

基于子图结构的局部社区发现算法

陈湘涛 赵美杰 杨 梅

湖南大学信息科学与工程学院 长沙 410000



摘 要 局部社区发现算法通常选取种子节点进行社区发现,针对现有重叠社区发现算法中种子节点选取时有效性不足的问题,提出了一种基于子图结构的局部社区发现算法(Subgraph Structure Based Overlapping Community Detection, SUSBOCD)。该算法提出了一种新的节点重要性度量指标,不仅考虑了节点的邻居数量,同时也考虑了邻居间的链接紧密程度。首先,选取未被访问且重要性最大的节点以及与其最为相似的邻居节点,将该两个节点及其公共邻居节点合并形成一个初始种子子图,该过程迭代运行直到所有节点均被访问;其次,根据种子子图的邻域信息进行相似度判断,若相似则进行合并,从而形成初始社区结构,持续扩展该过程直到所有种子子图均被访问;最后,对社区进行优化处理,若存在未分配社区的节点,则将其加入到最相似的初始社区,再合并重叠度较高的初始社区结构。在人工数据集和真实数据集上,对所提算法进行实验验证,实验结果表明,与其他重叠社区发现算法相比,SUSBOCD 算法在 ONMI, EQ 和 Omega 这 3 个评价指标上均有所提升,即该算法能有效地提高重叠社区的划分质量。

关键词: 重叠社区发现;局部扩展;种子选取;社区扩展;社区优化

中图法分类号 TP391

Overlapping Community Detection Algorithm Based on Subgraph Structure

CHEN Xiang-tao, ZHAO Mei-jie and YANG Mei

College of Computer Science and Electronic Engineering, Hunan University, Changsha 410000, China

Abstract Local community detection algorithms usually select seed nodes for community detection. To improve the quality of effectiveness of seed node selection, we propose an overlapping community detection algorithm based on subgraph structure (SUSBOCD). This algorithm proposes a new measure of node importance, which not only considers the number of neighbors, but also considers the degree of density between neighbors. First, SUSBOCD selects the most important node that is not visited and the most similar neighbor node, and merges the two nodes and their common neighbor nodes to form an initial seed subgraph. The process runs iteratively until all nodes have been visited. Second, the similarity is judged according to the neighborhood information of the seed subgraph. If it is similar, it is merged to form the initial community structure. The process runs iteratively until all seed subgraphs are visited. Finally, we optimize the community. If there are nodes without assigned communities, they are added to the most similar community, and then the community structure with high overlap is merged. Experiments on real and artificial networks show that SUSBOCD can improve the quality of overlapping community partition effectively in the three evaluation indexes of ONMI, EQ and Omega.

Keywords Overlapping community detection, Local expansion, Seed selection, Community expansion, Community optimization

1 引言

在现实世界中,随着人类技术的不断发展,人与人、人与物以及物与物之间的联系越来越复杂。研究人员的研究对象从独立的个体拓展为错综复杂的整体,研究对象是通过复杂网络形式来进行概述的。复杂网络无处不在,例如社交网络、引文网络、蛋白质网络、交通网络等^[1-2]。复杂系统由不同的实体以及实体之间的联系组成。研究人员将复杂系统抽象为

复杂网络进行研究,将实体抽象为节点,而实体之间的关系抽象为节点之间的联系。研究人员通过研究复杂网络来揭示复杂网络潜在的重要特征。在复杂网络中,特征相同的节点会被分配到同一个集群中,即社区。一个社区的内部节点之间的联系较为紧密,而位于不同社区的节点间的联系相对较少^[3]。

至今为止,国内外研究人员已提出许多应用于非重叠以及重叠结构的社区发现算法,重叠社区结构发现对于错综复

到稿日期:2020-11-02 返修日期:2021-03-22 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(61873089);国家重点研发计划项目(2018YFC0910405);国家自然科学基金(61572180)

This work was supported by the National Natural Science Foundation of China (61873089), National Key R&D Program of China (2018YFC0910405) and National Natural Science Foundation of China(61572180).

通信作者:陈湘涛(xtchen2009@sina.cn)

杂的现实网络有着重大意义,相对于非重叠社区结构更接近现实网络的情况。但是,针对现实复杂网络规模越来越大以及实体之间的联系越来越紧密等现象,如何准确且有效地挖掘重叠社区结构仍是亟需解决的问题。

研究人员提出若干基于局部信息的社区发现算法,这些算法通常从种子节点出发,利用局部信息进行社区结构发现,不仅要保证初始种子的质量,而且还要根据不同方式对初始种子的邻域信息进行局部扩展,因此种子的选择直接影响了社区发现的质量。针对现有算法对种子结构重要性计算考虑不够全面和社区扩展策略的不足,本文提出了一种基于子图结构的局部社区发现算法(SUSBOCD)。首先根据节点的邻居数以及邻居之间链接的紧密情况来度量种子的重要性,其次根据节点重要性以及其邻域信息来确定种子子图结构,然后根据改进的节点与社区相似度公式来对种子子图的邻域信息进行扩展,最后对种子子图进行优化处理。

本文的主要贡献包括两个方面:

(1)提出了一种基于节点度和节点邻居链接密度的节点重要性来衡量指标与节点间相似度的计算方法,从而更加准确地寻找种子结构;

(2)基于节点重要性和节点间相似性提出了一种基于子图结构的重叠社区发现算法 SUSBOCD。

本文第2节描述了社区发现的相关研究;第3节给出了本文算法的基本概念;第4节详细介绍了本文算法;第5节进行了实验并对结果进行了分析;最后总结全文。

2 相关研究

局部社区发现算法通常根据网络中的局部或者全局信息来选取节点作为种子,随后进行局部扩展。因此,如何选取种子节点是局部社区发现的关键所在。经典 LFM 算法^[4]通过随机选取网络中未被访问的节点进行局部扩展,该方法的稳定性差。LMD^[5]和 SGE^[6]算法根据局部度中心性选择种子节点,即指节点的邻居数量均大于其邻居节点的邻居数量。Wang 等^[7-8]受文献^[9]的启发,提出结构中心性来表示节点重要性,并据此进行局部社区发现。i-SEOC^[10]算法根据度和 Jaccard 系数计算节点影响力分数,进而选取种子节点进行种子扩展。

Zhang 等^[11]提出了 W-CPM,为了解决 CPM 算法对于复杂大型网络计算成本大的问题,还提出了一种弱 CPM,简称 W-CPM,用于大规模复杂网络中的重叠社区发现。W-CPM 的关键思路是找到网络中的弱团,然后合并弱团以发现更大的社区。现有的局部社区发现算法往往从种子节点出发进行扩展,而本文受 W-CPM 算法的启发,寻找初始种子子图,由种子子图向外扩展。

综上所述,现有局部社区发现算法具有不同的种子选取方案,但对种子重要性的计算考虑得不够全面。本文提出了一种结合节点度和密度来度量节点重要性的衡量指标,该指标充分考虑了节点的邻居数量以及邻居之间的链接情况,能够更加准确地衡量节点的重要性;同时,本文受 W-CPM 算法的启发,提出了一种基于密度和度的局部扩展算法,该算法根据网络局部信息计算节点的重要性,并根据节点重要性的全

局排名以及局部最大相似性来选取种子子图,从子图而非种子节点进行社区的扩展。

3 基本概念

$G=(V,E)$ 用于表示复杂网络结构。其中, $V=\{v_1,v_2,v_3,\dots,v_n\}$, $E=\{e_1,e_2,e_3,\dots,e_m\}$, n 和 m 分别表示网络结构中的节点集合和边集合。一般社区定义为,社区内部节点之间的链接关系较为紧密,而社区之间的节点则链接较为稀疏。对于重叠社区发现,最终结果是找出重叠社区,即 $C=\{c_1,c_2,c_3,\dots,c_n\}$, $c_1 \cup c_2 \cup c_3 \cup \dots \cup c_n \subseteq V$ 。

定义 1(基于度和密度的节点重要性) 节点 i 的重要性 $NodeImportantce(i)$ 定义为:

$$NodeImportantce(i) = degree(i) * (density(i) + 1) \quad (1)$$

$$density(i) = \begin{cases} \frac{2 m_{nel(i)}}{degree(i) * (degree(i) - 1)}, & degree(i) \neq 1 \\ 0, & degree(i) = 1 \end{cases} \quad (2)$$

其中, $nel(i)$ 表示与节点 i 存在链接关系的邻居节点集, $degree(i)$ 表示与节点 i 存在链接关系的节点数量, $density(i)$ 表示与节点 i 存在链接关系的邻居节点之间链接的紧密性, $m_{nel(i)}$ 表示与节点 i 存在链接关系的邻居节点之间的链接数量。该指标不仅考虑了节点的邻居数,而且还考虑了邻居之间链接的紧密程度。

定义 2(节点相似性) 节点 v 与邻居节点 i 的相似性 $Sim(v,i)$ 定义为:

$$Sim(v,i) = |nel(v) \cap nel(i)| * NodeImportantce(i) \quad (3)$$

其中, $Sim(v,i)$ 越大,表明两节点越相似。

定义 3(局部扩展函数) 节点 i 和社团 C 的局部扩展函数 $f(i,C)$ 定义为:

$$f(i,C) = \frac{|nel(i) \cap C| + E(i,C)}{\min(|nel(i)|, |C|)} \quad (4)$$

其中, $E(i,C)$ 表示节点 i 与社区结构 C 内部节点之间存在的链接数。

为了计算节点与社区之间的相似性,本文将采用文献^[10]中的相关定义进行计算,具体定义如下。

定义 4(节点与社区相似度^[10]) 节点 i 和社区 C 的相似度 $Sim(i,C)$ 的定义为:

$$Sim(i,C) = \frac{|nel(i) \cap C|}{|nel(i) \cup C|} \quad (5)$$

其中, $Sim(i,C)$ 越大,表明节点 i 属于社团 C 的概率越大。

定义 5(社区相似度^[10]) 社团 C_i 和 C_j 的相似度定义为:

$$Sim(C_i, C_j) = \frac{|C_i \cap C_j|}{\min(|C_i|, |C_j|)} \quad (6)$$

其中, $Sim(C_i, C_j)$ 的值越大,表明两社团的结构越相似。

4 基于子图结构的局部社区发现方法

4.1 算法描述

本文提出了一种基于子图结构的局部社区发现算法 SUSBOCD。该算法的伪代码如算法 1 所示。其中,第 1—4 行展示了节点重要性的计算,第 7—24 行确定了种子子图结构,第 25—32 行介绍了种子子图结构的局部扩展,第 33—36

行展示的是社区优化。

算法 1 基于节点相似性的局部社区发现算法

输入:图 $G=\langle V,E\rangle$;节点与社区相似度阈值 T ;

社区相似度阈值 S

输出:社区划分结果 $C=\{C_1,C_2,C_3,\cdots,C_k\}$

```
1. 初始化社区:  $C=\emptyset$ 
2. for each  $v$  in  $V$  do
3.     根据式(1)计算节点  $v$  的重要性
4. end for
5. 根据节点重要性降序排列得到  $nodes=\{v_1,v_2,\cdots,v_n\}$ 
6. 初始化种子子图集:  $seeds=\emptyset$ 
7. for each  $i$  in  $nodes$  do
8.     初始化节点相似度集:  $sim=\emptyset$ 
9.     if 节点  $i$  未访问 then
10.        标记节点  $i$  为已访问
11.        for each  $j$  in 节点  $i$  的邻居节点集
12.            if 节点  $j$  未访问 then
13.                利用式(3)计算节点  $i$  和  $j$  的相似性  $sim_{ij}$ 
14.                 $sim_{ij}$  加入  $sim$  集合中
15.            end if
16.        end for
17.        if  $sim$  存在 then
18.            未访问邻居节点中相似性最大的节点  $j_{Max}$ 
19.             $seed=i+j_{Max}+commonNeighbors(i,j_{Max})$ 
20.            将  $seed$  加入种子子图集合  $seeds$  中
21.             $seed$  中的节点均标记为已访问
22.        end if
23.    end if
24. end for
25. for each  $seed$  in  $seeds$  do
26.    for each  $nel$  in  $seed$  的邻居节点 do
27.        利用式(4)计算节点  $nel$  与种子子图结构的相似性  $f$ 
28.        if  $f$  大于阈值  $T$  then
29.            将邻居节点  $nel$  加入种子子图结构中
30.        end if
31.    end for
32. end for
33. 根据初始社区划分结果  $coms$  判断未分配社区的离散节点  $isolatedNodes$ 
34. if  $isolatedNodes$  存在 then
35.     根据式(5)将离散节点加入最相似的邻居社区
36. end if
37. 根据式(6)及给定的社区相邻度阈值  $S$  进行社区结构合并
38. 返回  $C$ 
```

4.2 时间复杂度分析

假设 n,m,d,c 分别代表网络中的所有节点数、链接数、节点的平均度数、初始子图结构数。第一阶段种子结构选取过程的时间复杂度为 $O(n+n\log n+nd)$ 。在第二阶段,基于种子子图结构的邻域信息进行局部扩展,时间复杂度为 $O(cd)$ 。在第三阶段的社区优化过程中,社区数量为 c ,离散节点数量为 s ,时间复杂度为 $O(sc+c^2)$ 。由于其中 s,c 和 d 这 3 个参数远远小于 n ,因此 SUSBOCD 算法的总体时间复杂度化简为 $O(n+n\log n)$ 。

5 实验

5.1 实验环境描述

实验测试结果均在 1.8 GHz Intel Core i5 的处理器以及内存为 8GB 的 MacBook Air 上实现,所有相关算法的代码均用 Python 2.7 撰写。

5.1.1 真实和人工数据集

表 1 详细列出了用于进行对比实验的 8 个真实数据集的信息,其中 n 表示网络结构中包含的节点数量, m 表示网络结构中节点之间所有的链接数量, k 表示网络结构中节点的平均邻居数量。

表 1 真实数据集信息

Table 1 Real dataset information			
Networks	n	m	k
Karate	34	78	4.58
Dolphins	62	159	5.13
PolBooks	105	441	8.4
Football	115	613	10.66
Email	1133	5451	9.62
NetScience	1589	2742	3.45
ca-HepTh	9877	25998	5.26
ca-CondMat	23133	93497	8.08

人工合成网络采用 Lancichinetti 提出的生成合成数据的程序,通过调节参数来生成不同网络,主要通过调节不同节点总数 n 、重叠节点数量 on 、网络中一个节点最多属于的社区数 om 以及混合参数 μ 进行对比实验。本文使用 LFR 基准生成 4 组网络,详细参数设置如表 2 所列。

表 2 4 组合成网络的参数设计

Table 2 Parameter design of four combination network								
LFR	N	K	MaxK	MinC	MaxC	μ	om	on
L1	1000	15	50	10	50	0.1	2	$0.1N\sim 0.5N$
L2	1000	15	50	10	50	0.1	2~8	100
L3	1000	15	50	10	50	0.1~0.6	2	100
L4	$1000\sim 10000$	15	50	10	50	0.1	2	0.1N

5.1.2 对比算法

将本文算法 SUSBOCD 与 LFM^[4], Copra^[12], CPM^[13], Demon^[14], SGE^[6], OCDNW^[15] 以及 i-SEOCD^[11] 这 7 种算法进行对比实验。

本文参考对比算法对应原文的参数设置,具体参数设置如下:LFM 算法中 α 的范围为 0.5~1.5,每次增加 0.1; Copra 算法中 v 的范围为 3~8;CPM 算法中 k 的大小范围为 3~8,每次增加 1;SGE 算法中 α 的范围为 0.8~1.2,每次增加 0.1;i-SEOCD 算法中 $p=0.9,s=0.05,\alpha$ 的范围为 0.8~1.2,每次增加 0.1, $\epsilon=0.4$ 。每次记录算法的最优实验结果。

5.1.3 参数敏感度分析

本文算法 SUSBOCD 在局部扩展过程中需要判断节点和种子子图的相似性,若其相似性大于设定相似度阈值 T ,则加入种子子图。图 1 给出了 6 个不同真实数据集在不同相似度阈值 T 下 EQ 值的变化情况。由图 1 可知,SUSBOCD 算法在不同数据集上均在 0.6~1.0 范围内较稳定,因此本文相似度阈值 T 的取值范围为 0.6~1.0。

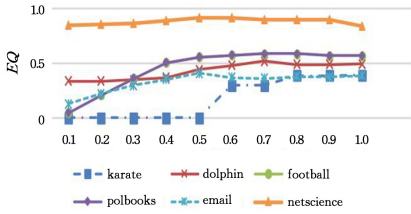


图 1 参数 T 的实验
Fig. 1 Experiments with parameter T

表 3 在不同真实数据集的 EQ 值比较
Table 3 EQ values in different real datasets

EQ	LFM	Copra	CPM	Demon	SGE	OCDNW	i-SEOCD	SUSBOCD
karate	0.3320	0.3519	0.1858	0.0376	0.3833	0.3947	0.3459	0.4000
dolphin	0.4078	0.3936	0.3612	0.2653	0.2915	0.5032	0.4787	0.5153
football	0.3799	0.5925	0.5549	0.2299	0.2825	0.5322	0.5410	0.5838
polbooks	0.4934	0.5085	0.4409	0.2217	0.2371	0.5146	0.4701	0.5110
email	0.1215	0.3641	0.2641	0.0676	0.1288	0.3945	0.3683	0.4181
netscience	0.8579	0.8392	0.7044	0.6358	0.7117	0.8716	0.8044	0.9065
ca-HepTh	0.2917	0.6000	0.3685	0.2373	0.2813	0.5575	0.4800	0.5725
ca-CondMat	0.2917	0.5670	0.3053	0.1630	0.2406	0.4810	0.4777	0.4972
平均值	0.3970	0.5271	0.3981	0.2323	0.3196	0.5312	0.4958	0.5506

图 2 给出了不同算法在已知真实分区数据集上 ONMI 和 Omega 两个指标的对比情况。SUSBOCD 算法在 Karate, Football 和 Polbooks 数据集上的表现最优,最接近于真实分

区情况。由于 LFM 算法存在随机性强和算法稳定性差等问题,而 Copra 算法在传播过程中也具有随机性,因此,综合考虑得出 SUSBOCD 的整体表现最好。

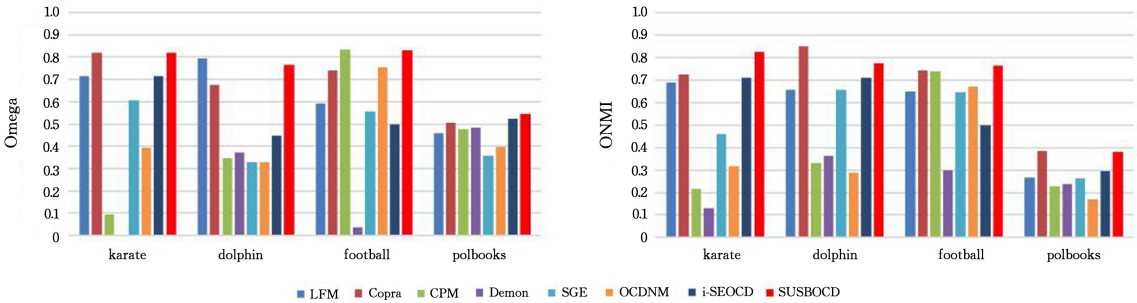


图 2 真实社区划分的 Omega 和 ONMI 值比较
Fig. 2 Omega and ONMI experiments on real community datasets

5.2.2 合成网络实验分析

(1)重叠节点 on 对算法的影响

图 3 给出了在 L1 组网络上不同算法在重叠节点数 on 不断增加的情况下社区划分结果的对比分析。由图 3 可知,随着重叠节点数不断增加,各算法的 EQ 均不断下降。随着网络

结构重叠节点数不断增加,在网络越复杂的情况下,SUSBOCD 的 EQ 值表现较好,Omega 和 ONMI 这两个指标较其他对比算法的表现最稳定,变化趋势最小。表 4 列出了 SUSBOCD 相比对比算法在 L1 组网络上的平均提升百分比。总体可得,随着重叠节点数的增加,SUSBOCD 算法的性能表现最好。

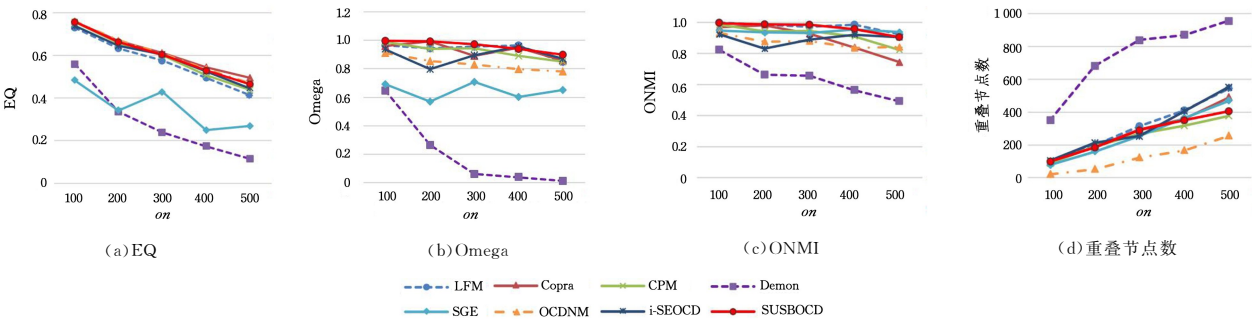


图 3 重叠节点 on 变化的对比实验图
Fig. 3 Experiments at on change

表 4 SUSBOCD 算法在 L1 组网络相比对比算法的平均提升百分比

Table 4 Average improvement percentage of the SUSBOCD algorithm in the L1 group network relative to the comparison algorithm

评价指标	LFM	Copra	CPM	Demon	SGE	OCDNW	i-SEOCD
EQ	6.1	-2.3	2.3	55.7	41.7	-1.0	2
Omega	2.7	3.3	5.8	2285.5	49.7	15.1	8.1
ONMI	-0.8	9.0	5.1	54.8	2.5	10.7	8.2

图 3(d)给出了随着重叠节点数的增加,各算法最终挖掘的重叠节点数的对比分析。由图可知,相比真实重叠节点数, Demon 和 OCDNW 算法分别挖掘出过多和过少的重叠节点,而 SUSBOCD 挖掘的重叠节点数接近真实情况。

(2)节点隶属不同社区数量 om 对算法的影响

图 4 给出了 L2 组网络上不同算法在节点社区隶属度增加的情况下性能的对比分析,其中 SUSBOCD 和 CPM 算法的表现最好。表 5 列出了 SUSBOCD 算法相比对比算法在 L2 组网络上的平均提升百分比,实验结果表明,随着 om 的增加,SUSBOCD 算法的性能表现最好。

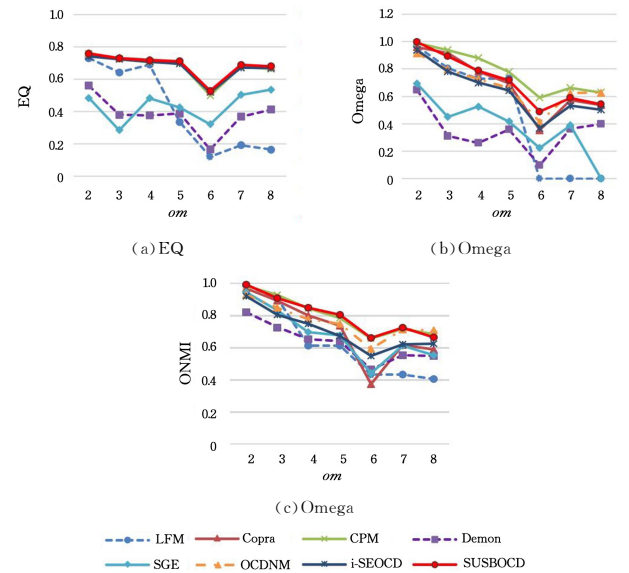


图 4 om 变化时 EQ, Omega 和 ONMI 的对比实验结果
Fig. 4 Experiments with EQ, Omega and ONMI at om change

表 5 SUSBOCD 算法在 L2 组网络上相比对比算法的平均提升百分比

Table 5 Average improvement percentage of the SUSBOCD algorithm in the L2 group network relative to the comparison algorithm

评价指标	LFM	Copra	CPM	Demon	SGE	OCDNW	i-SEOCD
EQ	149.3	-0.1	2.0	95.6	65.3	0.1	1.8
Omega	5	1.4	-5.6	135.4	65.7	9.5	11.0
ONMI	36.6	18.0	0	28.1	20.4	5.2	13.9

(3)混合参数变化对算法的影响

图 5 给出了 L3 组网络上,随着混合参数的不断增加即网络结构越来越复杂的情况下,各算法在不同指标上的对比情况。当混合参数大于 0.4 时,SUSBOCD 算法的下降趋势最小。表 6 列出了 SUSBOCD 算法相比对比算法在 L3 组网络

上的平均提升百分比,表明 SUSBOCD 算法在混合参数增加的情况下社区划分质量优于其他对比算法。

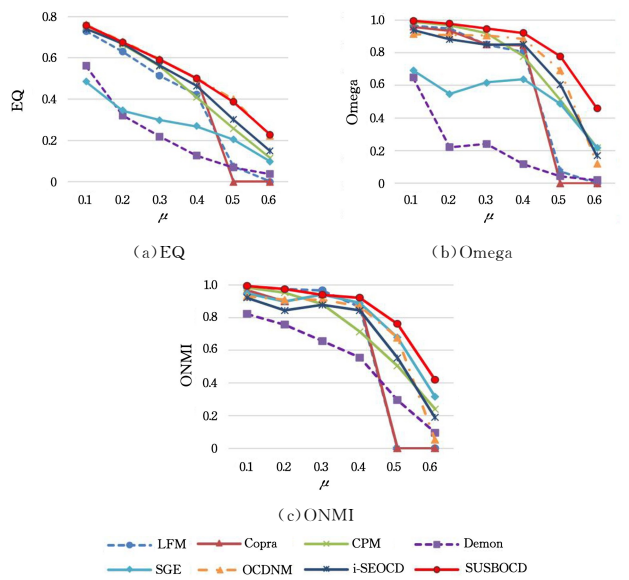


图 5 混合参数 μ 变化时 EQ, Omega 和 ONMI 的对比实验结果
Fig. 5 Experiments with EQ, Omega and ONMI at μ change

表 6 SUSBOCD 算法在 L3 组网络上相比对比算法的平均提升百分比

Table 6 Average improvement percentage of the SUSBOCD algorithm in the L3 group network relative to the comparison algorithm

评价指标	LFM	Copra	CPM	Demon	SGE	OCDNW	i-SEOCD
EQ	95.6	33.3	29.6	267.8	93.2	67.0	74.8
Omega	194.3	6.9	30.8	920.1	64.9	53.4	39.9
ONMI	1.1	3.6	27.3	108.5	10.3	127.8	33.4

(4)合成网络规模不同对算法的影响

图 6 给出了在 L4 组网络上,当合成网络数据规模不断增加时,各算法的性能情况。

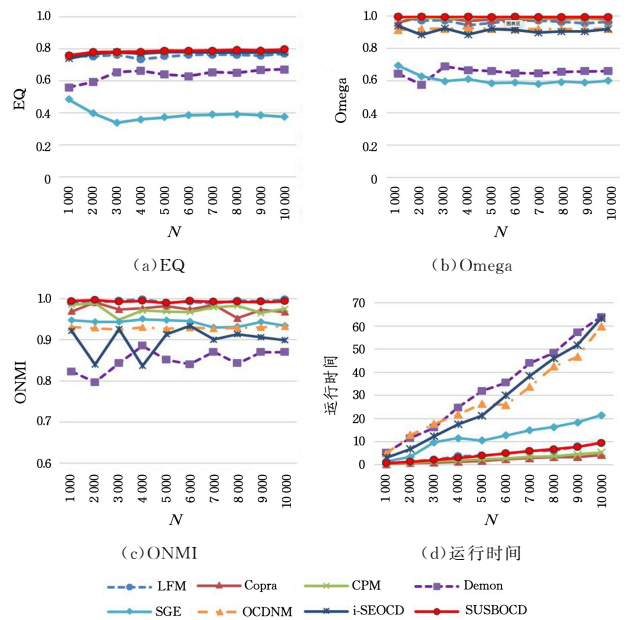


图 6 合成网络规模 N 变化时的对比实验结果图
Fig. 6 Experiments at N change

由图 6 可知,从整体情况看,随着网络规模的不断增大,SUSBOCD 算法的性能表现最稳定且最好。图 6(d)给出了在 L4 组网络社区规模不同时不同算法的运行时间的对比分析。表 7 列出了 SUSBOCD 算法在 L4 组网络上相比对比算法的平均提升百分比。整体来说,SUSBOCD 算法在合成网络规模不断增大的情况下,能够在较短时间内实现较好的社区分区。

表 8 列出了在 L4 组网络中不同算法的真实社区划分数量的对比情况,True 代表真实分区数量。由表清楚可得,SUSBOCD 算法能够较为准确地确定合成网络的社区分区数量。

表 8 不同规模合成网络的社区数量比较

Table 8 Comparison of the number of communities in different LFR benchmark network datasets

N	True	LFM	Copra	CPM	Demon	SGE	OCDNW	i-SEOCD	SUSBOCD
1 000	41	41	39	41	40	93	41	43	41
2 000	69	76	69	69	70	198	69	78	69
3 000	100	108	101	104	101	360	100	102	100
4 000	131	151	134	142	145	409	137	150	137
5 000	168	186	166	173	179	519	169	182	168
6 000	207	221	204	214	219	643	209	215	207
7 000	234	250	232	237	236	703	237	246	234
8 000	268	292	261	268	283	797	269	269	268
9 000	302	326	294	312	321	971	304	303	302
1 000	328	352	321	334	347	1 062	328	345	328

通过人工合成网络和真实网络的实验结果可得,SUSBOCD 算法不仅能够实现较好的社区划分结果,也能较为准确地发现重叠节点以及分区情况,并且有较好的时间复杂度,这主要是由于 SUSBOCD 采用节点重要性和局部相似性来确定种子子图,能够找到较好的初始子图结构,从而获得更准确的社区发现结果。

结束语 本文提出了一种基于子图结构的局部重叠社区发现算法 SUSBOCD。该算法根据节点重要性以及局部最大相似性来确定种子子图结构,并基于种子子图的邻域信息进行局部扩展,最后进行社区优化。

将 SUSBOCD 算法与一些性能较好的经典算法以及最近两年提出的局部社区发现算法在不同真实网络以及合成网络上进行了对比分析。实验结果表明,SUSBOCD 算法的社区发现效率明显更优,具有较好的表现。本文算法在规模较小的网络中效果较好,但当网络规模较大时还有待进一步的提升,下一步将考虑进行并行化处理。本文算法研究的网络为静态网络,但真实情况下网络将随着时间不断变化,下一步将进一步研究动态网络的社区发现。

参 考 文 献

[1] NEWMAN M E J. The structure of scientific collaboration networks[J]. Proceedings of the National Academy of Sciences, 2001,98(2):404-409.

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

[3] GIRVAN M,NEWMAN M E J. Community structure in social and biological networks[J]. Proceedings of the National Academy of Sciences,2002,99(12):7821-7826.

[4] LANCICHINETTI A,FORTUNATO S,KERTÉSZ J. Detect-

表 7 SUSBOCD 算法在 L4 组网络上相比对比算法的平均提升百分比

Table 7 Average improvement percentage of the SUSBOCD algorithm in the L4 group network relative to the comparison algorithm

(单位:%)

评价指标	LFM	Copra	CPM	Demon	SGE	OCDNW	i-SEOCD
EQ	3.5	-0.1	0.5	23.2	103.8	-0.2	1.5
Omega	2.9	1.6	0.8	53.6	64.7	8.1	9.5
ONMI	0	1.9	2.0	17.0	5.5	6.9	10.6

ting the overlapping and hierarchical community structure in complex networks[J]. New Journal of Physics, 2009, 11(3): 033015.

[5] CHEN Q,WU T T,FANG M. Detecting local community structures in complex networks based on local degree central nodes [J]. Physica A:Statistical Mechanics and its Applications,2013, 392(3):529-537.

[6] LI Y,HE J,WU Y X. Overlapping Community Discovery Method Based on Greedy Expansion of Seed Nodes[J]. Journal of Chinese Computer Systems,2019,40(5):1115-1119.

[7] WANG X,LIU G,LI J,et al. Locating structural centers: A density-based clustering method for community detection[J]. PloS one,2017,12(1):e0169355.

[8] WANG X,LIU G,LI J. Overlapping community detection based on structural centrality in complex networks[J]. IEEE Access, 2017,5:25258-25269.

[9] RODRIGUEZ A,LAIO A. Clustering by fast search and find of density peaks[J]. Science,2014,344(6191):1492-1496.

[10] YU Z Y,CHEN J J,GUO K. Overlapping Community Detection Based on Influence and Seeds Extension[J]. Acta Electronica Sinica,2019,47(1):153-160.

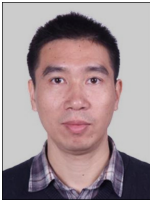
[11] ZHANG X,WANG C,SU Y,et al. A fast overlapping community detection algorithm based on weak cliques for large-scale networks[J]. IEEE Transactions on Computational Social Systems, 2017,4(4):218-230.

[12] GREGORY S. Finding overlapping communities in networks by label propagation[J]. New Journal of Physics, 2010, 12(10): 2011-2024.

[13] PALLA G,IMRE D,ILLÉS F,et al. Uncovering the overlapping community structure of complex networks in nature and society [J]. Nature,2005,435(7043):814-818.

[14] COSCIA M, ROSSETTI G, GIANNOTTI F, et al. DEMON: a Local-First Discovery Method for Overlapping Communities [C]//Proceeding of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2012;615-623.

[15] CHEN X, LI J. Overlapping Community Detection by Node-Weighting[C]//Proceedings of the 2nd International Conference on Compute and Data Analysis, ICCDA. 2018;70-74.



CHEN Xiang-tao, born in 1973, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include data mining, heterogenous information network, social network and network embedding.