计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200300034

基于中心团的重叠社区检测算法

薛 磊 唐旭清
江南大学理学院 江苏 无锡 214122
(1597013440@qq.com)

摘 要 社区检测已经成为了了解复杂网络结构和网络动态的一个重要途径。针对传统的节点聚类和链接聚类在发现重叠社区方面存在的两种固有缺陷,即参数依赖和结果不稳定,文中提出了一种基于中心团的局部扩展改进算法CLEM,用于检测重叠社区。该算法通过选取中心团为核心种子,并在种子扩展过程中惩罚被多次删除的节点,改善所得结果的稳定性;通过选取不依赖参数的适应度函数,改进其迭代计算过程,避免了适应度函数的参数限制,并降低了计算复杂度。在合成网络和现实网络上测试的结果表明,与已有算法相比,所提算法在计算时间和准确度上均有很好的表现。

关键词:中心团;局部扩展;重叠社区检测;种子扩展;社区优化
中图法分类号 TP399

Algorithm for Detecting Overlapping Communities Based on Centered Cliques

XUE Lei and TANG Xu-qing
School of Science,Jiangnan University,Wuxi,Jiangsu 214122,China

Abstract Community detection in complex network has become a vital way to understand its structure and dynamic characteristics. However,there are two inherent shortcomings that the parameter dependency and instability of using the traditional node clustering and link clustering to detect overlapping communities. This paper proposes an improving algorithm,that is,the local expansion method based on the centered clique(CLEM),for detecting overlapping communities. Firstly,in CLEM algorithm,the centered cliques is selected as the core seed and the nodes deleted by multiple times in the process of seed expansion are punished,so its stability of results is improved. Then,by selecting the fitness function with parameter-independent and improving its iterative calculation process,the parameter limitation of the fitness function is avoided and the computational complexity is quickly reduced. Finally,the test results on synthetic networks and real-world networks show that CLEM is good both in computing time and accuracy compared with some existing algorithms.

Keywords Centered clique,Local expansion,Overlapping community detection,Seed expansion,Community optimization

1 引言

随着复杂网络在生物组学、人类社交和交通运输等各个领域[1-3]的发展,社区检测已经成为了解网络结构和网络动态变化[4-6]的一种重要途径。在过去几年中,研究者们已经提出了许多不同的算法用于揭示网络中的社区结构,如层次聚类[7-9]、谱平分法[10]和基于优化的方法[11-12]等。这些方法限制节点只能属于一个社区,而许多真实网络中的社区往往带有重叠的节点[13-15]。例如在生物酵母复合物数据集CYC2008[16]中,1 628种蛋白质中的207种蛋白就参与了不止一种复合物的组成。重叠社区检测的目的就是发现这些重叠节点并将其合理归类到相应社区。关于重叠结构[17]的检测和分析已经成为复杂网络的热点问题。

现有的重叠社区检测算法大致可以分为两大类:基于节点的算法和基于链接的算法。基于节点的重叠社区检测算法通过节点的网络信息,直接将节点划分给相应的社区。目前已有许多成熟的算法被提出,一类是基于团渗透的方法,如2005年的团渗透算法(Cluster Percolation Method,CPM)[18],另一类是基于局部扩张的方法,如基于局部适应度的算法(Local Fitness Method,LFM)[19]。Lancichinetti等于2009年首次提出基于局部适应度的重叠社区检测算法(LFM),并给出了局部扩张的框架结构。LFM算法的特点是利用贪婪的局部函数来扩张网络中的随机节点以形成网络社区。但在实际应用中,LFM算法有结果不稳定和易死循环两个缺陷。随后,许多沿用局部扩张框架结构的改进算法被相继提出。BECKER等于2012年提出了一种用于蛋白质互作网络的重叠社区检测算法(Overlapping Cluster Generator,OCG)[20]。Nepusz等于2014年提出了一种新的用于解决加

到稿日期:2020-03-05 返修日期:2020-06-10 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金项目(11371174)
This work was supported by the National Natural Science Foundation of China(11371174).
通信作者:唐旭清(txq5139@jiangnan.edu.cn)

权网络的重叠检测算法(Clustering with Overlapping Neighborhood Expansion, ClusterONE)[21]。Ding 等于 2016 年提出了一种基于网络分解的重叠社区检测算法(Overlapping Community Detection Algorithm based on Network Decomposition, NDOCD)[22],该算法比 LFM 算法拥有更快的运行速度。这些改进算法不断拓宽 LFM 的应用领域和网络。此外,还有一些其他节点方法,如随机块[23]、多标签传播[24]和基于相似度指标的聚类方法[25]等。这些方法在不同的领域和网络中显示了它们的性能优势。

另一方面,不同于上述基于节点的检测,基于链接的重叠社区检测认为链接社区比节点社区更加直观,选择对链接进行相应的划分[26]。Ahn 等最初在 2010 年提出链路聚类(Link Clustering, LC)算法[27],并将其应用于大型网络,LC 算法测量相邻链接之间的相似度,并分层地对链接进行分类。还有一些基于 LC 的改进算法,例如扩展链路聚类(Expanded Link Clustering, ELC)算法[28]和基于遗传算法的网络检测算法(A new algorithm to discover overlapped communities in networks by employing genetic algorithms, GANET)[29]。链接聚类的缺陷是计算时间较长,且不能保证其比节点聚类拥有更好的检测效果[30]。此外,不管是节点聚类还是链接聚类,大多数算法都需要通过先验信息的参数来检测重叠社区。如 LFM 需要适当的参数 α 来控制社区的大小且 CPM 社区选取对参数 k 敏感,NDOCD 也依赖参数 $JS_{\max}$ 来扩展种子,链接聚类 LC 也需要一定的切割阈值才能获得较好的社区。

本文提出了一种基于中心团的局部扩展改进算法 CLEM (Local Expansion Method based on Centered clique)。本文的主要改进工作有:1)针对 LFM 算法的参数依赖问题,选取不依赖参数的局部模块度密度[31]作为适应度函数,并对其进行改进以缩短运算时间;2)针对 LFM 算法结果不稳定的缺陷,选取中心团[20]为核心种子,并引入惩罚指标改善算法的稳定性。本文第 2 节引入相关的基础概念;第 3 节给出 CLEM 算法的概述;第 4 节对算法的结果进行讨论;最后总结全文。

2 相关概念

2.1 网络社区

一个网络可以被建模为一个图 $G=(V,E)$,其中 V 是所有顶点的集合,E 是所有链接的集合,并且 $n=|V|$, $m=|E|$ 分别表示顶点和链接的总数。网络中的一个社区可以表示为一组顶点集,这些顶点之间有更稠密的链接。一个更加清晰且正式的定义由 Radicchi 等[32]借助顶点的度提出。定义节点 i 的度 $k_i=\sum_j \boldsymbol{A}_{ij}$,其中 $\boldsymbol{A}$ 是图 G 的邻接矩阵。对于一个无向无权图,如果 $(v_i,v_j)\in E$,那么 $\boldsymbol{A}_{ij}=1$,否则 $\boldsymbol{A}_{ij}=0$。给定一个子图 $S\subseteq G$,对于 $\forall i\in S$,节点 i 相对于 S 的度 k_i 可以被改写为:

$$k_i=k_i^{\text{in}}+k_i^{\text{out}} \tag{1}$$

其中,$k_i^{\text{in}}=\sum_{j\in S}\boldsymbol{A}_{ij}$, $k_i^{\text{out}}=\sum_{j\notin S}\boldsymbol{A}_{ij}$。那么,$S$ 被称为一个强意义上的社区,如果满足:

$$\forall i\in S, k_i^{\text{in}}>k_i^{\text{out}} \tag{2}$$

相对地,S 被认为是一个弱意义上的社区,如果满足:

$$k_S^{\text{in}}>k_S^{\text{out}} \tag{3}$$

其中,$k_S^{\text{in}}=\frac{1}{2}\sum_{i\in S,j\in S}\boldsymbol{A}_{ij}$, $k_S^{\text{out}}=\sum_{i\in S,j\notin S}\boldsymbol{A}_{ij}$。由式(3)可看出弱意义社区的内部链接大于外部链接,本文选择的是弱意义上的社区,从适应度函数上可体现(见式(4))。

2.2 适应度函数

适应度函数在局部扩展中十分重要,一方面需要通过适应度最大化来扩张,另一方面需要通过删除负适应度的节点来优化。本文选取的适应度函数为局部的模块化密度函数[31],这是由于该函数既没有参数的限制,也没有分辨率的限制[33]。给定一个社区 $S\subseteq G$,则社区 S 的局部模块化密度 D_S 定义为:

$$D_S=\frac{k_S^{\text{in}}-k_S^{\text{out}}}{|S|} \tag{4}$$

其中,$|S|$ 表示社区 S 中节点的总数;D_S 值越大,社区效果就越好。

而节点 i 对于社区 S 的局部模块化密度 D_S^i 定义为:

$$D_S^i=D_{S+i}-D_{S-i} \tag{5}$$

其中,$S+i$ 表示将节点 i 加入社区 S, $S-i$ 表示将节点 i 剔出社区 S。D_S^i 值越大,节点 i 对 S 就越重要。

3 CLEM 算法

3.1 CLEM 算法概述

CLEM 算法主要有 3 个步骤:种子选取、种子扩展和社区优化,具体如下。

Step1 种子选取。将节点按度降序排列,依次选取未在社区中的单个节点,计算以此节点为基础的中心团并将其选为核心种子。

Step2 种子扩展。通过局部最优密度函数对核心种子进行扩展。

Step3 返回 Step1,直到没有新节点可以选取。

Step4 社区优化。消除社区中负扩展模块度(见式(7))的节点和少于 3 个节点的社区。

3.1.1 种子选取

本文将中心团作为核心种子。核心种子的具体寻找步骤如下:

Step 1 将节点以度降序排列,并依次选取节点 x;

Step 2 以节点 x 为基础建立中心团,并选择其为核心种子。

在 Step1 中,以节点度降序为更新序列,由于节点的度越大,越有可能是网络的中心节点。文献[20]提供了计算所有中心团的算法,但由于本文只需计算局部的中心团,因此选择将原算法修改成对应 Step2 的算法 1。

算法 1 计算节点 x 的中心团

输入:G=(V,E), x

输出:节点 x 的中心团 C_x

1. 初始化 $C_x\leftarrow\{\ \}$
2. $L_x\leftarrow$节点 x 的所有邻居
3. $C_x\leftarrow\{x\}$
4. for all $y\in L_x$ do
5. 计算节点 y 限制于 L_x 的节点度

6. end for
7. 将 L_x 以节点度降序排序
8. for all y∈L_x do
9. if ∀z∈C_x，(y，z)∈E then
10. C_x←C_x∪{y}
11. end if
12. end for

选择图中所有的完全子图(团)作为核心种子是十分耗时的，因此本文仅选择寻找某一节点所在的局部完全子图(中心团)作为核心种子，其计算复杂度低，适合大规模网络，并且结构稳定，更利于社区的构建。

3.1.2　种子扩展

通过种子选取获得的核心种子需要通过加点和删点两个操作来扩展。加点操作需要遍历核心种子的所有邻居节点，计算它们的局部模块化密度并加入最大正值所对应的邻居节点。删点操作需要计算核心种子内每个节点的局部模块化密度并删除所有负值对应的节点。值得注意的是，每当加入一个节点，就需要进行删点操作，直到没有节点可以被加入到核心种子内。核心种子扩展的详细过程如算法 2 所示。

算法 2　核心种子扩展

输入：中心团 C_x
输出：局部社区 S_x
1. S_x←C_x
2. L_{S_x}←社区 S_x 的所有邻居
3. for all y∈L_{S_x} do
4. 计算节点 y 关于社区 S_x 的局部模块化密度 $D^y_{S_x}$
5. end for
6. 在所有满足 P_y<c 的 y 中找出最大局部模块化密度的节点 y′
7. if $D^{y'}_{S_x}$<0 then
8. break
9. else
10. S_x=S_x∪{y′}
11. for all y∈S_x do
12. 计算节点 y 关于社区 S_x 的局部模块化密度 $D^y_{S_x}$
13. if $D^y_{S_x}$<0 then
14. S_x←S_x\{y}，P_y=P_y+1，break
15. end if
16. end for
17. end if
18. 回到第 2 行

在算法 2 中，第 4 行和第 12 行均可通过存储计算局部模块化密度所需的 k_S^{in} 和 k_S^{out} 来简化适应度的计算。因在第 4 行中的 $D^y_{S_x}$ 考虑的是加入节点，故有：

$$D^y_{S_x}=D_{S_x+y}-D_{S_x}=\frac{k^{in}_{S_x+y}-k^{out}_{S_x+y}}{|S_x|+1}-\frac{k^{in}_{S_x}-k^{out}_{S_x}}{|S_x|} \tag{6}$$

将节点 y 关于社区 S 的链接和记为 M_{yS_x}，$M_{yS_x}=\sum_{i\in S_x}A_{yi}$，即有：

$k^{in}_{S_x+y}=k^{in}_{S_x}+2M_{yS_x}$，$k^{out}_{S_x+y}=k^{out}_{S_x}+k_y-2M_{yS_x}$

由于算法 2 中的第 12 行考虑的是删除节点，因此只需注意删除节点与加入节点在 $D^y_{S_x}$ 上的差异，可类似于第 4 行得到相似结论，此外略。

同时，算法 2 中的第 14 行中被剔除的点仍有可能在第 10 行被再次加入社区，从而导致死循环。本文引入惩罚指标 P 和参数 c 来解决易死循环的问题。当节点 y 在同一社区被剔除一次，则将该点的惩罚指标 P_y 值增 1，而为了保证正常剔除优化不受影响，须设定一个参数 c 作为惩罚指标的上限(见算法 2 第 6 行)。当 P_y>c 时，便认为节点 y 是异常节点，将导致无限循环，不再考虑节点 y 的加入。

3.1.3　社区优化

CLEM 中的社区优化主要包含两个优化策略：1)删除社区中所有扩展模块度 EQ[34] 为负的节点；2)删除所有小于 3 个节点的社区。与节点的局部模块化密度相同，将节点 i 关于划分 S 的扩展模块度记为 EQ^i_S，有 $EQ^i_S=EQ_S-EQ_{S-i}$。详细过程如算法 3 所示。

算法 3　社区划分优化

输入：初始的社区划分 S
输出：最终的社区划分 S
1. for all S_x∈S do
2. for all y∈S_x do
3. 计算节点 y 关于社区 S_x 的扩展模块度 $EQ^y_{S_x}$
4. if $EQ^y_{S_x}$<0 then
5. S_x←S_x\{y}，break
6. end if
7. end for
8. end for
9. for all S_x∈S do
10. 删除所有满足|S_x|≤2 的社区 S_x
11. end for

CLEM 算法由算法 1－算法 3 组成，并实现其功能。通过算法 1 可获得一个核心种子，通过算法 2 可获得一个初始社区；通过算法 1 和算法 2 循环操作可获得全部初始社区；通过算法 3 将全部初始社区优化后便可获取最终的社区。

3.2　算法复杂度

CLEM 算法复杂度计算如下：因为存在一个回溯的过程，建立一个具有 s 个节点的社区需要的时间复杂度约为 $O(s^2)$；检测所有社区的时间复杂度为 $O(s_1^2+s_2^2+\cdots+s_c^2)$，其中 c 表示社区的总数量。因 $s\leqslant n$ 且 $c\leqslant n$，故时间复杂度上界为 $O(n^3)$。CLEM 算法的内存消耗为 $O(4n)$，其中每 n 个空间分别用于存储更新序列以及每个节点的 k_S^{in}，k_S^{out} 和惩罚指标 P。

4　实验与讨论

本节通过人工合成网络和真实网络分别验证算法所需的运行时间和社区检测的质量。人工合成网络可通过模拟各种可控条件来观测算法的运行时间和社区还原程度，真实网络可检测算法的实际运行效果。为了评测算法在人工合成网络和真实网络上的性能，实验采用扩展模块度 EQ[34] 和扩展的归一化互信息 ENMI[35] 两个指标。同时，将 CLEM 算法与其他 5 种已有的算法进行比较：CPM，LFM，LC，NDOCD 和 ELSC。对于每种算法，在原作者的推荐参数范围内运行 10 次，将相应评价指标的平均值和最大值作为结果。CPM 的参数 k 的推荐范围为 3～6，步长为 1。LFM 的参数 α 推荐选取

0.8～1.6，步长为 0.1。LC 以链接相似度 S 为阈值进行切割。NDOCD 算法的参数 $JS_{\max}$ 推荐选取 0.3～0.4，步长为 0.1。ELSC 需要合并小于等于参数 λ 的小社区。CLEM 算法的适应度函数没有参数限制，但需要参数 c 作为惩罚指标的上限。所有算法都在同一台机器上运行，运行参数为 Intel(R)Core(TM)i5-7300HQ，2.5GHz，8GBRAM，操作系统为 Win10，运行环境为 MATLABR2014a。

4.1 评价指标

在实验中，用于评测重叠社区准确度和还原度的两个重要指标分别是扩展模块度 EQ 和扩展的归一化互信息 ENMI。

4.1.1 扩展模块度

扩展模块度最初由 Shen 等[34]提出。由于大多数真实网络的真实划分未知，因此本文选择扩展模块度 EQ 来评价算法在真实网络中的社区划分准确度。扩展模块度的定义如下：

$$EQ=\frac{1}{2m}\sum_{i}\sum_{v,w\in C_i}\frac{1}{O_vO_w}\left[A_{vw}-\frac{k_vk_w}{2m}\right] \tag{7}$$

其中，O_v 指节点 v 所在社区的个数；EQ 值域为[0,1]，且 EQ 值越大，社区划分的准确度就越高。

4.1.2 扩展归一化互信息

归一化互信息 NMI 是来自信息理论的相似性度量，由 Danon 等[36]引入，用于度量两个社区间的相似性。最初的表述没有考虑重叠节点，Lancichinetti 等[35]加入重叠模块的扩展，并提出了扩展归一化互信息 ENMI。因为人工合成网络拥有对应的真实划分，所以本文选择 ENMI 来评价社区的还原程度。

给出网络中的两个划分 X 和 Y，其中 Y 已知，扩展归一化互信息 $ENMI(X|Y)$ 定义为：

$$ENMI(X|Y)=1-\frac{1}{2}[H(X|Y)_{\text{norm}}+H(Y|X)_{\text{norm}}] \tag{8}$$

其中，

$$H(X|Y)_{\text{norm}}=\frac{1}{|X|}\sum_{i}\frac{\min_{j}H(X_i|Y_j)}{H(X_i)} \tag{9}$$

其中，X_i 为划分 X 的第 i 个社区，函数 $h(x)=-x\log x$，$H(X_i)$ 为社区 X_i 的熵，$H(X_i|Y_j|)$ 为社区 X_i 相对于社区 Y_j 的条件熵，$H(X_i,Y_j)$ 为社区 X_i 与 Y_j 的联合熵，即有：

$$H(X_i)=h[P(X_i=1)]+h[P(X_i=0)] \tag{10}$$

$$H(X_i|Y_j)=\begin{cases}H(X_i,Y_j)-H(Y_j),\\ \text{if } h(P_1)+h(P_2)\geqslant h(P_3)+h(P_4)\\ H(X_i),\quad \text{Otherwise}\end{cases} \tag{11}$$

$$H(X_i,Y_j)=h(P_1)+h(P_2)+h(P_3)+h(P_4) \tag{12}$$

其中，P_1,P_2,P_3,P_4 分别表示概率 $P(X_i=0,Y_j=0)$，$P(X_i=0,Y_j=1)$，$P(X_i=1,Y_j=0)$，$P(X_i=1,Y_j=1)$。$ENMI$ 值域为[0,1]，且取值越大，社区还原程度就越高。

4.2 人工合成网络的验证

本节基于 LFR 基准网络[37]构建人工合成网络，并将所提算法与 CPM，LFM，LC，NDOCD 和 ELSC 5 种相关算法进行比较研究。

实验中，生成 10 组不同规模的网络用于计算算法的时间效率。10 组实验网络的节点数分别为 1000～10000，步长为 1000，其他的网络参数一致，具体为：$k=10$，$maxk=50$，$minc=10$，$maxc=50$，$mu=0.1$，$on=0$，$om=0$。同时，在 8 大组规模 N 为 1000 的网络上计算扩展归一化互信息来比较社区的还原度，每一大组包含 7 个网络，分别对应不同的参数 om(2～8)。详细的参数设置如表 1 所列。表 1 中，平均度和最大度分别为 k 和 $maxk$；社区大小通过 minc 与 maxc 来模拟；网络混合系数为 mu，表示社区混合的程度；重叠节点个数为 on，单个节点的最多重叠社区个数为 om。

表 1 LFR 基准网络信息

Table 1 Information of LFR benchmarks

ID	k	$maxk$	$minc$	$maxc$	mu	on	om
S_1	10	50	10	50	0.1	100	2～8
S_2	10	50	10	50	0.1	500	2～8
S_3	10	50	10	50	0.3	100	2～8
S_4	10	50	10	50	0.3	500	2～8
S_5	10	50	20	100	0.1	100	2～8
S_6	10	50	20	100	0.1	500	2～8
S_7	10	50	20	100	0.3	100	2～8
S_8	10	50	20	100	0.3	500	2～8

不同规模合成网络下算法的计算时间如图 1 所示。通过对图 1 中各算法的实验数据进行分析发现，NDOCD 在运算时间上是最快的，但 NDOCD 在合成网络和真实网络的社区准确度较差(见图 2 与表 3)。CLEM 与 LFM 两种算法的运行时间中等，其中 CLEM 算法比 LFM 节省了不同程度的计算时间，且随着网络规模的增大，节省的计算时间也越多。这是由于 CLEM 算法比 LFM 算法额外存储了计算局部模块化密度所需的 k_S^{in} 和 k_S^{out}，从而简化了适应度的计算。LC，CPM 和 ELSC 算法耗费最多的计算时间，且 LC 和 ELSC 分别在 N 大于 5000 和 7000 后的网络中失效。实验数据表明，LC 和 ELSC 算法失效的主要原因是内存不足，因为 LC 和 ELSC 算法均需要大量的内存记录相似度指标。例如，LC 算法最少需开辟 $[m(m-1)/2]*8$ 字节的内存，而所用 7000 规模网络的网络链接 m 为 34296，因此仅存储相似度指标便需约 4.3816GB 的内存，而所用机器实际内存不足 8GB，从而导致失效。这表明 LC 和 ELSC 并不适合检测链接复杂的网络。CPM 算法需计算网络中全部的最大团，往往十分耗时，且随着网络规模的增大，会出现算法无法终止的情况。

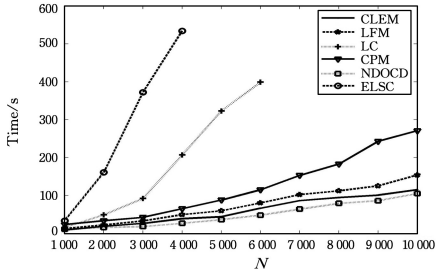


图 1 不同规模合成网络中不同算法的计算时间比较

Fig. 1 Comparison of computation time of different algorithms on synthetic networks with different sizes

图 2 为 6 种算法分别在 8 组 LFR 基准网络(S1-S8)上的社区发现结果，横坐标表示 om，纵坐标表示 $ENMI$。我们分别将 CLEM 与其他算法进行比较。

1)与 LFM 算法进行比较。在图 2 中所有 $on=100$ 的低

重叠网络（S1，S3，S5 和 S8）中，LFM 和 CLEM 的差距最小，但随着混合系数 *mu* 的增大，CLEM 算法的表现越来越好。在 *on*＝500 的高重叠网络（S2，S4，S6 和 S8）中，CLEM 在大多数情况下仍优于 LFM。这也证明了前文的分析，即 CLEM 的中心团种子比 LFM 的随机节点种子的结构更稳定，更适合在复杂网络中发现重叠社区。

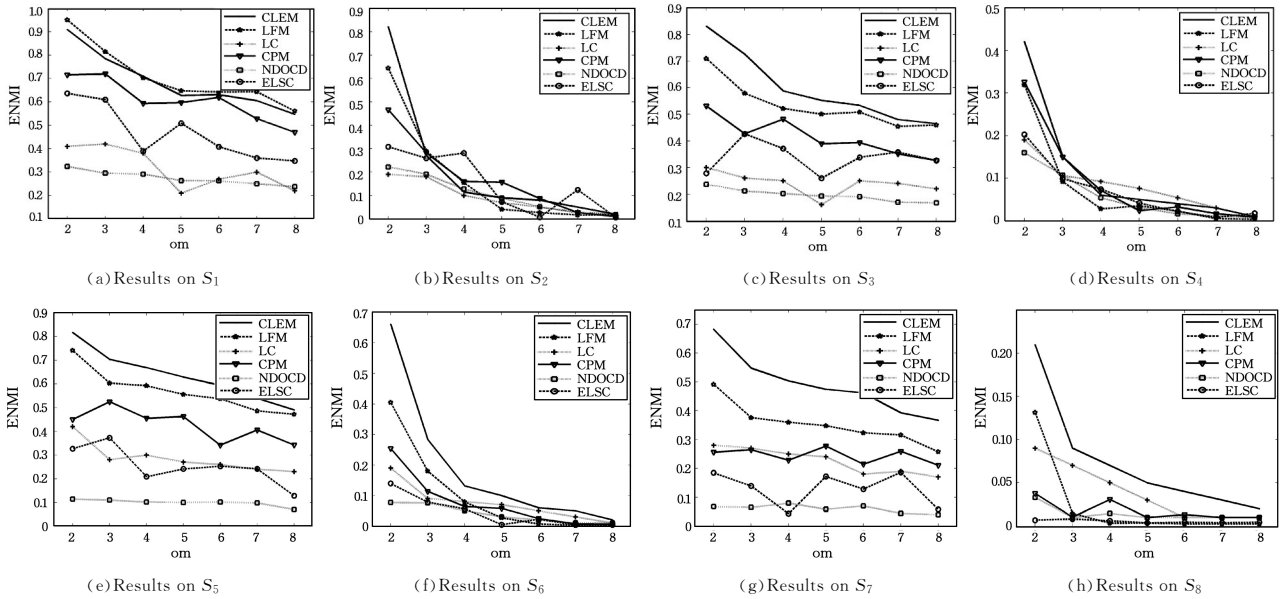
2）与 CPM 算法进行比较。在低重叠网络（S1，S3，S5 和 S8）中，CLEM 的结果全都优于 CPM。在绝大多数合成网络上 CLEM 的结果都优于 CPM 算法。在高重叠网络（S2，S4，S6 和 S8）中，除了少数情况（如 *S*2 的 *om*＝5，S4 的 *om*＝4）外，CLEM 仍有更好的结果。

3）与 LC 算法进行比较。LC 算法除了在高重叠网络 *S*4 的 *om* 为 4～6 时结果最好，其他情况都劣于 CLEM，且 LC 算法在低重叠网络的结果起伏较大（如 *S*1 的 *om* 为 4～7 和 *S*3 的 *om* 为 4～6），这也说明了 LC 算法的不稳定性。

4）与 NDOCD 算法进行比较。CLEM 算法在各个网络上的结果都好于 NDOCD。LFM，LC 和 CPM 也在大多数情况下优于 NDOCD，但也有一些劣于 NDOCD 的情况，如 LC 在 *S*1 和 *S*3 的 *om* 为 5，LFM 在 *S*2 的 *om* 为 5～6 时。

5）与 ELSC 算法进行比较。ELSC 算法除了在高重叠网络 *S*2 的 *om*＝4 时结果最好，在其他情况下都劣于 CLEM，且 ELSC 算法与 LC 算法一样，在低重叠网络的结果起伏较大（如 *S*1 的 *om* 为 3～5、*S*3 的 *om* 为 2～5 和 S7 的 *om* 为 3～8），这说明了 ELSC 算法的结果不稳定。

总的来说，CLEM 算法在合成网络上拥有较短的计算时间和较优的社区检测效果。



(a) Results on S_1　(b) Results on S_2　(c) Results on S_3　(d) Results on S_4

(e) Results on S_5　(f) Results on S_6　(g) Results on S_7　(h) Results on S_8

图 2　人工合成网络下的社区划分结果

Fig. 2　Community detection results on artificial synthesis networks

4.3　真实网络验证

在现实网络中，实验计算各算法检测到的社区的扩展模块度来评判算法的表现。实验所用的 8 个真实网络的详细描述和下载来源如表 2 所列。为探究 CLEM 算法的最大值 *EQ*，将种子节点的更新规则从度降序变为随机序列更新，以此增加 CLEM 算法的随机性。6 种算法在 8 个真实网络上的实验结果如表 3 所列。表 3 中第 2 列－第 6 列的数据格式为 a/b，算法在原作者推荐参数内运行 10 次后取平均值，a 为其中最大的平均值，b 为所有结果中的最大值。由于 LC 和 ELSC 算法参数设置的特殊性，算法每次运行的结果相同。表 3 中第 7 列记录的是算法取最大值时的参数（k 是 CPM 的参数，α 是 LFM 的参数，S 是 LC 的参数，$JS_{\max}$ 是 NDOCD 的参数，λ 是 ELSC 的参数，c 是 CLEM 的参数），加粗字体表示各网络中最好的结果。

由表 3 可以得到如下结论：

1）在 Karate 上，ELSC 算法在平均值 *EQ* 上表现最好，CLEM 算法在最大值 *EQ* 上表现最好。实际上，Karate 网络拥有真实社区，其真实社区的 *EQ* 为 0.371。CLEM 算法不仅检测到其真实社区，还找到了社区交界处的重叠节点，因此 *EQ* 更高。CLEM 算法在同为小数据的 Dolphins 上的社区发现情况与 Karate 一致。

2）在 Pol. books 上，LC 算法在平均值 *EQ* 上表现最好，LFM 算法在最大值 *EQ* 上表现最好。LFM 算法的随机种子更利于获得优异的结果，如 LFM 也在 C. elegans 和 PGP 上获得最大值 *EQ*。但是，随机种子也会得到一些较差的结果，如 LFM 在 7 组网络中的平均值 *EQ* 均不高，这说明 LFM 算法的结果是不稳定的。

3）在 Football 和 Email 上，CLEM 算法在平均值 *EQ* 和最大值 *EQ* 上都表现最好。这表明了 CLEM 算法的中心团种子在社区构建中的合理性。

4）在 Power 上，LC 和 ELSC 算法在 *EQ* 上的表现远超其他算法。Power 数据集的链接十分稀疏，平均度仅为 2.669，这表明 ELSC 和 LC 算法适合检测链接稀少的网络，而一旦网络链接复杂，LC 和 ELSC 的结果就稍差一些，如在平均度最高的 Football 和 Email 网络上，LC 算法的结果在 6 种算法中最差；ELSC 算法在链接复杂的 PGP 网络上会出现算法无法终止的情况。

表 2 实验所用的真实世界网络

Table 2 Real-world networks used in experiments

Network	Nodes	Edges	Average degree	Description	Source
Karate	34	78	4.588	Zachary's karate club	文献[38]
Dolphins	62	159	5.129	Dolphins society network	文献[38]
Pol. books	105	441	8.4	Books about US politics	文献[38]
Football	115	613	10.66	American college football	文献[38]
C. elegans	453	2025	8.94	C. elegans metabolic	文献[39]
Email	1133	5451	9.622	Email network URV	文献[39]
Power	4941	6594	2.669	Power grid	文献[38]
PGP	10680	24316	4.554	PGP networks	文献[39]

表 3 8 个真实世界网络的实验结果

Table 3 Experimental results on eight real-world networks

Network	CPM	LFM	LC	NDOCD	ELSC	CLEM	$(k,\alpha,S,JS_{\max},\lambda,c)$
Karate	0.15/0.186	0.285/0.382	0.179/0.179	0.187/0.208	**0.371**/0.371	0.361/**0.401**	(3,1.2,0.3333,0.3,5,6)
Dolphins	0.325/0.361	0.338/0.447	0.284/0.284	0.27/0.305	**0.472**/0.472	0.3568/**0.498**	(3,1.1,0.2,0.3,6,6)
Pol. books	0.374/0.436	0.329/**0.481**	**0.403**/0.403	0.167/0.180	0.3737/0.373	0.338/0.431	(3,1,0.2,0.3,6,6)
Football	0.396/0.559	0.476/0.527	0.110/0.110	0.293/0.311	0.401/0.401	**0.574**/**0.598**	(4,1,0.2,0.3,4,6)
C. elegans	0.065/0.095	**0.154**/**0.201**	0.109/0.109	0.128/0.136	0.153/0.153	0.098/0.142	(4,1,0.2081,0.3,6,6)
Email	0.137/0.264	0.181/0.273	0.06/0.06	0.141/0.171	0.264/0.264	**0.283**/**0.355**	(4,0.9,0.1707,0.3,5,6)
Power	0.066/0.157	0.496/0.570	0.768/0.768	0.307/0.32	**0.819**/**0.819**	0.575/0.584	(3,0.8,0.1176,0.3,5,6)
PGP	0.311/0.363	0.483/**0.598**	0.458/0.458	0.309/0.319	—	**0.548**/0.567	(3,0.8,0.1408,0.3,—,6)

从总体上看,CLEM 算法在 8 组真实网络上检测重叠社区的质量优于其他算法。

4.4 参数分析

CLEM 算法引入惩罚指标 P 和参数 c 来解决原 LFM 算法易陷入无限循环的问题。在 LFM 算法的社区扩展中,一些节点会被重复地加入和删除,从而导致社区扩展失败。这些引发死循环的节点称为异常节点,其他节点称为正常节点。CLEM 算法在 8 组真实网络上的异常节点数量如表 4 所列,其中第 1 行记录了在社区中正常节点被删除的最大次数(Maximum Number of Deletions, MND),第 2 行记录了在社区中出现异常节点的总数(Total of Abnormal Nodes, TAN)。算法通过 MND 来分辨异常节点和正常节点,在扩展过程中,当某一节点被删除的次数超过 MND 时,便认为该节点是异常节点,选择不再加入该节点。显然,在这 8 组真实网络中,参数 c 选取该 8 组网络 MND 的最大值 6 便可解决易死循环的问题。实际上,该参数值在本文所用的人工合成网络中也有效。值得注意的是,参数 c 的选取与数据集相关。一般地,当数据集节点数越大时,MND 和 TAN 也越大,参数 c 也应该越大。

表 4 有关惩罚指标的结果

Table 4 Results about penalty indicator

	Karate	Dolphins	Pol. books	Football	C. elegans	Email	Power	PGP
MND	1	1	2	1	4	2	6	6
TAN	0	0	0	0	0	1	16	25

结束语 本文提出了一种新的基于中心团的局部扩展算法 CLEM,用于检测重叠社区。CLEM 算法一方面引入中心团和惩罚指标,改善了 LFM 算法的不稳定性;另一方面改进适应度函数,解决了传统重叠算法的参数依赖问题,并缩短了计算时间。在合成网络和现实网络上的测试结果表明,与已有算法相比,无论是在计算时间还是在检测准确度方面,CLEM 算法均有很好的表现。

参 考 文 献

[1] SPORNS O, CHIALVO D R, KAISER M, et al. Organization, development and function of complex brain networks [J]. Trends in Cognitive Sciences, 2004, 8(9): 418-425.

[2] MEHTA R L, KELLUM J A, SHAH S V, et al. Acute Kidney Injury Network: report of an initiative to improve outcomes in acute kidney injury [J]. Critical Care, 2007, 11(2): R31.

[3] NEWMAN M E J, PARK J. Statistical mechanics of networks [J]. Physical Review E, 2004, 70(6): 266117.

[4] NEWMAN M E J. The structure and function of complex networks [J]. Siam Review, 2003, 45(2): 167-256.

[5] BULLMORE E, SPORNS O. Complex brain networks: graph theoretical analysis of structural and functional systems [J]. Nature Reviews Neuroscience, 2009, 10(3): 186-198.

[6] NEWMAN M E J, GIRVAN M. Modularity and community structure in networks [J]. Proc. Natl. Acad. Sci. USA, 2006, 103(23): 8577-8582.

[7] WHITE S D M, FRENK C S. Galaxy formation through hierarchical clustering [J]. The Astrophysical Journal, 1991, 379(1): 52-79.

[8] TANG X Q, ZHU P. Hierarchical clustering problems and analysis of fuzzy proximity relation on granular space [J]. IEEE Transactions on Fuzzy Systems, 2013, 21(5): 814-824.

[9] TAO H, TANG X Q. Clustering Structural Analysis on Fuzzy Proximity Relation [J]. Computer Science, 2013, 40(1): 263-267.

[10] KRZAKALA F, MOORE C, MOSSEL E, et al. Spectral redemption in clustering sparse networks [J]. Proceedings of the National Academy of Sciences, 2013, 110(52): 20935-20940.

[11] ZHANG X S, WANG R S, WANG Y, et al. Modularity optimization in community detection of complex networks [J]. Europhysics Letters, 2009, 87(4): 49901.

[12] ZHANG Q, LI H. MOEA/D: A multiobjective evolutionary al-

gorithm based on decomposition [J]. IEEE Transactions on Evolutionary Computation,2008,11(6):712-731.
[13] MAITY S,RATH S K. Extended clique percolation method to detect overlapping community structure [C]// Proceedings of the International Conference on Advances in Computing,Communications and Informatics. Washington,USA:IEEE,2014.
[14] WEN X,CHEN W N,LIN Y,et al. A maximal clique based multiobjective evolutionary algorithm for overlapping community detection [J]. IEEE Transactions on Evolutionary Computation,2017,22(3):363-377.
[15] NEWMAN M E J. Finding and evaluating community structure in networks [J]. Physical Review E,2004,69(2):026113.
[16] PU S Y,WONG J,TURNER B,et al. Up-to-date catalogues of yeast protein complexes [J]. NuCLEMic Acids Research,2009,37(3):825-831.
[17] XIE J,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):1-35.
[18] 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.
[19] 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.
[20] BECKERE,ROBISSON B,CHAPPLE C E,et al. Multifunctional proteins revealed by overlapping clustering in protein interaction network [J]. Bioinformatics,2012,28(1):84-90.
[21] NEPUSZ,TAMÁS,YU H,et al. Detecting overlapping protein complexes in protein-protein interaction networks [J]. Nature Methods,2012,9(5):471-472.
[22] DING Z,ZHANG X,SUN D,et al. Overlapping community detection based on network decomposition [J]. Scientific Reports,2016,6:24115.
[23] KARRER B,NEWMAN M E J. Stochastic blockmodels and community structure in networks [J]. Physical Review E,Statstical,Nonlinear,and Soft Matter Physics,2010,83(2):016107.
[24] GREGORY S. Finding overlapping communities in networks by label propagation [J]. New Journal of Physics,2010,12(10):103018.
[25] YANG X H,SHEN M. Community detection algorithm based on local similarity of feature vectors [J]. Computer Science,2020,47(2):58-64.
[26] EVANS T S,LAMBIOTTE R. Line graphs,link partitions,and overlapping communities [J]. Physical Review E,2009,80(1):016105.
[27] AHN Y Y,BAGROW J P,LEHMANN S. Link communities reveal multiscale complexity in networks [J]. Nature,2010,466:761-764.
[28] LAN H,GUISHEN W,YAN W,et al. Link Clustering with Extended Link Similarity and EQ Evaluation Division [J]. PLoS ONE,2013,8(6):e66005.
[29] PIZZUTI C. Overlapped community detection in complex networks [C]// Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation. New York,USA:ACM,2009:859-866.
[30] FORTUNATO S. Community detection in graphs [J]. Physics Reports,2009,486(3):75-174.
[31] ZHANG X,ZHANG S,WANG R,et al. Quantitative function for community detection [J]. Physical Review E,2008,77:036109.
[32] RADICCHI F,CASTELLANO C. Defining and identifying communities in networks [J]. Proceedings of the National Academy of Sciences of the United States of America,2004,101(9):2658-2663.
[33] FORTUNATO S,BARTHELEMY M. Resolution limit in community detection [J]. Proceedings of the National Academy of Sciences,2007,104(1):36-41.
[34] SHEN H,CHENG X,CAI K,et al. Detect overlapping and hierarchical community structure in networks [J]. Physica A,2009,388(8):1706-1712.
[35] LANCICHINETTI A,FORTUNATO S,RADICCHI F. New benchmark in community detection [J]. Physical Review E,2008,78(4):561-570.
[36] DANON L,DÌAZ-GUILERA A,DUCH J,et al. Comparing community structure identification [J]. Journal of Statistical Mechanics,2005,DOI:10.1088/1742-5468/2005/09/p09008.
[37] LANCICHINETTI A,FORTUNATO S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities [J]. Physical Review E,2009,80(1):016118.
[38] NEWMAN M E J. Network Data[EB/OL]. [2013-04-19]. http://www-personal.umich.edu/~mejn/netdata/.
[39] ARENAS A. Data sets[EB/OL]. http://deim.urv.cat/~alexandre.arenas/data/welcome.htm.

XUE Lei, born in 1996, postgraduate. His main research interests include intelligent computing and bioinformatics.

TANG Xu-qing, born in 1963, Ph.D, professor. His main research interests include intelligent computing, bioinformatics, modeling and simulation of ecological systems.