)} \quad (2)$$

其中,若 $n = 2$,则 $D_{C_i} = 0$ 。

社区密度是一种社区识别质量的度量函数,若社区密度增加,则其社区识别质量也将提高,



根据社区密度, 在此给出如下定理与函数。

定理 1 设 C_i 是一个边社区, m_i 、 n_i 为 C_i 所含的边数及节点数, 若将 $e_{v,v'}$ 加入其紧密连接边所在的社区 C_i , 使得社区 C_i 中的紧密连接组数量增加, 则 $e_{v,v'}$ 加入社区使 C_i 形成的新社区 $C_{i'}$ 对应的社区密度 $D_{i'}$ 大于 C_i 对应的社区密度 D_i 。

证明 社区 $C_{i'}$ 与社区 C_i 的社区密度增量值 $\Delta D_{C_{i'}-C_i}$ 可表示为:

$$\Delta D_{C_{i'}-C_i} = D_{i'} - D_i \quad (3)$$

$$\Delta D_{C_{i'}-C_i} = \frac{m_{i'} - (n_{i'} - 1)}{n_{i'}(n_{i'} - 1) / 2 - (n_{i'} - 1)} - \frac{m_i - (n_i - 1)}{n_i(n_i - 1) / 2 - (n_i - 1)} \quad (4)$$

根据定义 2 可知, 若将边 $e_{v,v'}$ 加入社区 C_i 形成社区 $C_{i'}$, 使得 $m_{i'} = m_i + 1$, $n_{i'} = n_i$ 。

因此, $\Delta D_{C_{i'}-C_i} > 0$, 故 $D_{i'} > D_i$ 。

证毕。

通过定理 1 可知, 将边加入其紧密连接边所在的社区, 以形成紧密连接组, 可以使社区密度增加, 又由于社区密度函数是评价边社区的度量函数, 因此, 可以说明将边加入其紧密连接边所在的社区并形成紧密连接组, 可以提高社区的识别质量。然而, 若一条边拥有多个紧密连接边, 而这些紧密连接边分别隶属于不同社区, 会使该边的隶属社区存在歧义。因此, 根据社区特性, 最合理的做法是将边加入与其紧密连接边紧密连接的社区中。鉴于此, 给出边社区增量函数的定义。

定义 5 (边适应函数) 设社区 C_i 是一个边社区, $|\text{TG}(C_i)|$ 表示社区 C_i 中的紧密连接组的数量, $|E_{C_i}|$ 表示 C_i 中边的数量, 其 C_i 的边适应函数 $\text{EF}(C_i)$ 可被定义为:

$$\text{EF}(C_i) = \frac{|\text{TG}(C_i)|}{|E_{C_i}|} \quad (5)$$

定义 6 (边社区增量函数) 设社区 C_i 是一个边社区, $\text{EF}(C_i)$ 是社区 C_i 的边适应函数, 若将边 $e_{v,v'}$ 加入社区 C_i , 形成社区 $C_{i'}$, 其 $C_{i'}$ 的边适应

函数为 $\text{EF}(C_{i'})$, 则 $e_{v,v'}$ 加入 C_i 的边社区增量函数可定义为:

$$\Delta \text{EF}(e_{v,v'}, C_i) = \text{EF}(C_{i'}) - \text{EF}(C_i) \quad (6)$$

在设计 CDHNE 算法时, 将沿着使边社区增量函数增加的方向将网络中的边识别到不同社区中。

定义 7 (紧密连接度量) 设社区 C_i 中包括 m_i 条边与 n_i 个节点, 则紧密连接度量 F_{C_i} 可表示为:

$$F_{C_i} = 3n_i - 2m_i - 4 \quad (7)$$

定理 2 设边 $e_{u,v}$ 为社区 C_i 中的一条边, 若将其邻居边归入社区 C_i 中可以与 $e_{u,v}$ 构成紧密连接组 $\text{TG}(e_{u,v})$, 形成新社区 $C_{i'}$, 此时若 $C_{i'}$ 的紧密连接度量 $F_{C_{i'}} > 0$, 则 $C_{i'}$ 对应的社区密度 $D_{i'}$ 大于 C_i 对应的社区密度 D_i 。

证明 设 C_i 中所包含的边数与节点数分别为 m_i 、 n_i , $C_{i'}$ 中的边数与节点数为 $m_{i'}$ 和 $n_{i'}$, 那么社区 $C_{i'}$ 对于社区 C_i 的社区密度的增量 $\Delta D_{C_{i'}-C_i}$ 可表示为:

$$\Delta D_{C_{i'}-C_i} = \frac{m_{i'} - (n_{i'} - 1)}{n_{i'}(n_{i'} - 1) / 2 - (n_{i'} - 1)} - \frac{m_i - (n_i - 1)}{n_i(n_i - 1) / 2 - (n_i - 1)} \quad (8)$$

由紧密连接组的定义可知, 若将边 $e_{u,v}$ 的邻居边加入社区 C_i 构成紧密连接组 $\text{TG}(e_{u,v})$ 形成新社区 $C_{i'}$, 可以使得 $m_{i'} = m_i + 2$, $n_{i'} = n_i + 1$, 将其代入式 (8), 可得到:

$$\Delta D_{C_{i'}-C_i} = \frac{m_i + 2 - n_i}{(n_i + 1)n_i / 2 - n_i} - \frac{m_i - (n_i - 1)}{(n_i - 1)(n_i - 2) / 2} \quad (9)$$

通过计算可以得到:

$$\Delta D_{C_{i'}-C_i} = \frac{(m_i + 2 - n_i)(n_i - 2)}{n_i(n_i - 1)(n_i - 2) / 2} - \frac{(m_i - n_i + 1)n_i}{n_i(n_i - 1)(n_i - 2) / 2} \quad (10)$$

对式 (10) 进行整理可得到:

$$\Delta D_{C_{i'}-C_i} = \frac{3n_i - 2m_i - 4}{n_i(n_i - 1)(n_i - 2)} \quad (11)$$

由社区密度定义可知, 若社区中节点的数量

是2, 社区密度为0, 而在一个社区中其节点数 n_i 通常大于2, 因此式(11)的分子大于0; 此时若 $3n_i - 2m_i - 4 > 0$, 则 $\Delta D_{C_i, -C_i} > 0$, 即可增加社区密度。

证毕。

根据定理2, 若一个社区中的边能够与其邻居组成紧密连接组, 并且将这些边加入社区中, 这能够使得紧密连接度量增加, 则可以增加社区密度。

定义8 (节点社区增量函数) 设 $|m_{in}|$ 是社区 C_i 内的边数, $|m_{out}|$ 是不属于社区 C_i 并与社区 C_i 内节点相连的边数, 若将节点 i 加入社区 C_i 形成社区 $C_{i'}$, 其社区 $C_{i'}$ 内的边数为 $|m_{in+i}|$, 社区 $C_{i'}$ 外的节点与社区 $C_{i'}$ 内节点相连的边数是 $|m_{out+i}|$, 则节点 i 加入社区 C_i 的增量 $\Delta T(i, C_i)$ 可表示为:

$$\Delta T(i, C_i) = \frac{m_{in+i}}{m_{out+i}} - \frac{m_{in}}{m_{out}} \quad (12)$$

根据节点社区增量函数定义可知, 若某节点加入社区后, 如果使得社区增量函数增加, 则会相应增加社区内边数与社区外边数的比例值, 进而提高社区内部的连接紧密度。

在CDHNE算法中, 在完成边社区识别任务时, 若网络中依然存在未被识别到任何社区的节点(以下称为孤立节点)或“过度重叠”现象时, 本文运用节点社区增量函数分析节点所在社区, 即沿着使节点社区增量函数增加的方向将网络中的孤立节点划分到不同的社区中; 另外, 针对重叠节点展开分析, 将重叠节点划分到能够使节点社区增量函数增加的社区中, 以进一步提高社区识别质量。

2 CDHNE 算法

第1节已给出CDHNE算法涉及的相关定理与定义, 本节将围绕这些定理与定义完成CDHNE算法的设计, 主要包括初始化边社区、扩展边社区及扩展节点社区3个阶段。

2.1 初始化边社区

由于局部中心节点往往被认为是在局部范围内拥有较高节点度的节点^[2], 因此, 初始化边社区算法首先搜索网络中节点度较高的若干节点作为初始化边社区阶段的初始节点, 然后将与中心节点相连的边归属为同一社区, 完成初始化边社区阶段。具体过程如算法1所示。

算法1 边社区初始化算法

输入: 复杂网络 $G=(V, E)$ // V 为 G 的节点集合, E 为 G 的边集合

输出: LC //初始化边社区集合

begin

所有节点的 isNotVisit 设置为 true;

while $v_i \in V$ and v_i .isNotVisit

v_i .isNotVisit $\leftarrow$ false;

$\text{adj}(v_i) \leftarrow \{v_u \mid v_u \in V \wedge e_{v_i, v_u} \in E\}$;

$v_j \leftarrow \text{adj}(v_i)$ 中节点度最大的节点;

if v_j .degree $\geq v_i$.degree

$(\forall v_k \in \text{adj}(v_i) - \{v_j\}).\text{isNotVisit} \leftarrow$
false; //在 $\text{adj}(v_i)$ 中去除 v_j 后, 剩余节点的 is Not
Visit 标记为 false

$v_i \leftarrow v_j$;

else

$\text{LC}_i \leftarrow \{e_{v_i, v_j} \mid v_j \in \text{adj}(v_i)\}$;

$\text{LC} \leftarrow \text{LC} \cup \text{LC}_i$;

$(\forall v_k \in \text{adj}(v_i)).\text{isNotVisit} \leftarrow$ false;

if $v_s \in V \wedge v_s$.isNotVisit

$v_i \leftarrow v_s$;

end if

end if

end while

end

算法1首先从网络中任意节点 v_i 开始搜索, 若其周围邻居节点中度最大节点 v_j 的度大于或等于当前节点 v_i 的度, 则将 v_j 作为当前节点, 继续搜索邻居节点集合中度更大的节点, 反复执行这



一过程,直到当前节点为邻居节点集合中度最大的节点,从而将当前节点作为一个中心节点,并将与中心节点存在连接的边集合归属于一个初始社区。然后,在算法从未被搜索过的节点集合中,继续搜索中心节点,并形成初始社区,反复执行这一过程,直到网络中全部节点均被搜索过,则算法 1 停止执行。在此以图 1 为例更直观地说明本算法,在图 1 中随机取出节点 v_4 作为搜索的开始点,节点 v_4 的邻居节点 v_3 的度为 2,是节点 v_4 及其邻居节点集合中度最大的节点,此时以节点 v_3 为当前节点,搜索节点 v_3 的邻居节点,节点 v_3 的邻居节点集合中节点 v_1 的度为 4,是节点 v_3 及其邻居节点集合中度最大的节点,再以节点 v_1 为当前节点继续进行搜索,此时节点 v_1 已经是其邻居节点集合中度最大的节点,则将节点 v_1 作为 1 个中心节点,并将与节点 v_1 存在连接的边 e_{v_1,v_2} 、 e_{v_1,v_3} 、 e_{v_1,v_5} 、 e_{v_1,v_6} 归属为一个社区。在未被搜索过的区域中,反复执行上述过程,直到网络中所有节点均被搜索过,则算法停止执行。可以看出,算法 1 执行完成后网络中若干连接紧密的初始社区被找到,能够在扩展边社区阶段围绕初始社区进行扩展,使算法在运行过程中更具指向性,便于算法快速收敛。

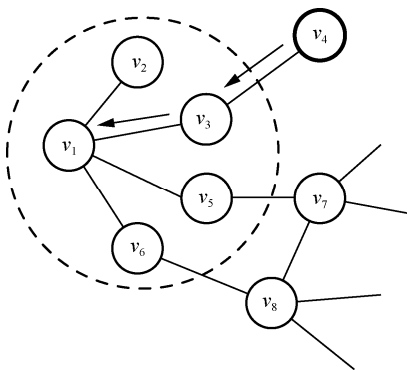


图 1 边社区初始化算法示意图

2.2 扩展边社区

由定理 1 可知,若将一条边归属于能够使边社区增量函数增加的社区,则能够提升社区密度;根据定理 2,若社区中的边与其周围邻居边能构成紧密连接组,并且加入社区后能够使紧密连接度

量大于 0,则社区密度也会增加。由定义 4 可知,社区密度是社区识别质量的度量函数,若社区密度增加,则其社区识别质量也将提高,因此扩展边社区阶段是在定理 1 与定理 2 的指引下完成的。具体过程如算法 2 所示。

算法 2 边社区扩展算法

输入: 复杂网络 $G=(V,E)$, LC // LC 为初始社区集合

输出: 扩展后的边社区 LC

begin

while $LC_i \in LC$

$Q_i \leftarrow \{e_{v_i,v_j} \mid v_i \in LC_i \wedge v_j \in \text{adj}(v_i) \wedge e_{v_i,v_j} \notin LC_i \wedge LC_i \in LC\}$

$Q_i \leftarrow Q_i$;

while $Q_i.\text{count} \neq 0$ and $e_{u,v} \in Q_i$

if $\Delta EF(e_{u,v}, LC_i) > 0 // \Delta EF(LC_i \leftarrow e_{u,v})$

是 $e_{u,v}$ 与社区 LC_i 的边社区增量值

$LC_i \leftarrow LC_i \cup \{e_{u,v}\}$;

$Q_i \leftarrow Q_i - \{e_{u,v}\}$;

end if

end while

while $Q_i.\text{count} \neq 0$ and $e_{u,v} \in LC_i$

if $TG(e_{u,v}).\text{count} > 0 \wedge TA(e_{u,v}) \subseteq$

$Q_i \wedge \Delta T_{LC_i \cup TA(e_{u,v})} > 0$

$LC_i \leftarrow LC_i \cup TA(e_{u,v})$;

$Q_i \leftarrow Q_i - TA(e_{u,v})$;

end if

end while

while $Q_i \neq \emptyset$

$Q_i \leftarrow \emptyset$;

end while

end while

end

算法 2 将算法 1 产生的初始化边社区作为输入,在算法 1 产生的初始化边社区基础上,将未被识别到任何社区中的边(以下称为孤立边)归

属到能够使边社区增量函数增加的社区；此时，若网络中依然存在孤立边，则将孤立边归入与其存在紧密连接组，同时可以使紧密连接度量大于0的社区中。算法2执行完成后，由定理1、定理2可知，将会产生社区密度较高的边社区结构。扩展边社区阶段的程序流程如图2所示。

2.3 扩展节点社区

经过扩展边社区后，网络中可能依然存在孤立边，以及由于边社区的特点，社区结构中会存在“过度重叠”现象。此时，将边社区转换为节点社区，利用节点较为灵活的特点分析孤立节点的归属社区，以及重叠节点归属社区的合理性。具体过程如算法3所示。

算法3 节点社区扩展算法

输入：复杂网络 $G=(V,E)$, LC // LC 为执行完算法2所得到的边社区结构

输出：最终社区 NC

begin

NC $\leftarrow$ LC 边社区转换为节点社区；

while $NC_i \in NC$

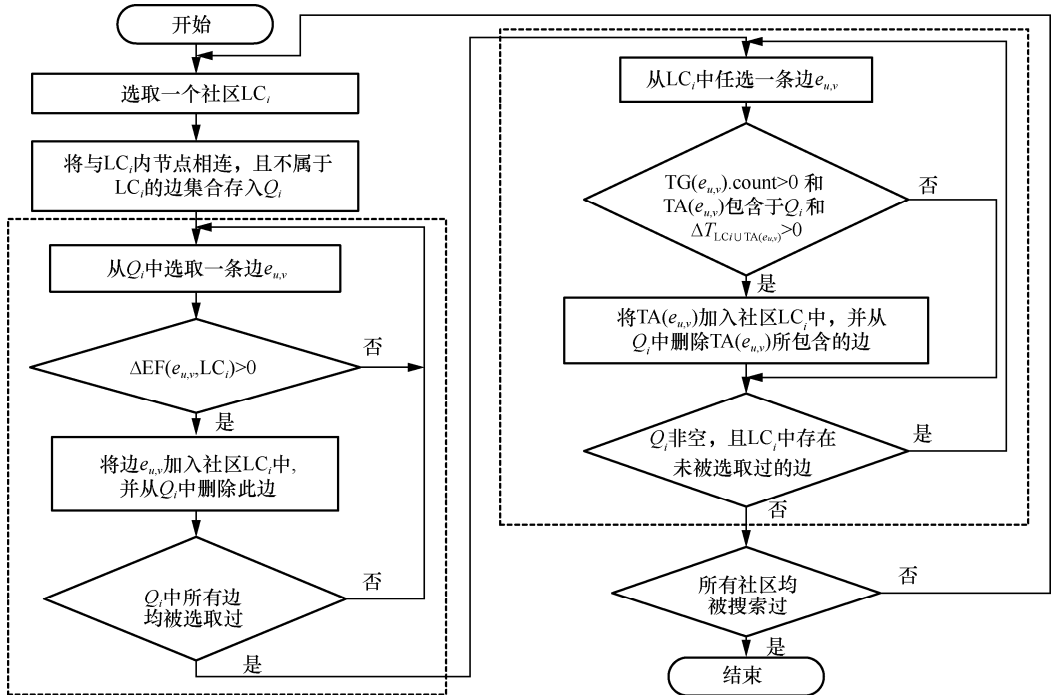


图2 扩展边社区阶段的程序流程

```

V_{C_i} ← {x | ∃ e_{x,y} ∈ E ∧ y ∈ NC_i ∧ x ∉ NC_i};
while v_i ∈ V_{C_i}
  if ΔT(v_i, NC_i) ≥ 0
    NC_i ← NC_i ∪ {v_i};
    V_{C_i} ← {x | ∃ e_{x,y} ∈ E ∧ y ∈ NC_i ∧ x ∉ NC_i};
  end if
end while
while ∃ v_k ∈ V ∧ {NC_i | NC_i ∈ NC ∧ v_k ∈ NC_i}.count ≥ 2
  C_n ← {NC_m | NC_m ∈ NC ∧ ΔT(v_k, NC_m) > 0}
  if C_n.count > 0
    C_m ← {NC_m | NC_m ∈ NC ∧ ΔT(v_k, NC_m) ≤ 0}
    while ∃ v_k ∈ NC_m ∧ NC_m ∈ C_m
      NC_m ← NC_m - {v_k}
    end while
  end if
end while
while ∃ v_i ∈ V ∧ v_i ∉ {NC_j | NC_j ∈ NC}

```



$$C_n \leftarrow \{NC_k \mid NC_k \in NC \wedge (\exists e_{v_i, v_j} \in E \wedge v_j \in NC_k) \wedge \Delta T(v_i, NC_k) < 0\};$$

$$C_m \leftarrow \{NC_k \mid NC_k \in C_n \wedge (\forall NC_t \in C_n \wedge \Delta T(v_i, NC_t) \leq \Delta T(v_i, NC_k))\};$$

$$NC_x \leftarrow \text{从 } C_m \text{ 中随机选取一个值};$$

$$NC_x \leftarrow NC_x \cup \{v_i\};$$

end while

end while

end

算法 3 首先反复计算孤立节点与周围邻接社区的节点社区增量函数，并将节点归入与其存在节点社区增量函数大于 0 的社区；然后，算法进一步分析社区中的重叠节点，计算重叠节点与其所属社区的节点社区增量函数，若存在重叠节点与某一所属社区的节点社区增量函数小于 0 的情况，则将此重叠节点从该社区中删除；经过上述操作，若依然存在孤立节点，则说明孤立节点与其周围邻接社区的社区增量函数均小于 0，此时将孤立节点加入使节点社区增量函数降低最小的社区中。节点社区扩展阶段的执行流程如图 3 所示。

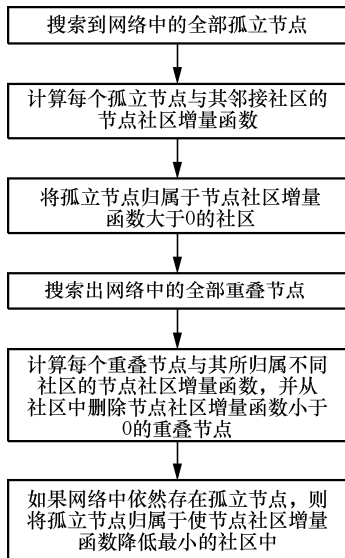


图3 节点社区扩展阶段的执行流程

2.4 CDHNE 算法描述

经过上述分析，CDHNE 算法具体包括 3 个阶段：初始化边社区阶段、扩展边社区阶段、扩展

节点社区阶段。

初始化边社区阶段以网络中若干度较高节点为算法的起始点，并将与度较高节点存在连接的边归入同一社区，其原因是度较高节点往往具有较大的吸引能力，能将周围邻居快速吸引，同时度较高节点周围通常较为稠密，便于算法辐射与扩展，因此 CDHNE 算法首先采用初始化边社区阶段作为算法的起始阶段。

扩展边社区阶段在定理 1 与定理 2 的引导下，通过对边社区增量函数与紧密连接度量的分析，不断优化社区密度函数，能够识别到连接紧密的边社区结构。

由于边社区识别特点，网络中依然会存在孤立节点以及“过度重叠”现象，此时，基于边的社区识别往往难以准确区分这些节点所在社区，因此，采用优势互补的思想，在社区结构已较为清晰时，通过节点社区增量函数分析节点归属社区的合理性，最终完成社区识别任务。

2.5 CDHNE 算法的时间复杂度

设 m 、 n 分别为网络 G 中的边数与节点数， k 为网络中节点的平均度， c 为网络中的核心节点数，每个社区中的边数可表示为 m_c ，社区数量为 t ，在完成边社区识别后，其孤立节点与重叠节点数为 o 。CDHNE 算法各阶段的时间复杂度分析见表 1。

由表 1 可知，CDHNE 算法的时间复杂度约为 $O(ck + nk^2 + 2mkt + 2om_c)$ ，由于核心节点数 c 、孤立节点数与重叠节点数 o 、社区中的边数 m_c 以及社区数量 t 远小于网络中的节点数 n 与边数 m ，从而 CDHNE 算法的时间复杂度约为 $O(q(nk^2 + mk))$ ，其中 q 为常数。

3 实验分析

3.1 实验方案

为验证 CDHNE 算法的有效性，本文在计算机生成网络 LFR benchmark^[17]以及真实网络中对各算法进行分析，并将 CDHNE 算法与基于 CPM

表1 CDHNE 算法各阶段的时间复杂度分析

算法阶段	分析过程	时间复杂度
初始化边社区阶段	每个节点需要分析 k 个邻居节点的度, 因此, 其时间复杂度为 $O(k^2)$, 那么分析网络 G 中 n 个节点的时间复杂度为 $O(nk^2)$, 将与 c 个核心节点连接的边归入社区的时间复杂度为 $O(ck)$, 因此, 初始化边社区的时间复杂度约为 $O(ck + nk^2)$	$O(ck + nk^2)$
扩展边社区阶段	分析每条边与其邻接社区存在紧密连接边与紧密连接组的时间复杂度约为 $O(2k)$, 计算 m 条边的边社区增量函数的时间复杂度约为 $O(mt)$, 因此扩展边社区阶段的时间复杂度不会超过 $O(2mkt)$	$O(2mkt)$
扩展节点社区阶段	分析孤立节点、重叠节点与其邻接社区的节点社区增量函数的时间复杂度约为 $O(2m_c)$, 分析 o 个孤立节点与重叠节点的时间复杂度约为 $O(2om_c)$	$O(2om_c)$

实现的派系搜索社区识别软件 (Clique FINDER, CFINDER)^[3]、多尺度 (multi-scale, MS) 算法^[18]、基于图流种子集扩展算法 (community detection via seed-set expansion on graph streams, CoEuS)^[10]以及标签传播控制算法 (overlapping community detection algorithm with label propagation control, OLPC)^[19]进行对比分析, 各算法相关信息见表2, 其中, CFINDER 的参数设置依据文献[3], 派系规模 $k=3\sim6$, 间隔为1, 选取算法在不同参数指标下的最大值作为最终结果。本文在每个网络中均对各算法运行10次, 选取各算法10次评价指标最优值相加取平均值的结果作为其最终对比结果。实验在 Intel Core i5-6200U CPU 2.30 GHz 以及 8 GB 内存的笔记本计算机上进行。

表2 各算法相关信息

算法	开发环境	提出时间
CFINDER	C++	2005年
MS	Java	2016年
CoEuS	Java	2017年
OLPC	MATLAB	2019年
CDHNE	MATLAB	2023年

为更好地评价各算法的性能, 本文选取统一化互信息 (normalized mutual information, NMI) 函数^[7]、重叠节点识别精度指标 F-Score 函数^[20]以及扩展模块度 EQ (extend Q)^[21]函数, 分别从社区识别准确率、重叠节点识别精度以及社区识别紧密度3个方面评价各算法的优劣。

(1) NMI 函数是将执行算法所得到的社区结构

与真实社区结构进行对比, 评价社区识别准确率的指标, 其中 NMI 的取值为0~1, 如果 NMI 取值是1, 则执行算法所获得的社区结构与真实社区结构完全一致, 反之, 如果 NMI 的取值是0, 则执行算法所取得的社区结构与真实社区结构完全不同。

(2) F-Score 函数是评价各算法识别重叠节点精度的社区识别评价指标, 其 F-Score 的取值为0~1, 执行算法所获社区结构的 F-Score 值越高, 则说明算法识别重叠节点准确率越高, 反之若 F-Score 值越低, 则算法识别重叠节点准确率越低。

(3) EQ 函数是评价社区内部节点连接紧密度的评价指标, 若识别社区内部的节点连接越紧密, 则 EQ 值越高。

在进行算法社区识别精度分析时, 需要将算法识别得到的社区结构与真实社区结构进行对比分析得出识别精度结果, 以判断算法的识别能力, 在 LFR benchmark 网络中社区结构是已知的, 因此, 本文在 LFR benchmark 网络中选用 NMI 和 F-Score 两个精度识别指标评价算法的社区识别能力。在真实网络中, 已知社区结构的网络相对较少, 因此, 无法通过精度识别指标对各算法进行分析, 本文选用 EQ 指标评价识别社区结构的紧密度, 以衡量各算法的识别效果。

3.2 基准网络数据集

由于 LFR benchmark 网络与真实网络的分布统计特性非常相似, 本文选择 LFR benchmark 网络作为本文评价各算法的实验测试数据集。从网



络规模、社区内部节点连接紧密性及重叠节点同时归属社区数几个角度出发,设计了4个网络,LFR benchmark网络设置见表3。其中, n 代表网络中的节点数; k 与 $k_{\max}$ 表示网络中节点的平均度

和最大度; $C_{\max}$ 、 $C_{\min}$ 表示网络中的最大社区数以及最小社区数; O_n 和 O_m 为网络中的重叠节点数以及重叠节点可同时归属的社区数; μ 表示混合比例数, μ 取值越大,社区结构越模糊。

表3 LFR benchmark网络设置

网络	n	k	$k_{\max}$	$C_{\max}$	$C_{\min}$	O_n	O_m	μ
R1	5 000	10	30	50	30	500	2	0.1~0.4
R2	5 000	10	30	50	30	2 500	2	0.1~0.4
R3	5 000	10	30	50	30	500	2~6	0.1
R4	5 000	10	30	50	30	500	2~6	0.3

各算法在R1与R2网络中社区识别的NMI值如图4所示。图4(a)给出了各算法在重叠节点同时归属于两个社区,并且重叠节点较少的R1网络中社区识别的NMI值。从图4(a)可以看出,在R1网络中,CDHNE算法在 $\mu=0.1$ 、0.2、0.3、0.4时,其NMI值分别为0.87、0.7、0.56、0.26,除在 $\mu=0.2$ 时,MS算法的NMI值与CDHNE算法相同外,在 μ 取其他值的情况下,CDHNE算法的NMI值均高于其他对比算法。图4(b)给出了各算法在重叠节点仅属于两个社区并且重叠节点较多的R2网络中社区识别的NMI值。从图4(b)能够看出,在R2网络中,CDHNE算法在 $\mu=0.1$ 、0.2、0.3、0.4时,其NMI值分别为0.33、0.29、0.23、0.10,均高于其他对比算法。通过分析可知,在社区结构较清晰的网络中,CDHNE算法的识别准确度可达到0.87,而无论是在重叠节点较多还是较少的网络中,当社区结构非常模糊($\mu=0.4$)时,CDHNE算法相对于其他对比算法,依然能识别出具有较高精度的社区结构,因此说明CDHNE算法在社区识别的过程中较为稳定。

在社区结构较清晰的R3网络,以及社区结构较模糊的R4网络中,随着重叠节点归属社区数量不断增加,各算法在R3与R4网络中社区识别的NMI值如图5所示。从图5可知,在R3网络中,CDHNE算法在 $O_m=2$ 、3、4、5、6时,其NMI值分别为0.87、0.8、0.77、0.75、0.68,在R4网

络中CDHNE算法在 $O_m=2$ 、3、4、5、6时,其NMI值分别为0.56、0.5、0.45、0.4、0.38,可以看出在R3与R4网络中,CDHNE算法的社区识别准确率均高于其他对比算法。

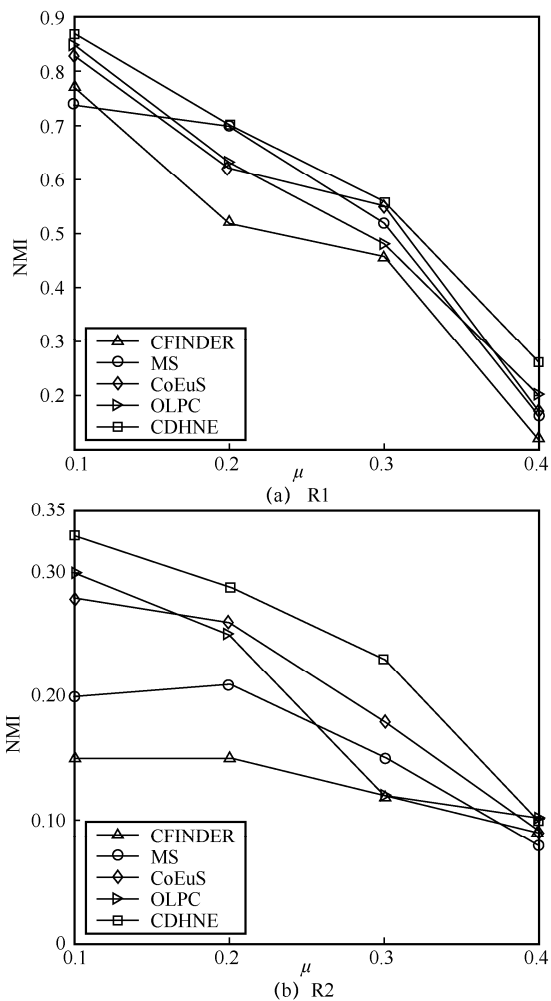


图4 各算法在R1与R2网络中社区识别的NMI值

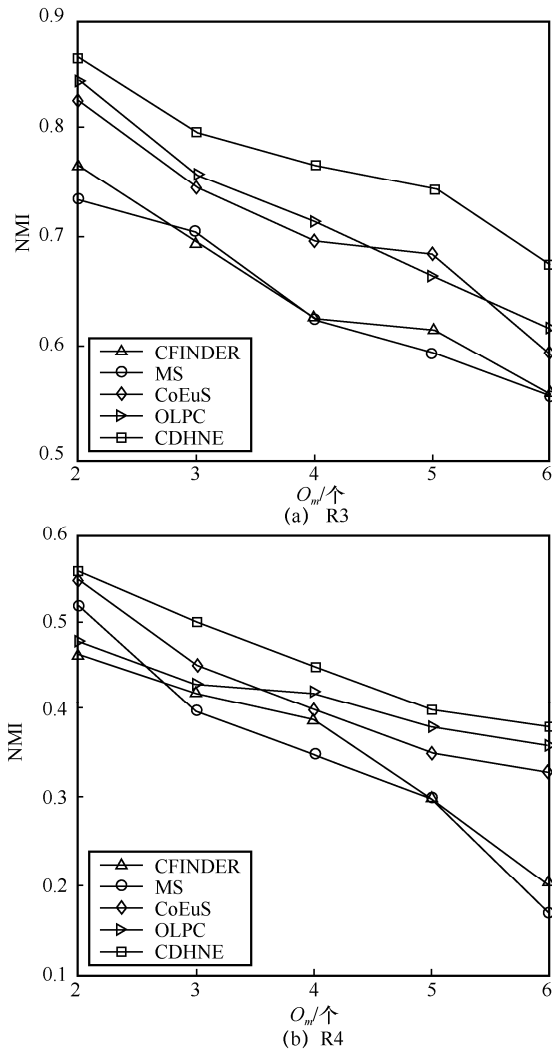


图5 各算法在 R3 与 R4 网络中社区识别的 NMI 值

从图4与图5能够看出, CoEuS 与 CFINDER 算法在识别社区的过程中需要网络中有连接较为紧密的节点集合, 进而根据这些连接紧密的节点集合完成社区识别, 虽然 CoEuS 与 CFINDER 具有较为稳定的社区识别结果, 但由于其对网络结构提出了较高要求, 这两个算法都没有取得较高的社区识别准确率; MS 算法在识别社区过程中, 需要从不同网络粒度进行分析, 但是不同网络粒度之间通常并不容易完成转换分析, 因此, 造成算法的社区识别准确率不高; OLPC 算法虽然在标签传播过程中加入了标签控制等策略, 减小了算法的振荡效应, 但该算法依然存在较高的随机性, 从而影响了算法的社区识别质量; CDHNE

算法吸收了节点型社区识别与边社区识别的优势, 将节点与边的优势应用在社区识别的不同阶段, 分别将边较为稳定以及节点较为灵活的特点运用在社区识别中, 从各算法的运行结果来看, CDHNE 算法较其他传统算法在 R1~R4 各种类型网络中 NMI 的整体平均值至少提高 10%。

各算法在 R1 与 R2 网络中社区识别的 F-Score 值如图6所示, 各算法在 R3 与 R4 网络中社区识别的 F-Score 值如图7所示。在 R1 网络中 $\mu=0.1$ 、0.2、0.3、0.4 时, CDHNE 算法的 F-Score 值分别为 0.58、0.53、0.3、0.3; 在 R2 网络中 $\mu=0.1$ 、0.2、0.3、0.4 时, CDHNE 算法的 F-Score 值分别为 0.66、0.63、0.6、0.55; 在 R3 网络中 $O_m=2$ 、3、4、5、6 时, CDHNE 算法的 F-Score 值分别为 0.58、0.57、0.52、0.51、0.5; 在 R4 网络中 $O_m=2$ 、3、4、5、6 时, CDHNE 算法的 F-Score 值分别为 0.3、0.33、0.28、0.26、0.23。可以看出, 仅在 R1 网络中 $\mu=0.3$ 和 R4 网络中 $O_m=2$ 时, CDHNE 算法的 F-Score 值略低于 CoEuS 算法, 在其他情况下 CDHNE 算法的 F-Score 值均优于其他对比算法。其原因是 CoEuS、MS 与 CFINDER 算法更关注网络内部结构以及如何产生社区, 而忽略了识别社区边界的分析, 因此, 算法所识别出的重叠节点精度不高; 虽然 OLPC 算法在社区识别过程中更注重对每个节点的分析, 也相应地加入了控制条件, 但由于节点过于灵活, 在识别过程中较难控制, 从而出现了识别重叠节点精度不够高的问题。CDHNE 算法利用边社区识别较为稳定的特点形成边社区结构, 避免了节点在社区形成的过程中过于灵活而难以控制的弱点。但在边社区识别中通常也存在识别重叠节点过多的现象, 因此, 在完成边社区识别后, 利用节点较为灵活的优势, 对边社区中的重叠节点进一步进行精确分析, 从实验结果来看, 起到了准确识别重叠节点的效果。从各算法的运行结果来看, 相较于其他传统算法, CDHNE 算法在 R1~R4 网络中 F-Score 的整体平均值至少提高 15%。

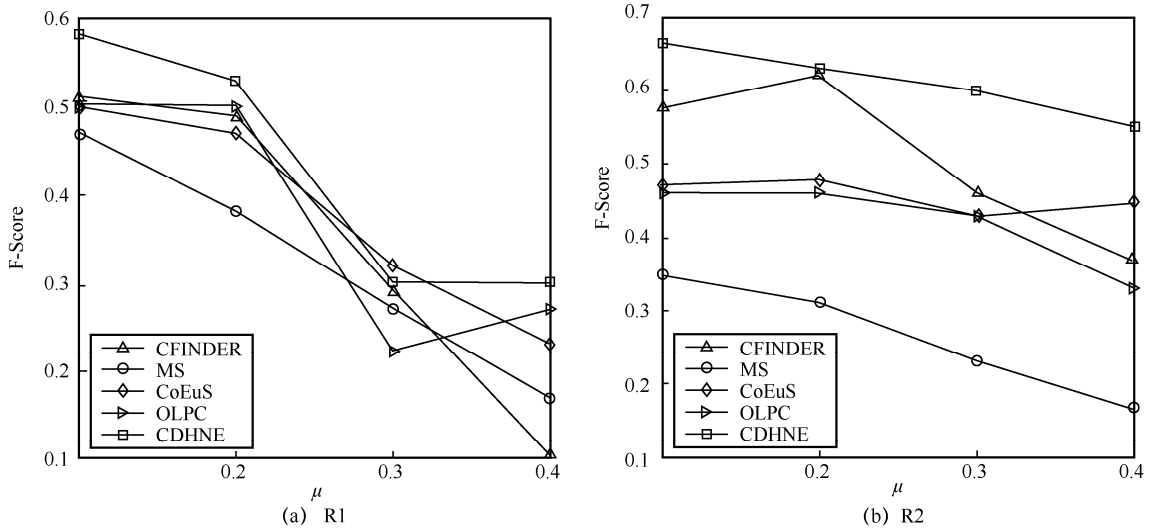


图6 各算法在 R1 与 R2 网络中社区识别的 F-Score 值

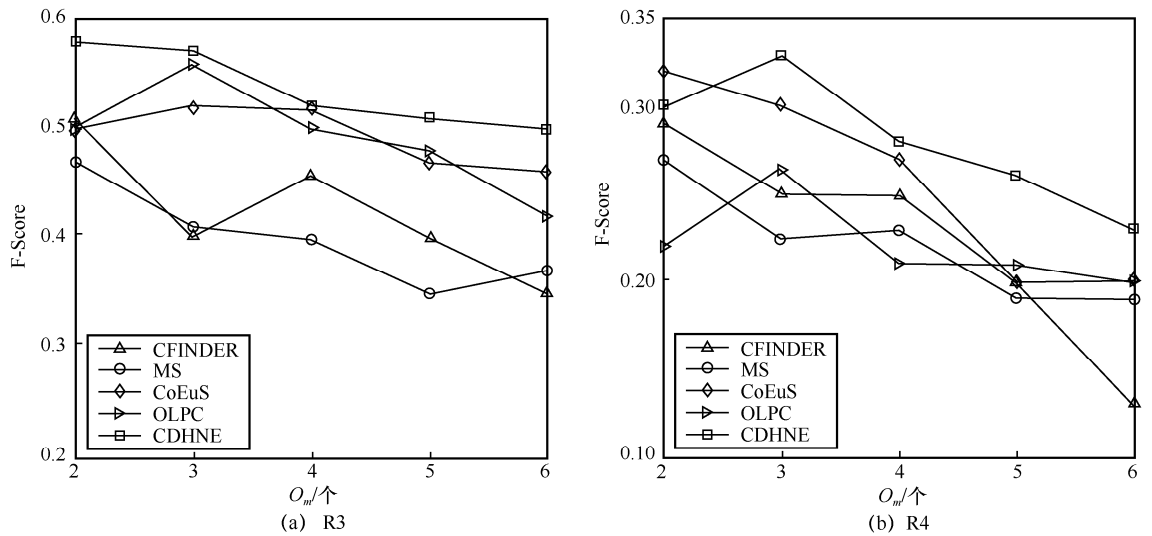


图7 各算法在 R3 与 R4 网络中社区识别的 F-Score 值

3.3 真实网络数据集

由于基准网络数据集与真实网络之间依然有所区别,因此,本文选用从几十个节点到十几万个节点的 7 个真实网络数据集对各算法进行进一步测试,真实网络数据集信息见表 4。

CDHNE 算法与其他对比算法在真实网络中的 EQ 对比见表 5。从表 5 能够看出,除了在 PGP 网络中 CDHNE 算法的 EQ 值低于 OLPC 算法,在其他真实网络中,CDHNE 算法的 EQ 值均优于其他对比算法。在 PGP 网络中,由于社区结构边界较为清晰,社区规模较大,OLPC

算法在社区识别过程中,可以将节点识别到一个相对较大的社区中,未对 OLPC 算法造成更大的影响。而 CDHNE 算法在 PGP 网络中所识别到的社区规模小于 OLPC 算法,损失了社区间的连接紧密性,因此在该网络中 OLPC 算法的 EQ 值高于 CDHNE 算法。但从算法的普适性方面来看,CDHNE 算法要优于其他对比算法,特别是在具有十几万个节点的 Facebook 网络中,仅有 CDHNE 算法能够顺利完成社区识别任务,说明 CDHNE 算法在大规模网络中依然有效。

表4 真实网络数据集信息

网络名称	节点/边/平均度
Lesmis (《悲惨世界》中人物的共现网络) ^[22]	77/254/6.6
Email (电子邮件交往网络) ^[22]	1 133/5 451/9.62
Netscience (科研作者合作网络) ^[22]	1 588/2 742/3.45
Word (语汇语义网络) ^[23]	7 207/31 784/8.82
PGP (信任网络) ^[22]	10 680/24 316/4.55
Internet (互联网快照网络) ^[22]	22 963/48 436/4.22
Facebook (脸书交互网络) ^[24]	134 833/1 380 293/10.24

表5 CDHNE 算法与其他对比算法在真实网络中的 EQ 对比

网络	CDHNE	CFINDER	MS	CoEus	OLPC
Lesmis	0.532 9	0.395 3 ($k=6$)	0.386 6	0.309 8	0.421 5
Email	0.358 2	0.264 1 ($k=4$)	0.151 3	0.183 5	0.305 5
Netscience	0.890 1	0.761 1 ($k=3$)		0.150 6	0.738 1
Word	0.162 3	0.164 9 ($k=4$)		0.012 8	
PGP	0.493 7			0.012 8	0.695 9
Internet	0.367 6			0.257 9	0.274 3
Facebook	0.412 1				

4 结束语

传统的社区识别方法通常仅从节点或边的角度研究社区识别问题,然而单纯从节点或边的角度分析社区结构均存在其自身的弱点,具体来说,基于节点的社区识别方法往往存在节点较为灵活的特点,在社区识别初期,社区结构较难快速形成;而基于边的社区识别方法往往存在“过度重叠”现象,使得社区识别的质量不高。基于这一考虑,本文整合了边与节点在社区识别过程中的优势,在社区识别的不同阶段从不同角度分析社区识别问题,提出了融合节点分析与边分析的复杂网络社区识别算法。通过实验分析可知,在人工网络(LFR benchmark)中CDHNE算法无论在社区识别精度(NMI)分析方面,还是在重叠节点精度(F-Score)分析方面均取得了较好的结果;此外,在真实网络中,CDHNE算法仅在PGP网络中的EQ值低于OLPC算法,而在其他真实网络中,其所取得的EQ值均优于其他对比算法,特别是在具有十几万个节点的Facebook网络,

CDHNE算法依然能够出色地完成社区识别任务,验证了算法在大规模网络中的识别能力,同时CDHNE算法无须设置任何参数,提高了算法的普适性。虽然CDHNE算法表现出了良好的性能,但是随着网络技术的不断发展,复杂网络也呈现出动态性特征,若要掌握动态网络的结构特征,就需要在动态网络中完成社区识别任务,然而动态网络中的节点与边会随着时间的变化而增加或删除,使得融合节点分析与边分析的复杂网络社区识别这一类算法无法针对动态网络完成社区识别任务。因此,在后续工作中,将针对动态网络开展边点融合分析的社区识别方法研究。

参考文献:

- [1] NEWMAN M E, GIRVAN M. Finding and evaluating community structure in networks[J]. Physical Review E, Statistical, Nonlinear, and Soft Matter Physics, 2004, 69(2): 026113.
- [2] FORTUNATO S. Community detection in graphs[J]. Physics Reports, 2010, 486(3): 75-174.
- [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] SATHIYAKUMARI K, VIJAYA M. Hybrid approach for com-



- community detection using k-core and clique percolation[J]. Zeitschen Journal, 2021,7(2): 187-201.
- [5] SATHYAKALA M, SANGEETHA M. Retraction note to: a weak clique based multi objective genetic algorithm for overlapping community detection in complex networks[J]. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(1): 259.
- [6] SUN L P, YE T, SUN J, et al. Density-peak-based overlapping community detection algorithm[J]. IEEE Transactions on Computational Social Systems, 2022, 9(4): 1211-1223.
- [7] LANCICHINETTI A, FORTUNATO S, KERTÉSZ J. Detecting the overlapping and hierarchical community structure in complex networks[J]. New Journal of Physics, 2009, 11(3): 033015.
- [8] ASMI K, LOTFI D, ABARDA A. The greedy coupled-seeds expansion method for the overlapping community detection in social networks[J]. Computing, 2022, 104(2): 295-313.
- [9] DILMAGHANI S, BRUST M R, RIBEIRO C H C, et al. From communities to protein complexes: a local community detection algorithm on PPI networks[J]. PLoS One, 2022, 17(1): e0260484.
- [10] LIAKOS P, NTOULAS A, DELIS A. COEUS: Community detection via seed-set expansion on graph streams[C]//Proceedings of 2017 IEEE International Conference on Big Data. Piscataway: IEEE Press, 2018: 676-685.
- [11] GREGORY S. Finding overlapping communities in networks by label propagation[J]. New Journal of Physics, 2010, 12(10): 103018.
- [12] ATTAL J P, MALEK M, ZOLGHADRI M. Overlapping community detection using core label propagation algorithm and belonging functions[J]. Applied Intelligence, 2021, 51(11): 8067-8087.
- [13] 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.
- DENG K, LI W P, CHEN L, et al. A novel algorithm for overlapping community detection based on label propagation in complex networks[J]. Control and Decision, 2020, 35(11): 2733-2742.
- [14] AHN Y Y, BAGROW J P, LEHMANN S. Link communities reveal multiscale complexity in networks[J]. Nature, 2010, 466(7307): 761-764.
- [15] 邓亚景, 张海, 郭晓, 等. 基于节点度和社区信息的复杂网络链接预测[J]. 工程数学学报, 2020, 37(2): 245-259.
- DENG Y J, ZHANG H, GUO X, et al. Link prediction in complex networks incorporating the degree and community information[J]. Chinese Journal of Engineering Mathematics, 2020, 37(2): 245-259.
- [16] LIU Z H, WANG H M, WANG G S, et al. Link community detection combined with network pruning and local community expansion[J]. Modern Physics Letters B, 2021, 35(5): 2150098.
- [17] LANCICHINETTI A, FORTUNATO S, RADICCHI F. Benchmark graphs for testing community detection algorithms[J]. Physical Review E, 2008, 78(4): 046110.
- [18] BRUTZ M, MEYER F G. A flexible multiscale approach to overlapping community detection[J]. Social Network Analysis and Mining, 2015, 5(1): 1-17.
- [19] DENG K, CHEN L, LI W P. An overlapping community detection algorithm with label propagation control for complex networks[J]. Instrumentation Measure Métrologie, 2019, 18(2): 105-109.
- [20] 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, 2013, 45(4): 1-35.
- [21] 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, 2009(7): P07042.
- [22] NEWMAN M. Network data from mark newman's home page[EB]. 2022.
- [23] GREGORY S. Network research[EB]. 2022.
- [24] LESKOVEC J. Stanford network analysis project[EB]. 2022.

[作者简介]



邓琨 (1980—), 男, 博士, 嘉兴学院信息科学与工程学院、嘉兴学院浙江省医学电子与数字健康重点实验室副教授、硕士生导师, 主要研究方向为网络结构分析、数据挖掘、异构网络分析等。



蒋庆丰 (1983—), 男, 博士, 常熟理工学院计算机科学与工程学院讲师, 主要研究方向为计算机网络及信息安全等。



刘星妍 (1980—), 女, 嘉兴学院信息科学与工程学院高级工程师, 主要研究方向为数据挖掘、网络结构分析等。