软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(6):1517−1530 [doi: 10.3724/SP.J.1001.2012.04076]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于带权图的层次化社区并行计算方法*

林旺群+, 卢风顺, 丁兆云, 吴泉源, 周 斌, 贾 焰

(国防科学技术大学 计算机学院,湖南 长沙 410073)

Parallel Computing Hierachical Community Approach Based on Weighted-Graph

LIN Wang-Qun+, LU Feng-Shun, DING Zhao-Yun, WU Quan-Yuan, ZHOU Bin, JIA Yan

(College of Computer, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: E-mail: linwangqun@nudt.edu.cn

Lin WQ, Lu FS, Ding ZY, Wu QY, Zhou B, Jia Y. Parallel computing hierachical community approach based on weighted-graph. *Journal of Software*, 2012,23(6):1517−1530. http://www.jos.org.cn/1000-9825/4076.htm

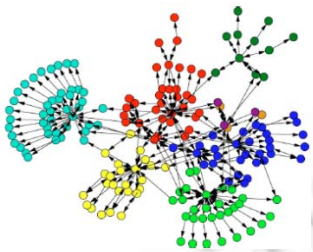
Abstract: This paper proposes a weighted-graph based hierarchical community detection approach, which defines the community structure with the use of graph partition. With the pre-defined structure, a novel parallel social network community discovery algorithm (P-SNCD for short) is designed. P-SNCD algorithm avoids the disadvantage of traditional modularity based methods, which tend to discover communities of similar scales. Moreover, it can efficiently mine communities in parallel with the CPU scale of $O(hmn)$ or $O(hn^2)$ and time complexity of $O(\log n)$, where h represents the density of the communities, m represents the total number of links and n represents the total number of nodes. Compared to the most of the existing methods, P-SNCD algorithm requires a few input parameters makes it even more practical. The accuracy and effectiveness of our algorithm is guaranteed by sufficient empirical studies in the later sections.
Key words: community discovery; weighted-graph; parallel computing; social network; hierarchical tree

摘 要: 提出了一种基于带权图并行分解的层次化社区发现方法,该方法采用图划分的方式定义社区结构,并在这种社区结构之上实现了社会网络社区发现并行算法 P-SNCD(parallel social network community discovery). P-SNCD 算法有效地避免了传统的基于“模块度”的社区发现方法倾向于发现相似规模社区的弊端.同时,该算法能够以可扩展的方式,在处理器规模为 $O(hmn)$或 $O(hn^2)$的条件下,以并行计算时间复杂度为 $O(\log n)$高效地挖掘大规模复杂社会网络中社区密度为 h 的社区,其中,n 为社会网络节点数,m 为边数,h 为用户指定的任意社区密度.所提出的算法对用户参数输入要求简单,从而使得算法具有较强的实用性.充分的实验数据验证了所提出算法的精确性和高效性.
关键词: 社区发现;带权图;并行计算;社会网络;层次化树
中图法分类号: TP391 **文献标识码**: A

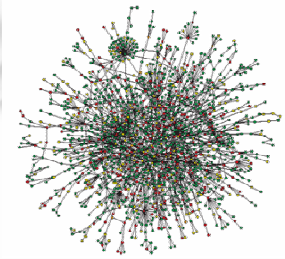
随着社会书签(social bookmark)、博客(blog)、微博(microblog)、论坛(bbs)、维基(wiki)、播客(podcast)、社交网络(social network)、协同内容标注(collaborative tagging)等各类新型“社会网络应用”的迅速普及,社会网

* 基金项目: 国家自然科学基金(60933005, 60873204); 国家高技术研究发展计划(863)(2010AA012505)
收稿时间: 2010-11-07; 修改时间: 2011-05-18; 定稿时间: 2011-07-01

络成为人们日常生活中不可或缺的一部分.社会网络分析不仅在社会学领域受到大家的热捧,在流行病传播、智能系统分析、科学论文引用和合作撰写、新陈代谢网络、电子商务市场和电子推荐系统等领域也受到人们越来越多的关注.因此,社会网络分析成为一个越来越重要的研究课题[1−3].近年来,通过对社会网络的研究,人们发现社会网络中普遍存在着小世界[1]、无标度[2,4]以及聚集[5]等基本统计特性.另外,社会网络的另一个重要特征就是网络中所呈现出的社区结构.通常认为[6],社区是一组彼此相似并与网络中其他节点存在差异的节点构成的集合,其内部连接稠密,相互之间则连接相对稀疏.一个典型的社区结构如图 1 所示.社会网络中的社区代表着特定对象的集合,如人类交往所形成的社区代表根据兴趣或背景而形成的真实的社会团体;引文网络中的社区代表针对同一主题的相关论文;万维网中的社区就是讨论相关主题的若干网站;而生物化学网络或者电子电路网络中的社区可以是某一类功能单元.社区发现对于社会网络的拓扑结构分析、功能分析和行为预测具有重要的理论意义及实用价值,已被广泛应用于恐怖组织识别、组织结构管理、新陈代谢途径(pathway)预测、蛋白质相互作用网络分析、Web 社区挖掘、搜索引擎等诸多领域[5,7−10].



(a) 某网站中具有社区结构的网络拓扑图[6]



(b) 具有社区结构的酶蛋白质交互网络[9]

Fig.1 Different types of network with community characters

图 1 具有社区特性的不同类型网络的拓扑结构

本文提出了一种基于带权图的层次化社区并行分解方法.该方法采用图划分的方式重新定义社区结构,基于这种新型的社区结构,实现了一种并行层次化社区发现算法 P-SNCD(parallel social network community discovery).本文的主要贡献如下:

(1) 提出一种新的社区结构模型,基于这种新型社区结构模型所实现的并行层次化社区发现算法 P-SNCD,有效地摆脱了传统的基于“模块度”的社区发现方法倾向于发现规模相似社区[11,13,16]的弊端;

(2) 本文给出的算法能够挖掘用户指定关系强度的社区,得到的社区结构能够更好地反映真实社会网络内部特性;

(3) 本文给出的算法具有较低的时间复杂度和良好的可扩展性.算法能够以可扩展的方式,在处理器规模为 $O(hmn)$或 $O(hn^2)$的条件下,以并行计算时间复杂度为 $O(\log n)$高效地挖掘现实社会中的大规模复杂网络中所有社区密度为 h 的社区.其中,n 为社会网络节点数,m 为边数,h 为用户指定的任意社区密度.

本文第 1 节介绍社会网络社区发现的相关工作.第 2 节给出社会网络社区发现的问题描述.第 3 节介绍面社会网络的并行层次化社区发现算法 P-SNCD.第 4 节通过大量数值实验,从不同角度验证算法的有效性和精确性.最后,在第 5 节总结全文并展望未来的工作.

1 相关工作

目前,在社会网络社区发现方面已有大量的研究成果.谱方法[11,12]将网络聚类问题转化为二次型优化问题,通过计算特殊矩阵的特征向量来优化预定义的“截”函数;KL(Kernighan-Lin)算法[13]、Fast-GN 算法[14]以及 GA (Guimera-Amaral)算法[15]将社区发现流程分为 3 个基本部分:目标函数的提出、候选解的搜索策略和最优解的搜索策略,并从不同角度采用局部搜索优化方法来挖掘潜在的网络社区.淦文燕等人[16]从数据场思想出发,将每

个社区视为拓扑势场的局部高势区,通过寻找被低势区域所分割的连通高势区域实现网络的社区划分,提出了一种基于拓扑势的社区发现算法.何晓东等人[17]提出了一种基于聚类融合的遗传算法用于复杂网络社区挖掘,该算法将聚类融合引入到交叉算子中,利用父个体的聚类信息辅以网络拓扑结构的局部信息产生新个体,避免了传统交叉算子单纯交换字符块而忽略了聚类内容所带来的问题.沈华伟等人[18]采用信息论的思想,提出了一种基于信息瓶颈的社区发现方法.该方法通过寻找网络的最优压缩表示来发现网络的社区结构,最优压缩表示尽可能多地保留原始网络拓扑特征.Wu 等人[19]提出了快速启发式算法 WH,该算法将复杂网络建模为电路系统,网络连接看成是具有电阻的线路,不同位置的网络节点具有不同的电位势.WH 算法的启发式规则是:当在不同的社区中分别选取两个节点作为正负极后,由于社区之间的电阻远远大于社区内部电阻,因此相同社区内的节点位势应近似相同,而不同社区之间的节点位势应具有显著差异.杨博等人[20]提出了基于马尔可夫随机游走模型的启发式符合网络聚类算法 FEC,该算法所采用的基本假设是:从任意给定的社区出发,网络中的随机游走过程达到起始社区内节点的期望概率将大于达到起始社区外节点的期望概率.通过计算在给定时刻随机游走过程到达所有节点的期望转移概率分布,进而根据该分布的局部一致性原理区别出各个不同的社区结构.

除了上述社会网络社区发现方法外,最新社区的发现研究成果还包括基于图的生成树分析的方法[22]、基于张量分解的方法[23]、基于粒子和密度演化聚集的方法[24]、基于贝叶斯推理的方法[25]等.

2 问题描述

社会网络社区挖掘即寻找网络中满足一定条件的节点集合,这些节点的集合通常要满足社区的基本条件:集合(社区)内边的密度高于集合(社区)之间边的密度.常用的社区内聚性评价方法为 Newman 和 Grivan[6]提出的网络模块性评价函数(又称 Q 函数).然而已有研究表明[27],Q 函数存在分辨极限(resolution limit)的问题,即采用 Q 函数作为社区结构内聚性启发式规则的相关算法倾向于发现规模相似的社区.除此之外,基于 Q 函数的社区发现方法强调社区的自然划分[5,11],而这种对社区的自然划分往往使得到的社区无法按照社区内部成员之间的关系紧密程度进一步进行子社区划分.比如对某个大学的所有课外研究小组而言,社会网络研究小组全体成员所形成的社区是个自然划分,但想要进一步得到社会网络研究小组中具有给定合作紧密度的不同子兴趣小组(如社区发现兴趣小组),传统的基于 Q 函数的社区发现方法就显得无能为力.另外,根据前面社会网络中社区的基本条件,对于社区的一个直观感觉就是成员关系越紧密的社区,社区内部应该以较少的点包含较多的边.根据这一原则,受文献[22]启发,定义 1 给出了一种新型的社区网络内聚性评价方法.

定义 1. 社会网络形成的带权图 $G_{sn}=(V,E,W)$的一个子图 L 具有密度(内聚性)$l(l\in Z^{+})$,当且仅当下式成立:

$$\min_{\forall P}\left\{\frac{|E(L/P)|}{|V(L/P)|-1}\right\}>l \tag{1}$$

其中,$E(L/P)$表示图 L'的所有边 E'的集合,其中,图 L'由对 L 进行一个数学意义上的 $P(|P|>1)$划分,然后将所有属于同一划分块的所有节点浓缩成一个新节点,同时,各个划分块之间原有的边连接保持不变所形成.对于$\forall e'\in E'$,如果 e'的权值为 $w_{e'}$,那么在计算$|E'|$时,e'等价于 $w_{e'}$条权值为 1 且与 e'平行的边.$E(L/P)$表示在上述 P 划分下形成的图 L'的所有新节点集合 V'.子图 L 密度可以理解为对 L 的所有划分中,使得公式(1)成立的最大整数 l.显然,只具有单个节点的带权图 G_{sn} 是没有社区划分意义的(这时对应的$|P|=1$),因此默认情况下,单节点 G_{sn} 具有任意的 l 密度[22].

在定义 1 中,将任意节点对之间的权值为 w 的边看成是 w 条权值为 1 的边.在余下的本文中,如不特别说明,在任意节点对之间进行边的删除(增加)操作都只是在原有边的基础上进行权值减 1(加 1)操作.当边的权值为 0 时,边的删除操作才表示物理意义上的删除操作.

定义 2. 子图 L 是社会网络形成的带权图 $G_{sn}=(V,E,W)$中的一个 l 级别社区,当且仅当 L 是 G_{sn} 中具有密度 l 的最大子图[22].

本文提出的社会网络社区发现算法就是要对给定的社会网络形成的带权图 $G_{sn}=(V,E,W)$和正整数 h,找出 G_{sn} 中所有 h 级别社区.

3 基于带权图的并行层次化社区发现算法

如果直接根据定义 1 和定义 2 来挖掘给定社会网络带权图 G_{sn} 的所有 h 级别社区,即首先枚举 G_{sn} 的所有子图,然后对每一个子图枚举其所有划分,接着计算每个划分的 DV 值,最后计算最小 l 值的方法来求解所有 h 级别的社区,那么其算法时间复杂度至少为 $O(2^n)$.分析过程是:假设 G_{sn} 的节点数$|V|=n$,那么 G_{sn} 的所有节点数大于 1 的子图数为 $\sum_{i=2}^{n}(C_n^i)$.对于节点数为 $m(m\leqslant n)$的子图 G_m,求其所有的划分是计算机领域著名的整数划分问题(http://en.wikipedia.org/wiki/Integer_partition).

假设这样的划分个数为$\rho(m)$(当 $m\to\infty$, $\rho(m)\approx(\exp(\pi\sqrt{2m/3}))/(4m\sqrt{3})$),那么所有划分个数为

$$\rho(n)=\sum_{i=2}^{n}(C_n^i)\rho(i).$$

因此,$O(\rho(n))\geqslant O(2^n)$恒成立.显然,这种基于穷尽枚举的方式存在组合爆炸问题,从而不适合复杂社会网络的社区挖掘.

3.1 问题转换

定义 3(生成树). 社会网络形成的带权图 $G_{sn}=(V,E,W)$的生成树是由图 G_{sn} 中全部节点组成且只包含 $n-1$(假设节点数$|V|=n$)条边的连通子图.

定义 4(边不相交生成树). 社会网络形成的带权图 $G_{sn}=(V,E,W)$的边不相交生成树是指由图 G_{sn} 生成的多棵生成树之间没有公共边的生成树.

定义 5(生成森林). 社会网络形成的带权图 $G_{sn}=(V,E,W)$的生成森林是由图 G_{sn} 全部节点组成且只包含若干棵生成树的图.

引理 1. 图 L 包含 l 棵边不相交生成树,当且仅当下式成立[28]:

$$\min_{\forall P}\left\{\frac{|E(L/P)|}{|V(L/P)|-1}\right\}\geqslant l \tag{2}$$

引理 2. 图 $L=(V',E',W')$为图 $G_{sn}=(V,E,W)$中一个至少具有 l 密度的子图,当且仅当删除 L 中任意一条边 e 所形成的新子图 $L\backslash\{e\}$包含 l 棵边不相交生成树[22].

引理 2 在文献[22]中已经得到初步证明.为了更好地了解本文所提出的算法原理,对引理 2 的证明阐述如下:一方面,假设图 $L=(V',E',W')$为 G_{sn} 的一个至少具有 l 密度的子图,根据定义 1 可知式(3)恒成立.对于$\forall e\in E'$,令 $L'=L-\{e\}$.因为有$|V(L'/P)|-1=|V(L/P)|-1$,且$|E(L/P)|\geqslant|E(L'/P)|$,所以公式(4)成立.由引理 1 可知,$L'=L-\{e\}$包含 l 棵边不相交生成树.

$$\min_{\forall P}\left\{\frac{|E(L/P)|}{|V(L/P)|-1}\right\}>l \tag{3}$$

$$\min_{\forall P}\left\{\frac{|E(L'/P)|}{|V(L'/P)|-1}\right\}\geqslant l \tag{4}$$

$$\min\left\{\frac{|E(L/P')|}{|V(L/P')|-1}\right\}\leqslant l \tag{5}$$

另一方面,对于$\forall e\in E'$,假设 $L'=L-\{e\}$包含 l 棵边不相交生成树,由引理 1 可知公式(4)成立.假设存在 L 的一个划分 P',使得公式(5)成立.令 $e_0\in E(L/P')$,$L''=L-\{e_0\}$.假设$\{T_1,\ldots,T_l\}$是 L''的 l 棵边不相交生成树集合,那么有 $E(T_i/P')\subseteq E(L''/P')$成立.又因为 T_i/P'包含$|P'|-1$ 条边,所以 L''/P'至少包含 $l(|P'|-1)$条边.进一步,$E(L/P')\geqslant l(|P'|-1)+1$,所以$|E(L/P')|/(|V(L/P')|-1)\geqslant(l(|P'|-1)+1)/(|P'|-1)>l$ 成立.而这与公式(5)矛盾,因此公式(3)成立.即 L 至少具有 l 密度子图. □

根据引理 1 和引理 2,要找出社会网络 G_{sn} 中所有 h 级别社区,等价于找出 G_{sn} 中所有最大子图 L,使得$\forall e_0\in L$,都有 $L'=L\backslash\{e_0\}$满足公式(2).

3.2 算法描述

定义 6(层次化社区树 HOT(hierachial community tree)). HOT 是这样的一棵树,HOT 中的第 l 层节点集合 L^l 中的任意一个节点 L_i^l 代表社会网络形成的带权图 $G_{sn}=(V,E,W)$节点集合 V 的一个子集 V_i^l .假设 V_i^l 及其在 G_{sn} 中对应的边集合 E_i^l 和权值集合 W_i^l 所构成的的子图为 $G_i^l=(V_i^l,E_i^l,W_i^l)$,如果:

1) L_i^l 为 HOT 内部节点,那么 L_i^l 对应的 G_i^l 表示 G_{sn} 的一个 l 级别社区;
2) L_i^l 为 HOT 的叶子节点,那么 L_i^l 对应的 G_i^l 如果满足公式(1)则为 G_{sn} 的一个 l 级别社区,否则不是 G_{sn} 的 l 级别社区.

层次化社区树 HOT 如图 2 所示.

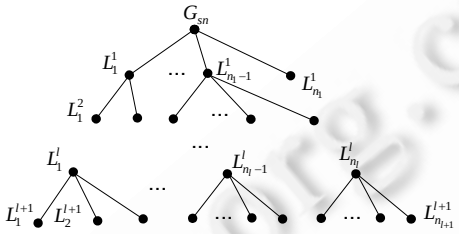


Fig.2 Hierarchy community tree (HOT)

图 2 层次化社区树 HOT

下面详细介绍社会网络 G_{sn} 的并行层次化社区发现算法 P-SNCD.为了更加清晰和方便地描述算法的核心思想以及执行流程,对算法的可并行部分并未结合某种特定并行计算模型进行描述,仅描述了算法的可并行部分及其与业务逻辑相关的关键步骤.下面首先介绍将要用到的符号系统.

生成森林 $T=\{T_1,\ldots,T_n\}$中的第 j 棵生成树用 T_j 表示;w_e 表示边 e 的权重;$Sa(\cdot)$为边的饱和度函数,并且对于 $\forall e\in E$,有 $0\leqslant Sa(e)\leqslant w_e$,每次生成一棵树 T_i,都会使得在 T_i 中出现的每一条边的饱和度加 1;E_{us} 表示所有不饱和边的集合;由第 n 条不饱和边 $e_0\in E_{us}$ 为种子边形成的环路集合用 C_n 表示;$S(e)$代表与边 e 可交换的边集合;$I(e)$代表与边 e 可交换的边所在的生成树集合;$\mathcal{Q}$表示所有处理器集合;$P(i)$表示$\mathcal{Q}$中第 i 个处理器;

下面将 P-SNCD 算法分为 3 部分进行描述.P-SNCD 算法的第 1 部分(Part 1)主要完成以下功能:

(1) 社会网络带权图 G_{sn} 的初始化工作;
(2) 实现多处理器之间的任务调度,使得每个处理器获取 HOT 最低层的一个叶子节点所对应的子图 L_n,并为每个 L_n 生成一座生成森林 T_l.

算法的第 1 部分(Part 1)形式化描述如下.

算法 1. 带权图并行分解社区发现算法.

输入:社会网络带权图 $G_{sn}=(V,E,W)$,社区级别 h;

输出:G_{sn} 中所有级别不高于 h 的社区,以及由这些社区形成的高度为 h 的层次化社区树 *HOT*.

Part 1

1) $\forall v_x,v_y\in V(x\neq y)$, $w'_e\leftarrow\sum e_i$, delete all e_i, where $e_i\in E(v_x,v_y)$; add e to v_x,v_y; $w_e\leftarrow w'_e$; /*完成任意节点对之间平行边的转换.*/
2) $l\leftarrow 0$; $L\leftarrow G_{sn}$; $L.proc\leftarrow 0$; $HOT.root\leftarrow L$; $T\leftarrow\varnothing$; $n\leftarrow 0$;
3) for $\forall e\in E(L)$, $Sa(e)\leftarrow 0$, $e.proc\leftarrow 0$, $e.flag\leftarrow 0$; /*完成初始化工作.*/
4) Let L_{leaf} be the l level leaf nodes of HOT.
5) if $l<h\cap\exists L_i\in L_{leaf}$, $L_i.proc==0$ { /*如果 HOT 的第 l 层存在没有处理的叶节点.*/
6) while $\forall P(i)\in\mathcal{Q}\cap P(i).status==idle\ parallel$ do

   6.1) Get L_i with $L_i.proc==0$ from the l level leaf nodes of HOT. /*获取 HOT 的第 l 层叶节点,完成多

处理器的任务调度工作.*/

6.2) $L_i.proc \leftarrow 1$; $L \leftarrow L_i$;

6.3) $l \leftarrow l+1$; $E_{us} \leftarrow \{e|Sa(e)<w_e \cap e \in E(L)\}$; /*保持 E_{us} 为 L 的不饱和边集合.*/

6.4) $E(L) \leftarrow E(L)-\{e|Sa(e)==w_e\}$;

6.5) $T_l \leftarrow ParalGenTree(L)$; /*采用文献[33]提出的方法生成由 L 的不饱和边形成的生成森林.*/

6.6) $T \leftarrow T+\{T_l\}$; $\forall e \in T_l$, $Sa(e) \leftarrow Sa(e)+1$; /*扩展生成森林集合 T,同时增加 T_l 相关边的饱和度.*/

6.7) $E_{us} \leftarrow E_{us}-\{e|e \in E_{us} \cap Sa(e)==w_e\}$; /*始终保持 E_{us} 为不饱和边的集合.*/

在第 1 部分的基础上,算法的第 2 部分(Part 2)主要完成以下功能:

(1) 实现多处理器之间的任务调度,使得每个处理器获取不饱和边集合 E_{us} 中的一条不饱和边 e_0;

(2) 以不饱和边 e_0 为种子边,将生成森林 T 中的每棵生成树(森林)T_j 与 e_0 形成的环路所包含的边集合存储到 C_n 中.为了提高这一过程的效率,对 C_n 生成过程进一步并行化.

算法第 2 部分(Part 2)形式化描述如下.

Part 2

6-8) if $\{e|e \in E_{us} \cap e.proc==0\}==\varnothing$ goto 6-11;

6-9) else {while $\forall P(i) \in \mathcal{Q} \cap P(i).status==idle\ parallel$ do /*并行处理不饱和边集合 E_{us};*/

6-9-1) $\forall e \in \{e|e \in E_{us} \cap e.proc==0\}$, $e.proc \leftarrow 1$; $e_0 \leftarrow e$;

6-9-2) $S(e_0) \leftarrow e_0$; $I(e_0) \leftarrow \varnothing$;

6-9-3) n++; $C_n \leftarrow \{e_0\}$;

6-9-4) if $\{e|e \in C_n \cap e.flag==0\}!=\varnothing$ { /*如果 C_n 中还有没有计算过的边.*/

6-9-5) while $P(n) \in \mathcal{Q}(n) \cap P(n).status==idle\ parallel$ do /*并行处理 C_n 中没有处理过的边.*/

6-9-5-1) $\forall e' \in \{e'|e' \in C_n \cap e'.flag==0 \cap e'.proc==0\}$, $e'.flag \leftarrow 1$; /*获取 C_n 中的一条边,完成多处理器任务调度.*/

6-9-5-2) if $(x',y' \in V(e') \cap x' \neq y'$, x', y' is in the same component of $T_l)$ {

6-9-5-3) for each $j \in \{1,\ldots,l\}$ {

6-9-5-4) Let p be the path of T_j between x' and y' where $T_j \in T$;

6-9-5-5) for $\forall e \in E(p)-C_n$ {

6-9-5-6) $S(e) \leftarrow S(e')e$; $I(e) \leftarrow I(e')j$;} /*记录可以交换的树和边.*/

6-9-5-7) $C_n \leftarrow C_n \cup E(p)$;}}

6-9-5-8) else {Let $S(e')=e_1,e_2,\ldots,e_t$, $I(e')=u_1,u_2,\ldots,u_{t-1}$.

6-9-5-9) for each $j \in \{1,\ldots,t-1\}$ { $T_{u_j} \leftarrow T_{u_j}+e_j-e_{j+1}$; $T_l \leftarrow T_l+e_1$;}

6-9-5-10) if $e==e_0$ $\{Sa(e) \leftarrow Sa(e)+1;\}$ /*更新边 e 的饱和度.*/

6-9-5-11) if $s(e_0)==w_{e_0}$, $E_{us} \leftarrow E_{us}-\{e_0\}$;

6-9-5-12) for each $e \in C_n$, $e.proc \leftarrow 0$; $C_n \leftarrow \varnothing$;}

6-9-6) goto 6-9-4};

6-9-7) else $E_{us} \leftarrow E_{us}-C_n$;

6-10) goto 6-8;}

在第 1 部分和第 2 部分的基础上,算法的第 3 部分(Part 3)主要完成以下功能:

(1) 合并集合 $C=\{C_1,C_2,\ldots,C_n\}$中具有交集的元素,使得合并后的每一个 C_r 就是一个 l 级别社区;

(2) 生成新的森林集合 T,为生成 HOT 的下一层节点所对应的社区做准备.

Part 3

6-11) for $\forall r,s \in \{1,2,\ldots,n\}$, $r \neq s$ {

6-12) if $V(C_r) \cap V(C_s) \neq \varnothing$ {

6-13) $C_r \leftarrow C_r + C_s$; delete C_s;}}

6-14) if $l<h$ {for every C_r {

6-15) for each $j \in \{1,\ldots,l\}$ $\{T_j \leftarrow T_j \cap L$; $T \leftarrow T - T_j + T_j'$;} /*生成新的生成树集合,用于下一轮迭代.*/

6-16) $L \leftarrow C_r$; $L.proc \leftarrow 0$;

6-17) for $\forall e \in E(L)$, $e.proc \leftarrow 0$, $e.flag \leftarrow 0$; /*初始化 L,为下一轮迭代做准备.*/

6-18) Add L as a new leaf node to the corresponding position at l level of HOT.}}

6-19) if $l==h$ output all C_r; /*每个 C_r 就是一个 h 级别社区.*/

7) goto 4;

从算法 P-SNCD 的执行流程可以看出,社区生成过程是一个从低级别社区生成高级社区的过程.第 l 级社区的输入依赖于第 l–1 级别社区的输出,并且分别对应 HOT 中第 l 层节点和第 l–1 层节点.

算法的基本执行流程如下:

(1) 将带权图 G_{sn} 相同节点对之间的多条平行边 $e_1,e_2,\ldots,e_n$ 用一条边 e 代替,e 的权重 w_e 为原来平行边 $e_1,e_2,\ldots,e_n$ 权重之和,即 $w_e = w_{e_1} + w_{e_2} + \ldots + w_{e_n}$;

(2) 并行处理 HOT 中最底层的叶子节点,对每个叶子节点 L 并行计算生成树.假设 $T_1,T_2,\ldots,T_{l-1}$ 是 L 的 l–1 棵边不相交的生成树,则 T_l 是由 L 的不饱和边生成的,并且与前面 l–1 棵树边互不相交地生成森林;

(3) 并行处理不饱和边集合 E_{us}.将任意不饱和边 $e_0 \in E_{us}$ 与任意生成树(森林)$T_i \in T$ 形成的环路以及该环路所包含的边与任意 $T_j \in T$ 迭代形成的环路所包含的边保存在 C_n 中;

(4) 并行处理边集合 C_n.对于 $\forall e' \in C_n \cap e'.flag=0$,找出 T_i 与 e' 形成环路所包含的边并将其保存在 C_n 中,同时对 e' 做"已处理"($e'.flag \leftarrow 1$)标记.重复上述过程,直到下面两种情况之一发生:(a) $\forall e' \in C_n$,e' 都已经被标记过;(b) e' 连接 T_l 中原来相互独立的两部份(连通子图).如果第(a)种情况发生,则表示 C_n 集合将无法得到进一步扩展,C_n 计算完毕,算法 P-SNCD 跳转到步骤(6-8)进行 C_{n+1} 的计算;如果第(b)种情况发生,则将生成森林 T_l 中相关的两棵树(连通子图)进行合并.程序跳转到步骤(6-9-4),重新开始计算 l 级别社区;

(5) 对最后形成的集合 $C=\{C_1,C_2,\ldots,C_n\}$ 中的元素进行交集测试,将具有交集的元素合并.经过合并后的每一个 C_r 是一个 l 级别社区,其中,$r \in \{1,\ldots,n'\}$(证明见第 3.3 节);

(6) 将生成树集合 $T \leftarrow T \cup \{T_j \cap L\}$ 与 l 级别社区作为算法新的输入,递归执行步骤(4),直到找到满足要求的所有 h 级别社区.

3.3 算法时间复杂度分析

为了尽可能挖掘算法的并行性,提高算法执行效率,算法 P-SNCD 通过对生成新社区 $C_r(r \in \{1,2,\ldots,n\})$和处理不饱和边集合 E_{us} 以及处理 HOT 中同一级别的叶节点 L 这 3 个层次采用并行计算.算法时间复杂度分析将从这 3 个层次进行展开.

(1) 以某一不饱和边 e_0 为种子边生成的第 l 级社区 C_r 采用并行计算.

根据所采用的并行模型,步骤(6-9-5-1)可以采用消息传递或者多处理器互斥访问共享变量 C_r 的方式,实现多处理器对 C_r 中的边 e' 的任务调度.对于以不饱和边 e_0 为种子边生成的 l 级别社区 C_r,采用$|E(C_r)|$个处理器并行计算.由于 C_r 至多包含全部 l 棵生成树所包含的边,因此有$|E(C_r)| \leqslant l(|V(L)|-1)$成立.由步骤(6-9-5-2)和步骤(6-9-5-8)可知,按照最终处理方式的不同,可以将不饱和边集合 E_{us} 分为两种类型:第 1 种类型的不饱和边通过步骤(6-9-5-7)直接生成第 l 级别的社区 C_r,第 2 种类型的不饱和边通过步骤(6-9-5-9)对生成森林 T_l 的非连通子图进行合并.由步骤(6-9-5-2)可知,计算第 1 类不饱和边只需要与 l 棵生成树进行环路测试,因此计算第 1 类不饱和边的时间复杂度为 $O(l)$.计算第 2 类不饱和边将触发与 t 棵生成树之间进行边交换,因此计算第 2 类不饱和边的时间复杂度为 $O(t)$.显然有 $O(t) \leqslant O(l) \leqslant O(h)$成立.

(2) 对不饱和边集合 E_{us} 的处理采用并行计算.

根据所采用的并行模型,步骤(6-9-1)可以采用消息传递或者多处理器对共享变量 E_{us} 进行互斥访问,实现多处理器处理不同不饱和边 e_0 的任务调度.一方面,由步骤(6-13)可知,由 L 生成的 l 级社区集合 C 中的不同社区之间不存在交集,即对于$\forall C_s,C_t\in C$,都有 $C_s\cap C_t=\varnothing$成立.假设不饱和边集合 E_{us} 中第 1 种类型的不饱和边条数为 $N_1(E_{us})$,因为 L 至多生成$|V(L)|$个 l 级别社区,而每条不饱和边 e_0 最多生成一个 l 级别社区,所以有 $N_1(E_{us})\leqslant|V(L)|$成立.另一方面,由 L 生成的生成森林 T_l 至多需扩展$|V(L)|-1$ 次后形成生成树.假设在不饱和边集合 E_{us} 中第 2 种类型的不饱和边条数为 $N_2(E_{us})$,显然有 $N_2(E_{us})\leqslant|V(L)|-1$ 成立.综合上述两个方面,有公式(6)~公式(9)成立.

$$\mathrm{Max}(N_1(E_{us}),N_2(E_{us}))\leqslant|E_{us}|\leqslant|E(L)| \tag{6}$$

$$\mathrm{Max}(N_1(E_{us}),N_2(E_{us}))<|V(L)| \tag{7}$$

以不饱和边 e_0 为种子边生成 C_r 的并行处理过程(第 3 级并行),是在并行处理 E_{us} 这个层次(第 2 级并行)之下.结合第 1 部分分析可知,在处理器规模为$|E(C_r)|.\mathrm{Max}(N_1(E_{us}),N_2(E_{us}))$的条件下,完成第 2 级并行处理的时间复杂度为 $O(h)$.

$$\mathrm{Max}(N_1(E_{us}),N_2(E_{us}))|E(C_r)|\leqslant|E(L)|.l(|V(L)|-1)\leqslant h|E(L)||V(L)| \tag{8}$$

$$\mathrm{Max}(N_1(E_{us}),N_2(E_{us}))|E(C_r)|\leqslant|V(L)|.l(|V(L)|-1)\leqslant h|V(L)|^2 \tag{9}$$

(3) 对层次化社区树 HOT 中同一级别的所有叶节点 L 进行并行处理.

根据所采用的并行模型,步骤(6-1)可以采用消息传递或者多处理器互斥访问共享变量 HOT 中叶节点集合$\{L_i|i=1,2,\ldots,b\}$的方式,将叶节点集合分成 b 路并行处理.步骤(6-1)采用 Halperin[26]提出的并行计算生成森林方法,在处理器规模为 $O((|E(L)|+|V(L)|).\log|V(L)|)$的条件下,算法复杂为 $O(\log|V(L)|)$.由于生成树 T_l 的计算与后续处理不饱和边集合 E_{us} 在程序流程上是串行的,因此当处理器规模为 $N(P)$时,完整的计算一个叶节点 L_i 的算法时间复杂度为 $O(h)+O(\log|V(L_i)|)=O(\log|V(L_i)|)$.根据公式(8)和公式(9)可得,$N(P)$如公式(10)所示.

$$\begin{aligned}N(P)&=\mathrm{Max}(O((|E(L_i)|+|V(L_i)|).\log|V(L_i)|),O(h.|V(L_i)|^2),O(h.|E(L_i)|.|V(L_i)|))\\&=\mathrm{Max}(O(h.|V(L_i)|^2),O(h.|E(L_i)|.|V(L_i)|))\end{aligned} \tag{10}$$

定理 1. 假设由社会网络形成的带权图 G_{sn} 的边规模为$|E|=m$,节点规模为$|V|=n$,那么当 G_{sn} 为稀疏图且处理器规模为 $O(hn^2)$,或 G_{sn} 为稠密图且处理器规模为 $O(hmn)$时,算法 P-SNCD 的计算复杂度为 $O(\log n)$.

证明:根据上面关于算法时间复杂度第 1 部分~第 3 部分的分析可知,计算 HOT 的一个叶子节点 L_i 所需要处理器规模为 $N(P)$,算法时间复杂度为 $O(\log|V(L_i)|)\leqslant O(\log n)$.因为同一级别的不同叶子节点计算是并行的,所以,处理第 l 层所有 b_l 个叶子节点所需要的处理器规模为 $\sum_{i=1}^{b_l}N(p)$,算法时间复杂度为 $O(\log|V(L_i)|)$.

假设 $m_l=\sum_{i=1}^{b_l}|E(L_i)|,n_l=\sum_{i=1}^{b_l}|V(L_i)|$,根据公式(11),有 $\sum_{i=1}^{b_l}N(p)\leqslant\mathrm{Max}(O(h.n^2),O(h.m.n))$ 成立.

$$\begin{aligned}\sum_{i=1}^{b_l}N(p)&=\sum_{i=1}^{b_l}\mathrm{Max}(O(h.|V(L_i)|^2),O(h.|E(L_i)|.|V(L_i)|))\\&\leqslant\mathrm{Max}(O(h.n_l^2),O(h.m_l.n_l))\\&\leqslant\mathrm{Max}(O(h.n^2),O(h.m.n))\end{aligned} \tag{11}$$

因此,定理 1 得证. □

3.4 算法讨论

P-SNCD 算法要求用户给出所要求的社区密度 h.由定义 1 和引理 2 可知,在社会网络 G_{sn} 中,对于具有密度为 h 的社区所对应的子图 G_h,可以将 G_h 看成是 G_{sn} 中去掉任何一条边 e 都有 h 棵边不相交生成树的最大子图.因此根据这一原则,用户可以根据实际问题的需要指定 h 大小.对于某个特定社会网络 G_{sn} 而言,根据第 3.2 节算法 P-SNCD 的描述可知,h 最大不超过 G_{sn} 中所有节点的最大节点度.

另外,算法给出层次化社区树 HOT 的并行计算方法.对于给定的带权图 G_{sn},在 HOT 中具有最佳社区结构位置的计算由公式(12)和公式(13)给出:

$$D_c^l = \frac{m_c^l - (n_c^l - 1)}{n_c^l(n_c^l - 1)/2 - (n_c^l - 1)} \tag{12}$$

$$D^l = \frac{2}{M^l}\sum_c m_c^l \frac{m_c^l - (n_c^l - 1)}{(n_c^l - 2)(n_c^l - 1)} \tag{13}$$

其中, m_c^l 表示第 l 级别社区的第 c 个社区中所有边的权值之和; n_c^l 表示第 l 级别社区的第 c 个社区中所包含的所有节点数; D_c^l 表示第 l 级别的 c 社区密度,很明显, D_c^l 的分母为社区 c 中的最大边数,分子为 c 社区为连通图情况下的最小边数;D^l 表示第 l 级别社区的密度;M^l 为第 l 级别社区的所有边的权值之和.由公式(13)可以知道,最终的 D^l 表示为每个第 l 级别社区密度的加权平均之和.由于某个社区的计算以及社区密度的评价都只与这个社区本身相关,而不涉及到待计算社区及以外的边和节点,因此,本文提出的方法不会存在基于模块度的分辨极限(resolution limit)限制[27].

4 实验分析

采用 BSP 并行计算模型,本文实现了 P-SNCD 算法,并通过模拟数据集和真实数据集对算法的效果和性能进行验证.在整个测试过程中,设置参数 $h=\infty$,目的是让程序计算完所有级别的社区后自动退出,以获取给定数据集中所有级别的社区集合.实验平台为由百兆以太网互联的 PC 机群,共包含 32 台 PC(双核处理器 2.8GHz,内存 2G).操作系统为 32 位 Windows XP 专业版,并行环境为 MPICH2-1.1.1.

4.1 聚类精度比较

将本文所提出的方法与 GN 算法[5]、Fast-GN 算法[14]以及 BGLL[30]算法进行对比.其中:GN 算法是经典的社区计算方法;Fast-GN 在是 GN 算法提出的模块度的思想上,采用从社区树的底端到顶端不断聚类的一种贪婪算法;BGLL 算法是 Blond 等人提出的一种快速聚类算法,能发现层次化的社区结构.由于 BGLL 算法的最终计算结果与节点的访问顺序相关,在本文中,我们对 BGLL 采用随机运行 10 次并取其其最好运行结果.计算精度评价指标我们采用社会网络社区计算精度对比常用的指标 *NMI*[31],其计算公式见公式(14).

$$NMI = \frac{-2\sum_{i,j} N_{ij}\log\left(\frac{N_{ij}N}{N_iN_j}\right)}{\sum_i N_i\log\left(\frac{N_i}{N}\right)+\sum_j N_j\log\left(\frac{N_j}{N}\right)} \tag{14}$$

其中,N 表示由两种不同算法所计算出来的社区对应的混合矩阵,N_{ij} 表示同时包含在社区 c_i 和 c_j 中的节点数,N_i 表示 N 的第 i 行元素之和.*NMI* 的取值范围为[0,1],其中,0 表示两个社区集合完全不同,1 表示两个社区集合完全相同.显然,*NMI* 值越大,两个社区集合越相似.

模拟数据集采用 LFR(Lancichinetti-Fortunato-Radicchi)程序[31]生成的基准模拟数据(以下简称 LFR).LFR 由 Lancichinetti 等人提出,是一个比较公认[32]的能够生成较高质量的模拟数据程序.该模拟程序的几个参数说明如下:n 表示模拟数据的节点个数,m 表示模拟数据的边数,k 表示每个节点的平均节点度,max$k$ 表示节点的最大节点度,u 表示每个节点与这个节点不在同一个社区中的节点相连的边占该节点所有边总和的比例,min$c$ 表示生成的模拟数据中最小社区拥有的节点数,max$c$ 表示生成的模拟数据中最大社区拥有的节点数.在该测试中,我们生成 4 个测试网络,各个网络的参数配置见表 1.

Table 1 Simulation data generated by LFR with different parameters

表 1 LFR 不同参数配置生成的模拟数据集

Dataset	M	N	k	min$c$	max$c$	max$k$
5 000 (small)	41 125	5 000	20	10	40	40
5 000 (big)	45 722	5 000	20	20	80	80
50 000 (small)	832 980	50 000	40	10	80	80
50 000 (big)	882 453	50 000	40	10	160	160

表 1 中,规模为 5 000 个节点的数据集有两个,分别为具有较小社区结构(min$c$=10;max$c$=40,max$k$=40)的数据集 5 000(small)和具有较大社区结构的数据集 5 000(big).同样,数据规模为 50 000 的数据集也生成两个,分别为 50 000(small)和 50 000(big).值得注意的是,u 值越大,模拟数据的社区结构越不明显,因而正确计算的难度越大.

图 3 给出了 LFR 生成的 4 个不同模拟网络下,u 取不同值所对应的 *NMI* 值.在 u 值较小的情况下(u<0.2),4 种算法所表现出来的计算精度差别不大(图 3(a)~图 3(d)),这种现象在 5 000 个节点规模的情况下表现较为突出.然而随着 u 值的上升,网络的社区结构逐渐模糊化,不同算法所体现出来的性能差别开始加剧.另外值得注意的是,如图 3(a)、图 3(b)所示,在 5 000 个节点规模的情况下,BGLL 算法和 P-SNCD 算法的计算精度均高于 GN 算法和 Fast-GN 算法.在 50 000 个节点规模下,社区大小由 10~80 个节点(图 3(c))变化到 10~160 个节点(图 3(d)),不同算法在所表现出来的计算精度由图 3(c)、图 3(d)所示.由图 3(c)、图 3(d)可知,P-SNCD 算法比其他 3 种算法的计算精度都要高,尤其是在 u>0.4 以后,这种现象更加明显.出现这种情况的原因是,GN 算法、Fast-GN 算法、BGLL 算法均为基于模块度的算法,这 3 个算法的不同之处在于启发式规则的不同.由于基于模块度的计算方法受到社区规模的限制,即这类算法总是趋向于发现规模相似的社区,因而在社区规模变化不大的情况下具有较高的计算精度.但是当社区规模变化较大的情况下,基于模块度的算法的计算精度将明显下降.GN,Fast-GN 和 BGLL 在图 3(a)~图 3(d)所表现出来计算精度变化明显体现出了这一点.另一方面,由于 P-SNCD 算法在社区计算时不受社区以外的其他节点和边的制约,而只是强调社区内部具有较高的内聚性,因而摆脱了基于模块度的分辨率限制.从图 3(c)、图 3(d)可以看出,尽管社区规模发生了较大的变化,P-SNCD 并不受这种变化的影响,并在一定的 u 值范围内保持较高的计算精度.

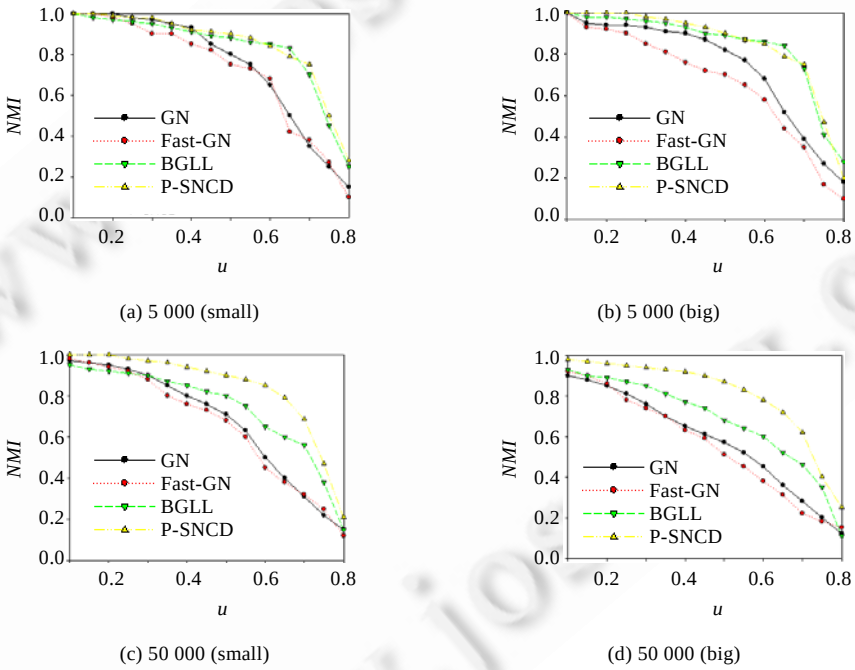


Fig.3 Accuracy comparison in different simulation datasets

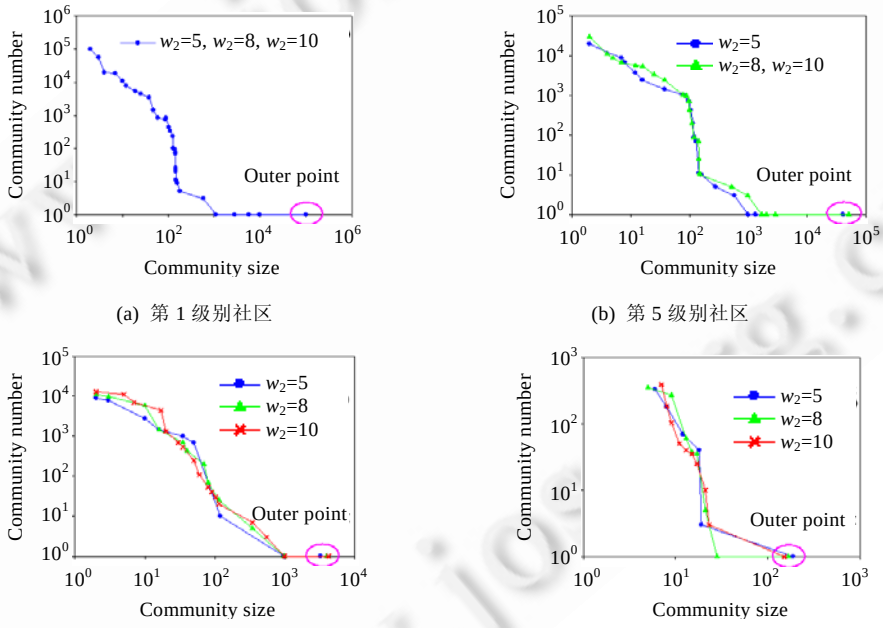
图 3 不同模拟数据集下算法精确度比较

4.2 案例学习

新浪微博关注和粉丝关系网络.新浪微博(http://t.sina.com.cn)是一个由新浪网推出、提供微型博客服务的类 Twitter 网站.为了获取新浪微博相关数据集,采用本课题组开发的 YHPODS 银河博思网络舆情管理平台进行网页抓取及解析处理.整个数据抓取过程从 2010.5.1~2010.5.3,历时 3 天,共抓取并成功解析新浪微博 ID(帐号)

号共 80 多万个,关注关系和粉丝关系 380 多万条.在新浪微博中,关注关系表示博主关注哪些人,而粉丝关系表示有哪些人关注该博主.为了更好地在新浪微博博主之间建立起一个拓扑网络,本文将关注关系和被关注关系平等对待,即任意两个博主 A 和 B,无论是博主 A 关注博主 B,还是博主 B 是博主 A 的粉丝,都会使得在发生这种行为的两个博主之间增加一条权重为 $w_1=1$ 的关联边.然而,如果博主 A 和博主 B 之间互相为关注关系(粉丝关系),由于这种关系表明博主 A 和博主 B 之间比单向的关注或被关注关系更加紧密,因此在这种情况下,本文将博主 A 和博主 B 之间的关系权重在原来的基础上增加 $w_2(w_2>0)$.

图 4(a)~图 4(d)给出了 w_2 分别取值 5,8,10 时,采用 P-SNCD 算法得到的在不同级别社区中不同大小的社区分布情况.在第 1 级别社区中(图 4(a)),$w_2=5$,$w_2=8$,$w_2=10$ 这 3 种不同的取值所形成的社区数量与社区规模关系曲线图完全一致.这是因为在不考虑边的权值差异情况下,w_2 的 3 种不同取值下的新浪微博关系网络拓扑结构相同.另外,由算法运行过程可知,3 种 w_2 取值使得那些互为关注(粉丝)关系的博主之间形成的不饱和边在计算第 1 级别社区过程成状态不会改变,仍旧为不饱和状态.因此在 w_2 分别取值 5,8,10 的情况下,第 1 级别社区的社区数量与社区规模关系曲线相同.同样,在第 5 级别社区中(图 4(b)),$w_2=8$,$w_2=10$ 的情况下,第 5 级别社区的社区数量和社区规模关系曲线完全重合.我们发现,在新浪微博关系网络中的同一级别的社区中,社区规模(博主数量)与具有这个规模的社区数量呈幂律分布,并且在每个级别的社区集合中同样出现一个包含较多博主的社区(对应图 4 的 outer point).在图 4 所示的两种不同 w 取值情况下,在第 20 级别社区中,博主数量不超过 15 位的社区数占该级别社区总数的平均值为 95.5%,并且这个比值随着社区级别的上升而略有增加,这一点与 Twitter[29] 中发现的规律类似.另外,在该数据集中出现的一个有意思的现象就是,在不同的赋值方案下,总是存在极少数社区成员较大的社区(在图 4 中用 outer point 表示),并且这些社区的大小与数量关系游离于幂律分布曲线之外.



(a) 第 1 级别社区 (b) 第 5 级别社区

(c) 第 10 级别社区 (d) 第 20 级别社区

Fig.4 Community number and community size distribution with different value of w_2

图 4 w_2 取不同值时的社区数量与社区大小分布

图 5 给出了 w_2 分别取值为 5,8,10 时,不同级别社区数量的分布情况.可以看出,w_2 的 3 种不同取值下,不同级别社区数量分布均呈现出长尾现象.另外,由图 5 可以看出,当 $w_2=10$ 时所生成的社区级别跨度最大,其最高级别社区为 61;而 $w_2=5$ 所生成的社区级别跨度最下,其最高级别社区为 45.这表明博主之间的相互关系随着 w_2

的上升而逐渐强化,也就是说,相互具有更牢固关系的博主将会出现在更高级别的社区之中.在 w_2=10 时,第 61 级别社区中,与地产界名人“任志强”处在同一社区的 ID(以下 ID 有可能就是博主的真实姓名)包括“潘石屹”、“刘春”、“陈志武”、“封新城”、“王冉”.

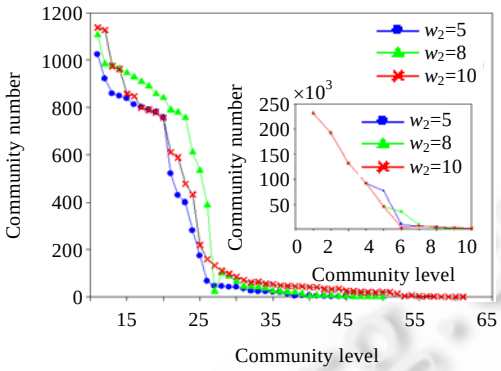


Fig.5 Community quantity distribution with different w_2

图 5 w_2 取不同值时不同级别社区数量分布

4.3 加速比测试

为了测试算法 P-SNCD 的可扩展性,对第 4.2 节中使用到的新浪微博关系网络(w_2=5)采用多处理器并行计算,图 6 给出了算法运行时间与处理器个数关系柱状图.由图 6 可以看出,算法的实际运行时间随着计算机规模的增长而减少.在计算机规模较小(低于 16 台)的情况下,算法能获得近似线性加速比;然而随着计算机的继续增加,算法的实际运行时间比预测时间有所增加.出现这种现象的原因是,随着计算机规模的增加,算法运行的并行层次上升,进一步使得计算机之间的通信开销和同步开销也迅速上升,从而在一定程度上增加了算法整体运行时间.算法运行过程中,在计算机规模为 32 的情况下,算法加速比约为 18.3.

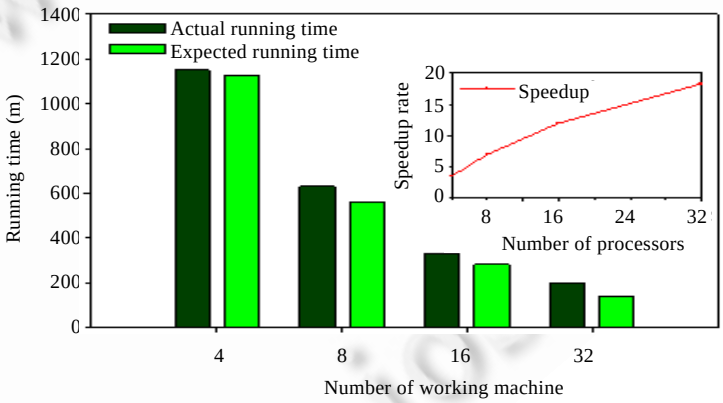


Fig.6 Speedups of P-SNCD on sina micro-blog

图 6 P-SNCD 算法在处理新浪微博关系网络的加速比

5 结束语

社区发现是社会网络研究领域的一个重要研究方向.针对现有的社区发现方法存在的弊端,本文提出一种基于带权图并行分解的层次化社区发现方法.该方法采用图划分的方式重新定义社区结构,基于这种新型的社区结构,实现了并行层次化社区发现算法 P-SNCD.该算法有效地避免了传统的基于模块度的社区发现方法倾向于发现相似规模社区的弊端[27],同时,算法能够以可扩展的方式,并行高效地挖掘大规模复杂网络中指定密度

的社区结构.本文提出的方法虽然是基于无向带权图模型,但是很容易稍加改进就扩展到有向图的场景,因而具有良好的可移植性.然而,在某些真实社会网络的社区之间可能存在重叠部分,本文提出的基于图划分的方法对于这种类型的社会网络社区挖掘存在固有的不足,今后的研究将针对具有社区重叠现象的社会网络进行展开.

References:

[1] Watts DJ, Strogatz SH. Collective dynamics of small-world networks. Nature, 1998,393(6638):440−442. [doi: doi:10.1038/30918]

[2] Barabási AL, Albert R. Emergence of scaling in random networks. Science, 1999,286(5439):509−512. [doi: 10.1126/science.286.5439.509]

[3] Albert R, Barabási AL, Jeong H. The Internet's Achilles heel: Error and attack tolerance of complex networks. Nature, 2000, 406(2115):378−382.

[4] Barabási AL, Albert R, Jeong H, Bianconi G. Power-Law distribution of the World Wide Web. Science, 2000,287(5461):2115a. [doi: 10.1126/science.287.5461.2115a]

[5] Girvan M, Newman MEJ. Community structure in social and biological networks. Proc. of the National Academy of Science, 2002, 9(12):7821−7826. [doi: 10.1073/pnas.122653799]

[6] Newman MEJ, Girvan M. Finding and evaluating community structure in networks. Physical Review E, 2004,69(2):026113. [doi: 10.1103/PhysRevE.69.026113]

[7] Kleinberg JM. Authoritative sources in a hyperlinked environment. Journal of the ACM, 1999,46(5):604−632. [doi: 10.1145/324133.324140]

[8] Palla G, Derenyi I, Farkas I, Vicsek T. Uncovering the overlapping community structures of complex networks in nature and society. Nature, 2005,435(7043):814−818. [doi: 10.1038/nature03607]

[9] Barabási AL, Oltvai ZN. Network biology: Understanding the cell's functional organization. Nature Reviews Genetics, 2004,11(2): 101−113. [doi: 10.1038/nrg1272]

[10] Yang B, Liu DY, Liu JM, Jin D, Ma HB. Complex network clustering algorithms. Journal of Software, 2009,20(1):54−66 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3464.htm [doi: 10.3724/SP.J.1001.2009.03464]

[11] Newman MEJ. Modularity and communities structure in networks. Proc. of the National Academy of Science, 2006,103(23): 8577−8582. [doi: 10.1073/pnas.0601602103]

[12] Shiga M, Takigawa I, Mamitsuka H. A spectral clustering approach to optimally combining numerical vectors with a modular network. In: Berkhin P, Caruana R, Wu X, eds. Proc. of the 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2007. 647−656. [doi: 10.1145/1281192.1281262]

[13] Newman MEJ. Detecting community structure in networks. European Physical Journal (B), 2004,38(2):321−330. [doi:10.1140/epjb/e2004-00124-y]

[14] Newman MEJ. Fast algorithm for detecting community structure in networks. Physical Review E, 2004,69(6):066133. [doi: 10.1103/PhysRevE.69.066133]

[15] Guimera R, Amaral LAN. Functional cartography of complex metabolic networks. Nature, 2005,433(7028):895−900. [doi: 10.1038/nature03288]

[16] Tu WY, He N, Li DY, Wang JM. Community discovery method in networks based on topological potential. Journal of Software, 2009,20(8):2241−2254 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3318.htm [doi: 10.3724/SP.J.1001.2009.03318]

[17] He DX, Zhou X, Wang Z, Zhou CG, Wang Z, Jin D. Community mining in complex networks—Clustering combination based genetic algorithm. Acta Automatic Sinica, 2010,36(8):1160−1170 (in Chinese with English abstract). [doi: 10.3724/SP.J.1004.2010.01160]

[18] Sheng HW, Cheng XQ, Cheng HQ, Liu Y. Information bottleneck based community detection in network. Chinese Journal of Computers, 2008,31(4):1706−1712 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2008.00677]

[19] Wu F, Huberman BA. Finding communities in linear time: A physics approach. European Physical Journal B, 2004,38(2):331−338. [doi: 10.1140/epjb/e2004-00125-x]

[20] Yang B, Cheung WK, Liu J. Community mining from signed social networks. IEEE Trans. on Knowledge and Data Engineering, 2007,19(10):1333−1348. [doi: 10.1109/TKDE.2007.1061]

[21] Pons P, Latapy M. Computing communities in large networks using random walks. In: Yolum P, ed. Proc. of the 20th Int'l Symp. on Computer and Information Sciences. Berlin: Springer-Verlag, 2005. 284−293. [doi: 10.1007/11569596_31]

[22] Zhang CQ, Ou Y. Clustering, community partition and disjoint spanning trees. ACM Trans. on Algorithms, 2008,4(3):35. [doi: 10.1145/1367064.1367075]

[23] Lin YR, Sun JM, Castro P, Konuru R, Sundaram H, Kelliher A. MetaFac: Community discovery via relational hypergraph factorization with propinquity dynamics. In: Kolum L, ed. Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Paris: ACM Press, 2009. 527−536. [doi: 10.1145/1557019.1557080]

[24] Kim MS, Han JW. A particle-and-density based evolutionary clustering method for dynamic networks. PVLDB, 2009,2(1): 622−633.

[25] Lin YR, Chi Y, Zhu S, Sundaram H, Tseng BL. Facetnet: A framework for analyzing communities and their evolutions in dynamic networks. In: Proc. of the 17th Int'l World Wide Web Conf. Beijing: ACM Press, 2008. 685−694. [doi: 10.1145/1367497.1367590]

[26] Halperin S, Zwick U. Optimal randomized EREW PRAM algorithms for finding spanning forests. Algorithms, 2001,39(1):1−46. [doi: 10.1006/jagm.2000.1146]

[27] Fortunato S, Barthelemy M. Resolution limit in community detection. Proc. of the National Academy of Sciences of the United States of America, 2007,104(1):36−41. [doi: 10.1073/pnas.0605965104]

[28] ST. Nash-williams C. Edge-Disjoint spanning trees of finite graphs. Journal of the London Mathematical Society, 1961, 36(1):445−450. [doi: 10.1112/jlms/s1-36.1.445]

[29] Kwak H, Lee C, Park H, Moon S. What is twitter, a social network or a news media? In: Proc. of the 19th Int'l World Wide Web Conf. Raleigh, 2010. 591−600. [doi: 10.1145/1772690.1772751]

[30] Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10):P10008. [doi: 10.1088/1742-5468/2008/10/P10008]

[31] Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms. Physical Review E, 2008,78(4): 046110. [doi: 10.1103/PhysRevE.78.046110]

[32] Huang J, Sun H, Han J, Deng H, Sun Y, Liu Y. SHRINK: A structural clustering algorithm for detecting hierarchical communities in networks. In: Proc. of the 2010 ACM Int'l Conf. on Information and Knowledge Management (CIKM 2010). Toronto, 2010. 219−228. [doi: 10.1145/1871437.1871469]

附中文参考文献:

[10] 杨博,刘大有,刘济明,金弟,马海宾.复杂网络聚类方法.软件学报,2009,20(1):54−66. http://www.jos.org.cn/1000-9825/20/3464.htm [doi: 10.3724/SP.J.1001.2009.03464]

[16] 淦文燕,赫南,李德毅,王建民.一种基于拓扑势的网络社区发现方法.软件学报,2009,20(8):2241−2254. http://www.jos.org.cn/1000-9825/3318.htm [doi: 10.3724/SP.J.1001.2009.03318]

[17] 何东晓,周栩,王佐,周春光,王喆,金弟.复杂网络社区挖掘——基于聚类融合的遗传算法.自动化学报,2010,36(8):1160−1170. [doi: 10.3724/SP.J.1004.2010.01160]

[18] 沈华伟,程学旗,陈海强,刘悦.基于信息瓶颈的社区发现.计算机学报,2008,31(4):1706−1712. [doi: 10.3724/SP.J.1016.2008.00677]

林旺群(1983－),男,湖南岳阳人,博士生,主要研究领域为社会网络数据挖掘.

吴泉源(1942－),男,教授,博士生导师,主要研究领域为人工智能,分布式计算,网络安全.

卢风顺(1982－),男,博士生,CCF 学生会员,主要研究领域为 GPU 并行计算,大规模科学与工程计算.

周斌(1971－),男,博士,副研究员,CCF 会员,主要研究领域为分布式计算,网络安全,数据挖掘.

丁兆云(1983－),男,博士生,主要研究领域为社会网络数据挖掘,信息安全.

贾焰(1960－),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为分布式计算,网络安全,数据挖掘.