

多点种子预划分的二阶段社区发现算法

佟 帅, 陈德运, 杨海陆

(哈尔滨理工大学 计算机科学与技术学院, 哈尔滨 150080)

摘 要: 社区发现是在线社交网络研究领域中的重要内容, 基于种子扩张的社区发现算法具有时间复杂度低、识别精度高以及不受社区形态限制等特点, 近年来在网络局部社区发现任务中得到了广泛的应用。然而, 该方法在种子选取时没有考虑种子之间的关联性, 因此识别出的社区结构个数较多、结构松散。针对这一问题, 提出一种基于多点种子预划分的二阶段社区发现算法。首先识别网络中的高影响力节点, 利用 K -means 算法将高影响力节点加以聚合, 得到高影响力社区簇。然后提出一种吸引力度量函数, 选择性的将网络中的剩余节点合并到社区簇以完成社区识别任务。实验结果表明, 二阶段社区发现方法能够发现尺寸较大, 个数较少的社区结构, 进而在中观层面捕捉群组之间的关联性。

关键词: 复杂网络; 局部社区发现; 种子扩张; 节点影响力; K -means 算法

DOI: 10.15938/j.jhust.2021.04.011

中图分类号: TP391.4 文献标志码: A 文章编号: 1007-2683(2021)04-0078-09

Two-stage Community Discovery Algorithm for Multi-point Seed Prepartition

TONG Shuai, CHEN De-yun, YANG Hai-lu

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

Abstract: Community detection is an important content in the field of online social networks research. The community detection algorithm based on seed expansion has the characteristics of low time complexity, high recognition accuracy, and is not restricted by the shape of the community. In recent years, it has been widely used in local community discovery of complex networks. However, this method does not consider the correlation between seeds when selecting seeds, so the number of identified community structures is large and the structure is loose. Aiming at this problem, a two-stage community discovery algorithm for multi-point seed prepartition was proposed. First, high-impact nodes in the network are identified, and high-impact nodes are aggregated using the k -means algorithm to obtain high-impact community clusters. Next, an attractiveness measurement function is proposed to selectively merge the remaining nodes in the network into the community cluster to complete the community identification task. The experimental results show that the two-stage community discovery method can find community structures with larger sizes and fewer numbers, and then capture the association between groups at the meso level.

Keywords: complex network; local community detection; seed expansion; node influence; K -means algorithm

收稿日期: 2020-06-01

基金项目: 国家自然科学基金(61402126, 61602133); 黑龙江省自然科学基金面上项目(F2016024); 黑龙江省博士后资助经费(LBH-Z15095); 黑龙江省普通高等学校创新人才培养计划(UNPYSCT-2017094)。

作者简介: 佟 帅(1992—), 男, 硕士研究生;

杨海陆(1985—), 男, 博士, 副教授, 硕士研究生导师。

通信作者: 陈德运(1962—), 男, 博士, 教授, 博士研究生导师, E-mail: chendeyun@hrbust.edu.cn.

0 引言

现实世界的复杂网络通常可以抽象为图模型,图中节点代表现实世界的实体,两个节点之间的链接代表实体之间的关系。社区^[1]是复杂网络中的稠密子图,保证了社区内部节点之间的链接较为紧密,社区之间节点的链接较为稀疏^[2-3]。探索社区结构有助于人们理解复杂网络的自组织以及群聚特性,是复杂网络中观层次最重要的属性之一。

从社区的层次化角度来看,已有的社区识别算法可分为全局优化算法和局部优化算法两种。前者在宏观角度寻找社区结构,描述了宏观层面网络节点的自组织特性,适合以数理统计为目的的网络特征分析^[1-2]。后者从微观层面刻画节点的局部倾向性,更适合发现社区的演化规律及形成过程。基于种子节点扩张的社区发现算法^[4],就是局部优化算法的典型代表。

2009年,Lancichinetti等^[5]提出重叠社区发现算法 LFM,根据定义的自适应函数 Fitness 进行基于种子扩张的社区识别。Coscia等^[6]提出的 DEMON 算法以整个网络节点为起始种子,利用标签传播算法识别网络中的局部社区结构。Baumes等^[7]提出一种链接聚合方法。先按照递增或递减原则对节点进行排序,作为初始社区,如果节点的添加不能提高任何社区的密度,则该节点将作为新的种子节点,并生成新社区。李婕等^[8]采用基于派系过滤的算法选择种子节点进行社区识别。Su等^[9]基于随机游走算法,使用紧密连接子图作为社区识别的初始种子社区。

提高基于种子扩张的社区识别性能的关键在于种子节点的选取,近年来有研究者提出借助影响力分析方法,增加初始种子节点位于社区内核的概率。Clauset等^[12]依次探索网络节点,推断给定节点所在社区。Luo等^[13]将度的概念从单节点扩展到子图,在此基础上给出了网络模块化的定义,以此进行社区识别。Chen等^[14]提出一种新的局部社区测度,先提取所有可能的候选社区,然后对社区层次进行优化。吴英俊等^[15]提出 LS 算法,通过分析社区与节点之间的链接相似性来寻找局部社区结构。Fan-rong等^[16]提出一种基于最大团扩展的局部社区检测算法 LCDMC,首先找到包含源节点的最大派系集合,然后利用贪婪优化方法扩展社区。Yao等^[17]分析了高影响节点在社区检测中的作用。齐金山

等^[18]提出了一种结合 Jaccard 系数的节点影响力计算公式,提高了算法对星形社区的匹配性。

上述社区识别方法虽然在特定领域具有一定的性能优势,但普遍存在以下两方面问题。首先,上述方法在种子选取时没有考虑种子之间的关联性,导致识别出社区结构个数较多、稳定性较差,不利于在中观层面上挖掘群组之间的关联性。其次,由于社会网络结构的多样性,上述方法在影响力识别时忽略了网络的结构特性,导致网络中的高影响力节点呈现出较低的影响力评分的假象,这使得识别出的社区结构较为松散,内聚性较差。

为了解决上述问题,提出一种多点种子预划分的二阶段社区发现算法(two-stage community detection algorithm, TSCDA),其创新之处主要体现在以下三方面。首先,在节点的影响力计算时融入边介数属性,增强节点影响力的结构因素;其次,通过预划分高影响力节点生成骨干网络,增强种子的结构密度;最后,以稠密子图代替节点作为种子进行扩张,增加了社区结构的稳定性。仿真结果表明,TSCDA 能够挖掘出数量较少、规模较大的社区结构,并在模块度以及 F-Score 等指标上表现出一定的性能优势。

1 基于节点影响力识别的种子节点选取方法

用无向图 $G = (V, E)$ 表示社交网络,其中 V 代表网络中个数为 n 的节点集合, E 代表网络中个数为 m 的链接关系集合。图 1 给出了一种具有强聚合特性的社交网络结构(社区网络)。

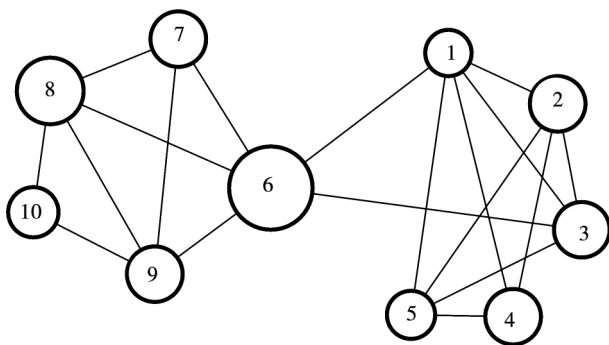


图 1 社区网络

Fig1 Community network

社会网络中,节点 u 的邻居集合定义为:

$$N(u) = \{v: v \in V, (u, v) \in E\} \quad (1)$$

这里 u 和 v 代表网络节点, V 代表网络中的节

点集,如果 u 和 v 之间存在边 (u, v) , 则 (u, v) 属于边集 E 。节点 u 的节点度 $D(u)$ 定义为与 u 直接相连的边的个数(或邻居元素的个数), 满足 $D(u) = |N(u)|$ 。

为了在影响力计算中引入结构的相似属性, 提出从节点相似性以及链接相似性双重角度构造影响力评价函数。

Jaccard 系数 J_{uv} 用于比较社会网络中节点之间的结构相似性。其定义为:

$$J_{uv} = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|} \quad (2)$$

$N(u)$ 和 $N(v)$ 分别代表节点 u 和节点 v 的邻居节点。 J_{uv} 取值范围为 $[0, 1]$, 其值越大, 节点 u 和节点 v 的共有邻居节点就越多, 邻域结构也就相对越稠密。

边介数用来度量社会网络中所有节点间的最短路径中经过该边的路径的数目占最短路径总数的比例。边介数值越大, 该边越有可能成为连接社区内部节点的重要途径。其定义为:

$$E^{BET}(e_{uv}) = \sum_{s < t} |g_{st}^{uv}| \quad (3)$$

其中 g_{st}^{uv} 表示网络中任意节点 s 与任意节点 t 之间的最短路径同时经过节点 u 和 v (即边 e_{uv}) 的条数。根据式(3), 可以计算出图1各边的边介数指数, 例如节点8和节点10, 即 $(e_{8,10})$ 之间边强度为 $E^{BET}(e_{8,9}) = 11$; 节点9和节点10, 即 $(e_{9,10})$ 之间边强度为 $E^{BET}(e_{9,10}) = 11$ 。边介数反映了社区内部链接的稠密程度, 是增加社区稳定性的关键参数。

受牛顿万有引力定律的启发, 提出一种基于Jaccard系数以及边介数的影响力计算方法。社会网络节点 u 对节点 v 的影响力评分定义为:

$$NG(u, v) = G \frac{D(u) D(v) E^{BET}(e_{uv})}{d^2(u, v)} \quad (4)$$

这里 $D(u)$ 和 $D(v)$ 分别表示节点 u 和节点 v 的度。由于节点度在局部环境反映了节点链接能力, 因此在本文中被用来衡量社会网络节点的质量。节点之间的距离 $d(u, v)$ 用节点间的相异程度加以衡量, 满足 $d(u, v) = 1 - J_{uv}$ 。其中 J_{uv} 代表节点 u 和节点 v 的 Jaccard 相似度。 G 是影响力常量, 在实验中通常取值为1。

节点在局部社区的影响力评分 $I_{score}(u)$ 定义为该节点对其所有邻居的节点影响力评分之和。进而有:

$$I_{score}(u) = \sum_{v \in N(u)} G \frac{D(u) D(v) E^{BET}(e_{uv})}{(1 - J_{uv})^2} \quad (5)$$

式(5)中, $v \in N(u)$ 表明节点 v 是节点 u 的邻居节点。 $I_{score}(u)$ 的值越大, 说明节点 u 在局部环境的影响强度越大, 表明越有可能成为种子节点。根据式(5), 可以计算出图1各节点的影响力分数。例如节点10的影响力分数:

$$I_{score}(u_{10}) = \frac{2 \times 4 \times 11}{(1 - \frac{1}{5})^2} + \frac{2 \times 4 \times 11}{(1 - \frac{1}{5})^2} = 275 \quad (6)$$

因此节点10在网络中影响力分数 $I_{score}(u_{10}) = 275$ 。

下面给出基于节点影响力识别的种子节点选取过程。算法通过式(5)计算社会网络每个节点的影响力评分。其伪代码如下:

算法1 具有高影响力的初始种子选取算法

输入: 社会网络 $G = (V, E)$

输出: 高质量节点集合 S

1) 初始化集合 S 为空集;

2) 遍历社会网络的所有节点, 根据公式(5)计算社会网络所有节点的节点影响力评分;

3) 对于任意节点, 如果它的影响力评分大于其任一邻居的影响力评分, 则为高影响力种子节点, 将其存储到集合 S ;

4) 输出集合 S 。

根据算法1, 计算图1中节点1到节点10的影响力评分, 具体如表1所示。在本文中, 如果一个节点的影响力分数大于其任一邻居的影响力评分, 则为高影响种子节点。例如, 节点8的邻居节点为6, 7, 9, 10。通过表1可知, 节点8的影响力评分大于节点6, 7, 9, 10的影响力评分, 因此节点8为高影响力节点。

表1 图1中各节点影响力评分排序

Tab. 1 In Figure 1, influence rating of each node is ranked

顶点	$I_{score}(u)$
3	2 189
1	2 155
5	1 253
8	1 249
9	1 010
7	673
2	607
4	573
6	437
10	275

2 多点种子预划分的二阶段社区发现算法

本节给出多点种子预划分的二阶段社区发现算法。首先,根据算法1选取网络中的高影响力种子节点,对这些节点利用 K -means 聚类算法进行初始划分,识别出由高质量节点组成的种子社区即骨干社区;然后提出一种基于吸引力度量的社区识别算法将社会网络中其余节点按特定规则有选择性的加入骨干社区中,从而完成社区划分。该方法的优势在于充分考虑了种子节点之间稠密性以及关联性,有助于提高社区的稳定程度。

2.1 基于 K -means 聚类的骨干社区识别

本文采用 K -means 算法作为骨干社区的初始化算法,其距离度量公式在式(7)给出。

$$Dis(A, B) = \frac{\min Dis(A, B)}{comNeighbor(A, B) + C} \quad (7)$$

式中: $\min Dis(A, B)$ 为在社会网络中节点 A, B 间最短路径长度; $comNeighbor(A, B)$ 为节点 A, B 间共同邻居数量; C 为迭代中心,初始值为0; $Dis(A, B)$ 为 A, B 两点的距离,该值与 A, B 两点的最短路径长度呈正相关,与 A, B 两点的共同邻居数量呈负相关。

K -means 算法在迭代时依靠簇内所有节点的平均坐标设定新中心。然而社区不存在坐标,因此需要重新定义中心选取方式。本文选取到簇内其他节点的距离之和最小的节点作为新的中心节点,具体如式(8)所示。

$$C_{new} = \arg \min_p \left(\sum_{k \in N(i)} Dis(p, k) \right) \quad (8)$$

式中: $N(i)$ 表示第 i 个社区内的节点集合。 K -means 算法的最终输出1棵骨干社区层次化树,采用模块度函数在层次化树上进行分割,获取质量最高的社区划分。模块度函数 Q 是 Newman^[9] 在2004年提出的概念,用以评价社区划分的质量,其定义为:

$$Q = \sum (e_{ii} - a_i^2) \quad (9)$$

式中: e_{ii} 代表社区 i 中边的期望; a_i 代表链接到社区 i 中的边数的期望。模块度的形式化定义是社区内的边数减去随机产生的边的期望,因此模块度越高,社区内外边比例就越大,社区划分结果就越好。

我们发现,算法在层次聚类时模块度函数 Q 的取值并不是单调递增或单调递减的,因此每当有社区进行合并操作,就需要重新计算 Q 值,能对产生最大 Q 值的两个社区合并。为简化计算,每次仅计

算 Q 的变化部分,即向 Q 值增大最多或者减少最小的方向进行社区合并。 Q 函数的变化量计算公式如下所示。

$$\Delta Q = e_{ij} + e_{ji} - 2a_i a_j = 2(e_{ij} - a_i a_j) \quad (10)$$

对能产生最大 Q 值的两社区进行合并,直到所有节点合并至同一社区。算法的伪代码如下:

算法2 利用 K -means 聚类算法实现种子节点初始划分

输入: 3个种子节点集合,社会网络 G

输出: 骨干社区集合 $BC = \{BC_1, BC_2, \dots, BC_n\}$

1) 选取这3个种子节点进行初始聚类;

2) 根据距离度量式(7)计算任意两节点之间的距离;

3) 比较任意两点间的距离,将节点划分至距离最近的节点簇内;

4) 对于每个簇内,分别计算簇内每个节点与簇内的其他节点距离之和,取和最小的节点作为新的中心节点;

5) 迭代步骤4)、5),直到中心节点不发生改变;

6) 对 K -means 算法产生的社区按照式(10)计算两两之间合并产生的 ΔQ ,将使 ΔQ 增加或减少的两个社区进行合并,并按照式(9)计算当前 Q 值,保存 Q 值及当前社区划分;

7) 对步骤7)进行迭代,直到所有节点都包含在一个社区或不能进行社区合并;

8) 比较每次合并的 Q 值大小,取 Q 值最大的社区作为最终的结果。

2.2 基于吸引力度量的社区识别过程

算法2结束后,会对社会网络中影响力较高的前3个节点进行预划分,识别出由高影响力节点组成的种子社区即为骨干社区。而网络中影响力较低的 $n-3$ 个零散节点可能处于孤立状态,因此需要将其合并到骨干社区之中。提出一种基于影响力分析的吸引力度量函数,将网络中零散节点添加到对其吸引力较高的骨干社区。对于任意骨干社区 BC 和节点 u , BC 对 u 的吸引力度量函数 $attract(BC, u)$ 的定义为:

$$attract(BC, u) = \frac{\sum_{v \in BC \wedge v \in N(u)} NG(u, v)}{I_{score}(u)} \quad (11)$$

式中: $\sum_{v \in BC \wedge v \in N(u)} NG(u, v)$ 表示节点 u 对当前所在骨干社区的所有邻居节点的影响力评分之和; $I_{score}(u)$ 表示节点 u 的影响力评分。 $attract(BC, u)$ 的值越大,节点 u 受骨干社区 BC 吸引而加入社区 BC 的可能性就越大,反之节点 u 成为骨干社区 BC

内部节点的概率较低。

设定了吸引力阈值 β , 对于节点 u 以及任意社区 BC , 若 $\text{attract}(BC, u) > \beta$, 则节点 u 属于骨干社区 BC 。节点 u 可能同时加入多个社区结构, 因此社区具有重叠性。在图 1 中, 假设骨干社区 BC_1 包含的节点集合为 $\{1, 2, 3, 4, 5\}$, 骨干社区 BC_1 对节点 6 的吸引力函数为:

$$\text{attract}(BC_1, 6) = \frac{N(6, 1) + N(6, 3)}{F(6)} \approx 0.25 \quad (12)$$

假设 BC_2 的骨干社区包含节点集合为 $\{7, 8, 9, 10\}$, 骨干社区 BC_2 对节点 6 的吸引力为:

$$\text{attract}(BC_2, 6) = \frac{N(6, 7) + N(6, 8) + N(6, 9)}{F(6)} \approx 0.52 \quad (13)$$

若 $\beta = 0.4$, 则节点 6 不属于骨干社区 BC_1 , 属于骨干社区 BC_2 。

下面给出基于种子扩张的社区发现算法, 算法的伪代码如下:

算法 3 基于种子扩张的社区发现算法

输入: 高影响力节点构成的社区集合 BC , 社区网络 G

输出: 社区划分结构

- 1) 将网络中高影响力节点标记为 true, 其余 $n - 3$ 个节点标记为 false;
- 2) 将网络中标记为 false 的节点按照式 (11) 依次对骨干社区集合 $BC = \{BC_1, BC_2 \cdots BC_n\}$ 进行计算;
- 3) 对每个标记为 false 的节点按照不同的骨干社区的计算, 选取 $\text{attract}(BC, u)$ 最大的值并将该节点加入到对应的骨干网络;
- 4) 将加入骨干社区网络的节点标记为 true;
- 5) 重复执行步骤 2), 步骤 3) 和步骤 4), 直到社区网络中所有节点都标记为 true;
- 6) 输出社区划分结构。

为了挖掘网络的社区结构, 首先将网络除 3 个高影响力节点外的节点都标记为 false, 若节点划分到一个社区中, 则该节点标记为 true。

在基于吸引力度量的种子扩张过程的伪代码中, 第一步为遍历 n 个高影响力评分节点社区集合。第二步以及第三步为计算集合 BC_i 中对 $n - 3$ 吸引力度量函数值 $\text{attract}(BC_i, v)$ 不小于给定的阈值 β 的节点。若节点 $\text{attract}(BC_i, v)$ 值大于或等于阈值, 将该节点合并到当前的社区中, 若一个节点属于多个

社区, 则称该节点为重叠节点。将该节点加入到 $\text{attract}(BC_i, v)$ 值最大的骨干社区中。最后将加入骨干社区的节点标记为 true, 得到最终的社区结构划分结果。社区合并本质上是一种贪心策略, 然而由于网络本身存在稀疏性, 在实际的种子扩张中零散节点邻接社区要远小于其邻接的社区数, 这有效的控制了算法的时间开销。

2.3 算法的时间复杂度分析

假设社会网络 G 包含有 n 个节点和 m 条边, 算法 1 在求解节点的影响力强度时, 时间复杂度为 $O(n)$ 。在算法 2 中基于 K -means 聚类算法将 $O(l)$ 个节点进行合并得出骨干网络, 其时间复杂度为 $O(kl)$, 最坏的情况下将 $O(n)$ 个节点进行合并, 故算法 2 的时间复杂度为 $O(kn)$ 。在算法 3 中, 在社区扩展阶段将 $O(n)$ 个节点加入到 $O(k)$ 个骨干社区中, 该过程重复执行 $O(n)$ 次, 所以找到所有的最终社区的时间复杂度为 $O(kn^2)$, 若 $k < n$, 则时间复杂度为 $O(n^2)$, 在最坏的情况下有 $O(n)$ 个骨干网络, 则找到所有的最终社区的时间复杂度为 n^3 。经过以上分析, 本算法最差情况下时间复杂度为 $O(n + kn + n^3) = O(n^3)$ 。

3 实验结果与分析

本节给出算法在人工合成网络以及真实数据集上的运行结果。实验的运行环境为 Intel Core i5-7300HQ 2.5 GHz 处理器, 8 GB 内存, Windows 10 操作系统, 算法采用 Python 与 Matlab R2016a 混合编程。

3.1 NMI 指数

Danon 等^[24] 提出归一化互信息 (normalized mutual information, NMI), 度量社区集合之间的相似性。NMI 基于混淆矩阵 N , 其中行表示真实的社区结构, 列表示算法生成的社区。 N_i 表示矩阵 N 第 i 行所有元素的总和, N_j 表示矩阵 N 第 j 列所有元素的总和。 N_{ij} 是矩阵 N 的元素, 表示同时属于真实社区 i 和算法生成社区 j 的节点数量。NMI 公式定义为:

$$NMI = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} N_{ij} \log\left(\frac{N_{ij} N}{N_i N_j}\right)}{\sum_{i=1}^{C_A} N_i \log\left(\frac{N_i}{N}\right) + \sum_{j=1}^{C_B} N_j \log\left(\frac{N_j}{N}\right)} \quad (14)$$

式中: C_A 代表网络中真实社区的数量; C_B 代表由算法生成社区的数量。如果探测出的社区结构和真实

网络的社区结构相同则 $NMI = 1$; 反之, 如果所探测的社区结构和真实网络完全不同, 则 $NMI = 0$ 。传统意义上的归一化互信息被用来量化两个分布之间的差异性, 为了使之适用于社区差异性量化, 实验中采用了 NMI 的简化版本^[21]。

3.2 F-SCORE

F -Score 也是常用的社区发现算法度量指标。首先定义算法生成社区在全部社区中的比例。

$$Recall = \frac{C_R \cap C_F}{C_R} \quad (15)$$

式中, C_R 代表高质量节点所在的真实社区, C_F 代表高质量节点所在的探测社区。

$Precision$ 是正确划分节点的数量除以节点在 C_F 中的数量, 其定义为:

$$Precision = \frac{C_R \cap C_F}{C_F} \quad (16)$$

结合式(15) 以及式(16), F -Score 定义为:

$$F\text{-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (17)$$

3.3 人工合成网络社区识别结果

利用 LFR-Benchmark 提供的 MATLAB 数据生成器生成网络, 网络中的部分重要参数设置如下。首先, 生成网络规模为 $N = 1\,000$ 的复杂网络, 其中最小社区尺寸为 10、最大社区尺寸为 100。接下来设置网络的平均度为 5、最大节点度为 25、混合参数 u 的取值范围为 0.1 ~ 0.8。在 LFR Benchmark 中, 混合参数是一个独立参数, 其值越大社区越难以发现, 因此通常用来刻画算法的鲁棒性。

首先验证了不同混合参数下 K -means 聚类算法中的 k 取值不同时 NMI 指标变化情况, 实验结果如图 2 所示。可以看出, 初始时社区质量会随种子社区的不断增多而不断上升, 但达到某一极值时, 社区质量开始下降。这是因为初始种子过多会导致社区个数增加, 进而降低社区内部的紧密程度。另一个发现是, 社区结构较为模糊的网络中 ($u = 0.8$), 需要更多的种子社区才能达到较好的发现效果。

接下来, 验证 TSCDA 算法的性能。与 Clauset^[12]、LWP^[13]、Chen^[14]、LS^[15]、LCDMC^[16] 以及 VI^[17] 6 种社区识别算法在人工合成网络社区进行比较, 为了保证算法结果的客观性, 将相关参数按原有文献加以设定, 算法的 NMI 指标以及 F -Score 在图 3 以及图 4 给出。

如图 2 及图 3 所示, 当混合参数 u 等于 0.1 时,

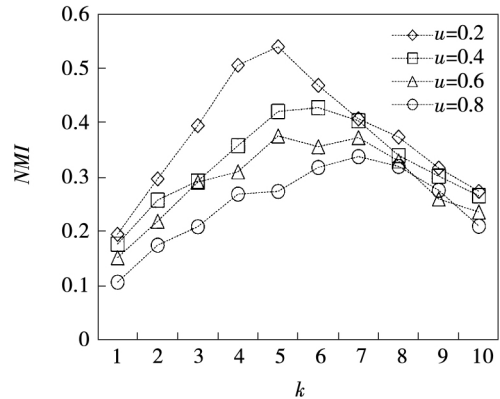


图2 不同 k 值 Q 函数的变化

Fig. 2 Function Q under different value of k

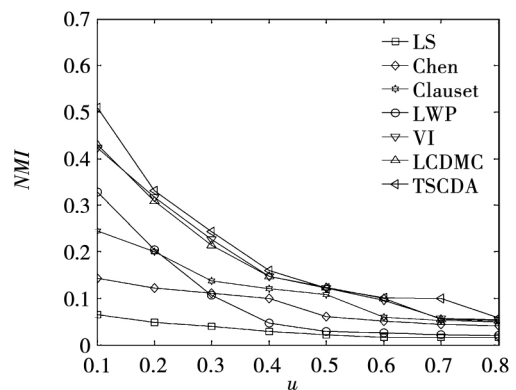


图3 各算法在人工数据集上的 NMI 指标

Fig. 3 NMI index of each algorithm on artificial data set

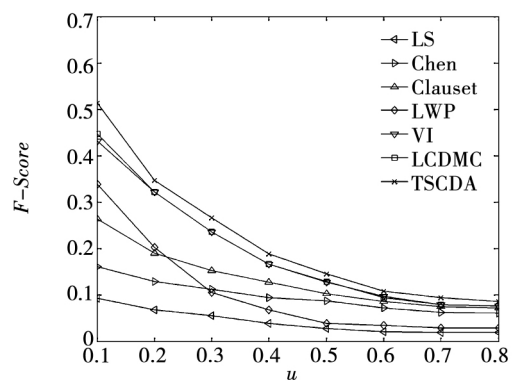


图4 各算法在人工数据集的 F -Score 指标

Fig. 4 F -Score index of each algorithm on artificial data set

TSCDA 在 NMI 以及 F -Score 的性能指标明显优于其它算法, 具有一定的性能优势。随着混合参数 u 的不断加, TSCDA 算法和其他 6 种算法的 NMI 指标以及 F -Score 均呈现明显的下降趋势, 当混合参数 u 等于 0.5 以及 0.6 时, TSCDA 算法略优于 VI 算法和

LCDMC 算法, 优于 LWP 以及 LS 算法; 当混合参数 u 等于 0.7 以及 0.8 时, 各算法性能较为接近, 但总体来看, TSCDA 算法仍然要优于其他 6 种社区识别方法, 这表明 TSCDA 算法在结构指标上更贴近于内嵌社区。

3.4 真实网络社区识别结果

为了验证 TSCDA 算法在真实网络社区识别的性能, 本节选取了在 5 种真实网络上, 对比 TSCDA 算法与前文 6 种社区识别算法的性能差异。表 2 列出了所选择的真实网络的网络特征, 其中 $Node$ 代表节点个数、 $Link$ 代表链接条数、 $\bar{d}$ 表示网络的平均

度、 $|\bar{C}|$ 表示真实社区的平均大小、 $Community$ 表示社区个数。我们从 Amazon 社区中移除了前 5 000 个子社区, 保留了它的不同的社区。

还在 DBLP 计算机学科文献数据网络上测试了 TSCDA 算法与 Clauset 算法、LWP 算法、Chen 算法、LS 算法、LCDMC 算法以及 VI 算法的 NMI 指标以及 $F-Score$ 指标。DBLP 文献网络的特性如表 3 所示。 $|C|$ 代表社区规模的范围, 例如, $(0, 10]$ 代表样本网络包含的社区大小大于 0, 小于等于 10。我们同样移除了 DBLP 网络前 5 000 个社区中的子社区, 并保留了它们不同的社区结构。

表 2 真实网络的特点
Tab. 2 Characteristics of real networks

Network	Node	Link	$\bar{d}$	Community	$ \bar{C} $	References
Karate	34	156	4.58	2	17.00	[22]
Dolphins	62	318	5.12	2	31.00	[23]
Polbooks	105	880	8.38	3	35.00	[24]
Football	115	1 226	10.66	12	9.58	[25]
Amazon	16 716	97 478	5.83	1 163	15.16	[26]

表 3 DBLP 网络的特点
Tab. 3 The characteristics of DBLP network

Network(Lable)	$ C $	Node	Link	$\bar{d}$	Community	$ \bar{C} $
DBLP(0)	$(0, 10]$	24 210	131 300	5.42	3 524	7.28
DBLP(1)	$(10, 20]$	14 540	91 872	6.31	1 100	13.85
DBLP(2)	$(20, 30]$	3 240	21 708	6.70	136	24.12
DBLP(3)	$(30, 40]$	1 338	10 352	7.73	39	34.30
DBLP(4)	$(40, 50]$	611	4 204	6.88	14	43.64
DBLP(5)	$(50, 100]$	583	3 940	6.75	9	64.77
DBLP(6)	$(100, 200]$	1 492	9 670	6.48	10	150.50
DBLP(7)	$(200, 300]$	1 341	6 618	4.93	6	232.83
DBLP(8)	$(300, 400]$	2 133	9 964	4.67	6	355.66
DBLP(9)	$(400, 500]$	834	5 670	6.79	2	417.00
DBLP(10)	$(500, 100]$	10 705	70 208	6.55	14	785.50

表 4 给出了各算法在真实网络上的 NMI 指标以及 $F-Score$ 指标。相比其它 6 种方法, TSCDA 具有一定的性能优势。一方面, TSCDA 基于节点影响

力扩展社区, 因此在识别社区成员时准确度更高。另一方面, 本文在节点影响力计算中加入了边介数属性, 不仅考虑了节点自身的重要程度, 还考虑了链

接在局部区域中的重要程度。LWP 和 LS 算法在个别网络中显示出比 TSCDA 更好的性能,例如在 Football 网络中 LWP 算法的 *NMI* 指标高于 TSCDA 约 0.09、在 Amazon 网络中 LS 算法的 *F-Score* 高于 TSCDA 约 0.02。一个可能的原因是 Football 网络的平均节点度较高,使得各算法在社区识别时更容易

发现结构较为紧凑的局部结构。TSCDA 算法在 Dolphins 网的准确性有所下降,这是由于 Dolphins 网络中的链接关系依靠于节点之间的接触历史,因此影响力识别时准确度一般。但是相比其它 6 种算法 TSCDA 算法仍然表现出一定的稳定性,进一步说明了 TSCDA 算法具有较高的鲁棒性。

表 4 算法在真实网络上性能指标
Tab. 4 Algorithm performance indicators on real networks

Network	Criteria	TSCDA	Clauset	LWP	Chen	LS	LCDMC	VI
KArate	NMI	1.00	0.29	0.51	0.41	0.45	0.50	0.39
	F-Score	1.00	0.64	0.71	0.48	0.76	0.79	0.80
dolphins	NMI	0.45	0.18	0.28	0.06	0.08	0.44	0.37
	F-Score	0.73	0.42	0.41	0.19	0.25	0.70	0.73
polbooks	NMI	0.48	0.26	0.29	0.10	0.01	0.39	0.39
	F-Score	0.76	0.49	0.46	0.23	0.03	0.62	0.62
football	NMI	0.51	0.51	0.60	0.55	0.53	0.53	0.81
	F-Score	0.66	0.66	0.63	0.61	0.56	0.64	0.88
amazon	NMI	0.72	0.56	0.62	0.63	0.74	0.91	0.91
	F-Score	0.74	0.61	0.65	0.67	0.76	0.93	0.93

图 5 和图 6 给出了各算法在 DBLP 引文网络上的性能指标。实验结果显示,当社区规模在 0 到 10 时 (Label 0), TSCDA 在社区识别性能上略低于其他 6 种算法表现的差。然而当社区规模大于 10 时 (Label 1 ~ 10), TSCDA 的表现性能明显优于 Clauset、LWP、Chen、LS、LCDMC 和 VI 算法,这表明 TSCDA 善于挖掘个数较多的社区结果,社区粒度更加细致。

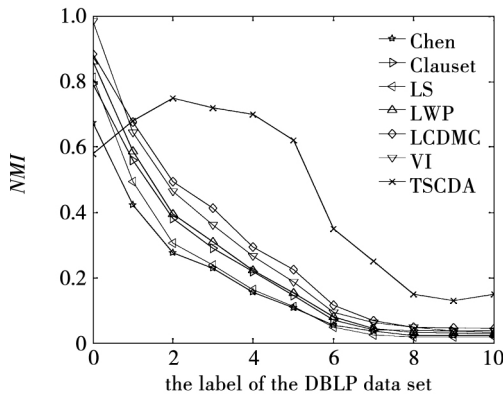


图 5 各个算法在 DBLP 网络上 *NMI* 指标
Fig. 5 *NMI* indicators of various algorithms on the DBLP network

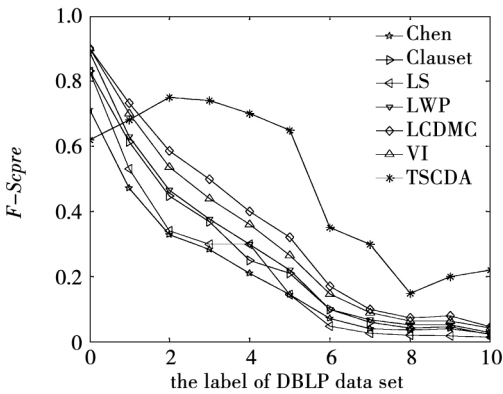


图 6 各算法在 DBLP 网络上 *F-Score* 指标
Fig. 6 *F-Score* indicator of each algorithm on DBLP network

4 结 论

为了解决传统社区识别方法社区稳定性较差这一问题,从节点相似性以及链接相似性两个角度入手,提出一种多点种子预划分的社区发现算法 TSCDA。首先,在种子扩张算法的基础上,提出利用高质量节点构建骨干社区进行社区识别,提高社区结构的致密性;其次,将边介数属性加入到节点影响力

的计算中,提高影响力计算中的结构相关性。人工合成网络和真实网络数据集上的仿真结果验证了TSCDA 算法的效率和识别性能。

参考文献:

- [1] SANTO Fortunato. Community Detection in Graphs [J]. Physics Reports, 2009, 486(3) : 26.
- [2] JAVED M A, YOUNIS M S, LATIF S, et al. Community Detection in Networks: A Multidisciplinary Review [J]. Journal of Network and Computer Applications, 2018, 108: 87.
- [3] AMOR B, VUIK S, CALLAHAN R, et al. Community Detection and Role Identification in Directed Networks: Understanding the Twitter Network of the Caredata Debate [J]. Dynamic Networks and Cyber-security, 2015, 25(2) : 19.
- [4] BAZZI M, PORTER M A, WILLIAMS S, et al. Community Detection in Temporal Multilayer Networks, with an Application to Correlation Networks [J]. Multiscale Modeling & Simulation, 2016, 14(1) : 1.
- [5] SHARMA S, SINGH A. An Efficient Method for Link Prediction in Complex Multiplex Networks [C] // 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) . IEEE, 2015: 452.
- [6] DING X, ZHANG J, YANG J. A Robust Two-stage Algorithm for Local Community Detection [J]. Knowledge Based Systems, 2018, 14(7) : 62.
- [7] LANCICHINETTI A, FORTUNATO S, KERTÉSZ, JÁNOS. Detecting the Overlapping and Hierarchical Community Structure in Complex Networks [J]. New Journal of Physics, 2009, 11(3) : 15.
- [8] COSCIA M, ROSSETTI G, GIANNOTTI F, et al. DEMON: a Local-First Discovery Method for Overlapping Communities [C] // Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012: 19.
- [9] BAUMES J, GOLDBERG M, MAGDON-ISMAL M. Efficient Identification of Overlapping Communities [C] // International Conference on Intelligence and Security Informatics, 2005: 11.
- [10] 李婕, 王兴伟, 郭静, 等. 面向移动通信网络的局部扩张群组构造方法 [J]. 东北大学学报(自然科学版), 2017, 38(12) : 1691.
- LI Jie, WANG Xingwei, GUO Jing, et al. Clique Percolation Based Local Fitness Method for User Clustering in Telecommunication Network [J]. Journal of Northeastern University Natural (Science) , 2017, 38(12) : 1691.
- [11] SU Y, WANG B, CHENG F, et al. An Algorithm Based on Positive and Negative Links for Community Detection in Signed Networks [J]. Scientific Reports, 2017, 7(1) : 10874.
- [12] CLAUSET, ARON. Finding Local Community Structure in Networks [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2005, 72(2) : 26.
- [13] LUO F, WANG J Z, PROMISLOW E. Exploring Local Community Structures in Large Networks [J]. Web Intelligence & Agent Systems, 2008, 6(4) : 387.
- [14] JIYANG Chen, OSMAR Zaiane, RANDY Goebel. Local Community Identification in Social Networks [C] // International Conference on Advances in Social Network Analysis & Mining. IEEE Computer Society, 2009: 79.
- [15] YING JUN, WU, HAN, et al. Local Community Detection Using Link Similarity [J]. Journal of Computer Science & Technology, 2012, 23(3) : 69.
- [16] FANRONG M, MU Z, YONG Z, et al. Local Community Detection in Complex Networks Based on Maximum Cliques Extension [J]. Mathematical Problems in Engineering, 2014, 36(6) : 1.
- [17] YAOY, WU W, LEI M, et al. Community Detection Based on Variable Vertex Influence [C] // IEEE International Conference on Data Science in Cyberspace. IEEE, 2016: 145.
- [18] 齐金山, 梁循, 王怡. 基于种子节点选择的重叠社区发现算法 [J]. 计算机应用研究, 2017(12) : 20.
- QI Jinshan, LIANG Xun, WANG Yi. Overlapping Community Detection Algorithm Based on Selection of Seed Nodes [J]. Application Research of Computers, 2017(12) : 20.
- [19] NEWMAN M E J. Fast Algorithm for Detecting Community Structure in Networks [J]. Physical Review E, 2004, 69(6) : 133.
- [20] DANON, LEON, DUCH, et al. Comparing Community Structure Identification [J]. Journal of Statistical Mechanics, 2005, 206(9) : 69.
- [21] BAGROW, JAMES P. Evaluating Local Community Methods in Networks [J]. Journal of Statistical Mechanics: Theory and Experiment, 2008, 8(5) : 26.

(编辑: 温泽宇)