一种新型的层次化动态社区并行计算方法

林旺群 邓 镭 丁兆云 吴泉源 贾 焰 周 斌
(国防科学技术大学计算机学院 长沙 410073)

摘 要 文中提出了一种可并行分解的层次化动态社区发现算法D-SNCD(Dynamic Social Network Community Discovery). D-SNCD算法充分利用复杂动态社会网络变化的局部性,对算法生成的层次化社区树HOT(Hierarchical cOmmunity Tree)的分枝进行选择性更新.与传统的对动态社会网络直接采用快照方式进行社区发现相比,D-SNCD算法在效率上取得了明显的提高.由于D-SNCD是对已有的静态社区并行计算方法P-SNCD(Parallel Social Network Community Discovery)的进一步扩展,因而D-SNCD保持着P-SNCD算法的高扩展性和高分辨率等优点.另外,D-SNCD算法对用户参数输入要求简单.严格的数学证明和充分的实验数据保证了整个算法的正确性和有效性.

关键词 社区发现;层次化社区结构;动态社会网络;并行计算;动态更新
中图法分类号 TP311 **DOI号**: 10.3724/SP.J.1016.2012.01712

Hierarchical Dynamic Community Detection by Parallel Computing

LIN Wang-Qun DENG Lei DING Zhao-Yun WU Quan-Yuan JIA Yan ZHOU Bin
(*School of Computer, National University of Defense Technology, Changsha* 410073)

Abstract We propose a parallel computing approach, namely Dynamic Social Network Community Discovery (D-SNCD), for detecting hierarchical community structures in dynamic social networks. Usually, in most of the dynamic social networks, the network structures change and evolve partially and dynamically in a unit time. By making use of this characteristic of the dynamic social network, D-SNCD updates the Hierarchical cOmmunity Tree (HOT) in a timely and choicely way. Compared to the existing community detection algorithms, which take advantage of the snapshots of the social network directly, the efficiency of D-SNCD is greatly improved. Since D-SNCD is an improved version of Parallel Social Network Community Discovery (P-SNCD), D-SNCD have the advantages of highly scalability and resolution. Moreover, the inputs of D-SNCD is simple and easy to control for users. The strictly mathematical proofs and experimental results further guarantee the correctness and effectiveness of our proposed algorithm.

Keywords community discovery; hierarchical community; dynamic social network; parallel computing; dynamic update

1 引 言

随着各类新型"社会网络应用"的迅速普及,社会网络成为人们日常生活中不可或缺的一部分.真实的社会网络中,大多数社区都表现出一定的层次性结构.例如在某些网站的社区结构中(如图1所示),某些特定的社区总是出现在某些层次更高的社

收稿日期:2011-08-29;最终修改稿收到日期:2012-04-08.本课题得到国家自然科学基金(60933005,60873204)、国家"八六三"高技术研究发展计划项目基金(2010AA012505)资助.**林旺群**,男,1983年生,博士研究生,主要研究方向为社会网络分析. E-mail: linwangqun2005@gmail.com. **邓 镭**,男,1984年生,博士研究生,主要研究方向为社会网络分析. **丁兆云**,男,1983年生,博士研究生,主要研究方向为社会网络分析. **吴泉源**,男,1941年生,教授,主要研究领域为人工智能、分布式计算、数据挖掘. **贾 焰**,女,1960年生,教授,主要研究领域为分布式计算、数据挖掘. **周 斌**,男,1971年生,研究员,主要研究领域为分布式计算、数据挖掘.

区之下. 如 Yahoo 社区下面可以分为“健康社区”、“情感社区”、“娱乐社区”等. 而在这些高层次的社区之下又可以进一步分为更加具体的社区结构，如“健康社区”之下又可以分为“运动养生社区”、“食物养生社区”等等. 甚至在这些社区之下，根据用户的交互行为，还可以进一步分为更加具体的用户交互社区. 另外一个真实的层次性化区结构的例子就是酶蛋白质的交互网络. 在酶蛋白质交互网络中，不同化学分子结合形成了具有不同功能结构的蛋白质，而不同蛋白质的结合又进一步形成了不同的功能组织. 挖掘隐藏在不同网络中层次化社区结构对进一步了解网络的特性和行为有着重要的意义. 然而，真实社会网络中的过大的数据规模（如现在流行的视频网站 Youtube 和社交网站 Facebook 所形成的社会网络节点规模都在 10^9 以上）使得现有的大多数社区发现算法无法有效地分析真实网络中的社区结构. 因此，一种有能有效地发现真实社会网络中的层次化动态社区结构，并且算法计算性能能够根据数据规模进行动态扩展的并行算法具有重要的意义.

图 1　具有层次化社区结构的社会网络①

2　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，总体来讲，可以将它们大致归纳成以下几类：

(1) 基于目标函数优化的社区发现方法

基于目标函数优化的社区发现方法将社会网络社区划分问题转化为优化问题，采用最优化定义的目标函数来挖掘社会网络的社区[1]. 例如：谱方法[2-3]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[4]、Fast-GN 算法[5]以及 GA(Guimera-Amaral)算法[6]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[7]从数据场思想出发，提出了一种基于拓扑势的社区发现算法，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. 何东晓等人[8]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[9]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. 林旺群等人[10]提出了一种新型的社区“内聚性”评价方法. 为了优化新型的“内聚性”目标函数，针对静态社会网络提出了一种并行计算方法.

(2) 基于启发式的社区发现方法

基于启发式方法的社区发现方法通过引入基本假设将社会网络社区发现问题转化为预定义启发式规则的设计问题[1]. 例如，GN 算法[1]以社区之间的边界数应大于社区内部连接的边界数这一基本假设作为启发式规则，通过重复删除具有最大边界数的边方式来划分社区. Tyler 等人[11]将统计方法引入基本的 GN 算法，提出一种近似 GN 算法. 该算法采用用蒙特卡洛方法估算出部分连接的近似边界数，从而有效地提高了算法的计算效率. Flake 等人[12]认为网络中的最大流量由网络“瓶颈”的容量决定，而在具有社区结构的网络中，网络“瓶颈”由社区之间的连接构成. 基于这一思想，Flake 等人提出的 MFC 算法能够有效挖掘社会网络中的潜在社区结构. Wu 等人[13]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. Palla 等人[14]提出了能够识别重叠网络簇结构的 CPM 算法. CPM 算法的启发式信息为：网络社区由多个相邻的 k-团(k-clique)组成，相邻的两个 k 团至少共享 $k-1$ 个节点，每个 k-团唯一地属于某个社区，但属于不同社区的 k-团可能会共享某些节点. Yang 等人[15]提出了基于马尔可夫随机游走模型的启发式复合网络聚类算法 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

① http://www.oslom.org/

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判别出各个不同的社区结构.Zhang 等人[16]基于相似度动力学(propinquity dynamics)特性,经过聚合网络拓扑结构和相似度(propinquity),基于 BSP 模型实现了针对大规模数据集的高效社区发现算法.

(3)其它社区发现方法

除了以上两类主要方法以外,还存在其它社会网络社区发现方法.例如,基于相似度的层次聚类方法.在这类方法中,节点间的相似度根据网络拓扑结构定义,如基于结构全等的相关系数(correlation coefficient)[17]、基于随机游走的相似度[18]和节点聚类中心度(clustering centrality)[19]等.另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[20]、基于张量分解的方法[21]、基于粒子和密度演化聚集的方法[22]、基于贝叶斯推理的方法[23]等.

虽然上述方法各有优势,并且在某些数据集上表现出了优异的准确性和计算性能.但是大多数算法的一个共同缺点在于算法性能不能随数据规模进行动态扩展,因而对某些真实网络中的大规模数据集不能有效地进行分析和处理.

3 问题描述

在现实社会网络中,多重图(Multi-graph)①,即社会网络的不同节点对之间可能存在多条平行边结构的带权图,能够很好地表示绝大多数社会关系.比如在一个科研论文合作撰写的社会网络中,每一个节点代表一位作者,而两个节点之间的一条边代表了一次论文合作撰写记录,边的权重代表两作者间的合作紧密程度.在该社会网络中,可以观察到以下现象:(1)同一篇科研论文可能有多位合作作者且其相互合作的紧密程度可能不同;(2)相同的两个作者之间可能存在多次合作记录.对于这两种现象,多重图的相同节点对之间具有独立权值的多条平行边恰好能够很好地表示这种论文合著关系.同样多重图也能很好地表示蛋白质相互作用、Web 社区交互等复杂网络形态.因此可以将社会网络建模成多重图 $G_{sn}=(V,E,W)$,其中 V 表示社会网络 G_{sn} 中所有实体集合,E 表示社会网络 G_{sn} 中实体之间的关系集合,W 表示社会网络 G_{sn} 中所有实体关系权重集合.

另外根据社会网络中社区的基本条件[10],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.由层次化的社区结构所形成的树状图我们可以知道,不同社区将出现在不同的分枝之下.这使得不同分枝的社区计算互不影响,因而对不同分枝的社区可以挖掘出其并行结构进行并行计算,从而有效地提高算法性能.本文在前续工作[10]的基础上,进一步提出一种基于多重图的层次化动态社区并行计算算法.

定义 1. 社会网络多重图 $G_{sn}=(V,E,W)$ 是指社会网络 G_{sn} 中的任意两个节点 $v_1,v_2\in V$ 之间可以存在多条边的图.其中,V 表示所有节点集合;E 表示节点之间的所有边集合;W 表示所有边的权重集合.

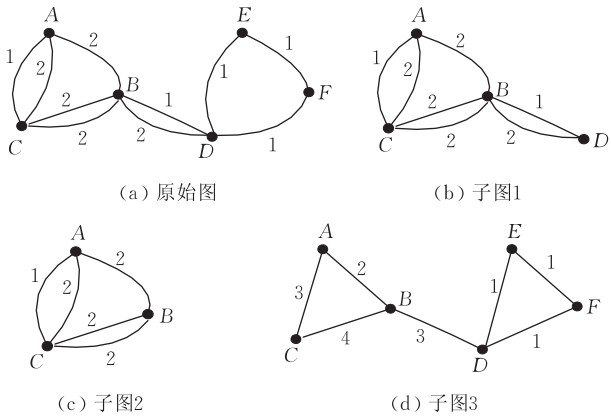
由定义 1 可知,社会网络多重图是多重图(Multi-graph)概念在在社会网络中的一个具体应用.为了论述的方便,在本文的余下章节中,统一将"社会网络多重图"简称为"多重图".

定义 2. 社会网络形成的多重图 $G_{sn}=(V,E,W)$ 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,$V(L/P)$ 表示在上述 P 划分下形成的图 L' 的所有新节点集合 V'[10,20].

为了进一步说明社区密度定义,图 2 给出了一个子图密度 $l=2$ 的一个例子.



(a)原始图 (b)子图1 (c)子图2 (d)子图3

图 2 不同密度的子图(其中图(b)~(d)同为图(a)的不同密度子图)

① http://en.wikipedia.org/wiki/Multigraph

如图 2 所示，图(b)是图(a)的一个子图. 假设 $DV=|E(L/P)|/(|V(L/P)|-1)$，则对子图(b)的所有划分及其对应的 DV 值(Density Value)如表 1 所列. 由表 1 可知，在图(b)的所有划分中，$\{\{A,B,C\},\{D\}\}$ 对应的 DV 值最小为 $DV_{\min}=3$. 因此，根据式(1)可知，图(b)的密度 $l=2$. 同样，枚举子图(c)的所有划分并计算其 DV 值可以得到图(c)的密度 $l=4$.

表 1　图 2(b)在不同划分下对应的 *DV* 值

	划分	*DV*	划分	*DV*	划分	*DV*	划分	*DV*
$\|P\|=2$	$\{\{A,B,C\},\{D\}\}$	3	$\{\{A,B,D\},\{C\}\}$	7	$\{\{A,C,D\},\{B\}\}$	9	$\{\{B,C,D\},\{A\}\}$	5
	$\{\{A,B\},\{C,D\}\}$	10	$\{\{A,C\},\{B,D\}\}$	6	$\{\{A,D\},\{B,C\}\}$	8		
$\|P\|=3$	$\{\{A,B\},\{C\},\{D\}\}$	5	$\{\{A,C\},\{B\},\{D\}\}$	4.5	$\{\{A,D\},\{B\},\{C\}\}$	6	$\{\{B,C\}\{A\},\{D\}\}$	4
	$\{\{B,D\},\{A\},\{C\}\}$	4.5	$\{\{C,D\},\{A\},\{B\}\}$	6				
$\|P\|=4$	$\{\{A\},\{B\},\{C\},\{D\}\}$	4						

定义 3.　子图 L 是社会网络形成的多重图 $G_{sn}=(V,E,W)$ 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[10,20].

在图 2 中，通过穷举图(a)中所有密度 $l=2$ 和 $l=4$ 的子图并结合定义 3 可知，图(b)是图(a)的一个 2 级别社区，图(c)是图(a)的一个 4 级别社区. 本文提出的社会网络动态社区发现算法就是要对给定的动态社会网络形成的多重图 $G_{sn}=(V,E,W)$ 和正整数 h，找出 G_{sn} 中所有的 h 级别社区.

4　基于多重图并行分解静态社区发现算法

在具体阐述层次化动态社区发现并行算法之前，先在本节简单介绍我们的前续工作，即基于多重图并行分解的静态社区发现算法 P-SNCD[10].

定义 4.　层次化社区树 HOT[10] (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多重图 $G_{sn}=(V,E,W)$ 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.

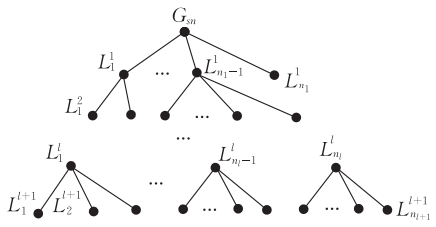


图 3　层次化社区树 HOT

图 4 给出了 P-SNCD 算法运行的并行关系层次示意图. 其中每个箭头代表一个处理器，相互平行的一组箭头代表执行并行计算任务的处理器. 箭头组之前的竖线代表处理器组并行工作时的输入，这个输入也是前一阶段的输出. 整个算法分 3 个并行计算层次. 第 1 层次并行处理 HOT 所有最底层叶子节点(对应图 4 中(a))；第 2 层次包括并行计算生成树和并行处理不饱和边集合 E_{us} 两部分(对应图 4 中(b)和(c)). 第 3 层次对以不饱和边 e_0 为“种子边”与生成树集合 T 中的每棵生成树形成的环路的边集合 C_n 采用并行计算(对应图 4 中(d)). 这种层次化并行结构充分地挖掘了算法潜在的并行性，最大限度提高了算法的效率. 然而随着并行层次的递增，多处理器之间在层次衔接位置的通信开销(假设采用 BSP 并行计算模型[26])和同步开销也随着上升. 在每个层次的并行计算开始时，可以根据实际采用的并行计算模型通过消息传递(如 BSP 并行计算模型)或多处理器互斥访问该层次的共享变量的方式(如 PRAM-CREW 并行计算模型)分发任务，这样不但自动完成了不同处理器之间的任务调度，而且还使得算法本身能够根据实际运行环境中处理器数量自动调整并行计算的层次. 这是因为第 1 层次的并行任务粒度大于第 2 层次的并行任务粒度，第 2 层次的并行任务粒度大于第 3 层次的并行任务粒度. 随着并行任务粒度的下降，每个处理器获得的并行计算量将减少，同时算法的并行度进一步增大，这将必然导致算法的通信开销和同步开销的上升. 然而，算法运行时层次间的时序关系也是从第 1 层次到第 2 层次再到第 3 层次，算法本身的这种层次间的执行时序关系决定了算法在处理器分配上总是先满足第 1 层次的并行，然后是第 2 层次，接着是第 3

层次.这种层次间的执行顺序除了能够最大限度地提高算法运行速度外,还使得在处理器数目不足的情况下,可以优先进行高层次的并行计算,从而有效降低算法的通信和同步开销.

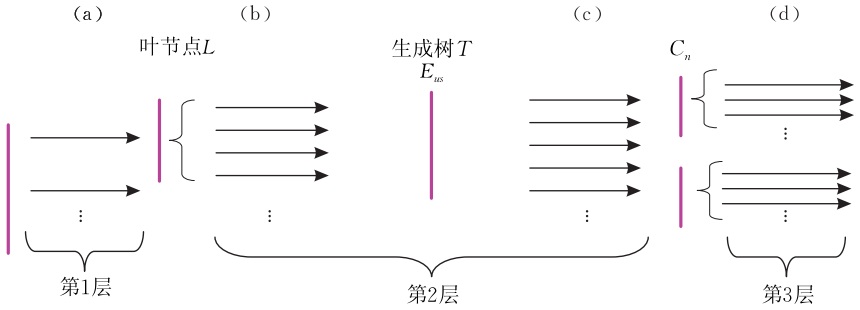


图 4 算法 P-SNCD 层次化并行结构示意图

关于 P-SNCD 算法的进一步详细描述,请参考文献[10].

5 基于多重图层次化动态社区并行计算方法

针对静态社会网络,算法 P-SNCD 给出了一种高效并行计算不同级别社区的方法[10].然而在大多数情况下,复杂系统中的社会网络拓扑结构是不断变化的.面对动态变化的社会网络,如果仅对网络拓扑结构进行快照并采用静态社会网络计算社区的方法对不同的快照进行处理,那么该社区划分方法必然是低效的.这是因为该社区计算方法没有考虑快照之间的内在联系,即动态社会网络拓扑结构变化对社区而言通常具有局部特性,也就是说一段时间内社会网络的动态变化只对部分社区的结构产生影响,而不会使得其它社区结构产生变化(相关证明见定理 1～定理 4).本小节在算法 P-SNCD 的基础上提出了一种针对演化的动态社会网络社区发现算法 D-SNCD(Dynamic Social Network Community Discovery).在详细介绍 D-SNCD 算法之前,先针对导致静态社会网络 G_{sn} 发生动态变化的 4 种行为,即"增加边"、"删除边"、"增加节点"、"删除节点"给出定理 1～定理 4.

定理 1. 如果社会网络 $G_{sn}=(V,E,W)$ 中的任意两个节点 v_x,v_y 之间的边为 e_{xy},权值为 $w_{e_{xy}}$(如果边 e_{xy} 不存在的话,可视边 e_{xy} 对应的权值 $w_{e_{xy}}=0$),那么计算在 G_{sn} 基础上在节点 v_x,v_y 之间新增加一条权重为 $\alpha(\alpha\geq 1)$ 的边 e_α 所形成的社会网络 $G'_{sn}=G_{sn}+e_\alpha$ 所对应的层次化社区树 HOT′,只需要在 G_{sn} 对应的 HOT 的基础上进行如下操作:

(1)沿着 HOT 的 *root* 节点向最底层节点方向寻找这样的节点 L_p^l,使得对于 L_p^l 的任意孩子节点 L_q^{l+1} 都有 $e_{xy}\in E(C_p^l)\bigcap e_{xy}\notin E(C_q^{l+1})$ 成立(其中,$1\leq p\leq n'_l$,$1\leq q\leq n'_{l+1}$,$1\leq l\leq h$).

(2)如果步骤(1)找到满足条件的 L_p^l,那么令 $T=T_q^{l+1}$,$z=l+1$;否则令 $T=T_i^1$,$z=1$.将不饱和边 e_{xy}(对应权值 $w'_{e_{xy}}=w_{e_{xy}}+\alpha$)及生成森林集合 T 作为算法 P-SNCD 的输入,从其步骤(6～8)开始执行生成新的 z 级别社区 $C_s^z(0<s\leq n'_z)$ 以及由 C_s^z 派生出来的所有后续高级别社区.将上述社区生成过程中形成的以 L_s^z(对应的社区为 C_s^z)为根节点的子树添加到 HOT 的 L_p^l(若 $z=l+1$)或 *root*(若 $z=1$)节点之下.

(3)若在 P-SNCD 步骤(6～13)参与 C_s^z 合并的 $C_t^z(0<t\leq n'_z\bigcap t\neq s)$ 在 HOT 中所对应的节点为 L_t^z,则删除 HOT 中以 L_t^z 为根节点的子树.

证明. 在社会网络 G_{sn} 中,任意两个节点 v_x,$v_y(v_x,v_y\in V)$ 之间新增一条权重为 α 的边 e_α 等价于将这条边所连接的两个节点之间原来的边 e_{xy}(如果 v_x,v_y 之间原来没有边,可以视 $w_{e_{xy}}=0$)的权值增加 α,即 $w_{e_{xy}}=w_{e_{xy}}+\alpha$.下面将就 e_{xy} 是否出现在某个 l 级别的社区 C_j^l(对应图 3 层次化社区树 HOT 中第 l 层的第 j 个节点 L_j^l)中分两种情况讨论.

(1)如果 e_{xy} 没有出现在任何 $C_j^l(1\leq l\leq h,1\leq j\leq n'_l)$ 中,那么只有可能是以下两种情况:(a)在 G_{sn} 中,$w_e=0$,即 v_x,v_y 两个节点之间原来是没有边连接的;(b)在 G_{sn} 中,$w_{e_{xy}}=1\bigcap e_{xy}\in T_{i,1}^1$,即连接 v_x,v_y 的边 e_{xy} 的权值为 1 且 e_{xy} 出现在生成森林集合 $T_i^1(0<i\leq n'_1)$ 的第一棵生成树(森林)$T_{i,1}^1$ 中.无论是(a)还是(b),增加边 e_{xy} 的权值都将使得 e_{xy} 成为一条不饱和边,从而需要将 e_{xy}(对应权值 $w_{e_{xy}}=w_{e_{xy}}+\alpha$)及 $T=T_i^1$ 作为算法 P-SNCD 的输入并且从其步骤(6～8)开始执行生成新的第 1 级别社区 $C_s^1(0<s\leq n'_1)$ 以及由 C_s^1 派生出来的所有后续高级别社区.在生成社区 C_s^1 的过程中,由其步骤(6～13)可

知，需要删除所有参与C_s^1合并的$C_t^1(0<t\leq n_1' \cap t\neq s)$以及由$C_t^1$生成的所有后续高级别社区. 由于第 1 级社区集合C^1中的任意两个社区之间不存在交集，所以在 1 级别社中，对社区C_s^1的操作将不会影响除C_t^1外的其它社区.

(2) 如果e_{xy}出现在某个C_j^l中，由算法 P-SNCD 的执行过程可知，任意一条不饱和边$e_0\in E_{us}$的饱和度$Sa(e_0)$随着 HOT 深度递增而非递减并且最终达到e_0的最大饱和度w_{e_0}，所以首先沿着层次化社区树 HOT 的 *root* 节点向最底层节点方向寻找，必然存在这样的节点L_p^l(对应的社区为C_p^l，其中$0<p\leq n_l'$，$0<l\leq h$)使得$e_{xy}\in E(C_p^l)\cap e_{xy}\notin E(C_q^{l+1})\cap L_q^{l+1}=L_p^l.child$，其中$L_q^{l+1}$(对应的社区为$C_q^{l+1}$，$0<q\leq n_{l+1}'$)为$L_p^l$在层次化社区树 HOT 的$l+1$层的任意孩子节点. 由于在$L_p^l$的祖先节点所对应的社区中，$e_{xy}$已经是这些社区中的一条边，所以增加$e_{xy}$的权值不会影响这些社区的结构. 下面对$e_{xy}$是否是$C_p^l$中的饱和边分两种情况进行分析. (a) $Sa(e_{xy})=w_{e_{xy}}$，即e_{xy}是C_p^l中的饱和边. 在这种情况下，当增加边e_{xy}的权值将使得e_{xy}重新成为不饱和边. 又因为$e_{xy}\in C_p^l$，所以e_{xy}即使成为不饱和边也不会对C_p^l的结构产生影响. 此时，只需在第$l+1$级别社区中重新计算以e_{xy}作为“种子边”的新社区. (b) $Sa(e_{xy})\neq w_{e_{xy}}$，即$e_{xy}$是$C_p^l$中的不饱和边. 因为$e_{xy}\notin E(C_q^{l+1})$，所以此时一定有$e_{xy}\in T_{q,l+1}^{l+1}\cap Sa(e_{xy})=w_{e_{xy}}-1$成立，其中$T_{q,l+1}^{l+1}$表示生成森林集合$T_q^{l+1}$的第$l+1$棵生成树(森林). 增加边$e_{xy}$的权值将使得$e_{xy}$成为第$l+1$级别的一条不饱和边. 无论是(a)还是(b)，这时只需要以生成森林集合$T=T_q^{l+1}$以及不饱和边e_{xy}(对应权值$w_{e_{xy}}=w_{e_{xy}}+\alpha$)作为算法 P-SNCD 的输入并且从其步骤(6～8)开始执行，生成新的$l+1$级社区$C_u^{l+1}(0<u\leq n_{l+1}')$以及由$C_u^{l+1}$派生出来的所有后续高级别社区，同时需要删除在其步骤(6～13)与C_u^{l+1}合并的所有相关$l+1$级社区$C_v^{l+1}(0<v\leq n_{l+1}'\cap v\neq u)$以及由这些社区派生出来的高级别社区. 由于在第$l+1$级社区集合$C^{l+1}$中，任意两个社区之间不存在交集，所以对社区$C_u^{l+1}$的操作将不会影响除被$C_v^{l+1}$外的其它$l+1$级社区.　　证毕.

定理 2.　如果社会网络$G_{sn}=(V,E,W)$中的任意两个节点v_x，v_y之间的边为e_{xy}，权值为$w_{e_{xy}}$(如果边e_{xy}不存在的话，可视边e_{xy}对应的权值$w_{e_{xy}}=0$)，那么计算在G_{sn}基础上删除v_x，v_y之间一条权重为β($1\leq\beta\leq w_{e_{xy}}$)的边$e_\beta$所形成的社会网络$G_{sn}'=G_{sn}-e_\beta$所对应的层次化社区树 HOT′，只需要在$G_{sn}$对应的 HOT 的基础上进行如下操作：

(1) 当$w_{e_{xy}}=\beta$时，如果对于$\forall i\in\{1,\cdots,n_1'\}$都有$e_{xy}\notin E(C_i^1)$，则无需执行额外操作就有 HOT′=HOT；否则(即$\exists i\in\{1,\cdots,n_1'\}$使得$e_{xy}\in E(C_i^1)$)假设$C_i^1-e_{xy}$所包含的不饱和边集合为$E_i'$，对应的生成森林集合为$T_1^{1\prime}=T_1^1-e_{xy}$，那么以不饱和边集合$E_{us}=E_i'$，生成森林集合$T=T_1^{1\prime}$作为算法 P-SNCD 的输入，从其步骤(6～8)开始执行生成新的第一级别社区$C_1^{1\prime},\cdots,C_x^{1\prime}(0\leq x\leq|E_i'|)$及其派生社区，将$x$棵分别以$L_1^{1\prime},L_2^{1\prime},\cdots,L_x^{1\prime}$(分别对应社区$C_1^{1\prime},\cdots,C_x^{1\prime}$)为 *root* 节点的子树加入 HOT 的 *root* 节点之下，删除 HOT 中以L_i^1为 *root* 的子树.

(2) 当$w_{e_{xy}}>\beta$时，沿着 HOT 的 *root* 节点向最底层节点方向寻找这样的节点L_r^l(对应的社区为C_r^l，$1\leq l\leq h$，$1\leq r\leq n_l'$)，使得$e_{xy}\in E(C_r^l)\cap Sa(e_{xy})=w_{e_{xy}}-\beta$并且对于$L_s^{l+1}(1\leq s\leq n_{l+1}')$有$e_{xy}\in C_s^{l+1}\cap Sa(e_{xy})=w_{e_{xy}}-\beta+1$，其中$L_s^{l+1}$为$L_r^l$在层次化社区树 HOT 的$l+1$层的某孩子节点. 假设$C_s^{l+1}-e_{xy}$所包含的不饱和边集合为$E_s'$，$C_s^{l+1}$对应的生成森林集合为$T_s^{l+1}$，那么以不饱和边集合$E_{us}=E_s'$，生成森林集合$T=T_s^{l+1}$作为算法 P-SNCD 的输入，从其步骤(6～8)开始执行生成新的第$l+1$级别社区$C_1^{l+1\prime},\cdots,C_y^{l+1\prime}(0\leq y\leq|E_s'|)$及其后续派生社区，将$y$棵分别以$L_1^{l+1\prime},L_2^{l+1\prime},\cdots,L_y^{l+1\prime}$(分别对应社区$C_1^{1\prime},\cdots,C_y^{1\prime}$)为 *root* 节点的子树加入 HOT 的$L_r^l$节点之下，并删除 HOT 中以$L_s^{l+1}$为 *root* 的子树.

证明.　在社会网络G_{sn}中，在任意两个节点v_x，$v_y(v_x,v_y\in V)$之间删除一条权重为β的边e_β等价于将这条边所连接的两个节点之间原来的边e_{xy}的权值减少β，即$w'_{e_{xy}}=w_{e_{xy}}-\beta\geq 0$. 下面将就$w'_{e_{xy}}=w_{e_{xy}}-\beta$分两种情况讨论：

(1) 如果$w'_{e_{xy}}=0$，即原来连接v_x，v_y两个节点的e_{xy}的权值$w_{e_{xy}}=\beta$. 在这种情况下将e_{xy}的权值减少β相当于直接删除e_{xy}，这使得v_x，v_y在G_{sn}中不再成为直接邻居. 下面分两种情况进行讨论. (a) 当$\exists i\in\{1,\cdots,n_1'\}$使得$e_{xy}\in E(C_i^1)$时，即$e_{xy}$出现在第 1 级别的第$i$个社区中. 因为删除边$e_{xy}$将影响$C_i^1$这个社区及其由$C_i^1$派生出来的后续社区的结构，因此需要重新生成$C_i^1$及其后续派生社区. 假设$C_i^1$所包含的不饱和边集合为$E_i$，对应的生成树集合为$T_1^1$，$C_i^1-e_{xy}$所包含的不饱和边集合为$E_i'$，对应的生成树集合为$T_1^{1\prime}=T_1^1-e_{xy}$，那么显然有$E_i'\subseteq E_i$，$T_1^{1\prime}\subseteq T_1^1$成立. 以不饱和边集合$E_{us}=E_i'$，生成森林集合$T=$

$T_1^{1\prime}$作为算法 P-SNCD 的输入，从其步骤(6～8)开始执行. 假设在步骤(6～13)的输出为 $C_1^{1\prime},\cdots,C_x^{1\prime}$ $(0\leq x\leq|E_i'|)$. 因为 $E_i'\subseteq E_i\cap T_1^{1\prime}\subseteq T_1^1$，所以必有 $C_1^{1\prime}\cup C_2^{1\prime}\cup\cdots\cup C_x^{1\prime}\subseteq C_i^1$，进一步 $C_1^{1\prime},\cdots,C_x^{1\prime}$是 x 个第 1 级别社区. 此时生成 $G_{sn}'=G_{sn}-e_\beta$所对应 HOT′只需将 x 棵分别以 $L_1^{1\prime},L_2^{1\prime},\cdots,L_x^{1\prime}$为 *root* 节点的子树加入到 G_{sn}所对应 HOT 的 *root* 节点之下，并删除 HOT 中以 L_i^1 为 *root* 的子树. (b) 当对于 $\forall i\in\{1,\cdots,n_1'\}$都有 $e_{xy}\notin E(C_i^1)$时，即 e_{xy}没有出现在任何第 1 级别社区中. 在这种情况下，即 e_{xy}也不会出现所有后续高级别社区中，因此删除 e_{xy}不会使得 G_{sn}的任何级别社区结构产发生变化.

(2) 如果 $w_{e_{xy}}'>0$，即原来连接 v_x，v_y两个节点的边 e_{xy}的权值 $w_{e_{xy}}>\beta$. 在这种情况下，沿着层次化社区树 HOT 寻找这样的节点 $L_r^l(1\leq r\leq n_l')$使得 $e_{xy}\in E(C_r^l)\cap Sa(e_{xy})=w_{e_{xy}}-\beta$并且对于 $L_s^{l+1}(1\leq s\leq n_{l+1}')$有 $e_{xy}\in C_s^{l+1}\cap Sa(e_{xy})=w_{e_{xy}}-\beta+1$，其中 L_s^{l+1}为 L_r^l在层次化社区树 HOT 的 $l+1$ 层的某孩子节点. 由于任意一条不饱和边$e_0\in E_{us}$的饱和度 $Sa(e_0)$随着 HOT 深度递增而非递减并且最终达到 w_{e_0}，所以上述 L_r^l和 L_s^{l+1}必定存在. 因为在 L_r^l及 L_r^l的祖先节点所对应的社区中，e_{xy}已经是这些社区中的一条边且 e_{xy}在这些社区中对应的饱和度 $Sa(e_{xy})\leq w_{e_{xy}}-\beta$，所以 e_{xy}的权值减少 β 不会影响这些祖先节点所对应社区的结构. 又因为社区 C_s^{l+1}及其后续包含边 e_{xy}的派生社区在 e_{xy}的权值减少 β 后将不再包含边 e_{xy}，所以 C_s^{l+1}及其派生社区需要重新计算. 假设 C_s^{l+1}所包含的不饱和边集合为 E_s，对应的生成树集合为 T_s^{l+1}，$C_s^{l+1}-e_{xy}$所包含的不饱和边集合为 E_s'，对应的生成树集合为 $T_s^{l+1\prime}=T_s^{l+1}$，那么显然有 $E_s'\subseteq E_s$成立. 以不饱和边集合$E_{us}=E_s'$，生成森林集合$T=T_s^{l+1\prime}$作为算法 P-SNCD 的输入并从其步骤(6～8)开始执行. 假设在步骤(6～13)的输出为 $C_1^{l+1\prime},\cdots,C_y^{l+1\prime}(0\leq y\leq|E_s'|)$，因为 $E_s'\subseteq E_s\cap T=T_s^{l+1}$，所以必有 $C_1^{l+1\prime}\cup C_2^{l+1\prime}\cup\cdots\cup C_y^{l+1\prime}\subseteq C_s^{l+1}$成立，进一步 $C_1^{l+1\prime},\cdots,C_y^{l+1\prime}$是 y 个第 $l+1$ 级别社区. 此时类似于(1)(a)生成 $G_{sn}'=G_{sn}-e_\beta$所对应 HOT′，只需将 y 棵分别以 $L_1^{l+1\prime},L_2^{l+1\prime},\cdots,L_y^{l+1\prime}$为 *root* 节点的子树加入到 G_{sn}所对应 HOT 的 L_r^l节点之下，并删除 HOT 中以 L_s^{l+1}为 *root* 的子树. 证毕.

定理 3. 假设社会网络 $G_{sn}=(V,E,W)$所对应的层次化社区树为 HOT，在 G_{sn}的基础上增加节点 v_a以及与 v_a相连的 $n(0\leq n\leq|E|)$条边集合$\{e_a^1,\cdots,e_a^n\}$后所形成的社会网络 $G_{sn}'=G_{sn}+v_a+\{e_a^1,\cdots,e_a^n\}$所对应的层次化社区树为 HOT′，那么构造 HOT′只需要在 HOT 的基础上进行如下操作：

(1) 如果 $n=0$ 时，HOT′=HOT；否则转入(2).

(2) 假设 G_{sn}'中 $e_a^1,\cdots,e_a^n$分别与第 1 级别社区集合 C^1中的 $x(0<x\leq n)$个社区 $C_1^{1\prime},\cdots,C_x^{1\prime}$中的节点相连接，$G_{sn}'$所对应的生成森林 $T_1^{1\prime}=T_1^1+v_a$并且 $\bigcup_{i=1}^x C_i^{1\prime}+\bigcup_{k=1}^n e_a^k$所包含的所有不饱和边集合为 E_i'. 以不饱和边集合 $E_{us}=E_i'$，生成森林集合 $T=T_1^{1\prime}$作为算法 P-SNCD 的输入，从其步骤(6～8)开始执行. 将新生成的社区 $C_1^{1\prime\prime},\cdots,C_y^{1\prime\prime}$ $(y\leq|E_i'|)$及其派生社区按照层次结构构成 y 棵树形成的森林加入到 HOT 的 *root* 节点之下以替换 HOT 中 x 棵分别以 $L_1^{1\prime},\cdots,L_x^{1\prime}$(对应的社区分别为 $C_1^{1\prime},\cdots,C_x^{1\prime}$)为 *root* 节点的子树.

证明. 分 $n=0$ 和 $n>0$ 两种情况进行证明.

(1) 当 $n=0$ 时，即增加节点 v_a到 G_{sn}没有引入任何新的边. 显然有 HOT′=HOT.

(2) 当 $n>0$ 时，即增加节点 v_a到 G_{sn}引入新的边集合$\{e_a^1,\cdots,e_a^n\}$. 假设新引入的边 $e_a^1,\cdots,e_a^n$分别与第 1 级别社区集合 C^1中的 $x(x\leq n)$个社区 $C_1^{1\prime},\cdots,C_x^{1\prime}$中的节点相连接. 一方面，对于 $\forall C\in(C^1-\bigcup_{i=1}^x C_i^{1\prime})$，因为 C 中不存在节点 v 使得 v 与 v_a相连且已经是最大扩展，所以添加边集合$\{e_a^1,\cdots,e_a^n\}$中的任何边都无法使得 C 进一步扩展. 另一方面，对于 $\forall C'\in\{C_1^{1\prime},\cdots,C_x^{1\prime}\}$，假设 $e_a\in\{e_a^1,\cdots,e_a^n\}$是连接节点 $v\in C'$和 v_a的边，那么对于 G_{sn}的生成森林 T_1^1，添加节点 v_a便得到了 $G_{sn}'=G_{sn}+v_a+\{e_a^1,\cdots,e_a^n\}$所对应的生成森林 $T_1^{1\prime}=T_1^1+v_a$，显然 $T_1^1\subseteq T_1^{1\prime}$成立. 如果 e_a的权值 $w_{e_a}\geq 2$，那么以 e_a为“种子边”形成的社区与 C'共享节点 v，所以增加节点 v_a将可能导致社区 C'进一步地扩展. 假设 $\bigcup_{i=1}^x C_i^{1\prime}+\bigcup_{k=1}^n e_a^k$所包含的所有不饱和边集合为 E_i'，以不饱和边集合 $E_{us}=E_i'$，生成森林集合 $T_1^{1\prime}$作为算法 P-SNCD 的输入，从其步骤(6～8)开始执行. 假设在步骤(6～13)的输出为 $C_1^{1\prime\prime},\cdots,C_y^{1\prime\prime}(0\leq y\leq|E_i'|)$，由于前面已经证明 $\forall C\in(C^1-\bigcup_{i=1}^x C_i^{1\prime})$都已经是最大扩展，所以 $C_1^{1\prime\prime},\cdots,C_y^{1\prime\prime}$亦无法进一步与 C 进行合并，因此 $C_1^{1\prime\prime},\cdots,C_y^{1\prime\prime}$是 y 个第 1 级别社区. 因此，将新生成的社区 $C_1^{1\prime\prime},\cdots,C_y^{1\prime\prime}$及其派生社区按照层次结构构成 y 棵树形成的森林加入到 HOT 的 *root* 节点之下，以替换 HOT 中 x 棵分别以 $L_1^{1\prime},\cdots,L_x^{1\prime}$(对应的社区分别为 $C_1^{1\prime},\cdots,C_x^{1\prime}$)为 *root* 节点的子树，将构造出社会

网络 G'_{sn} 所对应的新的层次化社区树 HOT′. 证毕.

定理 4. 假设社会网络 $G_{sn}=(V,E,W)$ 所对应的层次化社区树为 HOT，对于 $\forall v_d\in V$ 以及与 v_d 相连接的边集合 $\{e_d^1,\cdots,e_d^m\}(0\leq m\leq|E|)$，在 G_{sn} 的基础上删除节点 v_d 以及与其相连接的边集合 $\{e_d^1,\cdots,e_d^m\}$ 后所形成的社会网络 $G'_{sn}=G_{sn}-v_d-\{e_d^1,\cdots,e_d^m\}$ 所对应的层次化社区树为 HOT′，那么构造 HOT′只需要 HOT 的基础上进行如下操作：

(1) 如果 $m=0$ 时，HOT′=HOT；否则转入(2).

(2) 在 HOT 的第 1 层节点所对应的社区上搜索节点 v_d，如果对于 $\forall i\in\{1,\cdots,n'_1\}$ 都有 $v_d\notin V(C_i^1)$，那么有 HOT′=HOT 成立；否则(即 $\exists i\in\{1,\cdots,n'_1\}$ 使得 $v_d\in V(C_i^1)$)转入(3).

(3) 假设删除与 C_i^1 相对应的生成树(森林) T_i^1 中的节点 v_d 及其与 v_d 相连接的边后所形成的生成森林为 $T_i^{1\prime}=T_i^1-v_d-\{e_d^1,\cdots,e_d^m\}$，$C_i^1-\{e_d^1,\cdots,e_d^m\}$ 所包含的不饱和边集合为 E'_i. 以不饱和边集合 $E_{us}=E'_i$，生成森林集合 $T=T_i^{1\prime}$ 作为算法 P-SNCD 的输入，从其步骤(6～8)开始执行. 将新生成的社区 $C_1^{1\prime},\cdots,C_y^{1\prime}(y\leq|E'_i|)$ 及其派生社区按照层次结构构成 y 棵树形成的森林加入到 HOT 的 *root* 节点之下，删除 HOT 中以 L_i^1(对应社区 C_i^1)为 *root* 的子树.

证明. 对于 $\forall v_d\in V$，在删除节点 v_d 的同时也删除了与该节点相连接的所有边集合. 假设与节点 v_d 相连接的 $m(0\leq m\leq|E|)$ 条边形成的集合为 $\{e_d^1,\cdots,e_d^m\}$. 由算法 P-SNCD 生成社区的过程可知，社区生成过程是由低级别到高级别的过程，高级别社区所包含的节点集合是派生出这个社区的上一级社区所包含节点集合的子集. 因此，如果 v_d 出现在某个社区 $C_j^l(l\in\{1,\cdots,h\},j\in\{1,\cdots,n'_l\})$ 中，那么必定存在某个 1 级别社区 $C_i^1(1\leq i\leq n'_1)$，使得 $v_d\in V(C_i^1)$ 成立. 下面对是否存在这样的一个 1 级别社区 C_i^1 使得 $v_d\in V(C_i^1)$ 分两种情况讨论.

(1) 当被删除的节点 v_d 不在任何第 1 级别社区中出现($m=0$ 时属于这种情况)，即对于 $\forall i\in\{1,\cdots,n'_1\}$ 都有 $v_d\notin V(C_i^1)$ 成立，那么显然删除节点 v_d 不会对任何社区产生影响.

(2) 当被删除的节点 v_d 落在某个第 1 级别社区中，即 $\exists i\in\{1,\cdots,n'_1\}$ 使得 $v_d\in V(C_i^1)$ 成立. 由定义 2 和定义 3 可知，任何相同的节点不会出现在同一级别的两个不同社区中，因此在所有第 1 级别社区中，除社区 C_i^1 外的其它社区不会包含任意 $e\in\{e_d^1,\cdots,e_d^m\}$. 进一步，删除节点 v_d 不会对社区 C_i^1 及其派生社区外的任何其它社区结构产生影响. 假设删除生成树(森林) T_i^1 中的节点 v_d 及其与 v_d 相连接的边后所形成的生成森林为 $T_i^{1\prime}=T_i^1-v_d-\{e_d^1,\cdots,e_d^m\}$，$C_i^1$ 所包含的不饱和边集合为 E_i，$C_i^1-\{e_d^1,\cdots,e_d^m\}$ 所包含的不饱和边集合为 E'_i，那么必定有 $T_i^{1\prime}\subseteq T_i^1$，$E'_i\subseteq E_i$ 成立. 以不饱和边集合 $E_{us}=E'_i$，生成森林集合 $T_i^{1\prime}$ 作为算法 P-SNCD 的输入，从其步骤(6～8)开始执行将生成 $y(0\leq y\leq|E'_i|)$ 个第 1 级别社区 $C_1^{1\prime},\cdots,C_y^{1\prime}$ 以及这些社区所派生的高级别社区. 因为 $T_i^{1\prime}\subseteq T_i^1\cap E'_i\subseteq E_i$，所以必定有 $C_1^{1\prime}\cup C_2^{1\prime}\cup\cdots\cup C_y^{1\prime}\subseteq C_i^1$ 成立. 进一步，$C_1^{1\prime},\cdots,C_y^{1\prime}$ 不会与原来除 C_i^1 外的其它任何 1 级别社区存在交集. 因此，将新生成的社区 $C_1^{1\prime},\cdots,C_y^{1\prime}$ 及其派生社区按照层次结构构成 y 棵树形成的森林加入到 HOT 的 *root* 节点之下并删除以 L_i^1 为 *root* 节点的子树，将构造出社会网络 G'_{sn} 所对应的新的层次化社区树 HOT′. 证毕.

假设动态社会网络 G_{sn} 在 t_0 时刻为 $G_{sn}=(V,E,W)$，在 $t_1=t_0+\Delta t$ 时刻为 $G'_{sn}=(V',E',W')$，那么与 t_0 时刻相比，G'_{sn} 在 t_1 时刻新添加的节点集合 $V_a=V'-V$，被删除的节点集合 $V_d=V-V'$. 如果用 E_{Va} 表示在 G'_{sn} 中与节点集合 V_a 中任意节点有连接的边集合，用 E_{Vd} 表示在 G_{sn} 中与节点集合 V_d 中任意节点有连接的边集合，那么与 t_0 时刻相比，G'_{sn} 在 t_1 时刻新增加的边集合(不算增加节点集合 V_a 而增加的边) $E_a=E'-E-E_{Va}$，被删除的边集合(不算删除节点集合 V_d 而删除的边) $E_d=E-E'-E_{Vd}$. 算法 1 给出了动态社会网络社区发现算法 D-SNCD 的形式化描述. 算法 D-SNCD 运行时首先根据 t_0 时刻的 G_{sn} 和 t_1 时刻的 G'_{sn} 计算出新增的节点集合 V_a，新增的边集合 E_a，删除的节点集合 V_d，删除的边集合 E_d (对应步骤 1). 步骤 2～7 对结构可能发生变化的社区进行标识，对于不同标识的社区采用不同的计算方法. 由于定理 1～定理 4 保证了节点和边的变化对层次化社区树 HOT 结构影响的局部性，通过算法 D-SNCD 步骤 8～26，只对受节点和边的变化而使得结构受影响的第 1 级别社区及其后续派生社区重新计算，保证了算法本身的高效性. 由于整个算法执行过程不需重复计算与上一时刻相比社区结构不受影响的社区，因而 D-SNCD 算法可以看成对算法 P-SNCD 生成的 HOT 的一种高效更新算法.

算法 1. 动态社会网络社区发现算法D-SNCD.

Input: (a) Social network $G_{sn}=(V,E,W)$ at time t_0 and HOT;

(b) Social network $G'_{sn}=(V',E',W')$ at time $t_1=t_0+\Delta t$ and the level h

Output: The community set with the level below h in G'_{sn} and the new HOT′

1. $V_a \leftarrow V'-V$; $V_d \leftarrow V-V'$; $E_a \leftarrow E'-E-E_{Va}$; $E_d \leftarrow E-E'-E_{Vd}$;
2. for each $v_a \in V_a$, $v_d \in V_d$, $e_a \in E_a$, $e_d \in E_d$
3. for each $C \in C^1$ where C^1 is the 1st level community set of G_{sn} {
4. if $\exists e \in E_{Va} \cap \exists v \in C$ makes e connecting v_a and v {$C.tag \leftarrow C.tag \cup$ "Add-vertex"}
5. else if $v_d \in C$ {$C.tag \leftarrow C.tag \cup$ "Del-vertex"}
6. else if $e_d \in C$ {$C.tag \leftarrow C.tag \cup$ "Del-edge"}
7. else if e_a connects two vertexs of C {$C.tag \leftarrow C.tag \cup$ "Add-edge"}}
8. Let $C^1_{a1}, \cdots, C^1_{am}$ be the communities with $1 \le a1, a2, \cdots, am \le n_1$ and $\forall i \in \{a1, a2, \cdots, am\}$ $C^1_i \in C^1 \cap C^1_i.tag.contains$("Add-vertex"); Let E'_{us} be the unsaturated edge set of $C^1_{a1}, \cdots, C^1_{am}$; Let E'_a be the subset of E_a with each $e'_a \in E'_a$ connects two vertexs of $C^1_i (i \in \{a1, a2, \cdots, am\})$;
9. $T \leftarrow T^1_1 + V_a - V_d$; $E_{us} \leftarrow E'_{us} + E_{Va} + E'_a - E_{Vd} - E_d$;
10. Call P-SNCD from setp (6～8) with T and E_{us} as the input and generate the 1st level communities;
11. Let $C^1_{b1}, \cdots, C^1_{bn}$ be the communities with $1 \le b1, b2, \cdots, bn \le n_1$ and $\forall i \in \{b1, b2, \cdots, bn\}$ $C^1_i \in C^1 \cap C^1_i.tag.contains$("Del-vertex"); Let E'_{us} be the unsaturated edge set of $C^1_{b1}, \cdots, C^1_{bn}$; Let E'_a be the subset of E_a with each $e'_a \in E'_a$ connects two vertexs of $C^1_i (i \in \{b1, b2, \cdots, bn\})$;
12. $T \leftarrow T^1_1 - V_d$; $E_{us} \leftarrow E'_{us} + E'_a - E_{Vd} - E_d$;
13. Call P-SNCD from setp (6～8) with T and E_{us} as the input and generate the 1st level communities;
14. Let $C^1_{c1}, \cdots, C^1_{cp}$ be the communities with $1 \le c1, c2, \cdots, cp \le n_1$ and $\forall i \in \{c1, c2, \cdots, cp\}$ $C^1_i \in C^1 \cap C^1_i.tag.contains$("Del-edge"); Let E'_{us} be the unsaturated edge set of $C^1_{c1}, \cdots, C^1_{cp}$; Let E'_a be the subset of E_a with each $e'_a \in E'_a$ connects two vertexs of $C^1_i (i \in \{c1, c2, \cdots, cp\})$;
15. $T \leftarrow T^1_1$; $E_{us} \leftarrow E'_{us} + E'_a - E_d$;
16. Call P-SNCD from setp (6～8) with T and E_{us} as the input and generate the 1st level communities;
17. Let $C^1_{d1}, \cdots, C^1_{dq}$ be the communities with $1 \le d1, d2, \cdots, dq \le n_1$ and $\forall i \in \{d1, d2, \cdots, dq\}$ $C^1_i \in C^1 \cap C^1_i.tag.contains$("Add-edge"); Let E'_{us} be the unsaturated edge set of $C^1_{d1}, \cdots, C^1_{dq}$; Let E'_a be the subset of E_a with each $e'_a \in E'_a$ connects two vertexs of $C^1_i (i \in \{d1, d2, \cdots, dq\})$;
18. $T \leftarrow T^1_1$; $E_{us} \leftarrow E'_{us} + E'_a$;
19. Call P-SNCD from setp (6～8) with T and E_{us} as the input and generate the 1st level communities;
20. Let E'_a be the subset of E_a with each $e'_a \in E'_a$ connects two communities of C^1 and $C^1_{e1}, \cdots, C^1_{er}$ be the communites being connected by e'_a with $1 \le e1, e2, \cdots, er \le n_1$;
21. $T \leftarrow T^1_1$; $E_{us} \leftarrow E'_a$;
22. Call P-SNCD from setp (6～8) with T and E_{us} as the input and generate the 1st level communities;
23. Let $\{C^{1\prime}_1, C^{1\prime}_2, \cdots, C^{1\prime}_y\}$ be all of the 1st communities generated from step 10, 13, 16, 19, 22;
24. Delete nodes which correspond to the community of $\{C^1_{a1}, \cdots, C^1_{am}, C^1_{b1}, \cdots, C^1_{bn}, C^1_{c1}, \cdots, C^1_{cp}, C^1_{d1}, \cdots, C^1_{dq}, C^1_{e1}, \cdots, C^1_{er}\}$ from HOT;
25. Add nodes which correspond to the community of $\{C^{1\prime}_1, C^{1\prime}_2, \cdots, C^{1\prime}_y\}$ to the first level of HOT;
26. Call P-SNCD from step (6～8) with $C_r \in \{C^{1\prime}_1, C^{1\prime}_2, \cdots, C^{1\prime}_y\}$ and $l=1$ as the input and generate higher level communities.

算法时间复杂度分析

算法 D-SNCD 是对算法 P-SNCD 生成的层次化社区树 HOT 的一种动态更新算法. 在更新过程中对于与 t_i 时刻相比社区结构不会发生变化的社区, D-SNCD 算法在 t_{i+1} 时刻的计算任务上对 HOT 进行剪枝. 同时, 在 t_{i+1} 时刻对于 HOT 中需要重新计算的社区, D-SNCD 算法通过调用 P-SNCD 算法的方法进行. 显然 D-SNCD 的算法具有与 P-SNCD 算法相同的时间复杂度. 假设由社会网络形成的多重图 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 D-SNCD 的计算复杂度为 $O(\log n)$.

6 实验结果

针对 BSP 并行计算模型, 本文实现了 D-SNCD 算法, 并通过"DBLP"和"新浪微博"两个实际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 2.8 GHz, 内存 2 GB). 操作系统为 32 位 Windows XP 专业版, 并行环境为 MPICH2-1.1.1.

6.1 案例学习

DBLP 论文合著关系网络[①]是由德国特里尔(Trier)大学建立的一个计算机类期刊和会议论文集的数据库系统, 为用户提供权威的论文数据和方

① http://www.informatik.uni-trier.de/~ley/db/

便的查询服务.本文下载并解析该站点提供的在2010年7月份之前被40个与数据库系统、数据挖掘、数据流等领域相关的国际会议(包括sigmod、vldb、kdd、ckde等)所收录的论文49231篇,其中合著作者50396位.由上述合著作者之间形成的社会关系网络下面统称为合著关系网络.

为了对合著关系网络不同合著作者之间的权重进行合理赋值,本文将合著关系网络中作者之间的最终权值关系形成过程看作投票过程.每一篇合著论文相当于一轮投票,在任意一轮投票中,如果第i合著作者与论文的第1作者署名排序中靠得越近,则论文第i作者和第1作者之间的关系权重将获得较高的投票.另外,通过对合著关系网络进行统计发现,论文的合著作者数小于等于4的论文数占论文总数的88.5%.基于以上考虑,为了挖掘合著关系网络的社区及社区分布特性,表2分别给出了4种对合著关系网络中合著作者之间的关系权重赋值策略.

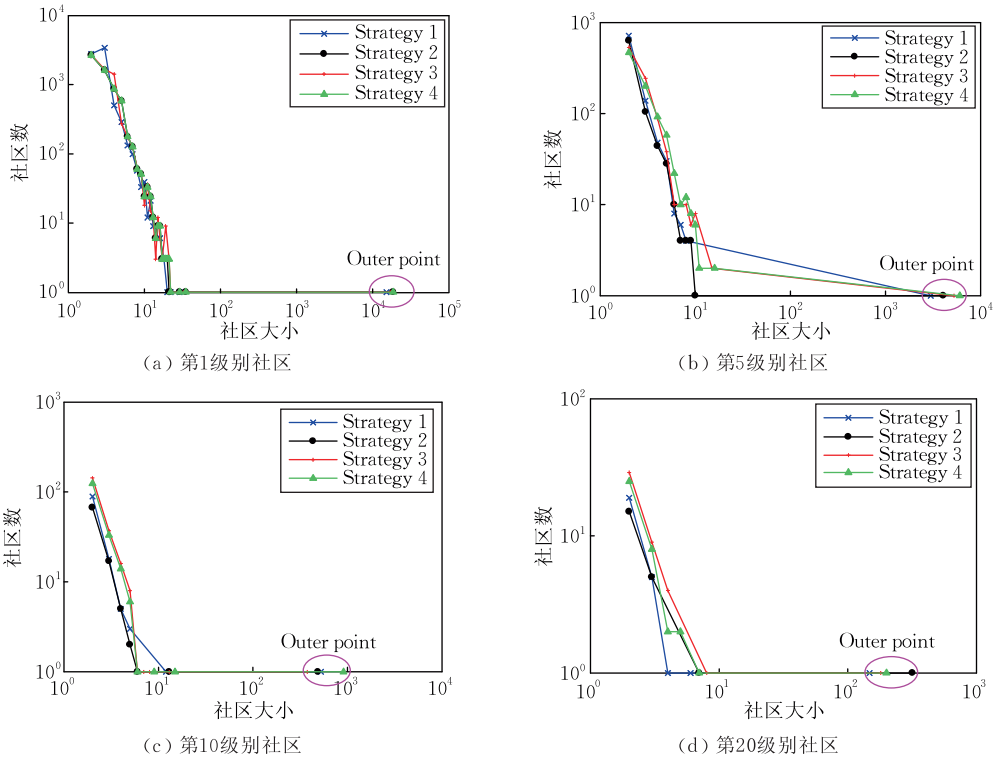
表2　不同策略下的权重赋值方案

Strategy	A1-B-A2	A1-B-A3	A1-B-A4	A2-B-A3	A2-B-A4	A3-B-A4	A1-B-OA
Strategy1	3	2	1	0	0	0	0
Strategy2	3	2	1	1	1	1	1
Strategy3	4	3	2	0	0	0	0
Strategy4	4	3	2	1	1	1	1

在表2中,Ai-B-Aj($1\leq i<j\leq 4$)对应的列为α($\alpha>Z^+$)表示对于DBLP数据集中任意一篇合著文章,如果文章的第j作者存在的话,第i作者与第j作者之间的关系权重值增加α;否者第i作者与第j作者之间的关系权重值维持不变(增加0).同样,A1-B-OA对应的列为β($\beta\in Z^+$)表示在合著作者个数大于等于5的情况下,第1作者与第5及其以后的合著作者之间的关系权重值增加β,否则第1作者与第5及其以后的合著作者之间的关系权重维持不变.由此可见,每一篇合著作者数大于1的文章都将使得这些合著作者之间的关系权重值增加.对于那些经常在一起合著论文的作者,通过采取表2中的关系权重赋值策略将会使得这些合著作者之间累计较高的关系权重.图5给出了采用D-SNCD算法分别在4种不同权重赋值策略下不同级别社区数量与社区大小的分布曲线.

从图5(a)～(d)不同策略下不同级别的社区数量与社区大小的分布曲线可以看出,采用D-SNCD算法得到的社区摆脱了传统的采用"模块度"定义存在内在的分辨率限制[24-25].如果不考虑社区规模最大的那个点(下面统称为Outer Point),在4种不同的权重赋值策略下,不同级别的社区大小与社区数量成明显的幂律分布,其中第1级别社区的Strategy1(图5(a))和第10级别社区的Strategy2(图5(c))表现得最为明显,幂指数分别为2.87和2.15.另外在4种不同策略下,其它不同级别的社区数量和社区大小也表现出这种相同的分布特性(因篇幅有限,未在这里一一给出).在每种策略下,不同级别的社区都包含一个作者数目比较大的社区(对应图5的Outer Point),而且这个最大社区所包含的作者数随着社区级别的上升而不断减少.比如在4种不同策略下第1级别,第5级别,第10级别,第20级别的最大社区平均所包含的作者个数分别为16104、4065、487、148.出现这种现象的原因有两个:首先,算法运行过程就是寻找包含指定数量的生成树(这里的生成树数量就是社区的级别)的最大子图过程,这使得算法在开始时将一些没有直接合作关系但是有间接合作关系的作者划分到同一个社区中,并且随着社区级别的上升,算法最终倾向于将具有直接合作且合作关系紧密的作者划分到同一个社区中;其次,在处理DBLP数据集时由于没有进行重名处理,这使得原本独立的多个社区因为存在同名作者而被合并成一个,并且随着计算的进行这种现象出现"叠加效应",最后形成个别规模极大的社区.另外,通过对图5(a)～(d)所示的不同级别社区中的合著作者进行统计分析发现,绝大部分社区中的合作作者数不超过6位.在图5所示的4种不同权重赋值下第20级别社区中,作者数量不超过6位的社区数占该级别社区总数的95.2%,并且这个比值随着社区级别的上升而略有增加.只有少数的社区拥有10位以上的作者,平均约占整个社区数目的1.8%,并且这个比值随着社区级别的上升而略有下降.这表明大多数作者都是以较小团队形式进行研究,并且团队成员之间合作较为紧密,而大团队"集体作战"的情况较少且大团队合作成员之间与小团队合作成员相比合作紧密程度较低.随着社区级别的上升,属于同一社区的作者之间将拥有更加紧密的合作关系.表3给出了在Strategy2和Strategy3

权重赋值方案下第 50 级别的所有社区(在 Strategy2 方案下的社区数量为 10 个,在 Strategy3 方案下的社区数量为 13 个)以及每个社区中的成员名单,其中同级别的相邻社区用不同颜色进行区分.



(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

图 5 在不同级别社区中的社区数量与社区大小分布

表 3 不同策略下的第 50 级别社区及其社区成员

Strategy2	Strategy3
Philip S. Yu, Ming-Syan Chen, Kun-Lung Wu, Charu C. Aggarwal, Bugra Gedik; Yufei Tao, Dimitris Papadias; Nikos Mamoulis, Man Lung Yiu; Wei Wang, Jiong Yang; Vivek R. Narasayya, Surajit Chaudhuri; Ramakrishnan Srikant, Rakesh Agrawal; Hans-Peter Kriegel, Peer Kröger; Lei Chen, Xiang Lian; Amr El Abbadi, Divyakant Agrawal; Shoji Hirano, Shusaku Tsumoto;	Philip S. Yu, Hong Cheng, Charu C. Aggarwal, Xiaoxin Yin, Jiawei Han, Ming-Syan Chen, Bugra Gedik, Dong Xin, Kun-Lung Wu, Jianyong Wang, Haixun Wang, Jian Pei, Wei Fan, Xifeng Yan; Nikos Mamoulis, Man Lung Yiu, Xiaokui Xiao, Yufei Tao, Dimitris Papadias; Ramakrishnan Srikant, Rakesh Agrawal; Nicolas Bruno, Vivek R. Naras ayya, Surajit Chaudhuri; Flip Korn, S. Muthukrishnan, Graham Cormode; Wei Wang, Jiong Yang; Christian Böhm, Elke Achtert, Peer Kröger, Hans-Peter Kriegel; Xiang Lian, Lei Chen; Amr El Abbadi, Divyakant Agrawal; Wen-Syan Li, K. Selçuk Candan; Jun Yan, Ning Liu; Shusaku Tsumoto, Shoji Hirano; Jim Melton, Andrew Eisenberg;

图 6 给出了在不同策略下,合著关系网络的不同级别社区所包含的社区数量分布曲线.由图可以看出,在 4 种不同的策略中,采取 Strategy4 所生成的社区级别跨度最大,其最高级别社区为 86;而采取 Strategy1 所生成的社区级别跨度最下,其最高级别社区为 67.这是因为在相同作者数目的合著网络中,采用 Strategy4 关系权重赋值方案使得这个合著网络中作者之间的关系表现得更加紧密(拥有权值更大的边),从而在生成的社区上表现出某些作者将同时出现在更高级别的社区中.另外,通过图 6 还可以看出在这个合著关系网络中,相同级别的社区数量随着社区级别的上升而递减并且大概在第 5 级别左右出现急剧下降,整个社区级别与社区数量关系曲线表现出明显的"长尾现象",这种特性也同时出现在很多其它类似的复杂系统中[27].另外从图 6 还可以看出,在 4 种不同权值赋值策略下所形成的曲线图前端都有一个先递增然后递减的"拐点",这个拐点大多出现在第 2 级别和第 4 级别之间.目前对于这种现象我们还不能给出很好的解释,造成这种现象的一个可能原因是在合著关系网络

中，一方面同一作者在不同论文中的署名的采用不同写法(国外作者名字很多时候会有名称的全拼写法和简写写法两种方式，如 Rodney C. Wolff 可能简写为 Rodney Wolff)将本属于同一社区的作者进行了“分割”，从而使得社区数量出现短暂的上升，另一方面如前面所述，随着算法的继续运行，极少数同名作者造成的“叠加效应”合并了一些原本独立的社区.

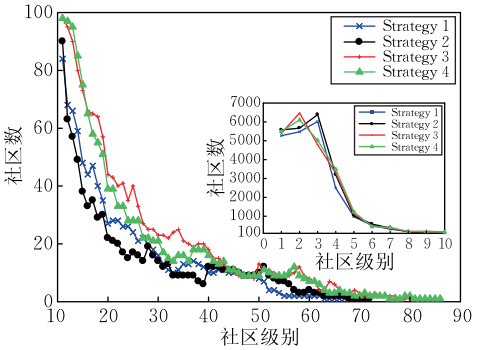


图 6　不同策略下的不同级别社区包含的社区数量分布

6.2　算法性能

新浪微博①是目前国内比较流行的一个社交网络平台. 采用本课题组开发的“YHPODS 银河博思网络舆情管理平台”对新浪微博进行解析. 从 2010.5.1～2010.5.3，历时 3 天，共采集 ID(帐号)80 多万个，“关注关系“和“粉丝关系”380 多万条. 在该社交网络中，对于任意的两个博主 A 和博主 B，假设博主 A 关注博主 B，那么博主 A 和博主 B 之间的增加一条权值为 1 的边. 如果博主 A 在关注博主 B 的同时，博主 B 也关注博主 A(即博主 A 和博主 B 互为粉丝关系)，那么博主 A 和博主 B 之间的权值在原来的基础上增加权值 w_2(本文取 $w_2=5$).

为了测试 D-SNCD 算法性能，利用“YHPODS 银河博思网络舆情管理平台”[10]对新浪微博关系网络 G_{s0} 每天进行一次快照(实际上就是控制网页爬虫重新爬取原来的新浪微博关系网络，为了保证快照的及时性，我们在工作主机数量和线程数量两个方面增加了爬虫的并行性.)，时间从 2010.5.4～2010.5.9 历时 6 天. 由于在 t_0 时刻的新浪微博关系网络中的部分好友关系或者粉丝关系随着时间的推移有可能发生变化，因而通过上述过程得到在 t_1，t_2，t_3，t_4，t_5，t_6 6 个时刻与原来(t_0 时刻)的新浪微博关系网络相关的不同时期的新浪微博关系网络 G_{s1}，G_{s2}，G_{s3}，G_{s4}，G_{s5}，G_{s6}. 图 7 给出了分别对 G_{s0}，G_{s1}，G_{s2}，G_{s3}，G_{s4}，G_{s5}，G_{s6} 采用 P-SNCD 算法和 D-SNCD算法，以及对G_{s0}，G_{s2}，G_{s4}，G_{s6}采用 D-SNCD 算法在 16 台计算机上并行处理时间关系曲线. 由图 7 可以看出在处理 G_{s0} 时，P-SNCD 算法和 D-SNCD 算法运行时间一样，这是因为 D-SNCD 算法在处理每一个快照时都要以已经处理过的最近快照为参考，对于 t_0 时刻的新浪微博关系网络 G_{s0}，D-SNCD 算法需要直接调用 P-SNCD 算法进行计算，以获得 G_{s0} 对应的层次化社区树 HOT 相关信息. 另外，在时间间隔为 1 天($interval=1$ day)的情况下，采用 D-SNCD 算法对连续 6 天的新浪微博关系网络(G_{s0}～G_{s6})进行计算比直接使用 P-SNCD 算法在运算时间上减少 60.9%，而在时间间隔为 2 天($interval=2$ days)的情况下，采用 D-SNCD 算法计算 6 天内的新浪微博关系网络(G_{s0}，G_{s2}，G_{s4}，G_{s6})比直接使用 P-SNCD 算法在运算时间上减少 44.1%，这充分说明了采用 D-SNCD 算法对 HOT 进行动态更新的有效性. 对于新浪微博关系网络采用 D-SNCD 算法运行时间随着间隔天数的拉长而增加的原因是：随着间隔时间的增加，在 t_i($1\leq i\leq 6$)时刻的新浪微博关系网络 G_{si} 发生动态变化的节点集合和边集合将逐渐布满了 t_{i-1} 时刻 $G_{s(i-1)}$ 对应的 HOT 的第 1 级别社区的各个社区，因而 HOT 中需要更新的分支也随之增加. 随着间隔时间的不断拉长，对于连续时间间隔的新浪微博关系网络，采用 D-SNCD 算法的运行时间最终趋向与直接采用 P-SNCD 算法的运算时间. 另外在计算相同的数据集时，分别采用 P-SNCD 算法和 D-SNCD 算法得到的结果完全一致，这进一步表明了 D-SNCD 算法的正确性.

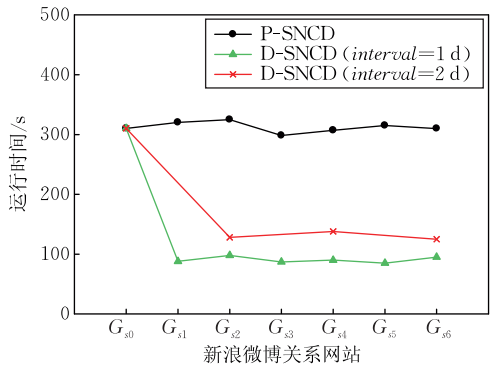


图 7　P-SNCD 算法和 D-SNCD 算法性能比较

7　结束语

社区发现是社会网络研究领域的一个重要研

① http://t.sina.com.cn/

究方向.针对现有的社区发现方法存在的弊端,本文在层次静态社区并行分解算法的基础上进一步提出了一种新型的层次化社区发现方法 D-SNCD.D-SNCD算法是对 P-SNCD 算法生成的 HOT 分支进行选择性的更新算法,能够并行高效地处理大规模动态社会网络.严格的数学证明与充分的实验数据保证了算法的正确性和有效性.

参 考 文 献

[1] Yang Bo, Liu Da-You, Liu Jiming, Jin Di, Ma Hai-Bin. Complex network clustering algorithms. Journal of Software, 2009, 20(1): 54-66(in Chinese)
(杨博, 刘大有, Liu Jiming, 金弟, 马海宾. 复杂网络聚类方法. 软件学报, 2009, 20(1): 54-66)

[2] Newman M E J. Modularity and communities structure in networks. Proceedings of the National Academy of Science, 2006, 103(23): 8577-8582

[3] Shiga M, Takigawa I, Mamitsuka H. A spectral clustering approach to optimally combining numerical vectors with a modular network//Berkhin P, Caruana R, Wu X eds. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2007: 647-656

[4] Newman M E J. Detecting community structure in networks. European Physical Journal (B), 2004, 38(2): 321-330

[5] Newman M E J. Fast algorithm for detecting community structure in networks. Physical Review E, 2004, 69(6): 066133

[6] Guimera R, Amaral L A N. Functional cartography of complex metabolic networks. Nature, 2005, 433(7028): 895-900

[7] Gan Wen-Yan, He Nan, Li De-Yi, Wang Jian-Min. Community discovery method in networks based on topological potential. Journal of Software, 2009, 20(8): 2241-2254(in Chinese)
(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)

[8] He Dong-Xiao, Zhou Xu, Wang Zuo, Zhou Chun-Guang, Wang Zhe, Jin Di. Community mining in complex networks—Clustering combination based genetic algorithm. Acta Automatic Sinica, 2010, 36(8): 1160-1170(in Chinese)
(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)

[9] Shen Hua-Wei, Cheng Xue-Qi, Chen Hai-Qiang, Liu Yue. Information bottleneck based community detection in network. Chinese Journal of Computers, 2008, 31(4): 677-686(in Chinese)
(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): 677-686)

[10] Lin Wang-Qun, Lu Feng-Shun, Ding Zhao-Yun, Wu Quan-Yuan, Zhou Bin, Jia Yan. Parallel computing hierarchcal community approach based on weighted-graph. Journal of Software, 2012, 23(6): 1517-1530(in Chinese)
(林旺群, 卢风顺, 丁兆云, 吴泉源, 周斌, 贾焰. 基于带权图的层次化社区并行计算方法. 软件学报, 2012, 23(6): 1517-1530)

[11] Tyler J R, Wilkinson D M, Huberman B A. Email as spectroscopy: Automated discovery of community structure within organizations//Huysman M, Wenger E, Wulf V eds. Proceedings of the 1st International Conference on Communities and Technologies. Dordrecht: Kluwer Academic Publishers, 2003: 81-96

[12] Flake G W, Lawrence S, Giles C L, Coetzee F M. Self-organization and identification of Web communities. IEEE Computer, 2002, 35(3): 66-71

[13] Wu F, Huberman B A. Finding communities in linear time. A physics approach. European Physical Journal B, 2004, 38(2): 331-338

[14] 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

[15] Yang B, Cheung W K, Liu J. Community mining from signed social networks. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(10): 1333-1348

[16] Zhang Y Z, Wang J Y, Wang Y, Zhou L Z. Parallel community detection on large networks with propinquity dynamics//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France: ACM, 2009: 997-1005

[17] Wasserman S, Faust K. Social Network Analysis. Cambridge: Cambridge University Press, 1994

[18] Pons P, Latapy M. Computing communities in large networks using random walks//Yolum P ed. Proceedings of the 20th International Symposium on Computer and Information Sciences. Berlin: Springer-Verlag, 2005: 284-293

[19] Yang B, Liu J. Discovering global network communities based on local centralities. ACM Transactions on the Web, 2008, 2(1): article 9: 1-32

[20] Zhang C-Q, Ou Y. Clustering, community partition and disjoint spanning trees. ACM Transactions on Algorithms, 2008, 4(3): 35

[21] Lin Yu-Ru, Sun Jimeng, Castro Paul, Konuru Ravi, Sundaram Hari, Kelliher Aisling. Parallel community detection on large networks with propinquity dynamics//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France: ACM, 2009: 527-536

[22] Kim Min-Soo, Han Jiawei. A particle-and-density based evolutionary clustering method for dynamic networks. PVLDB, 2009, 2(1): 622-633

[23] Lin Y-R, Chi Y, Zhu S, Sundaram H, Tseng B L. Facetnet: A framework for analyzing communities and their evolutions in dynamic networks//Proceedings of the 17th Interna-

tional Conference on World Wide Web. Beijing, China, 2008: 685-694

[24] Fortunato S, Barthelemy M. Resolution limit in community detection. Proceedings of the National Academy of Sciences of the United States of America, 2007, 104(1): 36-41

[25] Newman M E J, Girvan M. Finding and evaluating community structure in networks. Physical Review E, 2004, 69(2): 026113

[26] Chen Guo-Liang. Parallel Computing—Structure, Algorithm, Programming. Beijing: Higher Education Press, 2003 (in Chinese)
(陈国良. 并行计算——结构、算法、编程. 北京：高等教育出版社，2003)

[27] Kwak H, Lee C, Park H, Moon S. What is Twitter, a social network or a news media?//Proceedings of the 19th International Conference on World Wide Web. Raleigh, NC, USA, 2010: 591-600

LIN Wang-Qun, born in 1983, Ph. D. candidate. His research interests include social network and parallel computing.

DENG Lei, born in 1984, Ph. D. His research interests include social network analysis.

DING Zhao-Yun, born in 1983, Ph. D. His research interests include social network analysis.

WU Quan-Yuan, born in 1941, professor, Ph. D. supervisor. His research interests include social network analysis, data mining, parallel computing.

JIA Yan, born in 1960, professor, Ph. D. supervisor. Her research interests include parallel computing, data mining and social network analysis.

ZHOU Bin, born in 1971, professor, Ph. D. supervisor. His research interests include parallel computing and data mining.

Background

Social network analysis has emerged as one of the most important techniques for modern sociology research. It has been gaining a comprehensive attentions from many research fields including anthropology, biology, communication studies, economics, geography, information science, organizational studies, social psychology, and sociolinguistics. This makes social network analysis becomes a popular topic recently.

Community discovery is one of the most important steps for understanding the inner structures of social networks. Hence, an effective and efficient community discovery algorithm is quite challenging yet much desired. In this paper, we propose a multi-graph based hierarchical community detection approach, which defines community structure through graph partition. With the pre-defined structure, a novel algorithm for Dynamic Social Network Community Discovery (D-SNCD) is proposed. D-SNCD make full use of characteristics of dynamic social networks and update the Hierarchical cOmmunity Tree (HOT) in a timely and choicely way. Moreover, D-SNCD requires few inputs and easy to control for end users. The effectiveness and efficiency of our proposed algorithm is guaranteed by the strictly mathematical proofs and sufficiently experimental studies.

The work of this paper is supported by the National Natural Science Foundation of China under Grant Nos. 60933005, 60873204 and National High Technology Research and Development Program (863 Program) of China under Grant No. 2010AA012505.