面向大规模网络的快速重叠社团挖掘算法

李政廉,吉立新,黄瑞阳,兰巨龙
(国家数字交换系统工程技术研究中心,河南郑州 450003)

摘 要: 重叠社团在社交网络大数据中普遍存在.针对现有重叠社团挖掘算法易将重叠区域错误地划分为独立的社团且计算复杂的问题,提出了一种基于局部信息度量的快速重叠社团挖掘算法(Local information based Fast Overlapped Communities Detection,Li-FOCD).首先,为节点定义局部信息度量指标——社团连接度和邻居连接度,建模节点与社团的关系,缩小了计算范围;然后,每次并行地迭代执行缩减、扩展、去重等操作,并更新局部度量指标,通过松弛每次迭代的终止条件,发现近似最优社团集合而不是最优社团,最终算法复杂度为 $O(m+n)$.基于真实的大规模社交网络数据的试验分析表明:与当前流行的重叠社团挖掘算法相比,Li-FOCD 在不损失检测质量的前提下,大幅提升了计算效率.

关键词: 重叠社团挖掘;局部信息;社团连接度;邻居连接度
中图分类号: TP391.41 **文献标识码:** A **文章编号:** 0372-2112 (2019)02-0257-09
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.02.001

Fast Overlapping Communities Detection Algorithms for Large-Scale Social Networks

LI Zheng-lian,JI Li-xin,HUANG Rui-yang,LAN Ju-long
(*National Digital Switching System Engineering & Technological R&D Center,Zhengzhou,Henan* 450003,*China*)

Abstract: Overlap between community pairs is commonplace in large-scale social networks. The most existing overlapped community detection algorithms may falsely identify overlaps as communities because the overlap area is denser than others,and those algorithms are computationally demanding and cannot scale well with the size of networks. In this paper,we propose a fast overlapping community discovery algorithm based on some locally computed information—Local information based Fast Overlapped Communities Detection (Li-FOCD). Firstly,we introduce two local information metrics for each network node—community connectivity score and neighborhood connectivity score,to model the relationship between nodes and communities;secondly,based on local metrics,we can concurrently execute the iterations of reduction,expansion,and duplication removal to find the approximately optimal communities instead of the optimal community,and achieve a low complexity $O(m+n)$. Experimental analysis based on real large-scale social networking datasets shows that our algorithm outperforms some popular overlapped community finding algorithms in terms of computational time while not compromising with quality.

Key words: overlapping communities detection;Local information;community connectivity;neighborhood connectivit

1 引言

网络中社团结构的挖掘具有重要意义,已广泛应用于事件预测、商业情报、基础设施管理等众多领域[1].然而,网络中节点属性的多元化导致了社团结构的重叠性,例如,若在不同领域中出现有相同兴趣或目的人会导致重叠的社团.这种相互重叠的社团在社会图谱关系中很常见,传统未考虑重叠行的社团挖掘方法很难识别这种关系.此外,当前网络规模快速膨胀,往往包含数以百万计的节点以及数十亿的链路,对原本计算复杂的重叠社团挖掘算法提出了更高的要求.为应对这些问题,迫切需要研究一种高效的重叠社团挖

收稿日期:2017-12-12;修回日期:2018-05-15;责任编辑:蓝红杰
基金项目:国家自然科学基金(No.61521003)

掘算法.

因此,本文提出了一种基于局部信息度量的重叠社团快速挖掘算法(Local information based Fast Overlapped Communities Detection,Li-FOCD),为节点定义局部信息度量指标——社团连接度和邻居连接度,从而建模节点与社团的关系以缩小计算范围,解决大规模网络中重叠社团的快速发现问题. 基于真实的大规模社交网络数据的试验分析表明:与当前流行的重叠社团挖掘算法相比,Li-FOCD 在时间和效率方面都有更好的效果.

2 相关研究

社团挖掘是指对网络中的节点按如下原则进行分组检测:组内节点彼此密集地连接,而不同组的节点之间则为较稀疏地连接. 从拓扑角度,社团可看作拓扑聚类(clusters),然而,图聚类算法本质上是解决优化问题,属于计算密集型,主要应用于网络拓扑规模较小的场景[2]. 然而,近年来研究人员设计了一些快速计算方法,可以用于较大规模的社团挖掘. 文献[3]中描述了大量的图聚类方法. 例如,分层方法通常优化诸如电导、谱距离、模块性等全局得分(score),获得不相交的簇. Blondel 等在文献[5]中基于模块化优化,发现了大规模网络中层次化的不相交社团. 然而,将一个拓扑划分为多个不相交的簇的方法并不适用于当今的社交网络. 首先,社交网络中的社团一般是重叠的;其次,社交网络拓扑中的社团既有纵向的,也有横向的. 而仅通过节点的层次化聚类方法难以检测出这些社团[6].

另外,还有一些方法可以识别网络中的某些预先确定的结构,如派系(clique)[7]等. 这些方法虽然识别性能优异但是计算复杂度较高. 类似地,所有基于密集度的检测方法均无法识别该类社团[8]. 因为社交网络中的社团在重叠区域比非重叠区域的边更稠密[9].

文献[10,13]则从划分边的角度挖掘社团结构. 该类方法允许具有多个边的节点分别属于不同的社团. 但是该方法假设链路是同质的,也就是,两个节点因单一兴趣而连接在一起. 但是该假设与文献[9]中的统计规律不符,即一对节点之间存在连接的可能性随着它们共享的社团数目的增加而增加. 另外,利用非负矩阵分解算法识别社团也是常用的方法[15,16]. 文献[16]提出了一种基于非负矩阵分解的方法——BigClam 算法. 虽然 BigClam 声称其具有接近线性时间复杂度的运行效率,但是实验表明,对于大规模社交网络,BigClam 仍难以在短时间内计算出社团结构. 而且,BigClam 需要预先设置社团数量作为初始参数.

局部优化算法也是社团挖掘的一种重要方法. 局部优化算法成败关键是适应度函数的设计. 文献[4]通过对网络节点的局部信息传递过程进行建模,引入概率图有向矢量计算理论,实现社团的快速划分. 文献[17]将适应度得分设计为节点的邻居数量与整个社团中节点总数的比值. 文献[12,18~23]等均设计了不同的适应度函数.

标识传播算法(Label propagation algorithms,LPAs)是实现快速社团挖掘的一类重要方法. 标签传播算法初始化时为每个节点分配一个唯一标识,然后每个节点以接受具有最大共享值的邻居标识的原则驱动标识的传播[18,19,24~26]. COPRA[21]改进了经典的 LPA 算法[27],使得每个节点能够保留多个标识,以发现重叠社团结构. SLPA 算法[24]维护每个节点在迭代过程中接受的所有标识的出现频率. 但是,LPA 算法可能忽略规模较小的社团结构,而现实网络中小规模社团是普遍存在的. 文献[28]针对传统微博社区发现算法内聚低重叠度不可控制等问题,通过利用用户标签的共现关系等信息对标签加权,并对标签之间的内联及外联关系并将用户的标签进行扩充,提出了基于核心标签的可重叠微博网络社区划分方法. DEMON 算法[29]为每个节点提取局部网络,并将标识传播算法应用其中,最终可以发现已有社团之间的联盟,从而检测到重叠社团结构. 但是该算法仍不能避免 LPA 算法的限制.

B. Yang 等[30]聚焦于解决分布式和动态的网络社区的问题,提供了一种面向自治的计算方法来分布式和增量地挖掘网络社团. 文献[31]针对大型全局拓扑视图很难获得的问题,提出了一种局部社团挖掘算法,该方法可扩展应用于以检测非重叠和重叠的全局社区结构中. 文献[32]将在线社交网络中的每一个参与者都被认为是一个不合作博弈中的自私的参与者,提出了一种基于博弈论的图聚类框架挖掘大规模社交网络中的社团结构. 文献[33]从动态化的角度考虑可重叠社团发现,提出了基于隶属度的社会化网络重叠社区发现及动态集群演化分析.

综上可知,现有重叠社团挖掘算法一般计算复杂度较高,导致难以将其应用于大规模社交网络中. 为此,本文提出适用于大规模社交网络的社团结构检测算法——Li-FCOD,基于局部计算的得分值动态地发现重叠的社团结构. 该方法通过并发地选择多个近似最佳的社团,降低迭代次数,提高计算速度. Li-FCOD 算法检测的社团并不局限于某一个层次,而是在给定网络中包含所有有意义的社团. 而且,本文算法结果是稳定的,即,算法结果不依赖于迭代节点的顺序,而文献[6,18~23,25]均存在该问题.

3 Li-FOCD 算法

3.1 问题描述

设无向无权图 $G(V,E)$,不存在自环和并行边. 社

团挖掘问题建模为对子图 $S=\{S_i \mid S_i \subset V\}$ 搜索,使得子图 S_i 中的任意节点 v_j 与子图 S_i 连接紧密程度大于另一子图 S_j'. 其中 $S_j'=(S_k \mid v_j \notin S_k \wedge S_k \in S)$ 为 S 中任意不含节点 v_j 的子图. 每个子图 $S_i \in S$ 为一个社团. 对于任意节点 v_j, $\forall j \in \{1,2,\cdots,|V|\}$,令 $S(v_j)=\{S_i \mid v_j \in S_i \wedge S_i \in S\}$ 为包含节点 v_j 的一组社团. 此外,令 $S'(v_j)=S-S(v_j)$ 为不含节点 v_j 的一组社团. 如果每个节点 v_j 最多属于一个社团,也就是,$|S(v_j)| \leq 1$,那么称它为不相交簇,否则为重叠社团. 本文提出的算法 Li-FCOD 从给定的网络拓扑中检测重叠社团.

详细描述之前给出本文后续涉及的公式符号及描述,如表 1 所示.

表 1　符号描述

符号	描述
$G(V,E)$	无向无权图,其中 V 是节点集,E 是链路集
$S_i \subset V$	子图 i
$S(v_j)$	网络中包含节点 v_j 的一组社团
$\|V\|$	节点集 V 的元素
$S'(v_j)$	$S'(v_j)=S-S(v_j)$,网络中不含节点 v_j 的一组社团
$N(v_j)$	节点 $v_j \in V$ 的邻居节点的集合
$N_i(v_j)$	节点 v_j 在社团 $S_i \in S(v_j)$ 内的邻居
$\tilde{\xi}_j^i$	v_j 在社团 $S_i \in S(v_j)$ 中的社团连接度
ξ_j^i	v_j 在社团 $S_i \in S(v_j)$ 中的社团连接度,且 $\|N_i(v_j)\|>K$
dup	社团相似度门限
ζ_j^i	v_j 在社团 $S_i \in S(v_j)$ 中的邻居连接度
S^l	对于图 $G(V,E)$,阶段 l 时的社团结构,其中 S^0 为初始阶段的社团结构
Aug_i	社团 S_i 的外围节点集合
ξ_j^{th}	节点 v_j 的社团连接度的阶段门限
ζ_j^{th}	节点 v_j 对于邻居连接度的成员门限,用于衡量节点 v_j 是否为某社团的成员
$\langle \xi^i \rangle_j$	社团连接度序列,标识节点 v_j 加入社团结构 S^l 中所有社团后的社团连接度值的集合
$\langle \zeta^i \rangle_j$	邻居连接度序列,标识节点 v_j 加入社团结构 S^l 中所有社团后的邻居连接度值的集合
K	社团内节点的最小连接数
$\psi(C,C')$	社团 C 和 C' 的相似度

3.2　连接度

正如前一小节所述,对于任意节点 v_j, $\forall j \in \{1,2,\cdots,|V|\}$, v_j 与 $S(v_j)$ 中的所有社团的连接度大于与 $S'(v_j)$ 中社团的连接度. 因此,不妨认为,节点 v_j 与 $S(v_j)$ 中的所有社团具有等价的连接性. 那么,令 $N(v_j)$ 为节点 $v_j \in V$ 的邻居节点的集合,那么,

$$N(v_j)=\{v_k \mid (v_j,v_k) \in E\} \tag{1}$$

令 $N_i(v_j)$ 为节点 v_j 在社团 $S_i \in S(v_j)$ 内的邻居,记为

$$N_i(v_j)=\{v_k \mid (v_j,v_k) \in E \wedge v_k \in S_i\} \tag{2}$$

定义 1　社团连接度　节点的社团连接度定义为节点在社团内的邻居集合大小与社团规模大小减 1 的比值,即社团连接度 $\tilde{\xi}_j^i$ 为

$$\tilde{\xi}_j^i=|N_i(v_j)|/(|S_i|-1) \tag{3}$$

此外,为确保在任何社团中的任意节点在社团中至少存在 K 个邻居,公式(3)可重新定义为

$$\xi_j^i=\begin{cases}(|N_i(v_j)|-K+1)/(|S_i|-K), & 若|N_i(v_j)|>K \\ 0, & 其它\end{cases} \tag{4}$$

若 K 为非常大的值,小而稠密的社团将被忽略. 另一方面,若 K 为非常小的值,算法可以发现较稀疏的大型社团和微小的社团,且算法不会受 K 取较小值的影响. 当社团相似度门限 dup 取值为 0.6 时(关于社团相似度将在 3.3.4 小节定义),随着 K 的增加,图 1 给出了本文算法应用于 Amazon 网络[34]时检测的社团的统计变化. 可以看出:当 $K>5$ 之后,稀疏的拓扑将不存在社团结构. 除高密度网络拓扑结构之外,一般 $K=2$ 是较合理的取值.

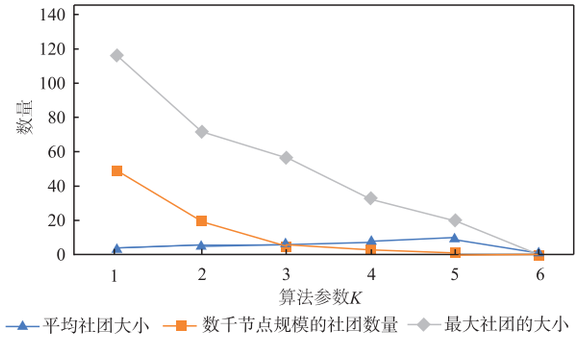

图1　社团数量统计

定义 2　邻居连接度　节点 v_j 在社团 S_i 中的邻居连接度 ζ_j^i 为节点在该社团内的邻居数量与节点在网络中的所有邻居数量之比,

$$\zeta_j^i=|N_i(v_j)|/|N(v_j)| \tag{5}$$

邻居连接度是节点 v_j 在社团 S_i 内邻居数量的比例,表示一个节点加入一个新社团的兴趣程度,社团连接度决定了节点对社团的归属关系. 表 2 给出了一个节点与社团之间的状态对照关系.

3.3　算法

算法 Li-FOCD 的驱动原则是社团由节点发起建立,并受到节点邻居和邻居社团的影响. 节点吸引其邻居节点加入它的社团. 若这些邻居节点在该社团中具有足够大的连接度,并选择留在该社团,否则离开该社团. 那么,随着算法的迭代更新,社团将不断地扩展或缩减. 因此,本文算法由初始化、缩减、扩展和去重等操作.

表 2 基于社团连接度与邻居连接度的社团中节点的状态

		社团连接度	
		低	高
邻居连接度	低	兴趣低，归属关系弱，节点未加入该社团	兴趣低，归属关系强，强连接社团但邻居少
	高	兴趣高，归属关系弱，节点不被社团接受	兴趣高，归属关系强，节点为该社团的中心

3.3.1 初始社团

初始化时，设节点 v_j，$\forall j \in \{1,2,\cdots,|V|\}$，至少具有 K 个邻居，与其邻居组成一个社团 S_i. 社团的数量等于节点度大于 K 的节点数量. 那么，当节点发起建立社团时，节点也成为该社团的一部分.

设网络拓扑 $G(V,E)$ 的初始社团结构记为 S^0，令 $Aug_i = \{v_k | v_k \in N(v_i) \wedge v_k \in S_i\}$，$\forall S_i \in S^0$，表示社团 S_i 的外围节点集合. 从物理意义上，Aug_i 是节点 v_i 的邻居节点集中属于 S_i 的所有节点. S_i 是节点 v_i 初始建立的社团，从社团 S_i 的角度，Aug_i 中的节点被划分给了社团 S_i，从 Aug_i 中节点 v_k 的角度，v_k 可以选择留在或离开社团 S_i. 因此，直观地，检测算法将在两个操作间迭代：缩减操作和扩展操作. 令包含这两个操作的每一次迭代称为一个阶段. 设阶段 l 的社团结构记为 S^l，对任意社团，后续阶段的缩减操作或扩展操作的对象均是外围节点. 算法 1 给出了初始化过程. 例如，设 $K=4$，v_1 的邻居集合为 $N(v_1) = \{v_3, v_5, v_7, v_8, v_9\}$，且 v_1 初始化社团 $S_1 = \{v_1\} \cup N(v_1)$，那么 $Aug_i = \{v_3, v_5, v_7, v_8, v_9\}$.

算法 1 函数 1 Initialization(G,K,S)

```
/* 对节点 v∈V,其邻居集合为 N(v),若|N(v)|≥K,那么,节点的社团由 v 和邻居集 N(v)初始化得到 */
For each i∈{1,2,…,n} do
  If |N(v_i)|≥K then
    S_i = {v_i} ∪ N(v_i);
    Aug_i ← N(v_i)
    S = S ∪ S_i;
  Else
    S_i = NULL, Aug_i = NULL;
  End if
End for
```

3.3.2 缩减操作

当某节点与其所在社团的其它节点的连接性不足时，缩减操作将该节点从该社团中删除. 对于每个节点 v_j，根据公式(4)和(5)可计算两个指标 ξ_j^i 和 ζ_j^i. 那么，对于任意节点 v_j 在 l 阶段社团结构 S^l 时，若加入多个社团时，可以获得一个社团连接度的序列 $\langle \xi^i \rangle_j = \{\xi_j^i | v_j \in S_i \wedge S_i \in S^l\}$，代表节点 v_j 加入社团结构 S^l 中所有社团后的社团连接度集合；邻居连接度序列 $\langle \zeta^i \rangle_j = \{\zeta_j^i | v_j \in S_i \wedge S_i \in S^l\}$，代表节点 v_j 加入社团结构 S^l 中所有社团后的邻居连接度集合. 若节点的社团连接度大于某门限时，那该节点可被认为与该社团具有足够的连接关系. 该门限决定了节点在该社团的去留操作，可以从社团连接度序列中计算得到. 为简化计算，本文采用了简单的桶排序方法快速确定阶段门限 ξ_j^{th}.

如图 2 给出了桶选择示例. 对于任意节点 $v_j \in V$，连接度取值的整个范围为 $[0,1]$，被分割为 $\max(20, N(v_j))$ 个相同大小的桶. 每个桶中的分数的初始计数被设置为 0. 当序列 $\langle \xi^i \rangle_j$ 中的取值位于某桶的范围内，那么该桶计数增加. 一旦计数完成，最右边的得分数大于 0 的桶被标记. 以此为起点，在桶列表中一直向左扫描，要么找到桶计数小于或等于标记桶，且该桶的左侧的计数大于或等于当前桶，要么已经达到了最左边的桶. 那么，对于节点 v_j，该桶的下限被选为阶段门限 ξ_j^{th}.

对于所有社团 $S_i \in S^l$，若其外围节点 $v_k \in Aug_i$ 的社团连接度 ξ_j^i 低于其阶段门限 ξ_j^{th}，则该节点将离开社团 S_i. 删除外围节点的操作可以确保社团核心节点不会被删除. 然而，任何小于 k 节点规模的社团会被认为连接度不足而被消除. 连接度的计算以及外围节点的删除在整个社团结构中迭代，直到没有节点离开社团.

算法 2 给出了缩减操作的执行流程. 对每个社团 S_i，其节点 $v \in S_i$，若该节点社团连接度小于阶段门限，则将 v 从该社团中去除. 根据邻居不小于 K 的原则更新社团. 首先，如果任意的 $S_i, S_j \in S$，存在节点 $u_j \in S_i$，$j \neq i$，且 $\psi(S_i, S'_i) > dup$，消除几乎完全重叠的社团 S_i. 其中，$\psi(S_i, S_j)$ 为两社团 S_i 和 S_j 之间的相似度(第 3.3.4 节公式(6)给出了定义)，dup 为相似度门限，判断两个社团是否重叠. 接着，任意的社团 $S_i \in S$，且节点 $v_j \in S_i$，计算社团连接度序列 $\langle \xi^i \rangle_j$ 以及邻居连接度序列 $\langle \zeta^i \rangle_j$；对于任意节点 $v_i \in V$，计算其阶段门限 ξ_i^{th}. 然后，遍历社团 $S_i \in S$，对于任意的节点 $v_k \in Aug_i$，判断其社团连接度

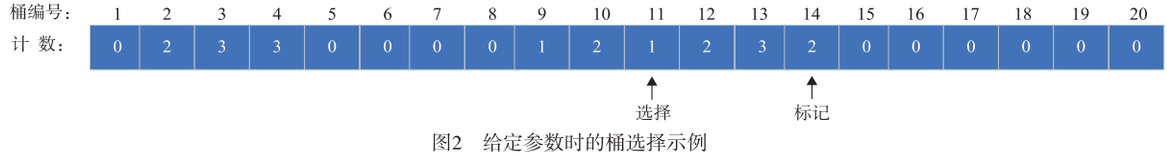

图2 给定参数时的桶选择示例

$\xi_k^i < \xi_k^{th}$,并执行缩减操作,将 v_k 从社团 S_i 删除,标记缩减操作指示变量 *reduce_tag*. 若 *reduce_tag* = 1 表示在下一阶段的迭代中,社团 S_i 可以继续执行缩减操作.

算法 2　函数 2 *reduceCommunities*(S,K,*dup*,*reduce_tag*)

如果 $\forall S_i \in S$, $\exists u_j \in S_i, j \neq i$,且 $\psi(S_i, S_j) > dup$,消除几乎完全重叠的社团 S_i;
$\forall S_i \in S$, $\forall v_j \in S_i$,计算社团连接度序列 $\langle \xi^i \rangle_j$ 以及邻居连接度序列 $\langle \zeta^i \rangle_j$;
$\forall v_i \in V$,计算阶段门限值 ξ_i^{th};
　For each 社团 $S_i \in S$ do
　　$\forall v_k \in Aug_i$,若 $\xi_k^i < \xi_k^{th}$, $S_i = S_i - \{v_k\}$;
　　If S_i 在前一步骤进行了更新 *then*
　　If $|S_i| \leq K$ then
　　$S_i = S - \{S_i\}$;
　　Else
　　reduce_tag←1;
　　End if
　　End if
　End for

3.3.3　扩展操作

缩减操作后,通过吸收邻居节点,可以实现社团的扩展. 所以,在每个社团 S_i 中,当节点满足满足如下两个条件时,才能将相邻节点 v_j 添加到外围节点集 Aug_i 中:

(1)该节点尚未被包含到 Aug_i;

(2)该节点对加入这个社团有很高的兴趣.

对于加入新社团的意愿程度可以由邻居连接度 ζ_j^i 度量. 当 $\zeta_j^i > \zeta_j^{th}$ 时,表明该节点具有较高的邻居连接度,其中 ζ_j^{th} 称为成员门限,用于衡量节点 v_j 是否为社团的成员. 成员门限 ζ_j^{th} 从邻居连接度序列 $\langle \zeta^i \rangle_j$ 中计算得到,方法与阶段门限的计算方法类似.

Aug_i 被设置为 S_i 中的新加入节点(若没有新加入节点,则为空集),用于下一阶段的缩减或扩展操作. 一方面,扩展外围节点可以重新吸收之前已删除的节点,另一方面,那些不满足该社团要求的节点将不会被重复添加,并在后续的缩减操作中被删除. 这也是本文算法收敛的关键.

扩展操作之后,将该阶段获取的社团结构作为输入,传递给下一阶段的缩减操作. 在每一阶段,一些特定社团的所有外围节点在缩减阶段将被删除. 本文算法的停止条件为:在当前阶段中,对于所有社团,没有剩余的外围节点.

算法 3 给出了扩展操作的执行过程. 对任意社团 S_i,节点 $v \in Aug_i$ 为社团 S_i 的成员,若其邻接点 $u \in N(v)$ 加入社团 S_i 后,邻居连接度大于其"成员门限",则将 u 扩展到社团 S_i 中. 首先初始化社团 S_i 的当前外围节点集为 $curAug_i = \varnothing$,然后,对于 $\forall v_j \in Aug_i$,遍历 v_j 节点的邻接点 $u_k \in N(v_j)$,若 $\zeta_k^i > \zeta_k^{th}$ 且 $u_k \notin S_i$,则将 u_k 扩展到社团 S_i,并更新 $curAug_i$. 然后,更新外围节点集 Aug_i 为 $curAug_i$,并判断外围节点集合是否空,若不空则标记社团 S_i 可扩展社团,在下一阶段可继续执行扩展操作.

算法 3　函数 3 *addCommunities*(S,*add_tag*)

$\forall v_j \in S_i$,计算"成员门限":ζ_j^{th};
For each 社团 $S_i \in S$ do
　$curAug_i = \varnothing$;
　For each $u_k \in N(v_j)$, $\forall v_j \in Aug_i$ do
　　If $\zeta_k^i > \zeta_k^{th}$ 且 $u_k \notin S_i$ then
　　$S_i = S_i \cup \{u_k\}$;
　　$curAug_i = curAug_i \cup \{u_k\}$;
　　End if
　End for
　$Aug_i = curAug_i$
　If $|Aug_i| \geq 1$ then
　　add_tag←1;
　End if
End for

3.3.4　去重操作

重叠社团挖掘算法允许网络拓扑中的每个节点同时属于多个社团. 因此,每次操作结束后,一些社团无法避免地近乎完全重叠. 这种"几乎完全"重叠的社团对 (C, C') 采用如(6)式的相似性度量识别.

$$\psi(C, C') = \frac{|C \cap C'|}{\min(|C|, |C'|)} \tag{6}$$

去重操作在每个阶段都会执行,执行时机一般在该阶段社团的缩减操作之前. 本文算法的去重操作以参数——社团相似度门限 *dup* 为输入. *dup* 为两个社团之间最大重叠范围的上限. 若两个社团的重叠区域超过该上限,则判断两者为一个社团. 一般设置 0.5 ≤*dup* ≤1,本文采用 *dup* = 0.6,同时也评估了不同 *dup* 时的试验结果. 图 3 给出了 *dup* 值为 0.5 ~ 0.7 时,采用 Amazon 网络数据试验时,社团大小的变化.

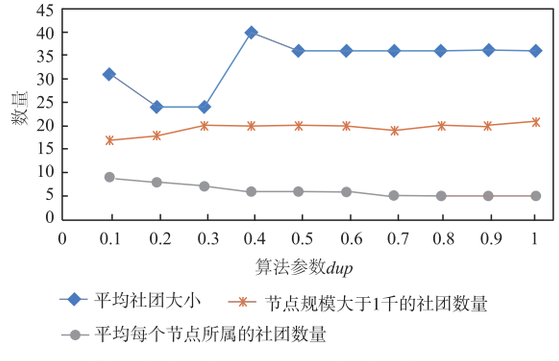

图3　随着参数*dup*变化时相关统计量的变化

3.3.5 算法流程

综上,算法 Li-FOCD 起始于节点组成的初始社团结构,并由它们的邻居以及邻近社团驱动迭代. 一个节点吸收其邻近节点成为其所在社团的一部分. 若被吸收节点在该社团中具有足够的连接性,则该节点可以选择留在该社团,也就是扩展操作,否则可以选择离开,也就是缩减操作. 然后,社团会随着迭代而不断膨胀和缩减,得到最终的社团结构. 给定静态网络拓扑 $G(V,E)$,Li-FOCD 算法描述如算法 4 所示,主要给出了 reduceCommunities 和 addCommunities 两个函数的迭代调用. 由于主函数的逻辑较简单,不再详细描述.

算法 4 基于局部信息的快速重叠社团挖掘算法(Li-FOCD)

输入:$G(V,E)$ 为输入拓扑;K 为社团内节点的最小连接数;dup 为判定两个社团之间重叠的相似度门限;
输出:$S=\{S_i \mid S_i \subset V$ 且 S_i 是一个社团$\}$
其它变量:$n=|V|$ 为拓扑节点数量,$N(v)$ 为节点 v 的邻居集合;Aug_i 为最后一次循环加入社团 S_i 的节点(作为外围节点);
主函数:*overlayCommunityDetection*(G,K,dup)

```
S = ∅;
Initialization(G,K,S);
add_tag←1;
While add_tag do
  reduce_tag←1;
  While reduce_tag do
    reduce_tag←0;
     reduceCommunities(S,K,dup,reduce_tag);
  End while
  add_tag←0;
  addCommunities(S,add_tag);
End while
Return S;
```

4 实验验证

4.1 时间复杂度分析

给定网络拓扑 $G(V,E)$,$n=|V|$ 表示节点数量,$m=|E|$ 表示边数量. 初始化时,每次扫描需要的时间为 $O(m+n)$. 然后,迭代地执行扩展操作和缩减操作. 设 l 为每个节点归属社团的平均数量. 那么,那么缩减操作的时间复杂度为 $O(nl^2+n+l)$. 对于扩展操作,社团 S_i 的每个外围节点,其邻接表将被遍历以确认邻接节点未包含在社团 S_i 中,需要时间 $O(nl^2)$;另外,对于每个邻接节点,计算邻居连接度需要遍历比较该节点的邻接表与社团 S_i,需要 $O(nl^3)$;去重操作需要将社团 S_i 与外围节点的其它社团进行比较,设社团数量为 n,每个社团具有 l 个外围节点,每个外围节点具有 l 个社团需要比较,则需要时间 $O(l)$;则扩展操作的时间为 $O(nl^3)$.

综上,Li-FOCD 算法的复杂度为 $O(m+nl^3)$,由于参数 l 为常数,因此,本文实际的算法复杂度为 $O(m+n)$.

4.2 实验分析

典型的仿真数据是 LFR 基准图[17]. LFR 基准图是人工仿真生成的网络拓扑,且可以模拟重叠社团. 但是,LFR 为不同的簇分配了相同数量的邻居,导致不同社团中的节点的邻居节点数量的总和服从节点度分布. 而这与实际网络不相符. 为此,本节采用真实网络图数据.

本节所有仿真试验机器配置为: ThinkCentre M8600T 系列,Intel i7-6700/8G. 为测试 Li-FOCD 算法的性能,本节使用了文献[34]中的 5 种不同的大规模真实网络数据,分别为 Amazon, DBLP, YouTube, LiveJournal, Orkut.

所有网络数据都被处理为无向、无权图. 各网络数据的具体参数如表 3 所示,参数包括网络节点数($|V|$)(单位:百万个)、边数($|E|$)(单位:百万条)、平均节点度(k)、实际社区数量(n_c)、平均社区大小(n_s)、每个节点所属的平均社区数(重叠)(n_o)、属于至少 1 个社团的节点的比例 $f_{\geq 1}$、属于超过 1 个社团的节点的比例 $f_{\geq 2}$.

表 4 给出了不同算法在五种真实网络数据中生成的社团数量以及执行时间. 其中"/"代表算法未在规定的时间范围内运行出结果(设定了 5 小时的限制). Li-FOCD 与 6 种典型的重叠社团挖掘算法,贪婪式派系膨胀算法(GCE)[7]、MOSES[14]、OSLOM[35]、COPRA[18]、SLPA[25]和 BigClam[16]. 其中 GCE 的最小簇大小为 3,其它参数默认;SLPA 和 COPRA 忽略规模不大于 2 的社团. COPRA 算法还需要设置每个节点可以参加的社团的最大数量 x. 算法 SLPA 主要依赖概率门限参数 r,本节设置为[0.01,0.5],输出社团的数量设定为接近真实社团的数量.

可以看出:GCE 和 OSLOM 没有在 YouTube 数据上生成结果. 随着网络规模的不断扩大,其它算法也均存在不能运行出结果的现象. 而本文提出的算法 Li-FOCD 则在所有的网络拓扑数据上生成了结果,尤其是两个较大规模数据集 LiveJournal 和 Orkut. 由于内存限制,COPRA 和 SLPA 未在数据集 LiveJournal 和 Orkut 运行出结果. 总之,从执行时间上,Li-FOCD 算法具有显著优势. 与此同时,Li-FOCD 算法具有较高的社团挖掘质量. 模块度是典型的描述社团挖掘质量的参数,趋于生成较大规模的社团,而且,对于不同的网络,模块度遵循相同的模式,无法检测具有多样化社团结构的真实网络数据. 本文采用算法检测出的社团与网络数据的真实(ground-truth)社团之间的标准互信息(normalized mutual information, NMI)进行性能评估,NMI 的计算参考了文献[6].

表 3　网络主要特征参数

网络数据	$\|V\|$(单位:百万)	$\|E\|$(单位:百万)	k	n_c	n_s	n_o	$f_{\geq 1}$	$f_{\geq 2}$
Amazon	0. 33	0. 9	5. 50	49,732	99. 86	8. 70	0. 91	0. 89
DBLP	0. 42	1. 34	6. 60	2,547	429. 79	2. 22	0. 83	0. 35
YouTube	1. 1	3	5. 27	8431	13. 5	0. 1	0. 05	0. 02
LiveJournal	4. 0	34. 9	17. 35	311,782	40. 06	1. 6	0. 29	0. 19
Orkut	3. 0	117. 2	76. 3	8,455,253	34. 86	29	0. 75	0. 70

表 4　各类算法与 Li-FOCD 算法执行时间比较

网络	社团数量(单位:K):执行时间(单位:秒)						
	MOSES	GCE	OSLOM	COPRA	SLPA	BigClam	Li-FOCD
Amazon	30:150	26:10	18. 9:700	8. 5:1200	30. 5:473	152:4320	21. 1:2
DBLP	47. 1:280	22. 6:21	22. 3:1260	15. 6:190	22. 2:590	40:1980	25. 2:2
YouTube	8:6840	/	/	12:245	40. 1:6204	8. 1:5040	7. 3:53
LiveJournal	/	/	/	/	/	/	200:320
Orkut	/	/	/	/	/	/	201:2880

表 5 给出了 Li-FOCD 与 6 种经典算法生成的社团数量与真实社团数量之间的标准互信息比较的标准互信息值. 根据表 4 中标准互信息值可知,Li-FOCD 算法在除了 Amazon 网络之外的其它数据上,展示了更优异的社团挖掘质量. 因为 Amazon 网络数据的重叠率较高,约 90% 以上的节点同时参与到 2 个以上的社团中. 因此,Amazon 的标准互信息值精确性要求检测方法设定的重叠社团数量足够大. 当 BigClam 算法的输入参数——社团数量参数的设置与实际一致时,其检测性能良好,但是时间复杂度较高. 最后,本节统计了 7 种算法随着网络边数量增加,其运行时间曲线趋势,如图 4 所示. 曲线展示了在 5 种数据集的运行时间统计,其中 Li-FOCD 采用了 Amazon、YouTube、DBLP、LiveJournal 和 Orkut 五种数据. 其它算法使用了数据集 Amazon, DBLP 和 YouTube. 可以看出 Li-FOCD 算法随着网络规模的增大,其算法具有显著的时间可扩展性.

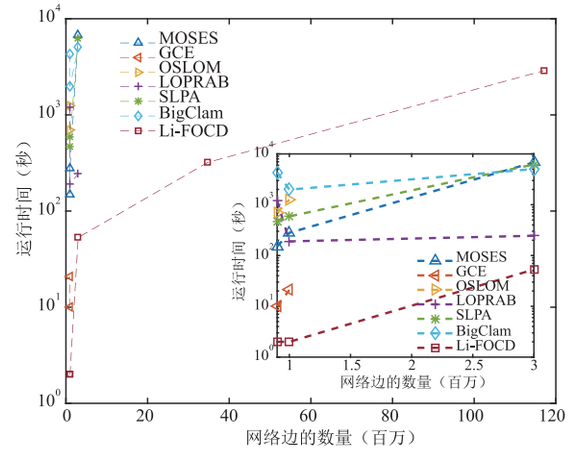

图4　随着网络边数量的增加Li-FOCD算法与其它算法的运行时间比较

表 5　各类算法生成的社团数量与真实社团数量之间的标准互信息比较

网络	标准互信息(NMI)						
	MOSES	GCE	OSLOM	COPRA	SLPA	BigClam	Li-FOCD
Amazon	0. 22	0. 21	0. 18	0. 20	0. 12	0. 25	0. 21
DBLP	0. 15	0. 13	0. 12	0. 15	0. 11	0. 15	0. 21
YouTube	0. 01	/	/	0. 01	0. 002	0. 01	0. 02
LiveJournal	/	/	/	/	/	/	0. 03
Orkut	/	/	/	/	/	/	0. 06

5 结论

社交网络结构复杂、数据规模大,Li-FOCD 算法通过选择局部连通性良好的节点快速检测重叠社团. 为描述节点与社团的关系或状态,本文为每个网络节点引入了社团连接度和邻居连接度. 这种局部属性的计算有利于算法根据网络拓扑规模扩展. 通过设定社团相似度门限、节点的邻居节点的最小数量,Li-FOCD 算法可以有效确定节点与任意社团的隶属关系. 采用真实数据的试验表明本文算法的有效性. 本文算法前提假设其发现的社团数量的最大值为网络拓扑节点的数量. 而在重叠社团存在的情况下,社团的实际数量可以大于网络拓扑节点规模,我们将在后续研究中改进.

参考文献

[1] DASGUPTA K, SINGH R, VISWANATHAN B, et al. Social ties and their relevance to churn in mobile telecom networks[A]. Proceedings of 11th International Conference on Extending Database Technology: Advances in Database Technology[C]. New York: ACM, 2008. 668 – 677.

[2] CAI Q, MA L, GONG M, et al. A survey on network community detection based on evolutionary computation[J]. International Journal of Bio-Inspired Computation, 2016, 8(2): 84 – 98.

[3] SCHAEFFER S E. Graph clustering[J]. Computer Science Review, 2007, 1(1): 27 – 64.

[4] 邓小龙,翟佳羽,尹栾玉. 基于矢量影响力聚类系数的高效有向网络社团划分算法[J]. 电子与信息学报,2017,39(9):2071 – 2080.
Deng Xiaolong, Zhai Jiayu, Yin Luanyu. Vector influence clustering coefficient based efficient directed community detection algorithm[J]. Journal of Electronics & Information Technology, 2017, 39(9): 2071 – 2080. (in Chinese)

[5] BLONDEL V D, GUILLAUME J L, LAMBIOTTE R, et al. Fast unfolding of communities in large networks[J]. Journal of Statistical Mechanics: Theory and Experiment, 2008, 2008(10): 10008.

[6] LANCICHINETTI A, FORTUNATO S and KERTESZ J. Detecting the overlapping and hierarchical community structure in complex networks[J]. New Journal of Physics, 2009, 11(3): 033015.

[7] LEE C, REID F, MCDAID A, et al. Detecting highly overlapping community structure by greedy clique expansion [J]. ArXiv, 2010, 1002: 1827.

[8] MISHRA N, SCHREIBER R, STANTON I, et al. Finding strongly knit clusters in social networks[J]. Internet Mathematics, 2008, 5(1 – 2): 155 – 174.

[9] YANG J, LESKOVEC J. Structure and overlaps of communities in networks[J]. ArXiv, 2012, 1205: 6228.

[10] 乔少杰,韩楠,张凯峰,等. 复杂网络大数据中重叠社区检测算法[J]. 软件学报,2017,28(3):631 – 647.
QIAO Shaojie, HAN Nan, ZHANG Kaifeng, et al. Algorithm for detecting overlapping communities from complex network big data[J]. Journal of Software, 2017, 28(3): 631 – 647. (in Chinese)

[11] CHAKRABORTY T, KUMAR S, GANGULY N, et al. GenPerm: a unified method for detecting non-overlapping and overlapping communities[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(8): 2101 – 2114.

[12] HUANG Faliang, LI Xuelong, et al. Overlapping community detection for multimedia social networks[J]. IEEE Transactions on Multimedia, 2017, 19(8): 1881 – 1893.

[13] AHN Y Y, BAGROW J P, LEHMANN S. Link communities reveal multiscale complexity in networks[J]. Nature, 2010, 466(7307): 761 – 764.

[14] MCDAID A, HURLEY N. Detecting highly overlapping communities with model-based overlapping seed expansion[A]. Proceedings of International Conference on Advances in Social Networks Analysis and Mining[C]. Piscataway, NJ: IEEE, 2010. 112 – 119.

[15] LIU Xiao, WEI Yiming, WANG Jian, et al. Community detection enhancement using no-negative matrix factorization with graph regularization[J]. International Journal of Modern Physics B, 2016, 30(20): 1650130.

[16] Yang J, LESKOVEC J. Overlapping community detection at scale: a nonnegative matrix factorization approach[A]. Proceedings of the Sixth ACM International Conference on Web Search and Data Mining[C]. New York: ACM, 2013. 587 – 596.

[17] NARAYANAM R, NARAHARI Y. A game theory inspired, decentralized, local information based algorithm for community detection in social graphs[A]. Proceedings of 21st International Conference on Pattern Recognition[C]. Piscataway, NJ: IEEE, 2012, 1072 – 1075.

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

[19] 刘功申,孟魁,郭弘毅,苏波,李建华. 基于贡献函数的重叠社区划分算法[J]. 电子与信息学报,2017,39(8):1964 – 1971.
LIU Gongshen, MENG Kui, GUO Hongyi, SU Bo, LI Jianhua. Overlapping-communities recognition algorithm based on contribution function[J]. Journal of Electronics & Information Technology, 2017, 39(8): 1964 – 1971. (in Chinese)

[20] CHEN W, LIU Z, SUN X, et al. A game-theoretic frame-

work to identify overlapping communities in social networks[J]. Data Mining and Knowledge Discovery,2010,21(2):224 – 240.

[21] ZHANG Lei, PAN Hebin, et al. A mixed representation-based multi-objective evolutionary algorithm for overlapping community detection[J]. IEEE Transactions on Cybernetics,2017,47(9):2703 – 2716.

[22] WEN X, CHEN W N, LIN Y, et al. A maximal clique based multi-objective evolutionary algorithm for overlapping community detection[J]. IEEE Transactions on Evolutionary Computation,2017,21(3):363 – 377.

[23] BAUMES J, GOLDBERG M, MAGDOM-ISMAIL M. Efficient identification of overlapping communities[A]. Proceedings of International Conference on Intelligence and Security Informatics[C]. Heidelberg, Berlin: Springer,2005. 27 – 36.

[24] XIE J and SZYMANSKI B K. Community detection using a neighborhood strength driven label propagation algorithm[A]. Proceedings of Network Science Workshop[C]. Piscataway, NJ: IEEE,2011. 188 – 195.

[25] XIE J, SZYMANSKI B K. Towards linear time overlapping community detection in social networks[A]. Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining[C]. Heidelberg, Berlin: Springer,2012. 25 – 36.

[26] 刘世超,朱福喜,甘琳. 基于标签传播概率的重叠社区发现算法[J]. 计算机学报,2016,29(4):717 – 729.
LIU Shichao, ZHU Fuxi, GAN Lin. A label-propagation-probability-based algorithm for overlapping community detection[J]. Chinese Journal of Computers, 2016, 29(4):717 – 729. (in Chinese)

[27] RAGHAVAN U N, ALBERT R and KUMARA S. Near linear time algorithm to detect community structures in large-scale networks[J]. Physical review E,2007,76(3):036106.

[28] 马慧芳,谢蒙,何廷年,等. 基于核心标签的可重叠微博网络社区划分方法[J]. 电子学报,2017,45(4):769 – 776.
MA Xiefang, XIE Meng, HE Tingnian, et al. An overlapping microblog community detection algorithm via core tags[J]. Acta Electronica Sinica,2017,45(4):769 – 776. (in Chinese)

[29] COSCIA M, ROSSETTI G, GIANNOTTI F, et al. Demon: a local-first discovery method for overlapping communities[A]. Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. New York: ACM,2012. 615 – 623.

[30] YANG B, LIU J, LIU D. An autonomy-oriented computing approach to community mining in distributed and dynamic networks[J]. Autonomous Agents and Multi-Agent Systems,2010,20(2):123 – 157.

[31] BU Z, WU Z, CAO J, et al. Local community mining on distributed and dynamic networks from a multiagent perspective[J]. IEEE Transactions on cybernetics, 2016, 46(4):986 – 999.

[32] BU Z, CAO J, LI H J, et al. GLEAM: a graph clustering framework based on potential game optimization for large-scale social networks[J]. Knowledge and Information Systems,2017:1 – 30.

[33] 国琳,左万利,彭涛. 基于隶属度的社会化网络重叠社区发现及动态集群演化分析[J]. 电子学报,2016,44(3):587 – 594.
GUO L, ZUO W L, PENG T. Overlapping community detection and dynamic group evolution analysis based on the degree of membership in social network[J]. Acta Electronica Sinica,2016,44(3):587 – 594. (in Chinese)

[34] YANG J, LESKOVEC J. Defining and evaluating network communities based on ground-truth[J]. Knowledge and Information Systems,2015,42(1):181 – 213.

[35] LANCICHINETTI A, RADICCHI F, RAMASCO J J, et al. Finding statistically significant communities in networks[J]. PloS One,2011,6(4):e18961.

作者简介

李政廉　男,1988 年生于河北三河. 国家数字交换系统工程技术研究中心博士生. 研究方向为数据可视化、复杂网络分析.
E-mail:lizhenglian@ yeah. net

吉立新　男,1969 年生于江苏淮安. 国家数字交换系统工程技术研究中心副总工程师,研究员. 研究方向为电信网安全、数据挖掘.