一种面向大规模社会信息网络的多层社区发现算法

康 颖[1),2)] 古晓艳[1)] 于 博[1)] 林 政[1)] 王伟平[1)] 孟 丹[1)]

[1)](中国科学院信息工程研究所 北京 100093)

[2)](中国科学院大学 北京 100049)

摘 要 社区发现旨在挖掘社会信息网络的社区结构,是社会计算及其相关研究的基础.随着交互式社会信息网络规模的快速增长,传统的社区发现算法难以满足大规模网络的可扩展分析需求.多层社区发现算法如 PMetis、Graclus 等虽然可以分析包含数百万节点规模的网络,但是小于 2 的粗化缩减比率以及社会信息网络的幂律分布特性极大地制约着该类算法的性能优势.该文提出了一种基于三角形内点同一社区性粗化策略的多层社区发现算法 TMLCD.TMLCD 不仅以大于 2 的粗化缩减比率加快了大规模社会信息网络的粗化过程,而且从基本拓扑结构上保持了初始网络的社区效应,提高了社区发现精度.基于 YouTube、Orkut 等真实网络的实验结果表明:TMLCD 在计算精度、内存占用以及运行时间方面的性能均优于目前典型的多层社区发现算法,适用于富含三角形的社会信息网络分析.

关键词 社会信息网络;社区发现;三角形;多层模式;粗化;大数据

中图法分类号 TP393 **DOI 号** 10.11897/SP.J.1016.2016.00169

A Multilevel Community Detection Algorithm for Large-Scale Social Information Networks

KANG Ying[1),2)] GU Xiao-Yan[1)] YU Bo[1)] LIN Zheng[1)] WANG Wei-Ping[1)] MENG Dan[1)]

[1)](*Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093)

[2)](*University of Chinese Academy of Sciences, Beijing* 100049)

Abstract Community detection aims at mining the community structures of Social Information Networks (SINs), which is the foundation of other related researches on social computing. Due to the rapid inflation of interactive SINs, traditional community detection algorithms encounter obstacles in analyzing large-scale networks with scalability. Although multilevel community detection algorithms such as PMetis, Graclus etc. have the capability to analyze networks containing millions of nodes, the coarsening shrink rate is less than 2 and the SINs follow the power-law distribution, which constrain these algorithms' performance enormously. This paper proposes a multilevel community detection algorithm TMLCD, based on the coarsening policy of triangle's inner nodes belonging to the same community. TMLCD accelerates the coarsening process of large-scale SINs at a coarsening shrink rate of greater than 2, and preserves the community effect of initial network from the point of basic topological structure and improves the accuracy of

收稿日期:2015-01-18;在线出版日期:2015-06-15.本课题得到国家科技支撑计划项目(2012BAH46B03)、国家核高基项目(2013ZX01039-002-001-001)、中国科学院先导专项(XDA06030200)、国家"八六三"高技术研究发展计划项目基金(2012AA01A401)和国家自然科学基金(61402473)资助.**康 颖**,女,1984 年生,博士研究生,主要研究方向为数据挖掘、社区发现等.E-mail:kangying@iie.ac.cn.**古晓艳**(通信作者),女,1987 年生,博士,助理研究员,主要研究方向为并行计算、分布式查询优化等.E-mail:guxiaoyan@iie.ac.cn.**于 博**,男,1985 年生,博士,助理研究员,主要研究方向为无线网络、无线传感器网络、分布式计算、查询处理等.**林 政**,女,1984 年生,博士,助理研究员,主要研究方向为自然语言处理、情感分析等.**王伟平**,男,1975 年生,博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为数据流、高性能数据库、并行处理等.**孟 丹**,男,1965 年生,博士,教授,博士生导师,主要研究领域为高性能计算机体系结构、分布式文件系统、系统安全等.

community detection. Experimental results from the real-world networks like YouTube, Orkut etc. indicate that, TMLCD outperforms the currently typical multilevel community detection algorithms in terms of computing precision, memory occupation and running time. It is obvious that TMLCD is appropriate for analyzing the SINs rich in triangles.

Keywords social information networks; community detection; triangle; multilevel paradigm; coarsening; big data

1 引 言

随着 Web2.0 概念的出现和不断深入,大量用户交互式社会媒体如微博、论坛、社交网络、社会新闻、维基等不断涌现.基于这些共享社会媒体平台产生的新型信息网络称为社会信息网络.社会信息网络具有强社区效应,社区发现是分析社会信息网络社区结构的一种有效方法,其广泛的应用场景引起了社会、经济、信息安全等多个学科领域的普遍关注.深入挖掘社会信息网络社区可为圈内好友推荐、个性化信息导向、未来经济走势预测、网络舆情监控等提供技术支持.

社区发现本质上是聚类在社会网络研究应用上的延伸.从定性的角度讲,社区发现就是挖掘一个网络中内部连接紧密而外部连接稀疏的群组(子团或簇).传统的社区发现算法有谱分析方法[1]、分层凝聚法[2]、Girvan-Newman 分裂法[3]、基于模块度(Modularity)的方法(该方法存在严重的分辨率受限问题)[4]以及结合其他学科理论如统计学[5]、遗传学[6]、信息论[7]等衍生的社区发现算法.这些算法虽然不断改进优化以适应新的数据计算需求,但 $\Theta(n^2)$ 或更高的计算复杂度制约着其分析大规模网络的能力.

为了能够有效挖掘大规模网络的社区结构,近年来,研究人员提出了大量可扩展的社区发现算法.如 Charikar 等人[8]在内存有限的情况下,通过流模式扫描大规模网络一次或几次得到其近似的网络拓扑结构,从而加快了社区发现过程.Satuluri 等人[9]采用随机抽样去边稀疏化的方法降低了网络规模,并在一定误差范围内识别出大规模网络的社区结构.上述两种算法虽然在一定程度上提高了大规模网络社区发现的效率,但随机抽样近似有时会给实际问题的求解带来较大误差,因此计算结果存在一定的不确定性,即社区发现精度不高.Li 等人[10-11]通过定量分析 Potts 模型的动力学特征来揭示网络中隐藏的社区结构属性,即通过马尔可夫过程来模拟基于 Potts 模型的社区发现过程,并采用马尔可夫过程的谱分析特征向量来推测对应多变量自旋组态的关键拓扑结构信息.该方法拥有扎实的数学理论推导,不仅计算精度高且完全解决了社区发现分辨率受限的问题,但是较高的计算时间复杂度使之难以分析包含百万节点以上的大规模社会信息网络.Yang 等人[12]基于亚稳态的两个基本属性"局部一致性和暂态不变性",提出了一种可以扩展分析大规模网络社区结构的有效社区发现算法(LM 算法),该算法同样采用网络的谱分析特征向量来推测网络中潜在的社区结构拓扑性.由于仅通过一个特征向量的二分迭代实现社区发现,LM 算法的计算速度很快,且易于扩展,但是二分迭代难以满足多分社区结构同时分析的应用需求,特别是在大规模社会信息网络的多分社区发现问题中,具有一定的适用范围局限性.

在大规模网络社区结构的分析研究中,多层社区发现是目前应用广泛且性能较好的一种方法,如 Karypis 等人[13-14]提出的 KMetis、PMetis 以及 Dhillon 等人[15]提出的 Graclus 等.该类方法并不直接挖掘大规模网络的社区结构,而是先将网络规模通过迭代粗化的方法逐层缩减,然后分析粗化后生成的小规模网络的社区结构,再通过反粗化逆推出原大规模网络的社区结构.已有的多层社区发现算法进行粗化时均采用一种基于极大匹配选边的粗化策略[14],即通过一定方法尽可能多的从网络中选取出可被粗化的边,然后将被选边上的两个端点融合形成一个复合节点来实现粗化.但是,这种粗化策略存在以下 3 个方面的问题:(1)极大匹配选边过程是一个 NP 难问题,且由于所选边集中任意两条边均不可以共点,因此被选作粗化边的数目最多为原网络边数的一半,严重制约着该类算法的粗化缩减比率[13](Coarsening Shrunk Ratio 为 $|G_i|/|G_{i+1}|$,其中 $|G_i|$ 和 $|G_{i+1}|$ 分别代表粗化前、后网络中的边数;粗化缩减比率越高,粗化过程越快,存储反粗化

所需的中间过渡图信息量越少.目前已有算法的粗化缩减比率实际值均小于 2),进而影响其计算时间复杂度和存储空间占用率;(2) 被选作粗化边上的两个端点,其自然社区归属性不确定,但因形成复合节点而被纳入同一个社区,这一结果将直接影响初始社区发现和最终社区发现的精度,因此反粗化阶段需要调用大量的调优操作来调整和提高算法整体的计算精度;(3) 社会信息网络的节点度一般服从幂律分布(power-law)[16],度高的节点会以其中心占优势屏蔽掉很多可被选作粗化的边,致使其粗化过程过早结束,加剧了算法的时间和空间复杂度,进一步限制了其分析大规模网络的性能优势.

针对上述问题,本文提出一种基于三角形的多层社区发现算法 TMLCD(Triangle-Based Multilevel Community Detection).该算法在粗化阶段采用一种新的粗化策略——基于三角形内点同一社区性的粗化策略,即将网络中遍历到的三角形 3 个顶点收缩融合形成一个复合节点来实现粗化.三角形因其 3 个顶点完全互连而具有强社区性[17],所以选择三角形作为粗化对象不仅可以确保被融合粗化的节点同属于一个社区,使得经逐层迭代粗化后生成的小规模网络可以保持原始大规模网络的基本社区结构,提高初始社区发现精度,而且可以节省大量反粗化调优工作并以高精确度反推出原始大规模网络的社区结构.实验结果表明,在分析真实的大规模社会信息网络时,TMLCD 可以大于 2 的粗化缩减比率加快网络的粗化过程,降低反粗化所需存储的中间过渡图信息量,同时能够以高精确度发现网络中隐含的社区结构,提升多层社区发现算法的性能,且适用于服从幂律分布的网络分析.

2　基于三角形的多层社区发现算法

多层社区发现算法的设计理念源于网络社区结构在不同划分度要求下呈现出的层次嵌套模式[18],是一种可高效分析大规模网络且扩展性能很好的方法.据 Abou-Rjeili 等人[16] 分析,即使在最坏情况下,通过多层社区发现算法间接分析得到的结果均优于采用某一算法直接挖掘大规模网络社区结构的结果.多层社区发现的一般计算过程分为粗化、初始社区发现、反粗化并调优 3 个阶段[14],如图 1 所示.本文秉承多层分析模型的计算特点并融入基于三角形内点同一社区性的粗化策略,提出一种可扩展分析大规模社会信息网络的多层社区发现算法——基于三角形的多层社区发现算法 TMLCD(Triangle-Based Multilevel Community Detection),下面将具体阐述 TMLCD.

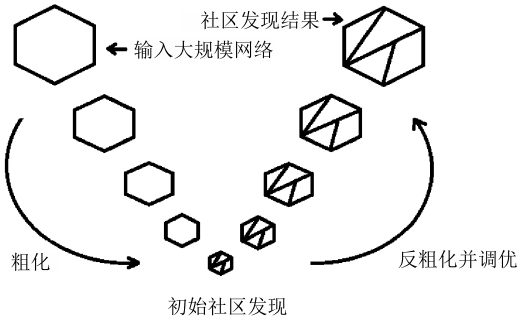


图 1　多层社区发现的计算过程

2.1　设计基础

三角形是组成复杂网络的基本拓扑结构之一,是复杂网络应用研究分析的核心[19-21],如典型的"聚类系数"计算等[22].在大规模社会信息网络中,表示用户的个体或组织节点常常会因共同的社会属性、兴趣爱好、价值取向、行为特征等形成具有强关联特性的三角形拓扑结构.而社会信息网络是人类社会活动在网络空间的虚拟映射,其能够以更短的时空距离增强人们之间的互动交流,映射出更加强烈的社会关系,且这种频繁交互易于引发两个关联节点的相邻节点集交集的迅速膨胀.另外,基于斯坦福大学 SNAP 网络的社会信息网络数据[23]的调研,绝大多数(至少 85% 以上)社会信息网络富含三角形拓扑结构(网络中三角形拓扑结构的数目至少与边的数目相当).因此,三角形以其固有的强社区效应(三角形内点完全互联,是最简单的完全子图),成为网络社区分层嵌套模式中的基本社区单元,是嵌套树中的叶子社区或者叶子社区的有机组成部分.并且,相对于高阶完全子图,大规模社会信息网络包含的三角形更为丰富,这为 TMLCD 粗化过程提供了必要的粗化源.

另一方面,随着粗化层级的不断深入,网络中的高阶多边形会逐渐降阶,进而演变成三角形,为 TMLCD 的逐层迭代粗化过程提供可持续的粗化源.例如图 2 中所示:粗化前社区 1 内的节点 6、7 与节点 1、2、3 或者节点 1、3、4 构成一个五边形,当由节点 1、2、4 构成的三角形被融合粗化形成一个复合节点时,上述五边形将降阶为下一层粗化图上的四边形.即如图 2 中粗化后的社区 1 内由节点 3、4、6、7 所构成的四边形(图中仅给出粗化局部效果图,未展示全图粗化结果),原因是五边形与被粗化三角形

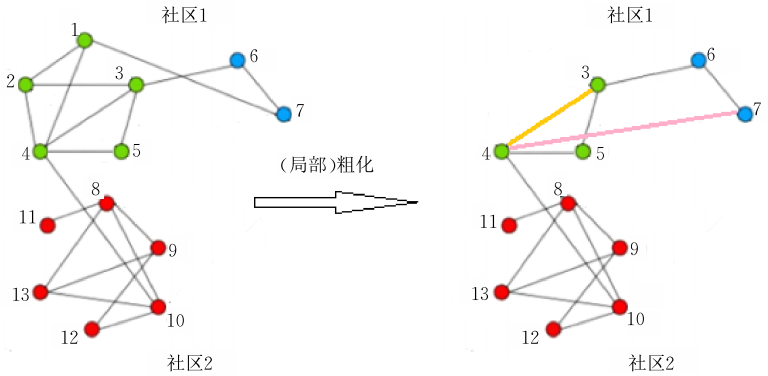


图 2 高阶多边形的降阶过程

共边;同理,粗化前社区 2 内由节点 9、10、12、13 构成的四边形也会随着由节点 8、9、13 构成的三角形被粗化而降阶为下一层粗化图上的三角形. 如此粗化迭代,高阶多边形逐步向低阶多边形演化,甚至转变成三角形,为后续粗化过程提供粗化对象,从而保证了 TMLCD 逐层迭代粗化的可持续性.

2.2 算法实现

本文先将基于链接结构的大规模社会信息网络表示成图 $G_0=(V_0,E_0)$,其中 V_0、E_0 分别表示节点集和边集,节点 $v\in V_0$ 表示个人、组织等信息,边 $e\in E_0$ 表示节点之间某种无向链接关系,如好友关系、合作关系等,边权重 ω_0 取单元值. 社区就是图上内连边数大于外连边数的局部强关联子图.

TMLCD 算法的具体实现过程如下.

2.2.1 粗化阶段

遍历大规模社会信息网络图 G_0 上的三角形,将每次遍历遇到的三角形 3 个顶点融合粗化形成 1 个复合节点 v_c,相应连接边参照复合节点作归一化处理,且累加求其边权重之和得到 ω_c,v_c 和 ω_c 将成为下一层粗化图的基本组成元素. 当图 G_0 遍历完全生成第 1 层粗化图 G_1 时,基于图 G_1 继续上述三角形遍历粗化过程生成第 2 层粗化图 G_2,然后再基于图 G_2 继续三角形遍历粗化过程生成下一层粗化图 G_3,如此循环迭代,生成一系列规模逐渐减小的粗化图 $G_0,G_1,\cdots,G_m$,并且满足节点数逐层递减 $|V_0|>|V_1|>\cdots>|V_m|$. 当图 G_m 的规模小于一定阈值 T(该阈值 T 一般为设定内存容量大小)或规模不再缩减时,粗化迭代过程终止,且定义此时生成的粗化图 G_m 为最简粗化图.

TMLCD 算法的多层粗化是基于单层粗化迭代实现的. 本文将该单层粗化过程定义为三角形遍历粗化算法 TTCA(Triangle Traversing Coarsening Algorithm),其具体实现如算法 1 所示.

算法 1. TTCA.

输入:位于磁盘上的图 $G(V,E)$的邻接表 $Adj(G)$

输出:经粗化生成的图 $G'(V',E')$的邻接表 $Adj(G')$

```
1.  Adjs(G)=SortedDegree(Adj(G));
2.  INIT(F(v));
3.  WHILE (Adjs(G)中有未读数据){
4.    buffer=READ(Adjs(G));
5.    FOREACH (node v in buffer) DO
6.      IF (F(v)>1) THEN
7.        CONTINUE;
8.      END IF
9.      N(v)=v的相邻节点集在内存?直接赋值:从磁盘读入内存;
10.     FOREACH (node u in N(v)) DO
11.       IF (d(u)<d(v) or F(u)>1) THEN
12.         CONTINUE;
13.       END IF
14.       N(v)∩N(u)=v和u的相邻节点集交集在内存?直接赋值:从磁盘读入内存;
15.       FOREACH (node w in N(v)∩N(u)) DO
16.         IF(d(w)<d(v) or d(w)<d(u) or F(w)>1) THEN
17.           CONTINUE;
18.         ELSE
19.           融合三角形形成复合节点 v_c,相应连接边聚合并求权重和 ω_c;设置复合节点 F(v_c)+=1,被复合节点 F(v)=2;
20.           BREAK;
21.         END IF
22.       END FOR
23.     END FOR
24.   END FOR }
25. 由复合节点、聚合边和未被粗化的节点、边联合构成粗化图 Adj(G');
```

算法中图 G 以邻接表的形式给出,节点信息包括标号、节点度 $d(v)$ 和标志位 $F(v)$,$N(v)$ 表示节点 v 的相邻节点集. 为了尽可能避免图上三角形遍历的随机性且使算法适合于分析节点度服从幂律分

布的网络，算法开始时先采用函数 *SortedDegree*() 对图 G 上所有节点按照其度值的升序排序. 这里有几点需要说明：(1) 为了避免重复操作，TTCA 先对节点进行排序并按照节点度值从低到高的顺序遍历粗化三角形，而且结合标志位 $F(v)$ 判断“已被粗化过的节点”是否可以进一步参与后续的三角形遍历粗化过程；(2)“已被粗化过的节点”包含两种，一种是被粗化融合后消失了的节点，如算法 1 中的节点 w 和节点 u，该类节点的标志位 F 值均为 2 且在后续操作中一律被忽略. 另一种是被粗化融合后形成的复合节点，如算法 1 中节点 v(亦是复合节点 v_c)，若其标志位 F 值小于 2，可继续参与后续三角形的遍历粗化过程，否则亦会被忽略，直到下一层粗化过程被重新考虑(对于标志位 F 值的设定还有一个重要考虑因素是避免过粗化现象，这一点可参见性能分析 3.3)；(3) 若节点排序过程中出现了度值相同的情形，可随机选择其中任一节点继续三角形的遍历粗化过程. 由于粗化过程本身是一个近似过程，本文所采取的策略是尽可能降低其遍历粗化的随机性(如节点排序以及标志位的辅助设置)，但当两个节点度值相同时，随机选择会存在一定的局部不确定性，但相对于整个网络而言，其基本的社区结构特性不会受到影响，因此选择差异不大. 若想完全排除这种随机性，可以结合模块度 Q(Modularity)，从与具有相同度值的节点相比邻的所有三角形中，选择使模块度 Q(或模块度增益 ΔQ)最大的三角形优先进行粗化融合.

另外，TMLCD 逐层迭代粗化过程中，生成的中间过渡图均需被存储，为后续反粗化还原、调优处理提供必要的信息.

2.2.2　初始社区发现阶段

调用社区发现算法对最简粗化图 G_m 进行初始社区发现. 原则上，任何高效的社区发现算法均可用于初始社区发现，因为图 G_m 包含的数据量已经完全被内存容纳. 但随着迭代粗化的深入，原始大规模社会信息网络图 G_0 中的边权重开始由单元值逐渐累加求和变成了图 G_m 中的数值，因此，边的权重成为决定初始社区发现算法质量的重要因素，选择初始社区发现算法必须考虑边权重的可计算性. 目前较受欢迎的初始社区发现算法是 kernel k-means 算法和谱分析算法，它们最大的特点是不受线性可分条件的限制. 由 Pizzuti[24] 提出的 GA-Net 算法也是一种不局限于分析凸社区结构的算法，且相对于 kernel k-means 和谱分析，GA-Net 还是一种社区自动检测方法，即不需要预先知道或假设社区的数目 k. 因此，本文采用基于遗传算法的社区发现算法 GA-Net 进行初始社区发现. GA-Net 运用遗传学机制缩小最优解目标搜索空间以降低算法的时间复杂度. 具体过程是，首先通过选择交叉算子将解空间锁定在使目标函数值趋优的范围内，避免全图逐点搜索，再结合变异算子将解空间适度变换到全局范围内，避免陷入局部最优解而提早终止计算过程，最终找到全局最优解.

GA-Net 算法定义了一个全局意义上的社区发现目标函数，如式(1)所示：

$$Q_c=\sum_{l=1}^{k}\frac{\sum_{i\in \boldsymbol{I}}(a_{i\boldsymbol{J}})^r}{|\boldsymbol{I}|}\times V_{C_l} \tag{1}$$

其中，Q_c 为社区发现的形式化定义，求解 Q_c 的最大值即是社区发现过程. $C=\{C_1,C_2,\cdots,C_k\}$ 表示图上的 k 个子图社区，C_l 表示其中某一子图社区；若将 $\boldsymbol{C}_l$ 表示为矩阵形式 $\boldsymbol{C}_l=(\boldsymbol{I},\boldsymbol{J})$，其中 $\boldsymbol{I}$ 表示图的邻接矩阵 $\boldsymbol{A}=\{a_{ij}=\{0,1\}\mid i,j=1,\cdots,n\}$ 中对应于社区 $\boldsymbol{C}_l$ 的行向量组成的子矩阵，$\boldsymbol{J}$ 表示图的邻接矩阵 $\boldsymbol{A}$ 中对应于社区 $\boldsymbol{C}_l$ 的列向量组成的子矩阵，则 $V_{\boldsymbol{C}_l}$ 表示 $\boldsymbol{C}_l$ 邻接矩阵中 1 的个数，即

$$V_{\boldsymbol{C}_l}=\sum_{i\in \boldsymbol{I},j\in \boldsymbol{J}}a_{ij} \tag{2}$$

令 $a_{i\boldsymbol{J}}$ 表示 $\boldsymbol{C}_l$ 邻接矩阵中第 i 行的均值，即

$$a_{i\boldsymbol{J}}=\frac{1}{|\boldsymbol{J}|}\sum_{j\in \boldsymbol{J}}a_{ij} \tag{3}$$

综上，即为 GA-Net 算法社区发现目标函数的完整定义. 但此时的 GA-Net 仅限于基于链接结构的图上社区发现，为了拓展其能够分析加权图 G_m，本文将边权重 ω 融入 $a_{i\boldsymbol{J}}$ 的定义式(3)中，得到如下定义式：

$$a^{\omega}_{i\boldsymbol{J}}=\frac{1}{|\boldsymbol{J}|}\sum_{j\in \boldsymbol{J}}\omega_{ij}a_{ij} \tag{4}$$

式中：a_{ij} 取值 0～1，表示节点间是否存在链接关系(同上述邻接矩阵)；ω_{ij} 表示对应边的权重. 经过逐层迭代粗化，边权重 ω_{ij} 由初始的单元值累加求和逐渐转变成为图 G_m 中的数值；ω_{ij} 值越大，表示节点间的关联度越高，其社区同属性越高，因此 ω_{ij} 成为初始社区发现的重要计算因子.

将式(4)重新代入式(1)中，得到 GA-Net 算法的加权社区发现目标函数：

$$Q_c=\sum_{l=1}^{k}\frac{\sum_{i\in \boldsymbol{I}}(a^{\omega}_{i\boldsymbol{J}})^r}{|\boldsymbol{I}|}\times V_{\boldsymbol{C}_l} \tag{5}$$

2.2.3　反粗化并调优阶段

初始社区发现结果并不能直接呈现出原始大规

模社会信息网络的社区结构，需经过逐层反粗化还原出图 G_0 上的社区结构，即将最简粗化图 G_m 上分析得到的社区结果通过中间过渡粗化图 $G_{m-1}\rightarrow G_{m-2}\rightarrow\cdots\rightarrow G_1$ 逐层反粗化还原到图 G_0 上，以得到原始大规模社会信息网络的社区结构. 在图 G_{i+1} 反粗化还原图 G_i 时，通常将复合节点展开的各节点简单地归属到复合节点所属社区内.

对于基于极大匹配选边粗化策略的多层社区发现算法，由于复合节点反粗化还原过程会产生随机自由节点(即社区归属性不确定的节点)，若不做适度调优，社区发现结果误差将在粗化图层间以链式反应逐级递增，进而降低原大规模网络图 G_0 上社区发现结果的精度. 因此，每次反粗化均需采用调优算法如 KL(Kernighan-Lin)、BKL(Boundary Kernighan-Lin)[25] 等重新调整随机自由节点的社区归属性. 而对于本文提出的 TMLCD 算法，由于采用基于三角形内点同一社区性的粗化策略，即从三角形所具有的内在社区性出发，将其 3 个顶点融合实现粗化，从基本拓扑结构上保持了初始网络的社区效应，因此反粗化还原展开的各节点可直接归属到复合节点所属的社区内，不会产生随机自由节点. 但存在一种特殊情况，即被粗化的三角形位于重叠社区部分. 如图 3 所示，由节点 1、2、3 构成的三角形位于重叠社区部分，其因强社区性同时属于社区 1 和社区 2. 如果优先选择该三角形作融合粗化，社区 1 和社区 2 会因过早聚合而形成一个大社区，模糊掉一定划分度要求下的自然社区结构. 因此，与基于极大匹配选边粗化策略的多层算法不同，TMLCD 反粗化调优工作的重点不是重新调整由复合节点展开的节点的社区归属性，而是分离因社区重叠而被聚合的大社区，从而发现不同划分度要求下的自然社区结构.

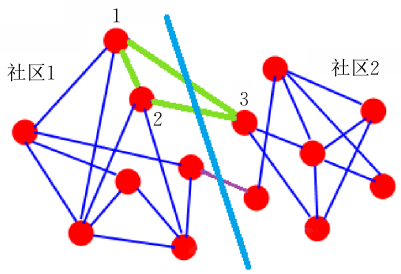


图 3 位于重叠社区部分的三角形

3 算法性能分析

3.1 复杂度分析

基于三角形内点同一社区性的粗化策略使得 TMLCD 算法在反粗化阶段无需进行特别的调优处理，且初始社区发现阶段所需分析的图规模远远小于原始大规模社会信息网络，因此 TMLCD 的复杂度主要取决于粗化阶段，即 TTCA 的有限次迭代粗化. 回顾算法 1 中的 TTCA，假设图 G 包含 n 个节点和 m 条边，第 1 行中的排序过程可在 $\Theta(n\log(n))$ 时间内完成，且空间复杂度为 $\Theta(n)$；当数据从磁盘写入内存，第 5 行至第 10 行通过两层嵌套循环遍历图上节点以及度大于该节点的相邻节点来遍历图上的边，因此时间复杂度为 $\Theta(m)$；第 14 行相邻节点集的交集计算完全可以在 $\Theta(m^{1/2})$ 时间内完成，这是因为节点已排序，因此第 15 行最内层循环的时间复杂度也为 $\Theta(m^{1/2})$；这 3 层嵌套循环的计算时间复杂度为 $\Theta(m^{3/2})=(\Theta(m)\ \Theta(m^{1/2}))$ 且占用 $\Theta(n+m)$ 大小的存储空间. 另外，算法中所描述的“遍历”图上三角形并不需要“完全遍历”图上的每一个三角形，如图 4 所示，若三角形 A 被融合粗化，与之相邻的三角形 B 和 C 会随之消失，因为边(1,4)、(3,4)以及边(1,5)、(3,5)会因节点 1 与节点 3 复合而被聚合形成一条边. 因此，后续遍历选择粗化对象时，可通过标志位的值($F\geq 2$)来判断并忽略所有与被粗化三角形相邻的三角形. 综上，TTCA 算法的时间复杂度将远小于 $\Theta(m^{3/2})$.

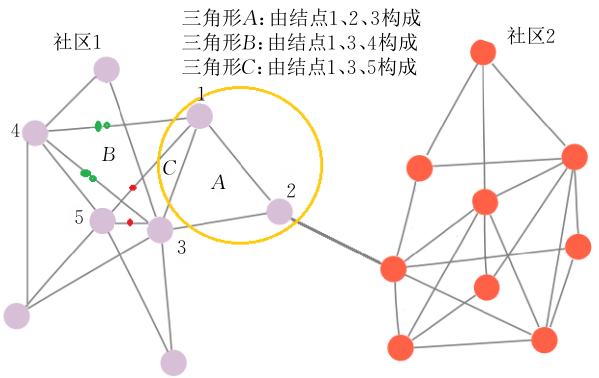
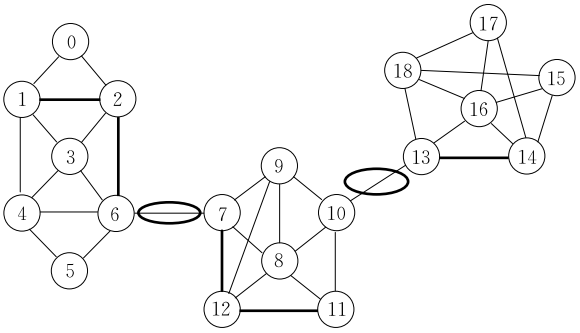


图 4 邻边三角形被聚合单边化

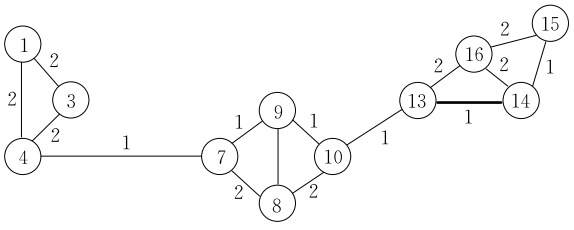
3.2 性能提升

TMLCD 算法聚焦于社会信息网络图上的三角形，并将被遍历选定的三角形 3 个顶点融合形成一个复合节点来实现粗化. 回顾前述内容，由三角形具有强社区效应[17] 可知，生成的复合节点所包含的被融合粗化了的节点必然属于同一个社区，完全不会产生基于极大匹配选边粗化策略中由复合节点展开的随机自由节点；且具有相同社区属性的节点经融合粗化后仍然同属于一个社区(仅规模缩减)，不会引起社区边界的混淆. 因此，粗化前后网络的基本社

区结构保持一致. 例如图 5(a)中所示，在包含 3 个明显社区结构的简单网络中，若基于极大匹配选边粗化策略选择分属于不同社区的边上的节点 6 和 7 或节点 10 和 13(如图 5(a)中圆圈标注的边)进行融合粗化，自然会因生成的复合节点的社区分划而混淆不同社区的边界，将节点 6 和 7、节点 10 和 13 归属到同一个社区，进而反粗化将产生随机自由节点；但如果选择网络中的三角形作为粗化对象，完全不存在被融合粗化了的节点社区属性不一致的情形，因此复合节点的社区划分完全等价于该复合节点包含的所有被融合粗化了的节点的社区属性，且社区边界不会被混淆，图 5(b)所示结果为经 TTCA(或 TMLCD)粗化后的生成图.



(a) 输入网络



(b) 经TMLCD粗化生成的网络

图 5　基本社区结构的一致性

综上，TMLCD 从基本拓扑结构上保持了生成粗化图 G_{i+1} 与被粗化图 G_i 的社区结构一致性($0\leq i\leq m$)，即经过 m 次迭代粗化后生成的最简粗化图 G_m，其拓扑结构近似于初始图 G_0，使得在图 G_m 上分析得到的社区结果可以最大限度地还原出原始大规模社会信息网络图 G_0 上的社区结构，减小分析误差，提高初始社区发现精度，且反粗化无需调优界定被展开节点的社区归属性，降低了系统的时间开销. 另外，相较于基于极大匹配选边粗化策略的多层社区发现算法，基于三角形内点同一社区性粗化策略的 TMLCD 算法可以大于 2(因为三角形 3 个顶点被融合粗化形成一个复合节点)的粗化缩减比率加快网络粗化过程，减少所需保存的中间过渡粗化图层数和每层粗化图的信息量，降低了多层社区发现算法的空间占用.

为了进一步提升社区发现性能，TTCA 算法采用了对节点的度进行排序并依次从度最低的节点开始遍历图上三角形的方法. 这样做有两方面的原因：一方面，节点度服从幂律分布的网络分布极不均衡，中心社区和边缘社区的密度相差较大[16]，若粗化阶段随机选择粗化三角形，则会造成度高的节点以其中心占优势将网络先向中心聚合粗化而模糊掉网络的边缘特征，从而覆盖掉很多有意义的边缘低密度社区；另一方面，与度高的节点相比邻的三角形数目较多，选择不同的三角形会引起图遍历粗化的随机性. 相反地，与度低的节点相比邻的三角形数目较少，当其中一个三角形被选作粗化时，与该粗化三角形相比邻的三角形的顶点会被逐渐标记(F)而变得不可选，这样从边缘向中心推进，逐渐减少度高节点相比邻的三角形中可被选作粗化对象的数目，降低网络图遍历粗化的随机性. 而且，从度最低的节点开始遍历，优先选择粗化边缘社区内的三角形，可以突出边缘低密度社区效应，弱化占优节点的中心覆盖性，进而提高 TMLCD 发现自然社区的能力.

3.3　过粗化现象

过粗化是一种极端的现象，一般不会发生，但当网络中被选作粗化对象的三角形很多都存在于高阶极大完全子图时，就会产生过粗化现象. 例如图 6 所示，粗化前社区 A、B、C 是 3 个高阶极大完全子图，若将其每个子图中包含的三角形均作融合粗化，就会产生粗化后的 1 个简单三角形，原有的社区效应完全消失. 这种导致社区内部连边数目大大减少或者消失，使得其与社区之间的连边数目相当，模糊掉社区边界，进而抵消网络的局部聚集特性(即社区内部边的密度高于社区之间边的密度的社区效应)的粗化现象就是过粗化. 过粗化将严重影响社区发现质量. 因此，本文在设计 TMLCD 算法时，采用设置标志位的值(如算法 1 中“$F<2$”，该限定值以 2 为临界且为经验值，其可通过简单地试探回溯方法进行确定)来控制一个节点所在的三角形中被选作粗化对象的数目，有效防止了过粗化现象的产生.

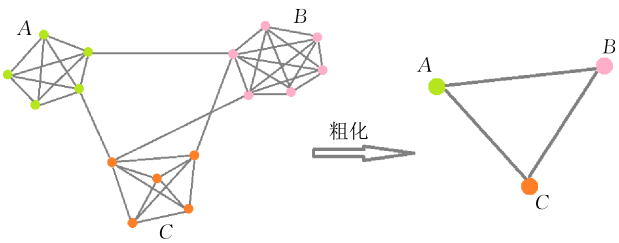


图 6　过粗化现象的产生

4 实 验

4.1 实验数据及运行环境

TMLCD 算法是基于三角形内点同一社区性粗化策略实现的,因此本实验将分成两组分别进行测试:第 1 组将测试基于三角形内点同一社区性粗化策略的性能及其对网络基本社区结构的保持能力,并对比基于极大匹配选边粗化策略,测试其对单层粗化过程性能的改善;第 2 组将对比目前可有效挖掘大规模网络社区结构的算法 Graclus、PMetis、MMLSC[26] 和 GEM[27],测试 TMLCD 的社区结构分析性能. 所有实验均运行于配置为 2.67 GHz CPU、24 核处理器且内存为 48 GB 的 Linux 机器上,且所有算法均基于 Java 语言实现.

实验数据采用斯坦福大学提供的真实社会信息网络数据负载[23],并根据实验需求分成 4 组,前两组用于第 1 组实验测试,后两组用于第 2 组实验测试,每组数据的具体信息分别列于表 1、表 2、表 3 和表 4 中. 目前的 TMLCD(或 TTCA)仅限于分析非重叠性社区结构或者重叠性较弱的社区结构. 因此,在运用上述各表中数据做社区发现之前,需先将各网络中的重叠社区部分作适度的去重叠化处理.

表 1 实验数据集(1)

数据-图	节点数	边数	三角形数
Collaboration-G_C	23133	93497	173361
Email-G_E	36692	183831	727044
Flickr-G_F	81306	768619	3985776

表 2 实验数据集(2)

数据-图	节点数	边数	$N_v=\alpha d^{\beta}$(幂律分布函数,N_v 为节点数,d 为节点度,α、β 为参数)	
			α	β
Actor	513165	1637860	2.56	−1.91
Google	216872	328917	1.39	−2.26

表 3 实验数据集(3)

数据-图	节点数	边数	三角形数
Facebook	4039	88234	1612010
GrQc	5242	14496	48260
HepTh	9877	25998	28339
HepPh	12008	118521	3358499
AstroPh	18772	198110	1351441
Brightkite	58228	214078	494728
Gowalla	196591	950327	2273138
Patents	3774768	16518948	7515023

表 4 实验数据集(4)

数据-图	节点数	边数	三角形数
DBLP	317080	1049866	2224385
Amazon	334863	925872	667129
YouTube	1134890	2987624	3056386
Orkut	3072441	117185083	627584181
LiveJournal	3997962	34681189	177820130
Twitter	17069982	476553560	1566092367

4.2 TTCA 性能评测

为了测试基于三角形内点同一社区性粗化策略的性能,并对比分析基于该策略实现的单层粗化算法 TTCA 的性能,本组测试先将已有的多层社区发现算法中基于极大匹配选边粗化策略的单层粗化过程定义为极大匹配选边粗化算法 MMSEA. 具体的实验步骤如下:先对表 1 中 3 个规模相对较小的社会信息网络数据(这里仅需对比单层粗化算法 TTCA 和 MMSEA 的性能,因此数据规模要求可适度降低)构建无向图 G_C、G_E 和 G_F;然后分别运用 TTCA 和 MMSEA 对图 G_C、G_E 和 G_F 进行粗化得到图 G_{TC}、G_{TE}、G_{TF} 和图 G_{MC}、G_{ME}、G_{MF}(这里要求经两种算法粗化后得到的图的规模相当);最后采用基于遗传算法的社区发现算法 GA-Net 对上述所有粗化图进行社区发现并对比实验结果.

由于表 1 中的数据规模较小,因此本测试可采用 GA-Net 直接分析 3 个网络图 G_C、G_E 和 G_F 上的社区结构,并将该社区发现结果作为基准社区以对比 TTCA 和 MMSEA 性能. GA-Net 的具体参数设定如下:种群规模为 500,一共遗传 50 代,每代交叉率为 0.8,突变率为 0.2,遵循轮转选择法则,并按照种群规模 10% 的比例选择精英个体直接复制到下一代. 为了对比 TTCA 和 MMSEA 的粗化结果对社区发现精度的影响差异,这里采用归一化互信息 NMI[28](Normalized Mutual Information)进行精度测量. NMI 是一种相似度度量指标,由 Danon 等人[28]证明可用于度量两种社区划分结果的相似程度,即若两个社区发现结果相同,NMI 值为 1,若完全不同,NMI 值为 0,其定义如式(6)所示:

$$NMI(\Omega,C)=\frac{I(\Omega,C)}{[H(\Omega)+H(C)]/2} \tag{6}$$

式中:Ω 和 C 分别代表两种不同社区发现结果,即 $\Omega=(\omega_1,\omega_2,\cdots,\omega_k)$,$C=(c_1,c_2,\cdots,c_j)$,$I$ 代表互信息,H 代表信息的熵,其形式化定义分别如式(7)和式(8)所示:

$$\begin{aligned} I(\Omega,C)&=\sum_k\sum_j P(\omega_k\cap c_j)\log\frac{P(\omega_k\cap c_j)}{P(\omega_k)P(c_j)}\\ &=\sum_k\sum_j(|\omega_k\cap c_j|/N)\log\frac{N|\omega_k\cap c_j|}{|\omega_k||c_j|} \end{aligned} \tag{7}$$

$$H(\Omega)=-\sum_k P(\omega_k)\log P(\omega_k)=-\sum_k\frac{|\omega_k|}{N}\log\frac{|\omega_k|}{N} \tag{8}$$

实验中，将经 TTCA 粗化得到的图 G_{TC}、G_{TE}、G_{TF} 和经 MMSEA 粗化得到的图 G_{MC}、G_{ME}、G_{MF} 分别通过算法 GA-Net 进行社区发现，再基于上述的基准社区分别代入式(6)，计算其各自对应的 NMI 值(NMI_{TTCA} 和 NMI_{MMSEA})并列入表 5 中. 对比表 5 中的每一列数据可以看到，NMI_{TTCA} 值均显著优于 NMI_{MMSEA} 值，且较接近于 1，说明在不做调优处理的情况下，基于 TTCA 粗化得到的网络图所发现的社区与基准社区很相近，即基于三角形内点同一社区性的粗化策略充分保持了网络粗化前后其基本社区结构的一致性，因此可以显著改善多层社区发现算法的性能，提高社区发现精度. 而 MMSEA 不能保证上述结论，其较低的 NMI_{MMSEA} 值显示，经过 MMSEA 粗化后的社区发现结果与基准社区相差较大，因此反粗化阶段需调用大量的调优工作以提高多层社区发现算法的计算精度.

表 5　粗化算法对社区发现结果的影响

NMI	Collaboration	Email	Flickr
NMI_{TTCA}	0.957	0.932	0.941
NMI_{MMSEA}	0.633	0.605	0.619

此外，为了证明 TTCA 在分析节点度服从幂律分布的社会信息网络时性能优于 MMSEA，测试又针对表 2 中列出的两个服从幂律分布的大规模网络 Actor 和 Google，分别采用 TTCA 和 MMSEA 作逐层迭代粗化. 两种粗化算法逐层迭代生成的粗化图 G_i 上的节点数和边数，相较于初始网络图 G_0 上的节点数和边数，其比率变化如图 7 所示(该比率是 $|V_{G_i}|/|V_{G_0}|$ 或者 $|E_{G_i}|/|E_{G_0}|$，而不是粗化缩减比率 $|V_{G_i}|/|V_{G_{i+1}}|$ 或者 $|E_{G_i}|/|E_{G_{i+1}}|$，但两者反映的算法粗化本质是一致的).

从图 7 中可以看到，同一横坐标下，经过相同层数的粗化，MMSEA 保留的节点数和边数大于 TTCA；同一纵坐标下，要达到相同的粗化程度，MMSEA 需经过更多次迭代粗化操作，这充分说明了采用基于三角形内点同一社区性粗化策略的 TTCA 其粗化幂律分布网络的能力高于 MMSEA，并以大于 2 的高粗化缩减比率(该值可通过计算 $(|V_{G_i}|/|V_{G_0}|)/(|V_{G_{i+1}}|/|V_{G_0}|)$ 或 $(|E_{G_i}|/|E_{G_0}|)/(|E_{G_{i+1}}|/|E_{G_0}|)$ 得到，而 MMSEA 的粗化缩减比率小于 2)加快了网络粗化过程. 另外，图 7 中 MMSEA 比率折线的渐近值明显高于 TTCA 比率折线的渐近值. 实验结果表明：TTCA 的基于三角形内点同一社区性粗化策略，以及依度值排序、从低到高地选择节点并开始由边缘向中心逐渐推进式的遍历三角形的粗化方式，有效地避免了 MMSEA 极大匹配选边的随机性以及因度高节点的中心占优势而引起的粗化过程提早终止的现象，适用于分析具有幂律分布特性的大规模社会信息网络.

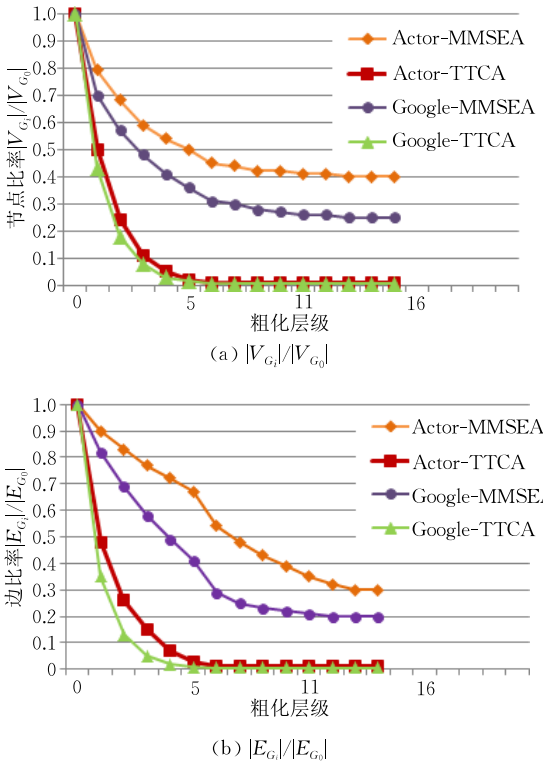


图 7　幂律分布网络粗化性能的比较

4.3　TMLCD 性能评测

首先，针对表 3 中的 8 个数据集，我们先来测试 TMLCD 的可适用范围，即网络中包含三角形的数目比例对于 TMLCD 社区发现性能的影响. 由于表 3 中的网络数据规模大小不一，且多数规模适中，我们仅通过 TMLCD 挖掘各网络社区结构的精度来分析和估量 TMLCD 的适用场景. 实验过程中我们采用两个社区发现质量评估函数：归一化互信息 NMI 和兰德指数 RI[29](Rand Index，其定义如式(9)所示，式中 TP、TN、FP 和 FN 的具体含义可参见文献[29]，且 RI 值越大表示算法的社区发现精度越高)来同时测量 TMLCD 的计算精度，并将结果 NMI 值和 RI 值列入表 6 中.

$$RI=\frac{TP+TN}{TP+FP+FN+PN} \tag{9}$$

分析表 6 中的计算精度 NMI 值和 RI 值，很明显 TMLCD 在各社会信息网络上的社区发现结果质量普遍很高，除了网络 Patents；再来回顾表 3 中

表 6 TMLCD 的计算精度

数据-图	*NMI*	*RI*
Facebook	0.973	0.941
GrQc	0.937	0.903
HepTh	0.902	0.864
HepPh	0.963	0.932
AstroPh	0.950	0.918
Brightkite	0.921	0.887
Gowalla	0.928	0.895
Patents	0.669	0.626

各网络的数据信息，对比于其他社会信息网络，Patents 包含的三角形数目明显少于边的数目，而其他网络包含三角形的数目均大于边的数目. 因此，根据网络数据信息的统计和计算精度结果的分析，可以粗略估计 TMLCD 适用于富含三角形的社会信息网络社区结构分析，即适用于所包含三角形的数目大于边的数目(或至少与边的数目相当)的社会信息网络分析.

其次，针对表 4 中所列的 6 个大规模社会信息网络，我们测试 TMLCD 计算精度、运行时间以及内存占用 3 个方面的性能，并对比于 4 个可扩展挖掘大规模社会信息网络社区结构的算法 Graclus、PMetis、MMLSC[26]（Graclus、PMetis 和 MMLSC 是 3 种广泛使用的基于极大匹配选边粗化策略实现的多层社区发现算法）以及 GEM[27]（GEM 是一种社区发现抽样近似算法，并非多层社区发现算法）的性能.

本组测试将选择如下两个具有代表性的社区评分函数[23]来度量各算法计算结果精度的优劣：

(1) Internal Density Function (IDF).

$$F(C)_{\mathrm{IDF}}=\frac{m_C}{n_C(n_C-1)/2} \tag{10}$$

(2) Normalized Cut Function (NCF).

$$F(C)_{\mathrm{NCF}}=\frac{b_C}{2m_C+b_C}+\frac{b_C}{2(m-m_C)+b_C} \tag{11}$$

以 $G=(V,E)$ 表示一个无向图，$n=|V|$ 为图中的节点数，$m=|E|$ 为边数，C 代表图上的一个子图社区，n_C 为社区 C 中的节点数，m_C 为社区 C 中的边数，b_C 为社区 C 中位于边界的边数，即 $b_C=|\{(u,v)\in E:u\in C,v\notin C\}|$. 上述式(10)中 $F(C)_{\mathrm{IDF}}$ 表示社区 C 内部边的连接密度的大小，其值越大表示社区划分效果越好；式(11)中 $F(C)_{\mathrm{NCF}}$ 表示社区 C 与其他社区之间的分离程度，其值越小表示社区划分效果越好.

通过计算上述评分函数来分别测试 TMLCD、Graclus、PMetis、MMLSC 以及 GEM 的社区发现质量，并对比各算法的时间和空间复杂度，其实验结果分别如图 8、图 9 和图 10 所示.

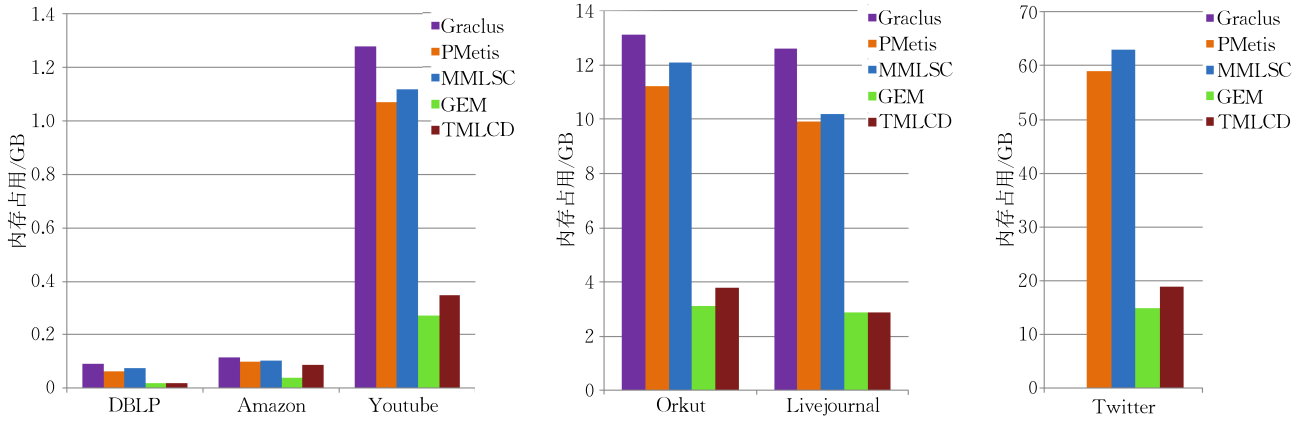


图 8 内存占用的比较

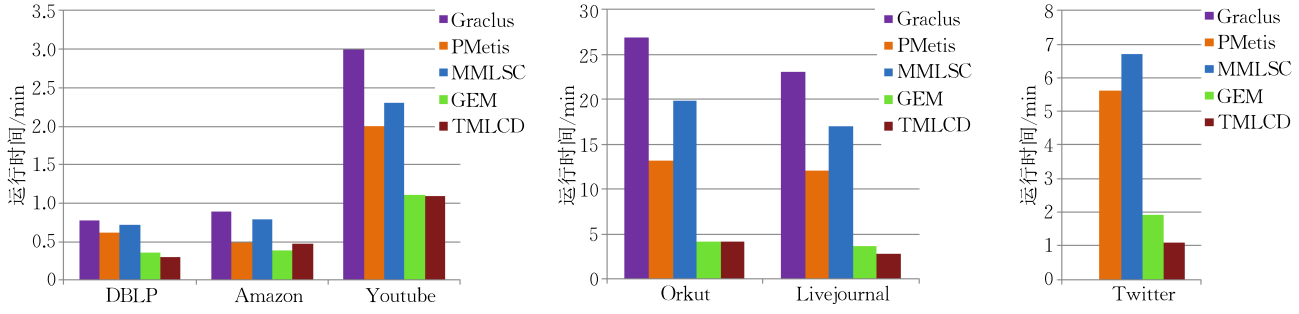
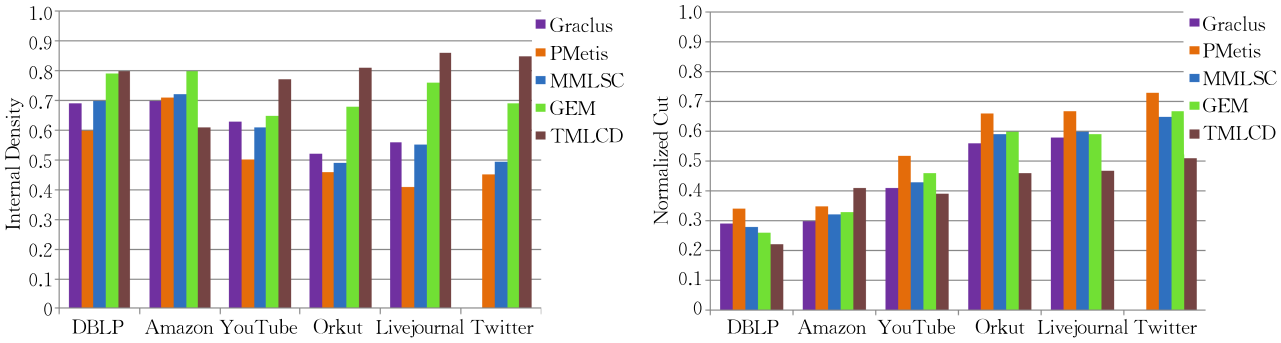


图 9 运行时间的比较



(a) Internal Density　　(b) Normalized Cut

图 10　社区发现质量的比较

分析图 8 中的实验结果，TMLCD 的系统内存占用远低于 Graclus、PMetis 和 MMLSC，其原因在于单层粗化算法 TTCA 的粗化缩减比率比 MMSEA 高，即在相同粗化程度要求下，TMLCD 所需的迭代粗化次数和每层需要保存的网络信息量要远小于 Graclus、PMetis 和 MMLSC. 而 GEM 的系统内存占用不仅低于 Graclus 和 PMetis，甚至低于 TMLCD，主要是因为 GEM 的抽样近似大大缩减了其所需存储的网络信息量. 另外，基于 Amazon 网络的实验结果显示，TMLCD 的内存占用较高，较接近 PMetis，原因是 Amazon 包含的三角形数目较少(明显少于网络中边的数目，与表 6 中实验结果的分析结论一致)，可选作粗化对象的三角形数目相对较少，因此极大地限制了 TMLCD 的性能优势，即系统需要经过更多次的迭代粗化来缩减网络规模和更多的内存空间来存储反粗化时所必需的网络信息. MMLSC 在粗化阶段基于模块度增益的选边策略需要不断选择和排除共边节点对，因此系统内存占用高于 PMetis，但相对于采用高计算复杂度的 kernel k-means 作为反粗化调优算法的 Graclus，其内存空间使用要低. 同理，在分析规模为 17 M 的 Twitter 网络时，由于 Graclus 内置的 kernel k-means 反粗化调优算法严重阻碍了其分析超大规模网络的可扩展性能，因此图 8 中的 Twitter 网络未显示 Graclus 的内存占用结果.

对比图 9 中的实验结果，TMLCD 的运行时间远小于 Graclus、PMetis 和 MMLSC，原因是实验预处理已将所有社会信息网络转变成了不包含重叠社区(或社区重叠性较弱)的网络，TMLCD 在分析这些网络时，可对初始社区发现结果直接作反粗化还原而无需任何的调优处理，加之 TMLCD 具有高粗化缩减比率和网络基本拓扑结构的高保持度，因此大大降低了其时间损耗；而 Graclus、PMetis 和 MMLSC 反粗化阶段需要大量的调优工作来重新界定由复合节点展开的自由节点的社区归属性，因此它们的时间复杂度较高. 图 9 中 Graclus 运行时间高于 PMetis 和 MMLSC 的主要原因是 Graclus 采用了更加复杂的 kernel k-means 算法分析网络，导致了高额的时间开销，而且 Twitter 网络仍未能显示 Graclus 的运行时间结果；MMLSC 运行时间高于 PMetis 的原因在于 MMLSC 需要额外计算模块度增益来匹配选择可被粗化的共边节点对. 虽然抽样近似加快了 GEM 发现大规模网络社区的速度，但图 9 中 GEM 的时间损耗仍高于 TMLCD，这是因为 TMLCD 省去了大量反粗化调优计算过程，而 GEM 却需要大量的调优工作来降低其抽样近似误差，以提高社区发现精度. 另外，基于 Amazon 网络的运行时间再一次验证了 TMLCD 适用于分析富含三角形的大规模网络(参见表 4 中数据，Amazon 包含的三角形的数目明显小于边的数目).

图 10 中对比显示了 Graclus、PMetis、MMLSC、GEM 和 TMLCD 在分析大规模社会信息网络社区结构时的质量评估函数值 IDF 和 NCF 的计算结果. 对比分析图 10 中的直方图，除了网络 Amazon 以外，TMLCD 的计算精度性能均高于 Graclus、PMetis、MMSLC 和 GEM. Amazon 网络不在此列的原因是 Amazon 包含的可选作粗化对象的三角形数目较少，限制了 TMLCD 的计算性能优势；而 Twitter 上 Graclus 性能结果的再次缺失，仍是因为其内置的 kernel k-means 算法具有过高的计算复杂度. 相对于 PMetis，Graclus 内嵌的 kernel k-means 反粗化调优算法和 MMLSC 内嵌的基于模块度增益选边算法使得其二者具有明显的计算精度优势. GEM 虽然具有较高的可扩展性，适合分析大规模甚至超大规模网络，但却是一种随机抽样近似方法，极易丢失网络中的社区结构信息，其社区发现结果

精度低于 TMLCD. 因此,基于三角形内点同一社区性的粗化策略以对网络基本社区结构的高保持度,极大地提高了 TMLCD 计算大规模社会信息网络社区结构的精度.

5 结论与展望

本文提出了一种面向大规模社会信息网络的基于三角形的多层社区发现算法 TMLCD. 该算法采用基于三角形内点同一社区性的粗化策略,将网络上遍历到的三角形 3 个顶点融合形成一个复合节点来实现粗化,不仅从基本拓扑结构上保持了初始网络的社区效应,提高了社区发现精度,而且以大于 2 的粗化缩减比率加快了网络粗化过程,节省了系统的运行时间与空间开销,同时其反粗化阶段无需大量的调优工作,进一步降低了算法整体的计算时间复杂度. 实验结果表明,TMLCD 在计算精度、内存占用以及运行时间三个方面的性能均优于 PMetis、Graclus 和 MMLSC,适用于分析富含三角形的大规模社会信息网络. 未来我们将在反粗化阶段引入分离重叠社区的调优算法,使得 TMLCD 可以发现不同划分度要求下的自然社区,或改进设计 TMLCD,使之可以分析重叠社团,拓展其应用需求,亦或基于 MapReduce 实现 TMLCD 的并行化计算,进一步提升其扩展分析超大规模社会信息网络的能力.

参 考 文 献

[1] Ng A Y, Jordan M I, Weiss Y. On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems, 2002, 2: 849-856

[2] Ravasz E, Barabsi A L. Hierarchical organization in complex networks. Physical Review E, 2003, 67(2): 026112

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

[4] Schuetz P, Caflisch A. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. Physical Review E, 2008, 77(4): 046112

[5] Handcock M S, Raftery A E, Tantrum J M. Model based clustering for social networks. Journal of the Royal Statistical Society: Series A (Statistics in Society), 2007, 170(2): 301-354

[6] Lipczak M, Milios E. Agglomerative genetic algorithm for clustering in social networks//Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation. Montreal, Canada, 2009: 1243-1250

[7] Ziv E, Middendorf M, Wiggins C H. Information-theoretic approach to network modularity. Physical Review E, 2005, 71(4): 046117

[8] Charikar M, O'Callaghan L, Panigrahy R. Better streaming algorithms for clustering problems//Proceedings of the 35th Annual ACM Symposium on Theory of Computing. San Diego, USA, 2003: 30-39

[9] Satuluri V, Parthasarathy S, Ruan Y. Local graph sparsification for scalable clustering//Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data. Athens, Greece, 2011: 721-732

[10] Li H J, Wang Y, Wu L Y, et al. Potts model based on a Markovprocess computation solves the community structure problem effectively. Physical Review E, 2012, 86(1): 016109

[11] Li H J, Zhang X S. Analysis of stability of community structure across multiple hierarchical levels. Europhysics Letters, 2013, 103(5): 58002

[12] Yang B, Liu J M, Feng J F. On the spectral characterization and scalable mining of network communities. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(2): 326-337

[13] Karypis G, Kumar V. Parallel multilevel k-way partitioning scheme for irregular graphs//Proceedings of the 1996 ACM/IEEE Conference on Super Computing. Washington, USA, 1996: 369103

[14] Karypis G, Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs. Society for Industrial and Applied Mathematics Journal on Scientific Computing, 1998, 20(1): 359-392

[15] Dhillon I S, Guan Y, Kulis B. Weighted graph cuts without eigenvectors: A multilevel approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(11): 1944-1957

[16] Abou-Rjeili A, Karypis G. Multilevel algorithms for partitioning power-law graphs//Proceedings of the 20th International Conference on Parallel and Distributed Processing. Rhodes Island, Greece, 2006: 124-134

[17] Kumpula J M, Kivelä M, Kaski K, Saramäki J. Sequential algorithm for fast clique percolation. Physical Review E, 2008, 78(2): 026109

[18] Sales-Pardo M, Guimerà R, Moreira A A, Amaral L A N. Extracting the hierarchical organization of complex systems. Proceedings of the National Academy of Sciences of the United States of America, 2007, 104(39): 15224-15229

[19] Albert R, Barabási A-L. Statistical mechanics of complex networks. Reviews of Modern Physics, 2002, 74: 47

[20] Stephen E, Kumar V S A, Marathe M V, et al. Structural and algorithmic aspects of massive social networks//Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia, USA, 2004: 718-727

[21] Schank T, Wagner D. Finding, counting and listing all triangles in large graphs//Proceedings of the 4th International Conference on Experimental and Efficient Algorithm. Santorini Island, Greece, 2005: 606-609

[22] Watts D J, Strogatz S H. Collective dynamics of small world networks. Nature, 1998, 393: 440-442

[23] Yang J, Leskovec J. Defining and evaluating network communities based on ground-truth//Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics. Beijing, China, 2012: 1-8

[24] Pizzuti C. GA-Net: A genetic algorithm for community detection in social networks//Proceedings of the 10th International Conference on Parallel Problem Solving from Nature. Dortmund, Germany, 2008: 1081-1090

[25] Hendrickson B, Leland R. A multilevel algorithm for partitioning graphs//Proceedings of the 1995 ACM/IEEE on Supercomputing. San Diego, USA, 1995: 28-es

[26] Rotta R, Noack A. Multilevel local search algorithms for modularity clustering. ACM Journal of Experimental Algorithmics, 2011, 16: 2.3:2.1-2.3:2.27

[27] Whang J J, Sui X, Dhillon I S. Scalable and memory-efficient clustering of large-scale social networks//Proceedings of the 12th IEEE International Conference on Data Mining. Brussels, Belgium, 2012: 705-714

[28] Danon L, Duch J, Diaz-Guilera A, Arenas A. Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment, 2005, 9: P09008

[29] Santos J, Embrechts M. On the use of the adjusted rand index as a metric for evaluating supervised classification//Proceedings of the 19th International Conference on Artificial Neural Networks. Limassol, Cyprus, 2009: 175-184

KANG Ying, born in 1984, Ph. D. candidate. Her current research interests include data mining, community detection etc.

GU Xiao-Yan, born in 1987, Ph. D., assistant researcher. Her current research interests include parallel computing, massive data query optimization etc.

YU Bo, born in 1985, Ph. D., assistant researcher. His current research interests include wireless networks, wireless sensor networks, distributed computing, query processing etc.

LIN Zheng, born in 1984, Ph. D., assistant researcher. Her current research interests include nature language processing, sentiment analysis etc.

WANG Wei-Ping, born in 1975, Ph. D., professor, Ph. D. supervisor. His main research interests include data stream, high performance database and parallel processing etc.

MENG Dan, born in 1965, Ph. D., professor, Ph. D. supervisor. His main research interests include high performance computer architecture, distributed file system and system security etc.

Background

Along with the expansion of the core concept of Web 2.0, a number of social media like Weibo, Social News, WikiPedia etc. emerge. Based on these interactive social media, a new kind of networks come up which are called Social Information Networks(SINs). At present, the analysis of SINs has been applied in many domains such as friend recommendation, personalized information navigation, resource classification, economic trend forecasting, electronic commerce, marketing management, network security and public opinion monitoring. However, the unparalleled explosion and increasing complexity of SINs make it infeasible to analyze SINs from the viewpoint of node or whole network. SINs have some notable properties like small world, free scale etc., and the most important one is community structure. Taking advantage of community structure, we can study SINs from the perspective of mesoscale, reducing the complication of the network modeling and analysis.

Community has become a breakthrough point for network structure and function analysis. In the past ten years, community detection, as a basic research topic, has attracted plenty of attention from related fields. So many kinds of community detection algorithms are proposed one after another, for example k-means, spectral algorithms, hierarchical clustering, divisive algorithms, modularity-based methods, dynamic algorithms etc. In spite of the efficient application to small networks, these traditional algorithms are unsuitable to analyze large-scale networks due to their computing complexity of $\Theta(n^2)$ or greater.

Facing up to tackle large-scale data networks, approximate computing is necessary. Multilevel community detection is one of the most popular methods to discover the community structure of large-scale networks recently. Although the

current multilevel community detection algorithms have the capability to analyze networks involving millions of nodes, the so-called maximum matching edge selecting coarsening policy makes the coarsening shrink rate less than 2 without exception, restricting the scalability of algorithms severely. In comparison, we adopt a remarkable characteristic of triangle, which is that the inner vertices belong to the same community, to design a new coarsening policy and propose a triangle-based multilevel community detection algorithm called TMLCD. During the coarsening phase, TMLCD merges three vertices when traversing a triangle in the network to promote the coarsening shrink rate more than 2 and accelerate the computing speed. Moreover, the performance of community detection has been improved, because TMLCD can keep the basic community effect of the initial network. Experimental results of real networks indicate that, TMLCD outperforms the currently classical multilevel community detection algorithms in terms of computing precision, memory occupation and running time, when analyzing the SINs that are rich in triangles.

This work is supported by the National Science and Technology Support Project of China under Grant No. 2012BAH46B03, the National HeGaoJi Key Project of China under Grant No. 2013ZX01039-002-001-001, "Strategic Priority Research Program" of the Chinese Academy of Sciences under Grant No. XDA06030200, the National High Technology Research and Development Program (863 Program) of China under Grant No. 2012AA01A401, and the National Natural Science Foundation of China under Grant No. 61402473.