基于树核度的社交网络影响最大化问题

朱恩强[1,2],吴艳蕾[2],许宇光[2],牛云云[2]
(1. 广州大学计算科技研究院,广东广州 510006;2. 北京大学信息科学技术学院,北京 100871)

摘 要: 社交网络中的影响最大化问题是指对于给定的 k 值,寻找 k 个在特定传播模型下能够使得传播范围达到最大的节点. 此问题在常用的几种传播模型中都是 NP-难的. 目前虽然已经有很多近似求解的算法,但如何在较低的算法时间复杂度下,保证较大的传播范围仍然是求解该问题的一个挑战. 为此,本文提出了一种新颖的基于图的树核度理论的方法来求解社交网络影响最大化问题,并相应地给出了一个多项式时间的算法. 所提算法综合考虑了网络的结构特征和传播特征. 另外,我们将该算法与传统的随机、度以及贪心算法进行了比较. 实验结果表明,所提算法可以较快地找到能够使得传播范围较大的节点集合.

关键词: 树核度; 树核; 社会网络; 算法; 影响最大化; 传播模型
中图分类号: TN95 **文献标识码:** A **文章编号:** 0372-2112 (2019) 01-0161-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.01.021

Tree-Coritivity-Based Influence Maximization in Social Networks

ZHU En-qiang[1,2], WU Yan-lei[2], XU Yu-guang[2], NIU Yun-yun[2]
(1. *Institute of Computing Science and Technology, Guangzhou University, Guangzhou, Guangdong* 510006, *China*;
2. *School of Electronics Engineering and Computer Science, Peking University, Beijing* 100871, *China*)

Abstract: Influence maximization problem in social networks deals with finding a small subset of nodes, which could maximize the spread of influence. It has been proved that this problem is NP-hard under the commonly used diffusion models. Although many algorithms have been proposed to solve this problem approximately, it is still a challenge to guarantee the spread of influence within a low time complexity. For this, we propose a novel method based on tree-coritivity theory and give a polynomial-time algorithm, for finding the initial active nodes required in the influence maximization problem. Our algorithm considers both the structure and the propagation characteristics of a network. Moreover, by experiment, we compare this algorithm with other conventional node-selection methods such as Random, Degree and Greedy. The results demonstrate that the proposed algorithm can find the node set that can widely spread the information efficiently.

Key words: tree-coritivity; tree-core; social network; algorithm; influence maximization; diffusion models

1 引言

社交网络是由代表人(或组织)的节点构成的复杂社会结构. 它用来描述群组中个体之间的关系和交互[1],是它的成员之间进行信息,思想,以及影响的传播的基本介质. 作为社交网络分析的一个方面,研究信息传播和扩散是非常有价值的. 通过对社交网络中有益信息扩散机制的认识,我们可以更好地知道如何使它们传播的更快更广. 同时,通过对社交网络中有害信息扩散机制的了解,还可以使我们更早更有效地对它们进行预防从而阻止它们传播到更大的范围.

为研究社交网络中信息的传播和扩散过程,我们需要挖掘最具影响力的某些用户. 这就是社交网络中的影响最大化问题,即如何选择 k 个初始活跃节点,使得从这些节点开始传播信息,在传播过程结束后,信息传播的范围能够达到最大. Domingos 和 Richardson[2,3] 最早将影响最大化问题引入社交网络,并结合概率方法将此问题归纳为一个算法问题. Kempe 等人[4,5] 把这个问题描述成离散优化问题,并证明此优化问题是 NP-难的. 他们提出了一种贪心算法近似地计算最优解,并

收稿日期:2015-04-22; 修回日期:2016-07-17; 责任编辑: 郭游
基金项目:国家 973 重点基础研究发展计划(No. 2013CB329600, No. 2013CB329602, No. 2013CB329606); 国家自然科学基金(No. 61572046, No. 61572492, No. 61472433)

证明所求近似解能保证约 63% 接近最优解. 但是使用该方法计算传播范围时效率不高,耗时较多. 为此,Leskoves[6]提出了“Lazy-forward”的思想来优化算法的速度. 为有效地计算社交网络中信息传播的范围,学者们还提出了基于独立级联模型的 SPM 算法和 PMIA 算法[7],MIA 和 PMIA 模型[8],基于社区[9]和路径[10]的方法等. 另外,关于应用启发式方法研究影响最大化问题,Chen 等人[11]在独立级联模型中提出了“Degree Discount”方法,Jiang 等人[12]利用模拟退火法启发式求解,Jung 等人[13]提出了 IRIE (Influence Rank Influence Estimation)方法. 近年来,Li 等人[14]提出了一种用社区挖掘的方法来求解影响力最大化问题的算法. Zhu 等人[15]将半定规划应用到求解影响力最大化问题上. 他们考虑了现实网络中信息传递对时间的敏感特性,提出了一种新的传播模型,并且针对不同情况设计了两种使用半定规划求解算法,Cheng 等人[16]提出了一种迭代的排序框架来解决独立级联模型下的影响最大化问题. Cohen 等人[17]提出了概括影响力公式的问题,Lucier 等人[18]提出一种在独立级联模型下估计节点级联影响力的方法.

考虑到图的树核度是一种反映图的结构及其连通性的参数[19],故我们在其基础上提出了一种求解影响最大化问题的方法. 本文的贡献主要有:(1)将图的树核度理论应用到影响最大化问题中,并给出了一种求解该问题的多项式时间的算法;(2)在各种不同数据集上进行了实验,结果表明,所提方法有很好的传播范围和覆盖率;(3)通过比较分析了不同节点的选择方法产生不同效果的原因.

本文第 2 节介绍树核与树核度的定义与一些相关性质;第 3 节给出求解树网络树核的方法,在此基础上给出了一种求解社交网络影响最大化节点集的算法;第 4 节给出实验结果,并分析不同节点选择方法的性能;第 5 节对本文成果进行了概括并探讨未来的工作.

2 树核与树核度

不难发现,任意一个网络中总是存在一些占有非常重要位置的要素. 如果从网络中删去这些要素,那么该网络的结构甚至稳定性将会受到很大的破坏. 可见,研究网络中的这些要素是非常有意义的. 为此,我们引入了图的树核度理论来研究此问题. 图的树核度理论通过判断从图中删去一些顶点及其关联边后所得之图中所含连通分支数与所删顶点数的差值以及各连通分支是否含有圈来衡量这些顶点在图中的重要性.

本文所言之图皆指有限无向简单图(无环无重边),所使用图论中的术语都是标准的[20]. 对于一个图 G,分别用 $V(G)$ 和 $E(G)$ (或简记为 V 和 E)表示 G 的顶点集和边集. 图 G 中一个顶点 v 的度 $d_G(v)$ 是指 G 中与 v 关联的边的个数. 图 G 的一条从 v_0 到 v_k 的途径是指一个有限非空序列 $W = v_0e_1v_1e_2v_2\cdots e_kv_k$,它的项交替地为顶点和边,使得对于 $1 \le i \le k$,e_i 的端点是 v_{i-1} 和 v_i. 若 W 的顶点 $v_0, v_1, \cdots, v_k$ 互不相同,那么称 W 为路. 若 $k \ge 3$ 且 $v_0, v_1, \cdots, v_k$ 中只有 v_0 和 v_k 相同,其余任意两个顶点都互不相同,那么称 W 为圈. 若 G 中任意两个顶点之间都存在一条路,那么称 G 是连通的;否则,称 G 是不连通的. 如果 G 是不连通的,那么 G 至少含有两个连通分支,用 $\omega(G)$ 表示 G 的连通分支的个数. 若图 G 是连通的,且 G 不包含圈,则称 G 为树.

令 $V' \subseteq V$,若 $G - V'$ 不连通,则称 V' 为 G 的顶点割,其中 $G - V'$ 表示从 G 中删去 V' 中的顶点以及与这些顶点关联的边所得到的子图. 我们用 $C(G)$ 表示 G 的所有顶点割构成的集合. 进一步,把 $G - V'$ 中含圈的连通分支(不是树)称为 G 的基于 V' 的圈分支,否则称为 G 的基于 V' 的树分支,分别简称为圈分支和树分支.

定义 1[19] **[树核与树核度]** 对于非完全图 G,令 $T(G)$ 表示 $C(G)$ 中满足 $G - S$ 的每个分支都是树分支的顶点割 S 构成的集合,即

$$T(G) = \{S \mid S \in C(G),\ G - S \text{ 不含圈分支}\}$$

则称

$$h_t(G) = \max\{\omega(G - S) - |S|; S \in T(G)\}$$

为图 G 的树核度. 若 $S_t^* \in T(G)$ 满足

$$h_t(G) = \omega(G - S_t^*) - |S_t^*|$$

则称 S_t^* 为图 G 的树核. 其中 $|S|$ 表示 S 中所含元素(顶点)的个数. 由于完全图没有顶点割,我们定义 n 个顶点的完全图 K_n 的树核度为 $2 - n$,并且任意 $n - 1$ 个顶点都构成它的一个树核. 相反,考虑到 n - 阶空图 $\overline{K}_n$ 的割集是空集,故定义 $h_t(\overline{K}_n) = n$.

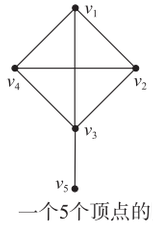

图1 一个5个顶点的图G

例如,对于图 1 所示的图 G,容易验证 $T(G) = \{S_1 = \{v_1, v_3\}, S_2 = \{v_2, v_3\}, S_3 = \{v_3, v_4\}, S_4 = \{v_1, v_2, v_3\}, S_5 = \{v_1, v_3, v_4\}, S_6 = \{v_2, v_3, v_4\}\}$. 另外,对于任意 S_i,$i = 1, 2, \cdots, 6$,$G - S_i$ 都恰好含有两个树分支,从而可以推出该图的树核度为 0,S_1, S_2, S_3 是它的树核.

定理 1 设 G 是一个 n - 阶简单连通图,$n \ge 3$. 如果 $h_t(G) \le 0$,那么 G 中必含圈.

证明 假设结论不成立,即当 $h_t(G) \le 0$ 时,G 中不含圈. 那么显然 G 是树. 因为当 $n \ge 3$ 时,G 中必含有割

点(树的每个非1度顶点都是割点),故删去 G 中任意割点后所得之图至少为 G 的两个树分支. 从而可以推出 $h_t(G)\geq 2-1=1$. 这与假设矛盾,从而结论成立.

3 算法

对于一个网络 G,为找到其最具影响力的 k 个顶点,本文的思想是:首先寻找 G 的树核,然后在其树核中挑选最具影响力的 k 个顶点(一般树核的节点数要比 k 大). 然而,求一个图的树核是 *NP*-完全的[14],这说明很难给出一个多项式时间的算法来精确求解此问题. 为此,本节给出一种优化算法来近似的求解. 首先我们考虑一个网络是树的情况.

设 T 是一棵树,$v\in V(T)$,如果 v 至少与 T 中两个1度顶点相邻,且至多与一个非1度点相邻,那么称 v 是 T 的一个外枝点[21]. 用 $N_T^+(v)$ 表示 T 中与外枝点 v 相邻的所有1度顶点构成的集合,并令 $T_v=T-\{\{v\}\cup N_T^+(v)\}$. 显然,若 v 是 T 的外枝点并且 $|V(T_v)|>0$,则 T_v 连通(既 T_v 是树). 对于至少含有4个顶点的树 T,$v\in V(T)$称为 T 的一个树叶如果 v 是1度顶点并且其邻点是2度顶点[21]. 注意,这里定义的树叶与常用的树的树叶有一定的区别,它更为特殊,不仅要求其度数为1,并且要求与其相邻的顶点的度数为2. 如图2所示的树 T,v_3,v_4 是外枝点,v_5 是树叶.

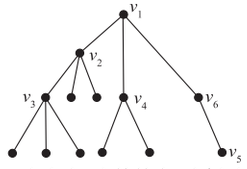

图2　一个含有2个外枝点1个树叶的树T

定理2[21]　设 T 是至少含有4个顶点的树. 如果 T 不含树叶,那么 T 中含有外枝点.

定理3　设 T 是至少含有4个顶点的树. 如果 v 是 T 的树叶,那么 $h_t(T)=h_t(T-v)$.

证明　令 v' 是与 v 相邻的2度顶点. 考虑树 $T-v$,v' 在 $T-v$ 中是1度顶点. 令 S 是 $T-v$ 的一个树核,那么 $v'\notin S$. 这是因为此时若 $v'\in S$,那么

$$\begin{aligned}&\omega(T-v-(S\backslash v'))-|S\backslash v'|\\&\geq\omega(T-v-S)-|S|+1>h_t(G)\end{aligned}$$

这与 S 是 T 的树核矛盾. 从而,可以推出

$$\begin{aligned}h_t(T-v)&=\omega(T-v-S)-|S|\\&=\omega(T-(S\cup\{v\}))-|S\cup\{v\}|+1\\&=\omega(T-S)-|S\cup\{v\}|+1\\&=\omega(T-S)-|S|=h_t(T)\end{aligned}$$

从而结论成立.

定理4　设 T 是至少含有4个顶点的树,v 是 T 的一个外枝点. 那么

$$h_t(T)=h_t(T_v)+|N_T^+(v)|-1$$

证明　令 S^* 是 T 的一个树核,因为 v 至少与 T 中两个一度顶点相邻,故总可取 S^* 包含 v. 从而

$$\begin{aligned}h_t(T)&=\omega(T-S^*)-|S^*|\\&\leq\omega(T-(S^*-\{v\}))+|N_T^+(v)|-|S^*|\\&\leq\omega(T-(S^*-\{v\}))-|S^*-\{v\}|+|N_T^+(v)|-1\\&\leq h_t(T_v)+|N_T^+(v)|-1\end{aligned}$$

另一方面,令 S_v^* 是 T_v 的一个树核,

$$\begin{aligned}h_t(T_v)&=\omega(T_v-S_v^*)-|S_v^*|\\&=\omega(T-(S_v^*\cup\{v\}))+|N_T^+(v)|-|S^*|\\&=\omega(T-(S_v^*\cup\{v\}))-|N_T^+(v)|-|S_v^*|\\&=\omega(T-(S_v^*\cup\{v\}))-|S_v^*\cup\{v\}|-|N_T^+(v)|+1\\&\leq h_t(T)-|N_T^+(v)|+1\end{aligned}$$

从而有 $h_t(T)\geq h_t(T_v)+|N_T^+(v)|-1$.

作为定理3和定理4的一个例子,我们考虑图2所示的树 T. 该图中共有2个外枝点 v_3,v_4,1个树叶 v_5. 首先,令 $T^1=T-v_5$. 由定理3,$h_t(T)=h_t(T^1)$;其次,T^1 中不含树叶,但含有2个外枝点 v_3,v_4. 任选其中一个外枝点 v_3,根据定理4,$h_t(T^1)=h_t(T^1_{v_3})+|N_{T^1}^+(v_3)|-1=h_t(T^1_{v_3})+2$;第三,令 $T^2=T^1_{v_3}$. T^2 中不含树叶,但含两个外枝点 v_2 和 v_4. 选择其中的一个 v_2,根据定理4,$h_t(T^2)=h_t(T^2_{v_2})+|N_{T^2}^+(v_2)|-1=h_t(T^2_{v_2})+1$;第四,在 $T^2_{v_2}$ 中,v_6 是树叶,故由定理3,$h_t(T^2_{v_2})=h_t(T^2_{v_2}-v_6)$. 令 $T^3=T^2_{v_2}-v_6$. T^3 中不含树叶,含有1个外枝点 v_4. 由定理4,$h_t(T^3)=h_t(T^3_{v_4})+|N_{T^3}^+(v_4)|-1=h_t(T^3_{v_4})+2$. 最后,因为 $T^3_{v_4}$ 是一个空图,即不含任何顶点,从而 $h_t(T^3_{v_4})=0$. 故 $h_t(T)=2+1+2=5$. 这里用到的符号 $T^i_{v_j}$,$i\in\{1,2,3\}$,$j\in\{2,3,4\}$,表示在 T^i 中删去 $\{\{v_j\}\cup N_T^+(v_j)\}$ 中的顶点及其关联边得到的图,即 $T^i_{v_j}=T^i-\{\{v_j\}\cup N_T^+(v_j)\}$.

根据定理3和定理4,下面给出求解树的树核算法.

算法1　树的树核算法 findTreeCoreOfTree(T)

输入:树 T,$|V(T)|\geq 4$,顶点集合
$S^*=\varnothing$,$h_t(T)=0$,$t=0$
输出:T 的树核 S^* 和树核度 $h_t(T)$
$T^0=T$;
$i=0$;
While ($|V(T^i)|\neq 4$&&T^i 含有叶子结点 v_i)
　$T^{i+1}=T^i-v_i$,$i++$;
if $|V(T^i)|=4$
　if T^i 含叶子结点
　　$h_t(T)=h_t(T)+1+t$;
　　$S^*=S^*\cup\{u\}$,其中 u 是 T^i 中任意一个2度顶点;
　else
　　$h_t(T)=h_t(T)+2+t$;

$S^* = S^* \cup \{u\}$,其中 u 是 T^i 中唯一一个 3 度顶点;
else
找到 T^i 的一个外枝点 v;
$T_v^i = T^i - \{\{v\} \cup N_{Ti}^+(v)\}$, $S^* = S^* \cup \{v\}$,
$t = N_{Ti}^+(v) - 1$;
if $|V(T_v^i)| \geq 4$
$T = T_v^i$
findTreeCoreOfTree(T);
else if $|V(T_v^i)| = 3$
$h_t(T) = h_t(T) + 1 + t$;
$S^* = S^* \cup \{u\}$,其中 u 是 T_v^i 中唯一的 2 度顶点;
else
S^* 不变,$h_t(T) = h_t(T) + t$;

在算法 1 的基础上,下面给出一个网络 G 中 k 个最有影响力节点的算法. 思想是:首先删除 l 个节点 $S = \{v_1, v_2, \cdots, v_l\}$ 使得 $G - S$ 的每个分支都是树,其中 v_i 是 $G - \{v_1, v_2, \cdots, v_{i-1}\}$ 中的最大度顶点;其次,利用算法 1 求 $G - S$ 中每个分支的树核 S_t^*,然后在 $S \cup S_t^*$ 中选择度数最大的 k 个顶点. 具体算法如下:对于一个网络 G 中的每个顶点 v,定义 $w_G(v) = d_G(v)$ 为顶点 v 在 G 中的权值.

算法 2 寻找 k 个最有影响力节点

输入:图 G,数 k
输出:大小为 k 的节点集 S
$G_0 = G$;
$S_0 = \varnothing$;
$i = 0$;
While(G_i 中存在分支不是树)
选择 G_i 中权值最大的一个顶点 v_i;
$S_{i+1} = S_i \cup \{v_i\}$;
$G_{i+1} = G_i - v_i$, $i++$;
if $k < i$
$S = S_k$;
else
求出各个分支 $T_1, T_2, \cdots, T_m$ 的树核 $S^*(T_1), S^*(T_2), \cdots, S^*(T_m)$;
令 $S' = S^*(T_1) \cup S^*(T_2) \cup \cdots \cup S^*(T_m)$;
if $|S'| \geq k - l + 1$
令 S_2 为 S' 中度数(在 G 中)最大的前 $k - l + 1$ 个顶点, $S = S_1 \cup S_2$;
else
$|S'| < k - l + 1$, $S_2 = S'$;
令 S_3 为 $G - (S_1 \cup S_2)$ 中所有的 2 度顶点的集合, $l'' = |S_3|$;
if $l'' \geq k - l - l' + 1$
在 S_3 中任意选择 $k - l - l' + 1$ 个顶点;
$S = S_1 \cup S_2 \cup S_3$;
else
在 $G - (S_1 \cup S_2 \cup S_3)$ 中任意选择 $k - l - l' - l'' + 1$ 个顶点,记为 S_4;
$S = S_1 \cup S_2 \cup S_3 \cup S_4$;

在算法 1 中,为找到一个叶子节点,我们最多将树中所有节点扫描一遍即可,而对一个节点进行判断其是否为叶子节点可在线性时间内完成,因此找到叶子节点的复杂度为 $O(n_T)$,其中 n_T 表示树 T 中的节点个数. 同理,寻找一个外枝点也最多只需将树中所有节点扫描一遍,同时对某个节点进行判断其是否为外枝点的复杂度不超过 $O(k_d)$,其中 k_d 表示树 T 中节点的最大度,因此找到外枝点的复杂度为 $O(k_d n_{|T|})$. 考虑到 *while* 循环每执行一次就减少一个节点. 另外,找到外枝点之后在调用算法 findTreeCoreOfTree 时节点个数也少一个,因此每减少一个节点的最大复杂度为 $O(k_d n_T)$,所以算法 1 总的复杂度为 $O(k_d {n_T}^2)$. 类似的分析,算法 2 中找一个权值最大的顶点复杂度为 $O(n)$(将所有节点扫描一遍,对一个节点的判断可在线性时间内完成),其中 n 表示图 G 中的节点数,所以 while 循环的时间复杂度不超过 $O(n^2)$. 后面的处理对于每个树调用算法 1,假设所有树中节点的最大度为 k_T,则这一步的复杂度为

$$\begin{aligned} &O(k_{d_1} n_{T_1}^2) + O(k_{d_2} n_{T_2}^2) + \cdots + O(k_{d_m} n_{T_m}^2) \\ \leq &O(k_T(n_{T_1}^2 + n_{T_2}^2 + \cdots + n_{T_m}^2)) \\ < &O(k_T(n_{T_1} + n_{T_2} + \cdots + n_{T_m})^2) \leq O(k_T n^2), \end{aligned}$$

因此总的时间复杂度不超过 $O(k_T n^2)$,从而是多项式时间的.

4 实验

为了证明树核理论在求解影响最大化问题中的有效性,我们在真实网络的数据集上进行了实验,比较了该方法与其他节点选择方法的传播效果并进行了相关的分析. 实验数据集为 E-mail 网络[22] 和 Jazz 音乐家形成的网络[23]. E-mail 网络是 Rovira i Virgili 大学的成员之间 E-mail 联系形成的网络,由 1133 个节点和 5451 条边组成,其中每个节点代表一个成员,每条边代表边连接的两个成员之间曾经使用 E-mail 联系过. Jazz 音乐家形成的网络由 198 个节点和 2742 条边组成,其中每个节点代表一个 Jazz 音乐家,每条边表示边连接的两个 Jazz 音乐家之间有联系. 这两个网络均为无向的,我们分别使用线性阈值模型、独立级联模型和加权级联模型三个传播模型对这两个网络进行实验.

线性阈值模型(Linear Threshold Model, LTM): 在线性阈值模型中,一个节点是否受到影响从不活跃状态变为活跃状态是由其邻居的共同影响力决定的. 对于节点 w 的邻居节点 v,将 v 对 w 的影响记为 $b_{w,v}$,则有 $\sum_v b_{w,v} \leq 1$. 在该模型中,如果节点 w 活跃邻居的影响总和超过某个阈值 θ_w,则 w 由不活跃状态变为活跃状态. 即满足公式

$$\sum_{v.\ activeneighborofw} b_{w,v} \geq \theta_w$$

时,节点 w 被激活(即由不活跃状态变为活跃状态). 可以看出,θ_w 越大,则节点 w 越不容易被激活. 因此 θ_w 可以反映出节点 w 被激活的倾向性[24].

独立级联模型(Independent Cascade Model,ICM):在独立级联模型中,节点只有在刚被激活时才可以尝试去激活其邻居. 假设 v 是一个在时刻 t 刚被激活的节点,则对于 v 的每个不活跃邻居 w,v 可以以概率 $p_{v,w}$ 激活 w. 若激活成功,则节点 w 在时刻 $t+1$ 变为活跃节点;若不成功,w 仍然保持不活跃状态. 无论 w 是否成功激活其邻居,在 $t+1$ 以后的时刻 v 都不能再尝试激活其他节点. 如果在时刻 t 不活跃节点 w 有多个邻居都刚被激活,则其邻居节点对其的激活顺序对于最后结果没有影响. 概率 $p_{v,w}$ 不依赖于之前所有对 w 的激活尝试. 传播过程以这种方式不断进行直到没有刚被激活的节点时停止[5].

加权级联模型(Weighted Cascade Model,WCM):加权级联模型可以看作是独立级联模型的一个特例. 在该模型中,节点 v 激活其不活跃邻居节点 w 的概率 $p_{v,w}$ 与节点 w 的度 $d(w)$ 有关,$p_{v,w} = 1/d(w)$[20]. 故当一个节点有很多邻居时,每个邻居对其的影响就会平均到一个非常小的值,这在某种程度上可以反映出真实世界中的人际关系. 例如,如果一个人只有一个朋友,那么这个朋友对他的建议就非常具有影响力,而如果他有很多朋友,那么其中一个朋友的建议对于他作何决定的影响并不大[5].

为了显示算法的实验效果,这里使用的作为对比的节点选择方法有:随机选择、按照度选择、贪心方法、树核方法.

随机选择(Random):一种简单的节点选择方法,这种方法完全随机地选择出 k 个需要的节点.

按照度选择(Degree):这种方法选择出网络中度最大的 k 个节点作为初始活跃节点集合.

贪心方法(CELFGreedy):使用 lazy-forward 优化的贪心算法,每次都选择能使活跃节点数增加最多的节点. 对于每个候选节点集 S,进行 10,000 次模拟来得到准确的传播范围 $\sigma(S)$.

树核方法(TreeCore):使用第 3 节中介绍的算法 2 选择初始活跃节点集合.

将这四种节点选择方法选出的节点作为初始的活跃节点,各自分别按照线性阈值模型、独立级联模型、加权级联模型的传播方式进行传播,对最终的传播效果进行比较.

4.1　E-mail

图 3 为在 E-mail 网络上各种节点选择方法选出的初始节点集合的传播效果图. 其中横坐标为初始节点集合的大小,纵坐标为最终传播到的节点集合的大小.

图 3(a)为在线性阈值模型(LTM)下的传播效果对比图. 由于缺少阈值信息同时为了实验的简洁性,实验中对各个节点的阈值均设为同一值 0.5,并且认为一个节点的各个邻居对其影响是相同的[5]. 这样当一个不活跃节点有一半的邻居是活跃状态时该节点即可被激活. 由图 3(a)可以看出,在线性阈值模型下贪心方法的效果最好,树核方法由于进一步利用了网络的连通性,因此效果比单纯使用度的方法效果更好,因为度大的节点之间可能互相连接也比较多,这样只选择其中一部分即可达到较好的效果而无须将所有度大的节点都选为初始活跃节点. 随机方法由于没有利用到网络的任何信息而只进行随机选择,因此效果最差.

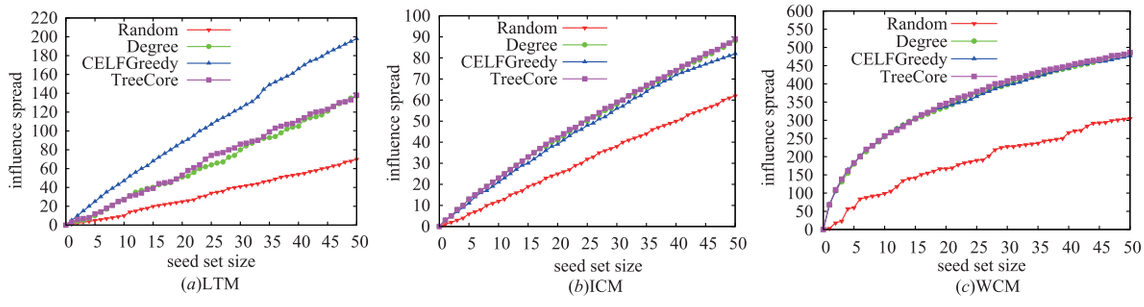

图3　E-mail网络的实验结果对比图

图 3(b)为在独立级联模型(ICM)下的传播效果对比图. 同样出于简洁性以及缺少相关信息的考虑,这里将节点之间的激活概率均设为相同的值 0.01,这也是该模型下通常[5]使用的概率值. 由于激活概率值较小,因此图中显示的传播范围并不大. 由图 3(b)可以看出,在独立级联模型下树核方法的效果会比度以及贪心方法稍好一些,这三种方法的效果都比随机方法好很多. 因为随机方法没有利用到网络的任何信息,而其他方法都有针对性地利用了网络的特性进行选择节点.

图 3(c)为在加权级联模型(WCM)下的传播效果

对比图. 由于网络中大部分节点的度并不大,导致激活概率相对 0.01 要大,传播范围也比图 3(*b*)中的广,而且节点数较少的时候传播范围的增长速度比图 3(*c*)中的快.

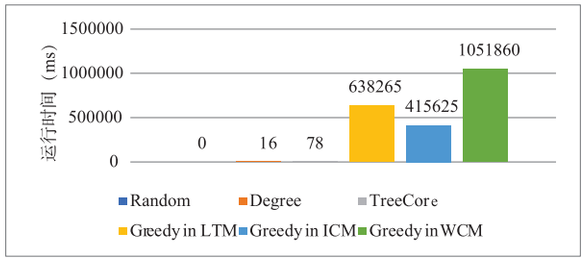

图4 E-mail网络各算法运行时间对比图

图 4 为随机、度、树核和贪心算法的运行时间对比图,从图中可以看出,贪心算法在各个模型中的运行时间均较长,而随机算法以及根据度进行选择的算法由于不需要进行各种计算而运行时间较短,本文中提出的基于树核的方法虽然运行时间比这两种方法稍长,但也可以在很短的时间内完成运算,时间效率也较高.

4.2 Jazz

图 5 为在 Jazz 音乐家网络上各种节点选择方法选出的初始节点集合的传播效果图. 其中横坐标为初始节点集合的大小,纵坐标为最终传播到的节点集合的大小.

图5(*a*)为在线性阈值模型(LTM)下的传播效果对比图. 同在 E-mail 网络上一样,实验中对各个节点的阈值均设为同一值 0.5,并且认为一个节点的各个邻居对其影响是相同的. 由图 5(*a*)可以看出,该数据集在线性阈值模型下的传播情况与 E-mail 网络不同. 使用度、贪心以及树核方法选择节点时在某些点处会出现传播范围的突然上升. 这与传播模型以及数据集有关,由于该模型下某个节点只有当其活跃邻居数累积到邻居总数的一半时才能被激活,因此当初始活跃节点较少时只有某些度比较小的节点可以被激活,随着初始活跃节点数的增多,节点的活跃邻居数也不断增加,当累积到某一程度时,很多节点受到活跃邻居的影响都达到阈值从而变为活跃状态,产生图 5(*a*)中的突变,而在独立级联模型和加权级联模型下就不会出现这种情况. 另外,在图 5(*a*)中,当初始活跃节点集合较大时,树核方法效果明显比其他方法更好,表明树核方法更适用于需要的初始节点集合较大时的情况.

图 5(*b*)、(*c*)分别为在独立级联模型(ICM)和加权级联模型(WCM)下的传播效果对比图. 同样出于简洁性以及缺少相关信息的考虑,这里将节点之间的激活概率均设为相同的值 0.01. 由于该网络的密度较大,因此在这两个模型下的传播范围相差不多. 从这两个图中可以看出,在 Jazz 数据集中这两个模型下,实验中使用的几种方法效果相差不多,而树核方法效果比其他方法会稍微好一些.

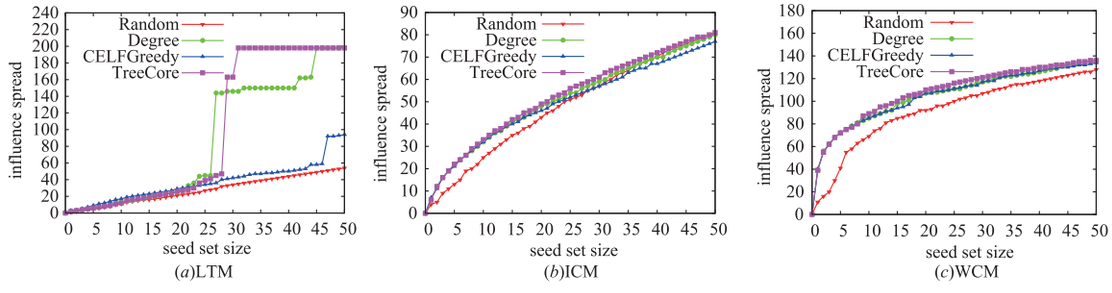

图5 Jazz音乐家网络的实验结果对比图

从实验的各个对比结果中可以看出,本文算法在选择节点数较多时效果较好. 这种现象出现的原因与算法的理论基础有关,在节点数较少时,其他节点直接选择影响力较高的少数几个节点进行信息传播,可以达到比较好的效果. 但是当节点数不断增多时,如果还按照节点自身单独的影响力进行选择,那么由于不同节点影响范围会存在很多交叉,增加的节点影响到的范围可能之前的节点已经可以影响到,故此时节点增加对于影响范围的作用并不大. 本文算法利用网络连通性相关信息,考虑到对整体的影响效果,节点数较多时节点影响范围交叉对于算法的负面影响更少一些,但是节点数较少时范围交叉对于算法的影响较少,因此出现在选择的节点数较多时效果较好的现象.

图 6 为随机、度、树核和贪心算法的运行时间对比图,与在 Email 网络上的结果类似,贪心算法在各个模型中的运行时间均较长,随机算法以及根据度进行选择的算法运行时间较短,本文中提出的基于树核的方法也可以在很短的时间内完成运算,时间效率也较高.

5 结束语

本文提出了一种基于树核度求解网络影响最大化问题的方法,并给出一种多项式时间的算法找到一个

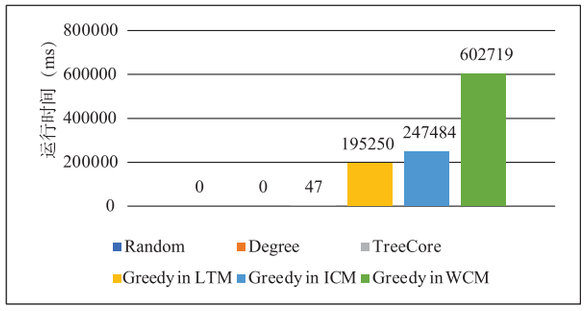

图6　Jazz音乐家网络各算法运行时间对比图

网络中 k 个初始活跃节点. 在实验部分,我们将所提方法与传统的随机、度以及贪心方法进行了比较. 结果表明,文中所提方法可以较快地找到能够使得传播范围较大的节点集合.

由于求解一般图的树核是非常困难的,故我们在算法 2 中首先给出了一种删点的方法使得所求网络变成一个森林,然后再分别求每颗树的树核. 但是该方法只是考虑了图中顶点度的影响,故具有一定的局限性. 为此,在后续的工作中,我们希望给出更好的删点方法. 也许在删点的同时将该点所在圈的数目考虑在内会得到更好的结果.

参考文献

[1] Stanley W, Katherine F. Social Network Analysis in The Social and Behavioral Sciences[M]. Social Network Analysis: Methods and Applications. Cambridge University Press. 1994. 1 – 27.

[2] Domingos P, Richardson M. Mining the network value of customers[A]. Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. ACM, 2001. 57 – 66.

[3] Richardson M, Domingos P. Mining knowledge-sharing sites for viral marketing[A]. Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. ACM, 2002. 61 – 70.

[4] Kempe D, Kleinberg J, Tardos E. Influential nodes in a diffusion model for social networks[J]. In International Colloquium on Automata, Languages and Programming, 2005, 32: 1127 – 1138.

[5] Kempe D, Kleinberg J, Tardos E. Maximizing the spread of influence in a social network[A]. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. Washington, USA, 2003. 137 – 146.

[6] Leskovec J, Krause A, et al. Cost-effective outbreak detection in networks[A]. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. ACM, 2007. 420 – 429.

[7] Kimura M, Saito K. Tractable models for information diffusion in social networks[A]. Knowledge Discovery in Databases: PKDD 2006[C]. Springer, 2006. 259 – 271.

[8] Chen W, Wang C, Wang Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks[A]. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. ACM, 2010. 1029 – 1038.

[9] Wang Y, Cong G, Song G, Xie K. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks[A]. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. ACM, 2010. 1039 – 1048.

[10] Goyal A, Lu W, Lakshmanan L V. SIMPATH: an efficient algorithm for influence maximization under the linear threshold model[A]. IEEE 11th International Conference on Data Mining (ICDM)[C]. IEEE, 2011. 211 – 220.

[11] Chen W, Wang Y, Yang S. Efficient influence maximization in social networks[A]. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. ACM, 2009. 199 – 208.

[12] Jiang Q, Song G, et al. Simulated annealing based influence maximization in social networks[A]. AAAI 2011[C]. San Francisco, California, USA, Augest 2011. 127 – 132.

[13] Jung K, Heo W, Chen W. IRIE: Scalable and robust influence maximization in social networks[A]. Proceedings of the 2012 IEEE 12th International Conference on Data Mining[C]. 2012. 918 – 923.

[14] Li J S, Yu Y Y. Scalable influence maximization in social networks using the community discovery algorithm[A]. Process of the 6th International Conference on Genetic and Evolutionary Computing[C]. Washington DC USA: IEEE Computer Society, 2012. 284 – 287.

[15] Zhu Y, Wu W, Bi Y, et al. Better approximation algorithms for influence maximization in online social networks[J]. Journal of Combinatorial Optimization, 2013: 1 – 12.

[16] Cheng S, Shen H et al. 2014. IMRank: influence maximization via finding self-consistent ranking[A]. In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval (SIGIR '14)[C]. ACM, New York, USA, 2014. 475 – 484.

[17] Cohen E, Delling D, Pajor T, Werneck R F. Sketch-based inuence maximization and computation: Scaling up with guarantees[A]. In Conference on Information and Knowledge Management, CIKM[C]. Shanghai, China, 2014. 629 – 638.

[18] Lucier B, Oren J, Singer Y. Influence at scale: distributed computation of complex contagion in networks [A]. Process of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. New York, USA: ACM, 2015.

[19] Zhu E Q, Li Z P, et al. Tree-core and tree-coritivity of graphs [J]. Information Processing Letters, 2015, 115 (10): 754 – 759 .

[20] Bondy J A, Murty U S R. Graph Theory [M]. Springer, 2008.

[21] 欧阳克智, 欧阳克毅, 于文池. 图的相对断裂度[J]. 兰州大学学报, 1993, 29 (3): 43 – 48.
Ouyang K Z, Ouyang K Y, Yu W C. Relative breaktivity of graphs [J]. Journal of Lanzhou University, 1993, 29 (3): 43 – 48 (in Chinese).

[22] Guimera R, Danon L, et al. Self-similar community structure in a network of human interactions[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2003, 68(2): 93 – 108.

[23] Gleiser P, Danon L. Community structure in JAZZ[J]. Advances in Complex Systems. 2011, 6(4): 565 – 573.

[24] Goldenberg J, Libai B, Muller E. Talk of the network: a complex systems look at the underlying process of word-of-mouth[J]. Mark Lett 2001, 12 (3): 211 – 223.

作者简介

朱恩强 男, 1983 年 6 月出生, 辽宁人. 2015 年获北京大学信息科学技术学院理学博士学位, 现为北京大学信息学院博士后, 主要从事图论与组合优化、社交网络信息传播及安全、DNA 序列编码理论的研究.
E-mail: zhuenqiang@ pku. edu. cn