

基于模块密度优化的标签传播社区发现算法^①

陈建军, 叶东毅

(福州大学 数学与计算机科学学院, 福州 350108)

摘要: 基于标签传播的社区发现算法(LPA)以其简单高效得到了广泛的研究, 然而当社区结构模糊时, LPA 得到的是一个单一的社区, 这是无意义的. 模块化标签传播算法(LPAm)则倾向于将网络划分为度数相近的社区且存在解极限问题. 为此提出基于模块密度的标签传播(LPAd)算法, 该算法通过对模块密度优化进行标签标记和传播, 以避免过大社区的形成, 且生成的社区满足 Radicchi 等人提出的弱社区定义. 多个真实数据集和人工网络数据的实验结果表明, 本文算法在不改变算法复杂度的情况下提高了所发现社区的质量, 与现有的若干基于标签传播的社区发现算法相比, 取得了改进的效果.

关键词: 社区发现; 标签传播; 模块密度; 复杂网络

Label Propagation Community Detection Algorithm Based on Modularity Density Optimization

CHEN Jian-Jun, YE Dong-Yi

(College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China)

Abstract: The simplicity and efficiency of the community detection algorithm based on label propagation (LPA) have been studied extensively, but when the community structure is not clear, a single community is obtained through the LPA, which is meaningless. Modularity-specialized label propagation algorithm (LPAm) tends to partition the network into communities with similar degrees and the problems of solving the limit of functions exist. Therefore, this paper points out label propagation algorithm based on modularity density optimization (LPAd), in order to avoid the formation of large communities, and the community meets the weak community definition proposed by Radicchi et al. Several real datasets and artificial network data experimental results show that, this algorithm raises the quality of the detected community without changing the algorithm complexity, and compared with the existing number of community detection algorithm based on label propagation, it has been improved effectively.

Key words: community detection; label propagation; modularity density; complex network

1 引言

网络的研究起源于瑞士数学家欧拉在 1736 年发表的一篇文章, 该文章讨论了哥尼斯堡(Konigsberg)七桥问题. 随后, 网络理论的研究经历了从规则网络到随机网络再到复杂网络三个阶段. 现实世界中的许多系统, 如信息网络、交通网络、社会网络和生物网络等, 都可以使用网络来描述. 这些网络通常具有很高的复杂性, 被称为复杂网络. 不同领域的许多学者以不同的视角对复杂网络展开了研究, 并取得了丰富的

成果. 1998 年, Watts 和 Strogatz 提出了 WS 小世界模型, 指出现实世界中的网络具有大的凝聚系数和短的平均路径长度的小世界特性^[1]. 1999 年, Barabasi 和 Albert 提出无尺度网络模型, 指出网络中节点度的分布服从幂律分布^[2]. 2002 年, Givan 和 Newman 指出复杂网络中普遍存在网络簇结构, 也被称为网络社区结构^[3]. 关于社区结构的定义, 在相关的研究中还没有得到普遍的认可, 但是通常认为社区结构具有这样的特点: 社区内的节点连接紧密, 而社区之间的节点连接稀疏^[4].

^① 基金项目:国家自然科学基金(71231003)

收稿时间:2014-06-28;收到修改稿时间:2014-09-02

对于复杂网络社区结构的研究,不同领域的学者从各自的研究角度出发,提出了许多社区发现的技术和方法.当前,社区发现已成为复杂网络研究的一个热点课题.

Raghavan 和 Albert 于 2007 提出了基于标签传播 (Label Propagation) 的社区发现算法 (简称 LPA)^[5], 该算法计算简单、易于实现, 且不需要事先知道网络中有多少个社区、社区规模如何, 能够在接近线性时间内查找出网络中的社区结构, 因此 LPA 受到越来越多的重视. LPA 也存在一个明显的不足, 即: 只有在网络的社区结构明显时, LPA 才能够准确快速地找出其社区结构, 而当社区结构模糊时, LPA 得到的结果往往是将所有节点视为相同的社区, 这就失去社区发现的意义, 算法失效. 针对社区结构模糊时 LPA 存在的这一问题, Barber 和 Clark 在 LPA 的更新策略中引入了一些限制条件来防止大社区的形成, 提出了模块化标签传播算法^[6] (LPAm), 将社区发现问题转化为模块度的优化问题. LPAm 虽然解决了形成大社区的问题, 但是也带来也新的问题, 即产生大量小社区. Liu 和 Murata 指出 LPAm 容易陷入局部极值, 结合多步贪婪凝聚算法 (multistep greedy agglomerative, MSG)^[7] 提出了 LPAm+ 算法^[8], LPAm+ 对 LPAm 产生的结果做进一步的处理, 合并那些可以使模块度增加的社区, 合并后继续使用 LPAm 进行优化直到不存在那样的社区为止. 显然, LPAm+ 得到社区划分的模块度是大于等于 LPAm 的, 不过其代价是大大增加了算法的复杂度. 赵卓翔等人指出 LPA 在标签传播的过程仅考虑了邻接节点中标签的个数, 而没有考虑节点间联系的紧密程度, 将基于无权图的 LPA 应用到加权图中, 选取种子集合并赋予集合中的节点唯一的标签, 同时按照节点度从大到小进行标签的更新, 提出基于标签影响值 (label-influence-based, LIB) 的社区发现算法^[9]. LIB 在计算标签影响值时需要用到边的权值, 而网络中边的权值往往很难确定. Zhao Y 等人认为 LPA 节点标签更新顺序的随机性带来了算法结果的不稳定性, 甚至得到不合理的社区结构, 提出了基于熵顺序的社区发现算法 (LPA-E)^[10]. LPA-E 相比 LPA 提高了算法的准确性和稳定性, 其代价是增加了排序的计算量, 不过仍无法解决社区结构模糊时算法失效的问题.

LPAm 以模块度为优化目标, 而文献[11]指出基于模块度优化的方法用于检测小于一定尺度的网络的

社区结构时会失败, 该尺度依赖于网络中边的数目和模块之间的关联性. 针对这问题, Zhenping Li 等人提出了一种新的衡量社区划分质量的函数, 称为模块密度 (modularity desity)^[12]. 本文借鉴 LPAm 优化模块度的思想, 提出对网络的模块密度进行优化的算法 LPAd, 实现网络社区结构的有效发现, 取得了优于 LPA、LPAm 等算法的社区划分质量.

2 基于模块密度优化的标签传播社区发现算法

2.1 模块密度

给定一个网络 $G = (V, E)$, V 为顶点集, E 为边集, A 为网络 G 的邻接矩阵. 假设 V_1 和 V_2 是 V 的两个互不相交的子集, 记

$$L(V_1, V_2) = \sum_{i \in V_1, j \in V_2} A_{ij}$$

网络 G 的一个划分: $G_1(V_1, E_1), \dots, G_m(V_m, E_m)$. 子图 $G_i(V_i, E_i)$ 的平均模块度定义如下:

$$d(G_i) = d_{in}(G_i) - d_{out}(G_i)$$

其中, $d_{in}(G_i)$ 表示子图 G_i 的平均内部度, $d_{out}(G_i)$ 表示子图 G_i 的平均外部度. 子图 G_i 的平均模块度可以表示为:

$$d(G_i) = \frac{L(V_i, V_i) - L(V_i, \bar{V}_i)}{|V_i|}$$

根据“社区内的节点连接紧密, 而社区之间的节点连接稀疏”这一特点, 对于一个有意义社区来说, 它的平均模块度越大越好. 一个划分的模块密度定义为其子图的平均模块度之和, 因此, 模块密度 D 可以表示为:

$$D = \sum_{i=1}^m d(G_i) = \sum_{i=1}^m \frac{L(V_i, V_i) - L(V_i, \bar{V}_i)}{|V_i|}$$

D 的值越大, 划分就越好. 因此, 社区发现问题可以转换为模块密度 D 的优化问题.

2.2 本文算法 LPAd

2.2.1 算法思想

LPA 在每次更新迭代过程中, 节点的标签更新为邻接节点中标签数最多的那个标签. 当社区结构明显时, 对于任意节点 x (记其实际所属的社区为 L), 则 x 的标签更新为社区 L 内节点 (记为 y) 的标签的概率更大, 同时与 x 、 y 相连的节点 z 属于社区 L 的概率偏大,

z 的标签更新为 x 、 y 的标签; 若 y 与 x 所属不同的社区, 同时与 x 、 y 相连的节点 z 属于 L 的概率偏小, 甚至不存在这样的节点, 这样就减少了 x 标签的错误更新带来的影响. 以此类推, 属于同一社区的节点最终会得到相同的标签. 而当社区结构不明显时, x 的标签更新为非社区 L 内节点的标签(记为 y)的概率更大, 同时与 x 、 y 相连的节点但又与 x 、 y 处于不同社区的节点也偏多, 这就使得 x 标签的错误更新带来的影响越来越大, 最终导致算法只得到一个社区.

模块密度定义为子图的平均模块度之和, 子图的平均模块度的大小与子图内部边的数量和子图节点数相关. 在 LPAd 的运算过程中, 对任意节点 x 进行标签的更新, 假设 x 的邻接节点中有 k 种标签, 标签为 i 的节点个数 N_i , 标签 i 表示的社区的大小 C_i . 将 x 的标签更新为 i 后, 网络的模块密度的增量与 N_i 成正比, 与 C_i 成反比. 对于 x 节点的任意两个标签 i 和 j , 当社区大小 C_i 和 C_j 相差不大时, 选择更新为标签数多的那个标签, 这与 LPA 的更新策略相同; 而当 N_i 与 N_j 相差不大时, 选择更新为小社区的标签. 因此, LPAd 对节点标签的更新是在相同标签个数与标签所表示社区的大小之间进行折中选择, 可以避免大社区的形成.

2.2.2 算法步骤

1) 初始标签设置: 与 LPA 一样, 给网络中的所有节点一个唯一的标签, 该标签表示节点所属的社区. LIB 在标签初始化时, 并不是所有节点都赋予标签, 而是通过选取种子节点集合, 只给种子节点集合中的节点赋予唯一的标签. 然而实验结果显示, 该方法并不是对所有网络都有效, 因此本算法还是采用原始的赋值方式.

2) 节点更新顺序: 按随机顺序进行更新. LPA 算法的不稳定性主要是由于节点标签更新顺序的随机性带来的. 而对于基于目标函数优化的标签传播算法, 顺序指定的方式会使算法陷入同一局部极值点. 随机顺序虽然可能使算法达到更差的极值点, 但也可能达到好的极值点. 因此, 本算法对节点标签的更新采用随机顺序.

3) 节点标签更新: 假设对节点 v_i 的标签进行更新, 首先忽略 v_i 的标签, 对 v_i 的邻接节点中出现的每一个标签 l 计算将 v_i 的标签更新为 l 后模块密度 D 的增量 Δ_l , Δ_l 的计算公式如下:

$$\Delta_l = \frac{(L(V_i, V_i) + 2N_l) - (L(V_i, \bar{V}_i) + k_i - 2N_l)}{|V_i| + 1} - \frac{L(V_i, V_i) - L(V_i, \bar{V}_i)}{|V_i|}$$

其中, N_l 表示邻接节点中标签 l 的个数, k_i 表示节点 v_i 的度数. 计算完所有增量后, v_i 的标签更新为:

$$label(v_i) = \arg \max_l \Delta_l$$

当最大增量有多个时, 随机取一个标签. 在文献 [13] 中, Radicchi 等人给出了社区结构的两种量的定义, 为了使得到的社区结构满足其弱社区结构的定义, 模块密度 D 的每一个分量 $d(G_i)$ 应为正数. 因此若 v_i 原本所属的社区的模块度小于零, 则必改变 v_i 的标签.

如图 1 所示, x 为将要进行标签更新的节点, 其标签为 3, 它的 6 个邻接节点中标签 1 的个数为 3、标签 2 的个数为 1、标签 3 的个数为 2. 记 A_i 为社区 i 节点的内部度之和, B_i 为社区 i 节点的外部度之和, C_i 为社区 i 节点的个数, $N(x, i)$ 为 x 的邻接节点中标签为 i 的节点个数. 此时有:

$$\Delta_1 = \frac{(A_1 + 2N(x,1)) - (B_1 + k_x - 2N(x,1))}{C_1 + 1} - \frac{A_1 - B_1}{C_1}$$

$$\Delta_2 = \frac{(A_2 + 2N(x,2)) - (B_2 + k_x - 2N(x,2))}{C_2 + 1} - \frac{A_2 - B_2}{C_2}$$

$$\Delta_3 = \frac{A_3 - B_3}{C_3} - \frac{(A_3 - 2N(x,3)) - (B_3 - (k_x - 2N(x,3)))}{C_3 - 1}$$

若 Δ_3 的前半部分小于零, 则只考虑将 x 的标签更新为 1 或者 2, 否则对比三个增量大小来决定 x 的标签.

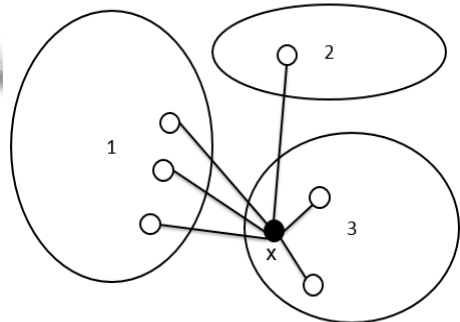


图 1 节点标签更新

4) 终止条件检查: 检查是否有节点的标签需要进行更新, 若都不需要, 则算法结束, 否则继续第 2 步.

算法结束后, 将所有具有相同标签的节点归为一个社区, 一个标签对应一个社区结构, 且所有社区结构都满足弱社区结构定义.

2.2.3 算法复杂度

LPA_d 与 LPA 相比, 差别仅在节点标签的更新上. 在更新时, LPA 仅需要计算邻接节点相同标签的个数, 而 LPA_d 还需要一些额外的计算. 但是这些计算所用到的变量都可以在节点标签更新时, 边更新边维护, 可以直接使用, 因此 LPA_d 与 LPA 一样, 可以在接近线性的时间复杂度内找到网络的社区结构.

3 实验

3.1 评价函数

模块度 Q 是最受广泛使用的衡量社区划分质量的函数, 它是由 Newman 和 Givan^[14]提出的. Q 的值等于网络中社区内部边所占的比例减去另一个随机网络 (节点的度保持不变, 边随机连接) 中社区内部边所占比例. Q 可以表示为:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(g_i, g_j)$$

其中, m 表示网络中的边数; 当节点 i 与节点 j 相连时, A_{ij} 等于 1, 反之为 0; k_i 表示节点 i 的度数; g_i 表示节点 i 所属社区; 当 $g_i = g_j$ 时, $\delta(g_i, g_j) = 1$, 否则, $\delta(g_i, g_j) = 0$. 当社区结构明显时, Q 接近 1; 当社区结构不明显时, Q 接近 0.

3.2 实验数据

在实验中选取了 6 个真实网络的数据集和一个人工生成的网络.

1) 真实网络数据集

选取了 6 个真实网络数据集, 分别为 karate^[15]、football^[16]、dolphins^[17]、lesmis^[18]、polbooks^[19]、sawmill^[20]. karate 数据集是一个由 34 个空手道俱乐部成员之间的关系构成的社会网络, 边表示两成员经常出现在俱乐部活动之外的其他场合, 由于出现了分歧, 俱乐部分为 2 个派系; football 数据集是美国橄榄球联赛中队伍的比赛关系构成的网络, 其中 115 支球队被分为 12 个小组, 因此可以认为该网络具有 12 个社区结构. dolphins 数据集由新西兰的一个海豚群体组成的海豚关系网, 边表示两个海豚之间接触频繁, 该海豚群体包含 2 个家族. Lesmis 数据集由小说悲惨世界中的人物关系网络构成, 网络中有 6 个主要人物, 边表示人物同时出现在一幕或多幕中; polbooks 数据集是由 Amazon 上销售的美国政治相关书籍页面上建立起

来的网络, 边表示一定数量的读者同时购买了这两本书, 书分为“自由派”、“保守派”和“中间派”3 类; sawmill 数据集为 sawmill 公司的通信网络构成, 边表示两者之间通信频繁. 6 个真实网络的基本信息如表 1 所示.

表 1 数据集基本信息

数据集	节点数	边数
karate	34	78
football	115	616
dolphins	62	159
lesmis	77	508
polbooks	105	411
sawmill	36	62

2) 人工生成的网络

LFR 网络^[21]是一种接近真实世界复杂网络的人工网络, 该网络节点的度分布和社区大小的分布均服从幂律分布. LFR 网络通过下面 8 个参数生成: 节点个数 N , 节点的平均度 k , 节点最大度 $kmax$, 节点度幂律分布指数 $t1$, 最小社区节点数 $cmin$, 最大社区节点数 $cmax$, 社区节点数幂律分布指数 $t2$, 混合参数 z . 其中 z 定义为所有节点与其他社区的边数除以节点的度数的平均值. 本文用于测试的 LFR 网络参数设置如下: $N=500$ 、 $k=5$ 、 $kmax=10$ 、 $t1=2$ 、 $cmin=5$ 、 $cmax=100$ 、 $t2=2$ 、 z 从 0.1 到 0.6 每次加 0.05, 一共生成 11 个 LFR 网络.

3.3 实验结果与分析

图 2 为 karate 的拓扑结构以及 LPA_d 对 karate 网络的划分结果. 在大多数情况下, LPA_d 将 karate 网络分为 A、B、C 和 D 四个部分, 而 karate 网络的实际划分为 A+B 与 C+D. 将网络分为四个部分相比实际划分而言, 具有更高的模块密度, 从网络的拓扑结构来看, 这样的划分也是合理的.

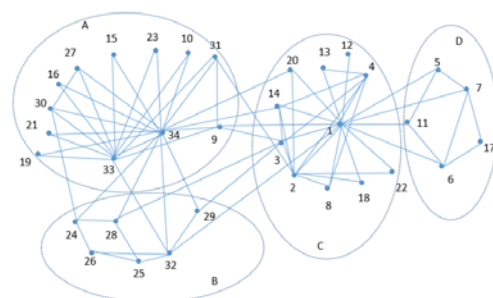


图 2 karate 网络拓扑结构及算法划分结果

表 2 football 网络实际分组

小组	编号
1	2 26 34 38 46 90 104 106 110
2	20 30 36 56 80 95 102
3	3 7 14 16 33 40 48 61 65 101 107
4	4 6 11 41 53 73 75 82 85 99 103 108
5	45 49 58 67 76 87 92 93 111 113
6	37 43 81 83 91
7	13 15 19 27 32 35 39 44 55 62 72 86 100
8	1 5 10 17 24 42 94 105
9	8 9 22 23 52 69 78 79 109 112
10	18 21 28 57 63 66 71 77 88 96 114
11	12 25 51 60 64 70 98
12	29 47 50 54 59 68 74 84 89 115

表 3 LPAd 对 football 网络的划分结果

小组	编号
1	2 26 34 38 46 90 104 106 110
2	20 30 31 36 56 80 81 83 95 102
3	3 7 14 16 33 40 48 61 65 101 107
4	4 6 11 41 53 73 75 82 85 99 103 108
5	45 49 58 59 60 64 67 76 87 92 93 98 113
6	-
7	13 15 19 27 32 35 37 39 43 44 55 62 72 86 100
8	1 5 10 17 24 42 94 105
9	8 9 22 23 52 69 78 79 109 112
10	18 21 28 57 63 66 71 77 88 96 97 114
11	12 25 29 51 70 91
12	47 50 54 68 74 84 89 111 115

表 2 和表 3 为 football 网络的实际分组与 LPAd 算法得到的分组. 表 3 中加粗的编号表示该编号所代表的队伍不在实际分组中. 从两张表格的对比分组可以看出, LPAd 得到的分组结果与实际分组结果相差不大.

使用 LPA^[5]、LPAm^[6]、LPA-E^[10]和 LPAd 四个算法对上述 6 个真实网络进行 1000 次重复实验, 计算其模块度和模块密度. 由于 LIB 使用的是加权网络, 而本文算法和其他 3 个算法使用的是无权网络, 不具可比性, 因此不加入对比实验. 实验结果如表 4 和表 5 所示.

表 4 平均模块度 Q

	karate	football	dolphins	lesmis	polbooks	sawmill
LPA	0.3450	0.5905	0.4788	0.5055	0.4953	0.4205
LPAm	0.3496	0.5780	0.4913	0.5212	0.4888	0.4065
LPA-E	0.3719	0.5934	0.4932	0.5271	0.5012	0.4611
LPAd	0.3901	0.6004	0.4953	0.5266	0.5205	0.4693

从表 4 可以看出, 除 lesmis 数据集外, LPAd 的平均模块度均比其他算法高, 而在 lesmis 数据集中, LPAd 的平均模块度仅比 LPA-E 低, 比其他算法高.

表 5 平均模块密度 D

	karate	football	dolphins	lesmis	polbooks	sawmill
LPA	6.7467	40.5969	10.5528	19.9296	18.2369	6.7103
LPAm	-0.3193	37.3021	5.3354	16.7693	14.0417	-1.7143
LPA-E	6.9864	41.1351	10.9083	20.4701	18.6761	7.2749
LPAd	7.1849	44.2573	11.1776	22.9400	21.5143	6.4359

从表 5 可以看出, 除 sawmill 数据集外, LPAd 的平均模块密度均比其他算法高, LPAd 的效果总体上优于 LPA、LPAm 和 LPA-E.

使用 LPA、LPAm、LPA-E 和 LPAd 四个算法对 11 个人工生成的 LFR 网络进行 100 次重复实验, 计算其模块度. 实验结果如图 3 所示.

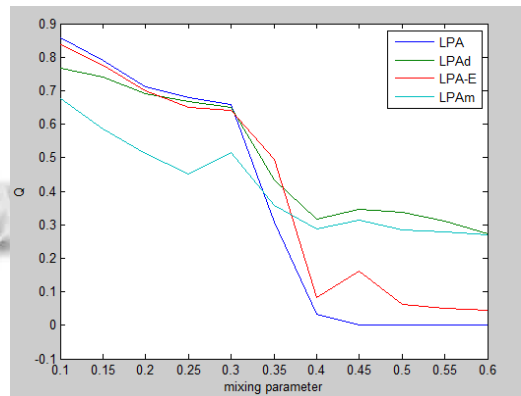


图 3 平均模块度

图 3 结果显示, 当网络社区结构明显时(混合参数小于 0.3), LPA、LPA-E 与 LPAd 的效果相当, 都优于 LPAm; 当网络社区结构模糊时(混合参数大于 0.3), LPAd 与 LPAm 的效果优于 LPA 与 LPA-E, LPAd 效果略好于 LPAm. 因此, 整体而言, LPAd 具有更好的鲁棒性, 而且适合于当网络社区结构模糊时的社区结构发现.

表 6 平均运算时间(ms)

	karate	football	dolphins	lesmis	polbooks	sawmill
LPA	0.11	0.62	0.30	0.37	0.69	0.13

LPA _m	0.13	0.55	0.28	0.33	0.75	0.11
LPA-E	0.25	1.20	0.66	0.96	1.71	0.27
LPA _d	0.23	2.91	0.61	0.61	1.80	0.18

表 7 算法在不同规模网络($z=0.3$)的平均运算时间(ms)

	LPA	LPA _d
500	4.30	13.75
1000	10.39	31.58
5000	68.37	213.2
10000	172.4	517.3
50000	1606	4027
100000	4286	9418

表 8 算法在不同规模网络($z=0.6$)的平均运算时间(ms)

	LPA	LPA _d
500	3.89	17.03
1000	9.84	38.86
5000	156.6	278.8
10000	602.5	560.3
50000	6599	4519
100000	18301	13206

表 6、7、8 为算法在不同网络的运算时间. 其中, 表 7 和表 8 测试的网络为 LFR 网络, 参数和上文所说的 LFR 网络参数相同, 表 7 中网络的混合参数 $z=0.3$ (社区结构清晰), 表 8 中网络的混合参数 $z=0.6$ (社区结构模糊). 总体而言, LPA_d 算法的运算时间约为 LPA 的 2 到 3 倍. 而从表 8 可以看到, 在网络的社区结构模糊且网络规模较大时, LPA_d 的速度比 LPA 快. 尽管两算法的运算时间有所不同, 但是它们在相同规模的网络上的运算时间为同一数量级, 因此从实验上验证了 LPA_d 与 LPA 具有相同的复杂度.

4 总结

LPA 以其简单高效而受到广泛的研究. LPA_m 通过改进其更新策略提高来算法的准确性, LIB 和 LPA-E 则通过确定更新顺序来提高算法的稳定性. LPA_m 倾向于将网络划分为若干个度数相近的社区, 当网络的社区大小差异较大时此方法效果不佳. 针对模块度优化方法存在解极限问题, Zhenping Li 等人提出模块密度及其优化方法, 不过该方法需要指定社区个数. 目前还没有直接优化模块密度的有效算法. 因此, 本文利用 LPA 社区发现算法的基本思想, 参考 LPA_m 的优化思想, 对模块密度进行优化, 能够快速地找出网络中的社区结构. 实验结果显示, 相比 LPA、LPA_m 和

LPA-E 而言, LPA_d 能够在网络社区结构模糊时得到具有更高的模块度的社区划分. 如何进一步提高 LPA_d 的准确性及稳定性并将其扩展到重叠社区的发现中将 是下一步的主要研究内容.

参考文献

- 1 Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. *Nature*, 1998, 393(6684): 440–442.
- 2 Barabasi AL, Albert R. Emergence of scaling in random networks. *Science*, 1999, 286(5439): 509–512.
- 3 Girvan M, Newman MEJ. Community structure in social and biological networks. *Proc. of the National Academy of Sciences*, 2002, 99(12): 7821–7826.
- 4 Fortunato S. Community detection in graphs. *Physics Reports*, 2010, 486(3): 75–174.
- 5 Raghavan UN, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 2007, 76(3): 036106.
- 6 Barber MJ, Clark JW. Detecting network communities by propagating labels under constraints. *Physical Review E*, 2009, 80(2): 026129.
- 7 Schuetz P, Caflisch A. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E*, 2008, 77(4): 046112.
- 8 Liu X, Murata T. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. *Physica A: Statistical Mechanics and its Applications*, 2010, 389(7): 1493–1500.
- 9 赵卓翔, 王轶彤, 田家堂, 周泽学. 社会网络中基于标签传播的社区发现新算法. *计算机研究与发展*, 2011, 3.
- 10 Zhao Y, Li S, Chen X. Community detection using label propagation in entropic order. 2012 IEEE 12th International Conference on Computer and Information Technology (CIT).
- 11 Fortunato S, Barthelemy M. Resolution limit in community detection. *Proc Natl Acad Sci USA*, 2007, 104(1): 36–41.
- 12 Li ZP, Zhang SH, Wang RS, Zhang XS, Chen LN. Quantitative function for community detection. *Phys Rev: E*, 2008, 77(3): 036109.
- 13 Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D. Defining and identifying communities in networks. *Proc. Natl. Acad. Sci.*, 2004, 101: 2658–2663.

- 14 Newman MEJ. Finding and evaluating community structure in networks. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, 2004, 69(2): 026113 .
- 15 Zachary WW. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 1977, 33: 452–473.
- 16 Girvan M, Newman MEJ. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* 99, 2002: 7821–7826.
- 17 Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM. *Behavioral Ecology and Sociobiology* 2003,54: 396–405.
- 18 Knuth DE. *The Stanford GraphBase: A Platform for Combinatorial Computing*. Addison-Wesley, Reading, MA, 1993.
- 19 Krebs V. A network of co-purchased books about U.S. politics. October 2008. [2010-03-28]. <http://www.orgnet.com/>.
- 20 Michael JH, Massey JG. Modeling the communication network in a sawmill. *Forest Products Journal*, 1997, 47, 25–30.
- 21 Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms. *Phys Rev E*, 2008, 78(4): 046110.