

基于节点中心性和社区相似性的快速标签传播算法

顾军华^{1,2*}, 霍士杰^{1,2}, 王守彬^{1,2}, 田 喆^{1,2}

(1. 河北工业大学 计算机科学与软件学院, 天津 300401; 2. 河北省大数据实验室(河北工业大学), 天津 300401)

(*通信作者电子邮箱 jhgu_hebut@163.com)

摘 要: 为了减少标签传播算法(LPA)中不必要的更新、解决算法准确率低且稳定性差的问题,提出了基于节点中心性和社区相似性的快速标签传播算法(FNCS_LPA)。按照节点中心性度量对网络的节点从低到高进行排序后加入节点信息列表,利用节点信息列表来指导更新过程,提高社区发现的稳定性并避免不必要的更新;采取基于社区相似性的更新规则,提高了社区发现的准确率。在真实社会网络和 LFR 基准网络上进行实验:相比 LPA 和三种较好的 LPA 改进算法, FNCS_LPA 在执行速度方面提升了几十倍,真实社会网络的模块度也相对较高,在社区结构比较模糊的 LFR 基准网络上的归一化互信息有明显的优势。实验结果表明 FNCS_LPA 在提高执行速度的基础上,提高了算法的稳定性和准确率。

关键词: 社区发现算法; 标签传播算法; 节点信息列表; 节点中心性; 社区相似性

中图分类号: TP301.6 **文献标志码:** A

Fast label propagation algorithm based on node centrality and community similarity

GU Junhua^{1,2*}, HUO Shijie^{1,2}, WANG Shoubin^{1,2}, TIAN Zhe^{1,2}

(1. School of Computer Science and Software, Hebei University of Technology, Tianjin 300401, China;

2. Big Data Laboratory in Hebei Province (Hebei University of Technology), Tianjin 300401, China)

Abstract: In order to reduce unnecessary update and solve the problem of low accuracy and poor stability of Label Propagation Algorithm (LPA), a Fast Label Propagation Algorithm based on Node Centrality and Community Similarity (FNCS_LPA) was proposed. According to the node centrality measure, the nodes of a network were sorted from low to high and added into node information list, which guided the update process to avoid unnecessary update and improve the stability of community detection. The accuracy of community detection was improved by a new update rule based on community similarity. Experiments were tested on a real social networks and LFR benchmarks. Compared with LPA and three improved LPA algorithms, the execution speed is improved by almost a dozen times, the modularities of the real social networks and the Normalized Mutual Information (NMI) of LFR (Lancichinetti Fortunato Radicchi) benchmark networks with more obscure community structure were significantly improved. The experimental results show that FNCS_LPA improves the accuracy and stability of community detection on the basis of improving execution speed.

Key words: community detection algorithm; Label Propagation Algorithm (LPA); node information list; node centrality; community similarity

0 引言

自然界与人类社会中的许多系统都可以用复杂网络模型表示,网络社区结构是复杂网络中最普遍和最重要的拓扑属性。社区发现算法可以通过分析网络中节点之间的关联性,挖掘复杂网络中的社区结构,从而揭示复杂网络中隐含的特性和功能,对于网络结构的理论研究和网络分析的实际应用有着重要的作用。

2007 年 Raghavan 等^[1]首次将标签传播算法(Label Propagation Algorithm, LPA)应用于社区发现。与传统的社区发现方法(基于层次聚类的算法^[2-3]、基于随机游走的算法^[4]等)相比, LPA 仅依赖于网络的传播特性,且具有线性的时间复杂度,适合用于对复杂网络进行社区发现和分析的优

点。但是 LPA 也有一些缺点:1)节点更新顺序具有随机性,并且当邻接节点中出现次数最多的标签(最大值标签)有多个时,会随机选择一个标签;2)存在不必要的标签更新过程;3)更新标签时仅仅考虑邻接节点中标签出现的次数,忽略邻接节点的局部结构信息。这些缺点会导致算法延迟收敛,社区发现的结果准确率低且稳定性差的问题。针对以上问题,2014 年, Xing 等^[5]提出了基于节点影响力的标签传播算法(Node Influence Based Label Propagation Algorithm for community detection in networks, NIBLPA),该算法使用 k -shell 分解方法计算每一个节点的影响力值,依据影响力值从高到低的顺序更新标签和选取标签,提高了算法的稳定性和准确率,但是时间复杂度也有所提升。2015 年, Sun 等^[6]提出了基于中心性的标签传播算法(Centrality-based Label Propagation

收稿日期:2017-12-14;修回日期:2017-12-10;录用日期:2017-12-14。

基金项目:天津市科技计划项目(16ZXHLSF0023);河北省科技计划项目(17210305D)。

作者简介:顾军华(1966—),男,河北赵县人,教授,博士,CCF 会员,主要研究方向:智能信息处理、数据挖掘; 霍士杰(1993—),男,河北石家庄人,硕士研究生,CCF 会员,主要研究方向:数据挖掘、计算机仿真; 王守彬(1994—),男,河南商丘人,硕士研究生,主要研究方向:数据挖掘、计算机仿真; 田喆(1994—),男,天津人,硕士研究生,主要研究方向:数据挖掘。

algorithm for community detection in networks, Cen_LP), 该算法为每一个节点定义了节点中心值和节点偏向值, 依据节点中心值从低到高的顺序更新标签并且利用节点偏向值选取标签, 算法在不增加时间复杂度的情况下, 提高了社区发现的准确率。2017年, Ma等^[7]提出了基于节点重要性和随机游走的标签传播算法(An improved Label Propagation Algorithm based on Node Importance and random walk for community detection, NILPA), 该算法依据节点的度数计算节点重要性, 并且依据随机游走理论形成节点相似度矩阵, 在两者的基础上形成新的度量标准来衡量节点之间的紧密程度, 依据该度量标准更新标签。算法提高了社区发现的准确率, 但由于计算节点相似性依赖于矩阵相乘, 在网络规模不断扩大时, 会消耗越来越多计算资源, 而且时间复杂度也较高。综上所述, 尽管上述算法在准确率上有所提高, 但是都提高了算法的时间复杂度, 降低了算法的执行速度。

为了在加快算法的执行速度的基础上提高算法准确率和稳定性, 本文提出了基于节点中心性和社区相似性的快速标签传播算法(Fast Label Propagation Algorithm based on the Node Centrality and Community Similarity, FNCS_LPA)。FNCS_LPA首先计算每一个节点的中心性度量值, 按照中心性度量值从低到高的顺序排序并加入到节点信息列表中, 用节点信息列表指导更新过程, 通过记录节点信息列表中每一个节点的状态信息, 判断当前节点是否需要更新, 从而避免了不必要的更新, 提高了运行速度。在更新过程中, FNCS_LPA利用基于社区相似性的更新规则, 即不仅仅考虑邻接节点中标签出现的次数, 还会评估每个邻接节点与待更新节点的相似性, 社区相似性依赖于节点相似性求和, 待更新节点会选择社区相似性最高的社区, 从而提高算法的准确率。

1 基于节点中心性的快速标签传播算法

1.1 基于节点中心性的更新顺序

在LPA的每一次迭代过程中, 节点的更新顺序是随机的, 这等同于平等对待网络中的所有节点。事实上, 网络中的每个节点的重要程度依赖于其在网络中的局部信息, 并不是同等的, 如果平等对待, 按随机的顺序更新标签, 可能会造成结果与实际不符, 导致算法准确率低且稳定性差的问题。基于此, 本文提出了基于节点中心性的更新顺序, 其中, 节点中心性依赖于节点的聚集系数(Clustering Coefficient)和节点密度。假设节点*i*和节点*j*是网络中的两个节点。

定义1 节点的聚集系数。节点*i*的聚集系数定义如下:

$$CC_i = \frac{E(i)}{|N_i| * (|N_i| - 1) / 2} \quad (1)$$

其中: N_i 表示节点*i*的邻接节点集合, $|N_i|$ 表示节点*i*的邻接节点数, $E(i)$ 表示节点*i*的邻接点之间实际的连边数。节点的聚集系数可表示为当前节点的所有邻节点之间实际连边数与所有可能连边数的比值, 由于节点的邻接点之间实际存在的边不会大于可能存在的最大值, 因此节点的聚集系数的取值范围在0~1; 当节点的度数为0或1时, 或者当节点的邻接点之间相互独立时, 该节点的聚集系数为0; 当节点的聚集系数为1时, 则表示该节点与其邻接点之间构成完全图。因此节点的聚集系数越高, 说明节点具有越高的聚集度。

定义2 节点密度。节点*i*的密度定义如下:

$$\rho_i = d_i / (N - 1) \quad (2)$$

定义3 节点中心性度量值。

$$\delta_i = CC_i * \rho_i \quad (3)$$

节点中心性度量值表示为节点密度和节点聚集系数的乘积, δ_i 的值越大, 表明节点在网络中的重要程度越高。在实际生活中, 重要程度高的节点为专家, 而重要程度低的节点为普通求知者, 普通求知者都是听取专家的知识, 从而完成知识传播。同样, 在标签传播过程中, 标签就是知识, 节点就是人, 重要程度低的节点往往要先采纳周围重要程度高的节点的标签, 从而完成标签传播。因此, 节点 δ_i 的值越小, 该节点就越先更新标签, 这样得到的结果才会与实际相符, 因此节点的更新顺序按照节点中心性度量值升序更新。

1.2 基于节点中心性的快速标签传播算法

LPA运行在线性时间内, 具有执行速度快的优点。LPA在前几次迭代过程中节点改变标签的概率是非常高的, 但是, 在5次迭代以后, 95%以上的节点都会被正确地划分(当前节点的标签已经变为邻接节点中最大值标签)。为了说明这一现象, 本文将LPA分别应用在CA-Hepth、CA-GrQc、Email和PGP这4种常用的真实网络数据集上进行社区发现, 本文实验将每一个真实网络都看作无向无权图, 每一个图都没有自循环的边并且只取图的最大连通子图。本文所有的实验均使用Python语言实现, 在Window 7, AMD Core A8-4555 CPU 1.6 GHz, 8 GB内存环境下进行。

1.2.1 实验一: LPA的收敛性

数据集详细信息如表1所示, 计算5次迭代以后已被正确划分的节点占总节点的百分比。每个网络重复100次实验, 4种网络的收敛信息如表2所示。

表1 第一部分数据集介绍

Tab. 1 Introduction of the first part datasets

网络名称	节点数	边数	描述
Email	1 133	5 451	Email 社交网络 ^[8]
CA-GrQc	4 158	13 422	广义相对论研究网络 ^[9]
CA-Hepth	8 638	24 807	高能物理学研究网络 ^[9]
PGP	10 680	24 316	PGP 算法的研究团体网 ^[10]

表2 网络的收敛信息

Tab. 2 Convergence information of networks

网络名称	5次迭代后正确划分节点所占百分比/%	算法收敛所需的迭代次数
Email	96.64	17.89
CA-GrQc	99.37	15.70
CA-Hepth	98.97	59.75
PGP	99.37	21.90

从表2可以看出, 在5次迭代以后, Email网络有96.64%的节点被正确划分, CA-Hepth网络有98.97%的节点被正确划分, Email和PGP网络有99%以上的节点都被正确划分。这说明无论总的迭代次数是多少, LPA在5次迭代以后, 95%以上的节点都已经得到了正确的划分, 而5次迭代后的每一次迭代, 仅仅较少的节点标签会发生改变, 而对于大部分的节点来说, 即使更新标签, 也不会改变标签, 这些不必要的更新拖慢了算法的收敛速度。针对这一问题, 本节提出基于节点中心性的快速标签传播算法。

1.2.2 实验二: LPA与FNCS_LPA的对比

LPA在每次迭代过程中, 对节点标签的更新有两种情况:

一种是邻接节点中的最大值的标签只有一个,另一种是邻接节点中的最大值标签有多个。如果当前节点的标签已经是邻接节点中的唯一最大值标签,这样的节点称为被动节点,被动节点在本次更新中不会再改变标签;如果邻接节点中的最大值标签有多个,当前节点的标签是其中之一,这样的节点称为干扰节点,根据 LPA 的更新规则,干扰节点会随机选择其中一个标签,干扰节点在本次更新中可能会改变标签。被动节点和干扰节点以外的其他节点是主动节点,主动节点在本次更新中一定会改变标签。如果能避免对被动节点的标签更新,算法会以更快的速度收敛,因此,本文构建一个节点信息列表,其中包含所有的节点和每个节点的状态信息(被动、干扰和主动)。每次只选择信息列表中的干扰和主动节点进行标签更新,标签更新完成后再进行状态更新。

基于节点中心性的快速标签传播算法(Fast Label Propagation Algorithm based on Node Centrality, FNC_LPA)的具体流程如下:

第一步 按照式(1)、式(2)和式(3)计算节点的中心性度量值,将所有的节点按照节点中心性度量值降序排序并加入节点信息列表当中。每一个节点都被初始化为一个独一无二的标签,都被设置为主动节点。

第二步 从节点信息列表中随机选择一个主动节点或干扰节点,按照节点标签更新规则,对当前节点的标签进行更新。标签更新完成后对该节点及其邻接节点进行状态更新,节点在更新状态时,可能会更新为被动节点、主动节点和干扰节点。

第三步 根据节点状态判断节点信息列表中是否还有主动节点:若有,继续执行第二步;否则,算法结束。

为了评估 FNC_LPA 的有效性,本次实验包括表 1 和表 3 中的 8 个真实网络数据集。对 8 种数据集分别使用 FNC_LPA 和 LPA 进行社区发现,对比算法收敛时所需的迭代次数和模块度^[11]。模块度是一种比较重要的衡量网络社区结构强度的方法,计算公式如下:

$$Q = \sum_{c \in L} \frac{L_c}{m} - (D_c/2m)^2 \quad (4)$$

其中: L_c 表示社区 c 内部的链接数, m 表示整个网络边的总个数, D_c 表示社区 c 内部节点的度数总和, L 表示整个网络的全部社区。

表 3 第二部分数据集介绍

Tab. 3 Introduction of the second part datasets

网络名称	节点数	边数	描述
karate	34	78	空手道俱乐部网络 ^[12]
dolphins	62	159	海豚数据网络 ^[13]
polbooks	105	441	亚马逊政治书销售网络 ^[14]
netscience	379	914	网络科学论文合著网络 ^[15]

表 4 LPA 和 FNC_LPA 平均模块度对比

Tab. 4 Comparison of average modularity between LPA and FNC_LPA

网络名称	LPA	FNC_LPA	网络名称	LPA	FNC_LPA
karate	0.345	0.332	Email	0.234	0.301
dolphins	0.483	0.478	CA-GrQc	0.775	0.774
polbooks	0.493	0.499	CA-Hepth	0.628	0.640
netscience	0.794	0.801	PGP	0.802	0.804

为了使算法具有可比性,FNC_LPA 与 LPA 使用相同的更

新规则(当前节点的标签选取邻接节点中的最大值标签)。FNC_LPA 的迭代次数记为算法总更新次数和节点总个数的比值。模块度和迭代次数为对每一个网络重复 100 次实验后所求得的平均值,迭代次数对比如图 1 所示,可以看出,因为 FNC_LPA 避免了很多不必要的更新,所以迭代次数明显少于 LPA。在 karate、dolphins、polbooks 和 netscience 网络上,FNC_LPA 的迭代次数是 LPA 的 1/5 至 1/2 左右,在 Email 和 CA-GrQc 网络上,FNC_LPA 是 LPA 的 1/6 左右,在 CA-Hepth 和 PGP 网络上,FNC_LPA 是 LPA 的 1/10 左右。平均模块度对比如表 4 所示,可以看出,与 LPA 相比,FNC_LPA 除了在 Email 和 CA-Hepth 网络上模块度有小幅提升以外,在其余网络上模块度几乎没有发生变化。因此,实验结果证明 FNC_LPA 在没有改变社区发现效果的基础上,能够大幅度降低迭代次数。

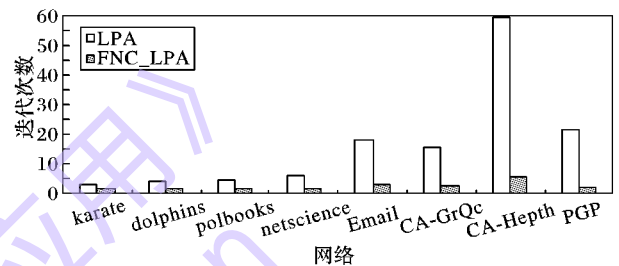


图 1 FNC_LPA 和 LPA 的迭代次数对比

Fig. 1 Comparison of iterations between FNC_LPA and LPA

2 基于社区相似性的更新规则

LPA 的更新规则仅仅考虑邻接节点的标签出现次数,并且当前节点的标签更新为邻接节点中最大值标签,具有相同标签的节点看作一个社区,这种更新规则默认是将每一个邻接节点与当前节点的相似性(社区相似性依赖于节点相似性求和)看作是相同的,LPA 最终是将最大值标签所代表的社区作为与当前节点相似性最高的社区。实际上,每一个邻接节点与当前节点的相似性并不是同等的,如果同等对待,会导致算法准确率降低且稳定性差的问题。本章提出基于社区相似性的更新规则,社区相似性是由节点相似性求和得到,其中节点相似性依赖于节点之间的公共节点个数和节点的度数,基于社区相似性的更新规则会使得更新节点选择邻接节点的社区中相似性最高的社区。

本章对算法中使用的主要概念进行形式化定义,如下所示:

定义 4 边缘度数比。假设节点 i 和节点 j 是图中的两个节点, i 与 j 的边缘度数比定义如下:

$$\rho_{ij} = (d_j - \text{sim}(i, j) - 1)/d \quad (5)$$

其中: $\text{sim}(i, j)$ 表示节点 i 和节点 j 的公共节点的个数, d_i 表示节点 i 的度。

定义 5 节点相似性。节点 j 对节点 i 的重要程度定义如下:

$$\text{close}(i, j) = c_1 \text{sim}(i, j) + c_2 \rho_{i, j} \quad (6)$$

其中: c_1 和 c_2 是取值在 0 ~ 1 的两个常数, ρ_{ij} 是节点 i 和节点 j 的边缘度数比。

定义 6 社区相似性。节点 i 与社区 l 的相似性定义如下:

$$S(i, l) = \sum_{j \in N_i \cap C_l} \text{close}(i, j) + |N_i \cap C_l| \quad (7)$$

定义 7 近似最大值标签。在 LPA 中,当前更新的节点的标签选取邻接节点中的最大值标签,将这个标签的出现次数记为

max ; 然而,如果邻接节点某类标签的出现次数(计为 n) 和 max 的值相差不大时,这类标签称为近似最大值标签。算法引入一个取值在 $0 \sim 1$ 的常量 $Radio$ 来衡量近似程度。

$$(max - n)/max < Radio \quad (8)$$

若邻接节点中标签出现次数 n 满足式(8),都称之为近似最大值标签。正在更新的当前节点也可能选择近似最大值标签,其中,在更新过程中为了不考虑节点数量较少的标签, $Radio$ 的值不宜设置过大,应在 $0 \sim 0.5$ 。近似最大值标签的引入,避免了对每一个社区进行相似性计算,提高了算法效率。

定义8 基于社区相似性的更新规则。假设集合 T 中保存了节点 i 的邻接节点中的最大值标签(社区)和近似最大值标签, $L(i)$ 表示节点 i 的标签,基于社区相似性的更新规则如式(9)所示:

$$L(i) = \arg \max_{l \in T} \{S(i, l)\} \quad (9)$$

式(9)是在集合 T 中,让节点 i 选择使得 $S(i, l)$ 最大的一类标签,如果有多类标签使得 $S(i, l)$ 最大,就从中随机选取。

为了验证该更新规则的有效性,本次实验在 LPA 中使用基于社区相似性的更新规则形成基于社区相似性的标签传播算法(Label Propagation Algorithm based on Community Similarity, CS_LPA),CS_LPA 的节点更新的顺序是随机的。其中式(6)中的 c_1 取值1, c_2 取值0.2,式(8)中的 $Radio$ 值取0.4。本次实验除了使用表1和表3中的8种真实网络数据集,还应用了 LFR(Lancichinetti Fortunato Radicchi)基准网络。LFR 基准程序是由 Lancichinetti 等^[16]提出的专门用于生成模拟网络的算法,该算法主要根据输入的参数,尽可能地生成符合真实网络特征的人工合成网络,因此具有较高的实验价值,是当前社区发现研究中最常用的模拟数据集。在生成 LFR 基准网络时常用的参数如表5所示,其中 mu 表示节点与社区外部连接的概率,其值在 $0 \sim 1$, mu 值越大,表明社区的结构越不明显,社区发现的难度也越大。本次实验使用的 LFR 基准网络数据集如表6所示,其中包含4组 LFR 基准网络,每组包含15个不同 mu 值的网络, mu 的取值在 $0.1 \sim 0.8$, 间隔为0.05。相比2000S(5000S)网络,2000B(5000B)网络中每一社区内的节点个数较多。

表5 LFR 基准网络的主要参数

Tab. 5 Parameters of LFR benchmark network

参数	含义	参数	含义
N	网络中的节点个数	c_{min}	社区内的最少节点个数
k	网络的平均度数	c_{max}	社区内的最多节点个数
k_{max}	网络的最大度数	mu	节点与社区外部连接的概率

表6 LFR 基准网络数据集描述

Tab. 6 Description of LFR benchmark network datasets

网络	N	k	k_{max}	c_{min}	c_{max}	mu
2000S	2000	10	50	10	50	$0.1 \sim 0.8$
2000B	2000	10	50	20	100	$0.1 \sim 0.8$
5000S	5000	10	50	10	50	$0.1 \sim 0.8$
5000B	5000	10	50	20	100	$0.1 \sim 0.8$

归一化互信息(Normal Mutual Information, NMI)^[17]能够量化算法对网络进行的划分和网络的真实划分之间的关系,是一种在社区发现领域常用的度量标准。NMI 的取值在 $0 \sim 1$,其计算式(10)所示:

$$NMI(A, B) = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} N_{ij} \log(N_{ij}n/N_i N_j)}{\sum_{i=1}^{C_A} N_i \log(N_i/n) + \sum_{j=1}^{C_B} N_j \log(N_j/n)} \quad (10)$$

其中: A 是网络的真实划分, B 是算法对网络所进行的划分, C_A 表示 A 种划分的社区个数, C_B 表示 B 种划分的社区个数, N 表示一个矩阵, N_{ij} 表示 A 种划分第 i 个社区中的节点出现在 B 种划分第 j 个社区中的节点个数, N_i 表示矩阵 N 第 i 行的求和, N_j 表示矩阵 N 第 j 列的求和, n 表示整个网络节点的个数。如果 A 种划分和 B 种划分是相同的,则 $NMI(A, B)$ 的值为1。 NMI 的值越高,表示算法社区发现的准确率越高。

将 CS_LPA 和 LPA 分别用在表1和表3的8种真实网络数据集和4种 LFR 基准网络数据集上进行社区发现,对每一个网络重复100次实验。在8种真实网络数据集上与 LPA 对比平均模块度,对比如表7所示;在所用的数据集上,CS_LPA 的模块度都比 LPA 要高;其中,在 Email、CA-Hepth、dolphins 和 netscience 网络上,CS_LPA 模块度有明显的提高。由于 LFR 基准网络已知真实的划分结果,因此在 LFR 基准网络数据集上与 LPA 对比平均 NMI 值。NMI 的对比如图2所示。

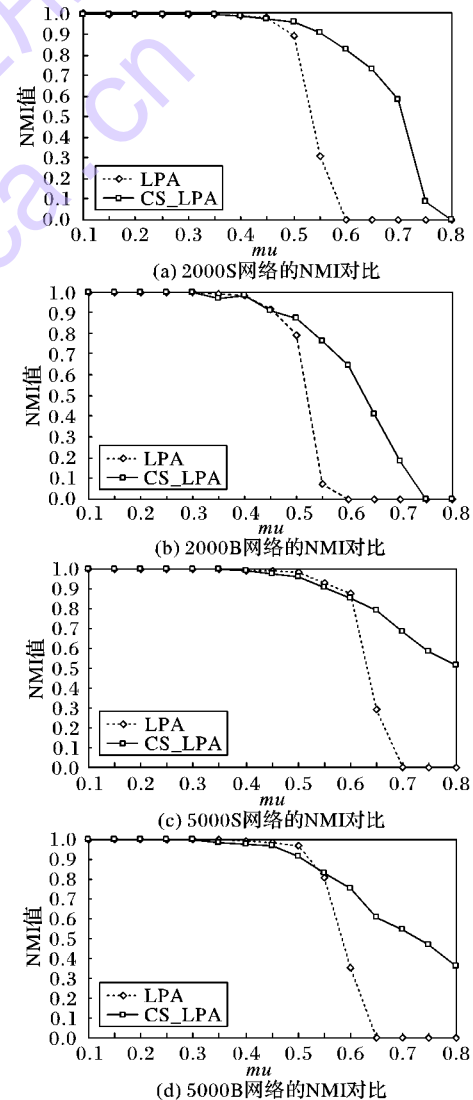


图2 LFR 基准网络上 CS_LPA 和 LPA 的 NMI 对比

Fig. 2 NMI comparison of CS_LPA and LPA on LFR benchmark network

从图2(a)可以看出, μ 的值在0.1~0.45时,两种算法都能较好地发现社区结构; μ 的值在0.45~0.6时,LPA的NMI值迅速下降,表明社区发现的准确率在下降,而CS_LPA的NMI的值远高于LPA; μ 的值在0.65~0.8,LPA已无法进行社区发现,而CS_LPA仍可进行。图2(b)产生了类似于图2(a)的效果, μ 的值在0.1~0.4时,两种算法都能较好地发现社区结构,而 μ 的值在0.45~0.8时,CS_LPA的NMI值高于LPA。在图2(c)和图2(d)中,当 μ 大于0.6时,CS_LPA的NMI明显高于LPA。

综上所述,相比LPA,CS_LPA提高了社区发现的准确率,尤其对于社区结构不清晰的网络,算法的优势更加明显。

表7 LPA与CS_LPA平均模块度对比

Tab. 7 Comparison of average modularity between LPA and CS_LPA

网络名称	LPA	CS_LPA	网络名称	LPA	CS_LPA
karate	0.345	0.369	Email	0.234	0.465
dolphins	0.483	0.519	CA-GrQc	0.775	0.788
polbooks	0.493	0.518	CA-Hepth	0.628	0.675
netscience	0.794	0.818	PGP	0.802	0.815

3 FNCS_LPA 算法

3.1 算法流程

在第1章和第2章中,本文分别引入了基于节点中心性的快速更新过程和基于社区相似性的更新规则,在基于节点中心性的快速更新过程中引入基于社区相似性的更新规则,形成基于节点中心性和社区相似性的快速标签传播算法(FNCS_LPA)。算法的伪代码如下:

FNCS_LPA:

输入 $G(V, E)$ 、 c_1 、 c_2 、Radio;

/* 输入图的邻接矩阵和式(6),式(8)的参数 */

输出 社区划分结果。

- 1) 为每个节点分配一个独一无二的标签;
- 2) 按式(1)~(3)计算所有节点的中心性度量值,按照升序的顺序添加到节点信息列表当中,将所有的节点标记为主动节点;
- 3) 从节点信息列表中随机选取一个主动节点或干扰节点,记为 i ;
- 4) 将当前节点 i 的邻接节点中出现次数最多的标签和近似最大值标签加入集合 T 当中;
- 5) 初始化 \maxLabel 集合为空; \maxLabel 记录与当前节点 i 相似性最高的一个或多个标签。
- 6) for each $l \in T$ do
- 7) $\maxS \leftarrow 0$;
- 8) 依据式(6)计算 $S(i, l)$;
- 9) if $S(i, l) = \maxS$ then
- 10) 将 l 标签添加到 \maxLabel 中;
- 11) else if $S(i, l) > \maxS$ then
- 12) $\maxS \leftarrow S(i, l)$;
- 13) 从 \maxLabel 中移除所有元素;
- 14) 将 l 标签添加到 \maxLabel 中;
- 15) end if
- 16) end for
- 17) 当前节点 i 从 \maxLabel 集合中随机选取标签;
- 18) for each $j \in N(i) \cap i$ do
- /* 更新当前节点 i 和 i 的邻接节点的状态 */
- 19) if node j is interference node then

- 20) 标记 j 为干扰节点;
- 21) else if node j is passive node then
- 22) 标记 j 为被动节点;
- 23) else
- 24) 标记 j 为主动节点;
- 25) end if
- 26) end for
- 27) 如果节点信息列表仍含有主动节点,跳到第3)步;
- 28) 将具有相同标签的节点划分到一个社区当中,算法结束

3.2 时间复杂度

与LPA相比, FNCS_LPA的总体时间复杂度没有改变。首先按照节点中心性度量值对网络节点进行排序,排序算法选择较快的桶排序,其时间复杂度接近于 $O(n)$, n 表示网络中的节点个数。初始化节点信息列表的时间复杂度是 $O(n)$;从节点信息列表中随机选择一个节点时间复杂度为 $O(1)$;更新当前节点的标签的时间复杂度是 $O(d_i)$, d_i 是节点 i 的度数,因此在整个更新过程中,时间复杂度是 $O(m)$, m 是边的个数。通过使用节点信息列表,算法的收敛比较简单,仅仅判断节点的状态标记即可,最差的情况是 $O(n)$ 。因此FNCS_LPA算法的总体时间复杂度是 $O(m)$ 。

4 实验结果与分析

为了验证FNCS_LPA的有效性,本次实验选取了第1章提到的NIBLPA^[5]、Cen_LP^[6]和NILPA^[7]三种较好的改进LPA,数据集选取了表1和表3中的8种真实网络数据集和表6中的4种LFR基准网络数据集。其中式(6)的 c_1 的值为1, c_2 的值为0.2,式(8)的Radio值为0.4。

4.1 执行速度对比

将FNCS_LPA、Cen_LP、NIBLPA和NILPA分别应用到真实网络数据集集中进行社区发现。为了具有可比性,每一个算法的终止条件都是节点标签成为邻接节点中的最大值标签,对每一个网络重复100次实验求平均的迭代次数,得到的结果如图3所示。

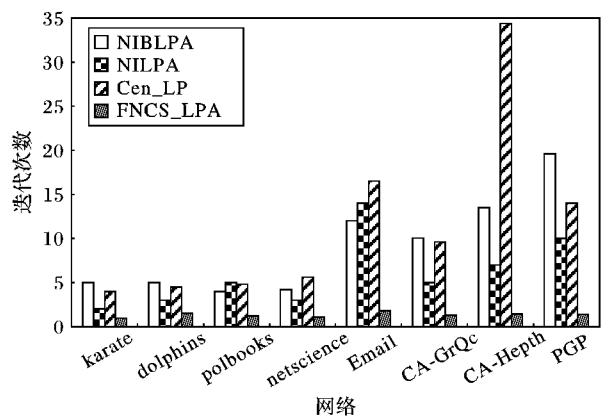


图3 FNCS_LPA与三种改进LPA的迭代次数对比

Fig. 3 Comparison of iterations between FNCS_LPA and other three improved LPA algorithms

从图3可以看出,与Cen_LP相比, FNCS_LPA在所有的数据集上的迭代次数明显要比Cen_LP少很多,这是因为FNCS_LPA避免了很多不必要的更新。在karate、dolphins、polbooks和netscience网络上, FNCS_LPA的迭代次数是Cen_LP的1/4至1/2左右;在Email、CA-GrQc和PGP网络中,算法的迭代次数是Cen_LP的1/10左右;而在CA-Hepth数据

集上,算法的迭代次数是 Cen_LP 的 1/30 左右。与 NIBLPA 相比, FNCS_LPA 在各个数据集上的迭代次数为 NIBLPA 的 1/16 到 1/3 左右。与 NILPA 相比, FNCS_LPA 在各个数据集上的迭代次数为 NIBLPA 的 1/14 到 1/2 左右。综上所述,与其他三种算法相比, FNCS_LPA 明显提高了算法的执行速度。

4.2 模块度对比

将 4 种算法应用在 8 种真实网络数据集上进行社区发现,并在最低模块度、平均模块度和最高模块度方面进行对比,模块度的计算公式如式(1)所示。平均模块度(average)是对每一个真实网络作 100 次实验求取的平均值,最低模块度(min)和最高模块度(max)分别是对每一个真实网络作 100 次实验后求取的最差结果和最优结果。模块度的对比如表 8 所示,其中加粗的数据表明 4 种算法中对应每一个网络的最好结果。

表 8 FNCS_LPA 与三种改进 LPA 算法模块度对比
Tab. 8 Comparison of modularity between FNCS_LPA and other three improved LPA algorithms

网络名称	范围	FNCS_LPA	Cen_LP	NIBLPA	NILPA
karate	min	0.371	0.416	0.423	0.371
	average	0.371	0.416	0.423	0.371
	max	0.371	0.416	0.423	0.371
dolphins	min	0.514	0.512	0.521	0.526
	average	0.514	0.519	0.521	0.526
	max	0.514	0.526	0.521	0.526
polbooks	min	0.518	0.511	0.497	0.520
	average	0.518	0.518	0.497	0.520
	max	0.518	0.523	0.497	0.520
netscience	min	0.831	0.772	0.747	0.807
	average	0.831	0.773	0.747	0.807
	max	0.831	0.774	0.747	0.807
Email	min	0.517	0.003	0.427	0.155
	average	0.517	0.427	0.427	0.155
	max	0.517	0.532	0.427	0.155
CA-GrQc	min	0.786	0.737	0.707	0.764
	average	0.786	0.740	0.707	0.764
	max	0.786	0.744	0.707	0.764
CA-Hepth	min	0.672	0.611	0.612	0.627
	average	0.672	0.622	0.612	0.627
	max	0.673	0.632	0.612	0.627
PGP	min	0.814	0.741	0.783	0.788
	average	0.814	0.750	0.783	0.788
	max	0.814	0.764	0.783	0.788

从表 8 中可以看出, FNCS_LPA 在所有的数据集上的最低模块度、平均模块度和最高模块度均完全一致。在 karate 网络上, FNCS_LPA 和 NILPA 都将网络划分为两个社区,求得的结果与真实划分完全相同;在 dolphins 和 polbooks 网络上, FNCS_LPA 在最低模块度、平均模块度和最高模块度上,均接近于最优解;尽管 Email 网络没有在最高模块度方面取得最好的结果,但是从平均模块度和最低模块度方面, FNCS_LPA 仍是最有优势的;而在 netscience、CA-GrQc、CA-Hepth 和 PGP 等网络上, FNCS_LPA 的模块度明显高于其他几种算法。综上所述, FNCS_LPA 相比其他 3 种算法提高了社区发现的准确率和稳定性;尤其在较大规模的网络算法,算法的优势更加明显。

4.3 归一化互信息对比

本次实验将 4 种算法应用在表 6 中的 LFR 基准网络数据集上进行社区发现,并在归一化互信息度量(NMI)方面进行对比, NMI 的计算公式如式(7)所示。对每一个网络重复 100 次实验求取平均 NMI 值。实验结果如图 4 所示。

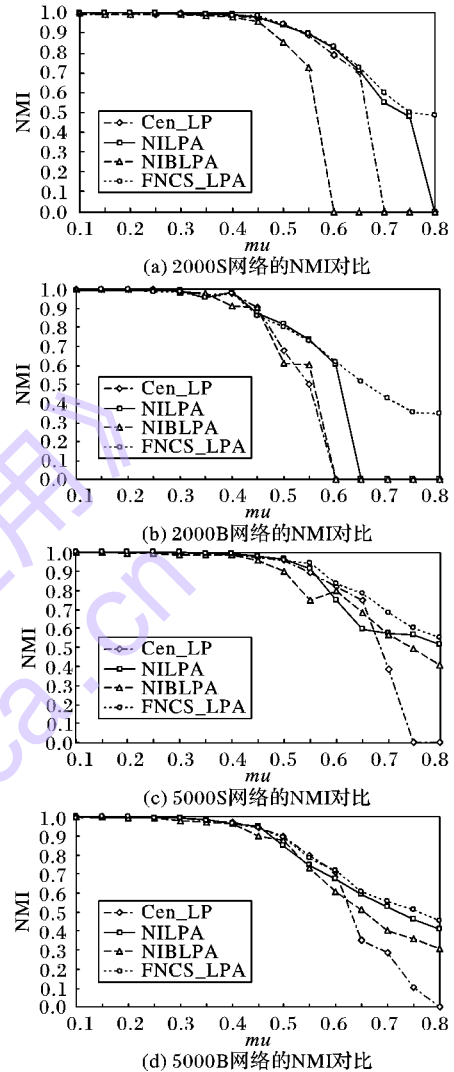


图 4 LFR 基准网络上 FNCS_LPA 与三种改进 LPA 算法的 NMI 对比

Fig. 4 NMI comparison of FNCS_LPA and other

three improved LPA algorithms on LFR benchmark network

从图 4 可以直观地看出,随着 μ 的值不断增大,社区结构越来越不清晰,4 种算法的社区发现准确率都在下降。从图 4(a)可以看出:当 μ 的值在 0.1~0.4 时,4 种算法都能较好地发现社区结构; μ 的值在 0.4~0.8 时, FNCS_LPA 的 NMI 的值高于其他 3 种算法。在图 4(b)中, μ 的值在 0.35~0.45 时, FNCS_LPA 略低于 NILPA;但 μ 的值在 0.6~0.8 时, FNCS_LPA 的 NMI 值都是 4 种算法中最高的。图 4(c)和图 4(d)产生了类似的效果,当 μ 的值大于 0.5 时, FNCS_LPA 社区发现的准确率明显高于其他 3 种算法。综上所述, FNCS_LPA 相比其他 3 种算法提高了社区发现的准确率;尤其对于社区结构不清晰的网络, FNCS_LPA 的效果更好,准确率更高。

5 结语

本文提出了基于节点中心性和社区相似性的快速标签传

播算法。首先,该算法计算每一个节点的中心度量值,依据节点中心性度量值升序将节点加入节点信息列表,利用节点信息列表指导更新过程,通过记录每一个节点的更新状态,避免了许多不必要的更新,减少了迭代次数,从而提高了社区发现的稳定性和执行速度。为了验证基于节点中心性的快速更新过程的有效性,实验使用了8种真实网络在迭代次数和平均模块度方面与LPA进行了对比。其次,算法引入社区相似性的更新规则,即当前节点会加入与当前节点社区相似性最高的社区,提高了社区发现的准确率,为了验证基于社区相似性的更新规则的有效性,算法在平均模块度、NMI方面与LPA进行了对比。最后,将基于节点中心性的快速更新过程和基于社区相似性的更新规则结合形成本文提出的基于节点中心性和社区相似性的快速标签传播算法,与目前3种改进的LPA在迭代次数、模块度和NMI三个方面进行了对比,实验结果表明FNCS_LPA在提升算法执行速度的基础上,提高了算法的稳定性和准确率。

参考文献 (References)

- [1] RAGHAVAN U N, ALBERT R, KUMARA S. Near linear time algorithm to detect community structures in large-scale networks[J]. *Physical Review E*, 2007, 76(3): 96–106.
- [2] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks[J]. *Proceedings of the National Academy of Sciences*, 2002, 99(12): 7821–7826.
- [3] NEWMAN M E J. Fast algorithm for detecting community structure in networks[J]. *Physical Review E*, 2004, 69(6): 66–73.
- [4] PONS P, LATAPY M. Computing communities in large networks using random walks[J]. *Journal of Graph Algorithms and Applications*, 2006, 10(2): 191–218.
- [5] XING Y, MENG F, ZHOU Y, et al. A node influence based label propagation algorithm for community detection in networks[J]. *The Scientific World Journal*, 2014, 2014(5): 210–223.
- [6] SUN H, LIU J, HUANG J, et al. CenLP: a centrality-based label propagation algorithm for community detection in networks[J]. *Physica A: Statistical Mechanics and its Applications*, 2015, 436(15): 767–780.
- [7] MA T, XIA Z. An improved label propagation algorithm based on node importance and random walk for community detection[J]. *Modern Physics Letters B*, 2017, 31(14): 98–116.
- [8] ZHANG X, CHEN S, JIA J, et al. An improved label propagation algorithm based on the similarity matrix using random walk[J]. *International Journal of Modern Physics B*, 2016, 30(16): 93–108.
- [9] LESKOVEC J, KLEINBERG J, FALOUTSOS C. Graph evolution: densification and shrinking diameters[J]. *ACM Transactions on Knowledge Discovery from Data*, 2007, 1(1): 2–9.
- [10] ZHANG X, FEI S, SONG C, et al. Label propagation algorithm based on local cycles for community detection[J]. *International Journal of Modern Physics B*, 2015, 29(5): 15–28.
- [11] NEWMAN M E J. Fast algorithm for detecting community structure in networks[J]. *Physical Review E*, 2004, 69(6): 66–78.
- [12] ZHOU K, MARTIN A, PAN Q, et al. Evidential label propagation algorithm for graphs[C]// *Proceedings of the 2016 19th International Conference on Information Fusion*. Piscataway, NJ: IEEE, 2016: 1316–1323.
- [13] LIU S, ZHU F, LIU H, et al. A core leader based label propagation algorithm for community detection[J]. *China Communications*, 2016, 13(12): 97–106.
- [14] LI W, HUANG C, WANG M, et al. Stepping community detection algorithm based on label propagation and similarity[J]. *Physica A: Statistical Mechanics and its Applications*, 2017, 472: 145–155.
- [15] SHANG R, ZHANG W, JIAO L, et al. Circularly searching core nodes based label propagation algorithm for community detection[J]. *International Journal of Pattern Recognition and Artificial Intelligence*, 2016, 30(8): 24–46.
- [16] LANCICHINETTI A, FORTUNATO S, RADICCHI F. Benchmark graphs for testing community detection algorithms[J]. *Physical Review E*, 2008, 78(4): 46–61.
- [17] NEWMAN M E J. Fast algorithm for detecting community structure in networks[J]. *Physical Review E*, 2004, 69(6): 66–78.

This work is partially supported by the Science and Technology Project of Tianjin City (16ZXHLSF0023), the Science and Technology Project of Hebei Province (17210305D).

GU Junhua, born in 1966, Ph. D., professor. His research interests include intelligent information processing, data mining.

HUO Shijie, born in 1993, M. S. candidate. His research interests include data mining, computer simulation.

WANG Shoubin, born in 1994, M. S. candidate. His research interests include data mining, computer simulation.

TIAN Zhe, born in 1994, M. S. candidate. His research interests include data mining.

(上接第1308页)

- [18] PAN S J, NI X, SUN J T, et al. Cross-domain sentiment classification via spectral feature alignment[C]// *WWW 2010: Proceedings of the 19th International Conference on World Wide Web*. New York: ACM, 2010: 751–760.
- [19] TOMMASI T, ORABONA F, CAPUTO B. Learning categories from few examples with multi model knowledge transfer[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014, 36(5): 928–941.
- [20] SZEGEDY C, IOFFE S, VANHOUCKE V, et al. Inception-v4, Inception-ResNet and the impact of residual connections on learning[EB/OL]. [2017-05-10]. <https://arxiv.org/abs/1602.07261>.

1602.07261.

This work is partially supported by the National Natural Science Foundation of China (41776142), the Support Project of Shanghai Science and Technology Commission (1439190400).

WANG Keli, born in 1990, M. S. candidate. His research interests include deep learning, image recognition, data mining, artificial intelligence.

YUAN Hongchun, born in 1971, Ph. D., professor. His research interests include intelligent computing, intelligent information processing.