doi:10.3772/j.issn.1002-0470.2019.05.004

基于覆盖阈值的影响最大化算法的研究①

陈 晶②*,**,*** 刘 贤*

(*燕山大学信息科学与工程学院 秦皇岛 066004)

(**河北省虚拟技术与系统集成重点实验室 秦皇岛 066004)

(***河北省软件工程重点实验室 秦皇岛 066004)

摘 要 针对影响最大化算法存在选取的种子节点影响力重叠、时间复杂度高等问题,提出了基于覆盖阈值的度最大启发式算法(CTMD)。该算法主要思想是利用改进的 k-shell 算法计算节点影响力以选取初始种子节点;计算两度以内节点的激活概率,基于覆盖阈值 θ,把易激活的节点标记为覆盖状态,更新节点的影响力值,直到选取到指定数量的种子节点;通过实验对核覆盖算法(CCA)、度最大(Max Degree)、影响力排名影响力估计(IRIE)和 CTMD 算法进行了比较与分析。实验结果表明,在独立级联(IC)模型和加权级联(WC)模型中,CTMD 算法的影响范围具有明显的优势。此外,通过对运行时间进行测试可知,随着网络规模的逐步增大,CTMD 算法具有较低的时间复杂度。

关键词 社交网络, 节点影响力, 影响最大化, 覆盖阈值, k-shell

0 引 言

随着互联网的迅速发展,大量的社交平台应运而生,例如 Twitter、Facebook 和微博等。越来越多的人们喜欢借助于这些平台发表言论、分享自己的观点与其他信息。如何使所分享的信息快速传播并扩大其影响范围等相关方面的研究已成为社交网络分析(social network analysis, SNS)中的热点问题。目前,较为流行的信息传播应用是口碑效应、广告投放和病毒式营销。利用社交网络平台,人们往往会以很小的成本获得巨大影响。而影响力最大化算法就是解决这些应用问题的关键因素。

影响最大化问题最早由 Domingos 等人[1]定义为如何寻找 t 个初始节点,使得信息的最终传播范围最广。2003 年 Kempe 等人[2]将影响最大化问题看作一个离散优化问题,提出了 Greedy 算法,证明了影响最大化问题是 NP-hard,并能够达到 63% 的近似解。由于 Greedy 算法时间复杂度过高不适合处理大规模的社交网络,因此,大多数学者致力于找到与 Greedy 算法影响效果相似且时间复杂度低的启发式算法。文献[3-6]提出了时间复杂度较低的启发式算法。其中,曹玖新等人[6]提出了核覆盖算法(core covering algorithm, CCA),该算法为解决影响力重叠问题,提出了覆盖距离的概念,并结合 k 核和度的方法来选取种子节点。与以往的方法不同之处在于,CCA 算法在大规模的社交网络中取得了较好的效果。Aybike 等人[7]利用群体智能算法对影响最大化问题进行研究。作者针对目前的影响最大化算法在时间和收敛速度方面存在的问题,基于社区结构的特点研究影响最大化问题的全局收敛最优。论文将社交网络中的个体作为节点,依据度量结果对节点进行降序排列,其目的是确保邻近的节点在状态空间中,有近似的影响程度。通过对提出

① 国家自然科学基金(61602401, 61472340)和河北省高等学校科学技术研究(QN2018074)资助项目。

② 女,1976 年生,副教授,博士;研究方向:对等网络,社会计算,Web 服务,CCF 会员;联系人,E-mail: xychenjing@ysu.edu.cn

(收稿日期:2018-08-24)

的算法进行测试取得了很好的效果。Tang 等人[8]提出了应用于大型社交网络影响力最大化的一种有效方法,并通过实验验证了该算法的效率和有效性。

本文基于上述研究基础可知,目前的大多数方法的关注点在寻找时间复杂度低而影响效果和贪心算法相近的启发式算法,忽视了影响力重叠问题。因此,本文提出了一种基于覆盖阈值的度最大启发式算法(coverage threshold maximum degree, CTMD)。本文的贡献是通过覆盖阈值 θ 来改善影响重叠的问题,并只考虑两级以内的邻居被激活的概率来降低时间复杂度,进而提高了种子节点对传播过程的影响效果。

1 相关工作

对于影响最大化问题的研究,国内外学者针对不同的信息传播模型提出了相关的种子集选取算法,主要包括贪心算法和启发式算法两类。由于贪心算法对于大型社交网络的时间复杂度较高,因此,近几年多数学者研究应用启发式算法来提高选取种子节点的速度。Leskovec 等人[9]提出了 CELF(cost-effective lazy forward)算法,该算法利用了影响最大化问题的单调性和子模性来优化 Greedy 算法。CELF 算法虽然比传统的贪心算法快700 倍,但是对于大型社会网络来说时间复杂度仍然太高。Vichaya 等人[10]提出了 IRIE(influence ranking influence estimation)算法,该算法基于信任传播,仅需要很少轮迭代就能对全部节点的影响力进行排序,然后选择排序最高的节点作为最有影响力的节点,该算法的精度与 PMIA(prefix excluding maximum influence arborescence)算法持平但在速度与内存上有优势,因此是综合实力最好的算法。刘晓东[11]根据贪婪算法的时间复杂度高、算法运行时间长、不适用大规模社交网络的缺点提出了基于并行架构的加速算法,该算法把 BUTA(bottom up traversal algorithm)算法映射到 CPU + GPU 的并行架构上,通过形成新的 IMGPU 架构来计算各节点的影响力,有效地改善了算法的时间效率。Litou 等人[12]研究了多级联模型中的影响最大化问题。论文针对现有研究方法忽视了多个传染病在社交网络中级联相关性问题,建立了一个新的相关传染病动态线性阈值 CCDLT(correlated contagions dynamic linear threshold)。该模型以竞争或互补的方式考虑许多传染病的相关性,通过设计的贪婪种子选择算法,使其传播最大化,并形式化地证明了它以 1 - 1/e 的比例逼近最佳解。Zareie 等人[13]提出了一种基于 TOPSIS(the technique for order of preference by similarity to ideal solution)的社会网络影响最大化方法。该方法针对新闻和消息传播领域中的用户影响力问题,考虑用户之间的距离,并基于用户"思想"相似度排序方法选择用户集合。通过对不同的数据集进行测试,验证了该方法选择的用户集合比传统方法具有更大的影响扩散能力。

除了对传统的社交网络影响最大化问题的研究,许多学者依据实际应用对影响最大化问题进行了延伸和变形。宋永浩等人[14]提出了团队构建的统一优化目标函数问题,在综合考虑协同作业任务所需技能集合覆盖约束和团队成员之间交流代价最小化约束的基础上,提出了不同的基于贪心策略的启发式团队构建算法。实验表明,该方法具有很高运行效率和精度。Yang 等人[15]借鉴了群体智能的思想,采用了蚁群优化算法来解决竞争影响最大化问题。Bozorgi 等人[16]提出一种新的传播模型来解决竞争影响最大化问题,该传播模型是线性阈值模型的扩展,并为节点提供了影响传播的决策能力。基于扩展后的传播模型,提出了一种有效的算法用于检索给定社交图中的影响节点,该模型利用图社区结构来计算每个节点在其社区内的局部传播。刘院英等人[17]提出了成本控制下的影响最大化算法,并利用动态规划的方法选择种子节点。Bucur 等人[18]以社会网络中多目标进化问题为出发点,提出了影响最大化问题的改进方案。实验测试结果表明,所提出的改进方法加快了优化过程,改善了影响力传播的过程。李小康等人[19]考虑到信息在多个网络中的传播,提出社交网络中多渠道影响最大化问题,从多个网络中选取 k 个种子用户,让其同时在多个网络中传播影响,使最终受种子用户影响的用户量最大化。张平等人[20]从实际应用场景出发,研

究了关注区域或人群的组合影响力问题,提出了团体传播模型 GIC(group independent cascade)。基于 GIC 模型,设计了贪心算法 CGIM(cascade group influence maximization),检索到最具影响力的 top-k 团组合。通过在人工数据和真实数据上验证了 GIC 模型的效率。Wang 等人[21]提出了一种价格性能比启发式算法 PPRank,该算法研究了如何在给定预算内经济地选择种子节点,并分析了最大化扩散过程。而本文研究的出发点是基于原核覆盖算法 CCA 的思想,提出了一种解决节点影响力重叠问题的算法 CTMD。该算法在独立级联(independent cascade, IC)和加权级联(weighted cascade, WC)模型下相对于 CCA 算法的影响效果均有提高,且时间复杂度比传统的贪心算法具有明显的优势。

全文组织如下。第 2 节介绍了影响最大化问题的相关工作;第 3 阐述了 k-shell 和核覆盖算法 CCA 的特点;第 4 节提出了基于覆盖阈值 θ 的影响最大化算法 CTMD,并论述了算法的实现步骤及算法实例;第 5 节基于选定的实验数据集对 CTMD 算法进行了验证,通过与相关算法的比较与分析,验证了 CTMD 的优势;最后对全文进行了总结与展望。

2 K-shell 算法和 CCA 算法介绍

由于 CTMD 算法是基于覆盖阈值的思想,本文采用改进的 k-shell 算法来估计节点影响力,借鉴并补充了 CCA 算法的特点。因此,本文首先对 k-shell 算法和 CCA 算法进行简单论述。

2.1 k-sehll 算法描述

Kitsak 于 2010 年提出了 k-shell 算法,该算法揭示了网络结构的层次性,能较好地估计网络中节点的影响力。依据 k-shell 算法,网络中所有节点都对应一个 *ks* 值,k-shell 算法认为 *ks* 值大的节点处于网络的核心位置,这些节点之间的连通性较强,因此它们的影响力也较大。从 k-shell 算法的描述过程可知,该算法存在两方面的不足:(1) k-shell 算法不能判断具有相同 *ks* 值的节点的影响力大小;(2) 网络中的边带有不同传播概率时,k-shell 算法不能准确地估计节点影响力。

2.2 CCA 算法描述

CCA 算法是曹玖新等人于 2015 年提出的,其目标是解决影响力重叠问题,因此 CCA 算法引入了覆盖距离参数 d 使得选取的种子节点保持一定的距离,该算法描述如下:

(1) 利用 k-shell 算法计算所有节点的 *ks* 值;

(2) 选取 *ks* 值最大的节点作为种子节点,当 *ks* 值相同时选取度最大的节点作为种子节点;

(3) 标记与种子节点距离为 d 的所有节点为覆盖状态,这些节点不能被选作种子节点;

(4) 重复步骤(2)和(3),直到选择到指定数量的种子节点集。

从 CCA 算法的描述过程可知,该算法存在以下不足之处:(1) 当一个节点 v 被选为种子节点后,如果覆盖距离 $d = 1$,则标记 v 的所有出边邻居为覆盖状态,这些被覆盖的节点不能被选为种子节点,然而在实际传播过程中,当各边的激活概率不同或较小时出边邻居不可能被全部激活,因此不能将节点 v 的出边邻居全部标记为覆盖状态;(2) CCA 算法将出度作为种子节点选择存在一定的弊端,其原因是由于该节点的某些出边邻居有可能已经被标记为覆盖状态,因此该节点的影响力可能会大打折扣。

3 CTMD 算法概述

CTMD 算法结合了 k-shell 的影响力估计和 CCA 算法覆盖的特性,由于 k-shell 算法和 CCA 算法存在的问题在很大程度上影响了种子节点的选择,因此,本文将分别论述改进的影响力估计方法和覆盖策略。

3.1 基于 *ks* 值和边传播概率的节点影响力估计

由于 CTMD 算法需要估计节点影响力,因此,本文提出了结合 *ks* 值和边传播概率的方法来实现。给定一个网络 $G(V,E)$,其中 V 表示网络中的节点集,E 表示边集。

定义 1 对于 V 中的每一个 v,计算 v 对其邻居的节点影响力 $inf(v)$:

$$inf(v) = \sum_{u \in N(v)} p(v, u) \tag{1}$$

其中,$N(v)$ 表示 v 的邻居节点集合,$p(v, u)$表

示边(v, u)的传播概率。

定义 2　对于 V 中的每一个节点 v, 计算 v 的影响力 $infs(v)$:

$$infs(v) = ks(v) + \lambda inf(v) \tag{2}$$

其中, $ks(v)$ 表示节点 v 的 ks 值,参数 λ 为平衡节点对其邻居的影响力和 ks 值之间的参数,在本文实验的网络中 λ 值大约取0.8。

设图 1 为包含 3 条出边的有向社交网络图,节点 1 的 3 条出边概率分别为0.01、0.02、0.03,设 λ =0.8,且节点 1 的 ks 值为 10,根据式(2),节点 1 的影响力 $infs(1) = 10 + 0.8 \times (0.01 + 0.02 + 0.03) = 10.048$。

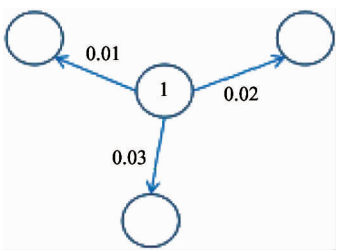


图 1　包含 3 条出边的有向社交网络影响力实例

3.2　CTMD 算法的覆盖策略

(1) 由于一般的社交网络传播概率通常较小,本文假设节点的影响力只在 2 级邻居以内(包括二级邻居),认为超过 2 级邻居的节点不易被影响,这样做的目的减少计算量,降低时间复杂度。

定义 3　节点 v 的 1 级邻居为 v 的出边邻居集 $N(v)$,v 的 2 级邻居为 $N(v)$ 的出边邻居集 $N2(v)$, $N2(v)$ 与 v 的距离为 2,如果节点 u 即属于 $N(v)$ 又属于第 $N2(v)$, 则算作属于 $N(v)$。

(2) 针对 CCA 算法中的覆盖状态问题,本文提出了覆盖阈值 θ 来代替覆盖距离 d, 其作用是:当一个节点 v 被选入种子节点集 S,计算 S 的 2 级邻居以内的节点被激活的概率,如果概率大于或等于 θ 则节点被标覆盖状态。被标记的节点不能被选为种子节点。

定义 4　如果节点 u 是种子节点集 S 两级邻居以内的节点,则 u 被激活的概率 $pact(u)$:

$$pact(u) = 1 - \prod_{v \in S \cap pr(u)} (1 - p(v,u)) + \prod_{v \in S \cap pr(u)} (1 - p(v,u)) \times (1 - \prod_{z \in pr(u) - S,\ w \in pr(z) \cap S} (1 - P(w, z) \times p(z, u))) \tag{3}$$

式中, v 表示从种子节点到节点 u 路径长度为 1 的路径的起点, $pr(u)$ 表示节点 u 的前驱集合, w 表示从种子节点到节点 u 路径长度为 2 的路径的起点, $\prod_{v \in S \cap pr(u)} (1 - p(v, u))$ 表示节点 u 不被集合 $S \cap pr(u)$ 激活的概率, $1 - \prod_{v \in S \cap pr(u)} (1 - p(v, u))$ 则表示 u 被集合 $S \cap pr(u)$ 激活的概率, $1 - \prod_{z \in pr(u) - S,\ w \in pr(z) \cap S} (1 - P(w, z) \times p(z, u))$ 表示 u 被集合 $S \cap pr(z)$ 激活的概率。如果 v、z、w 不存在时,其对应的传播概率为 0。

设图 2 为包含 5 个节点的有向社交网络图,边的传播概率分别为0.01、0.02、0.03、0.04,种子节点集 $S = \{1, 2\}$。S 的 2 级以内的邻居只有 3、4、5 节点,根据式(3)计算它们被 S 激活的概率分别为:

$pact(3) = 1 - (1 - 0.02) + (1 - 0.02) \times 0 = 0.02$

$pact(4) = 1 - (1 - 0.04) + (1 - 0.04) \times 0 = 0.04$

$pact(5) = 1 - (1 - 0) + (1 - 0) \times 0.9986 \approx 0.0014$

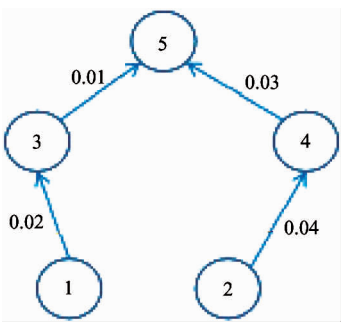


图 2　包含 5 个节点有向社交网络激活概率实例

(3) CTMD 算法在每次标记节点后,使用节点的度值来排名节点的影响力。由于不同的传播模型中每条边的传播概率不同,因此,本文提出了如下方法,即每次标记节点状态后,计算剩下节点的影响力:

$$infs(v) = \sum_{w \in Nei(v) - B,\ v \in V - B} p(v, w) \tag{4}$$

其中, $infs(v)$ 表示节点 v 的影响力,集合 B 表示种子节点集与被标记为覆盖状态的节点集的交集, $Nei(v)$ 表示节点 v 的邻居,在每次标记节点后选择,不使用节点的度值而是选择使得 $infs$ 最大的节点 v

作为种子节点。

3.3　CTMD 算法描述

本文针对影响最大化算法存在的影响力重叠问题，提出了一种新的影响力最大化算法 CTMD 算法，设有一个有向社交网络 $G(V, E)$，种子集 $S = \phi$，则该算法描述如下：

（1）计算所有节点的出度值；

（2）利用式（1）和（2）计算网络中所有节点的影响力；

（3）选择影响力最大的节点 v 作为第一个种子节点，将节点 v 标记为覆盖状态，添加到 S 中；

（4）计算 S 的 2 级邻居以内未被覆盖的节点被激活的概率，标记被激活概率大于覆盖阈值 θ 的所有节点为覆盖状态；

（5）除了被标记为覆盖状态的节点外，根据种子节点和被标记为覆盖状态的节点，更新节点的出度值；

（6）选择更新后出度值最大的节点 v 作为第二个种子节点，添加到 S；

（7）重复步骤（4）~（6）直到选择到 k 个节点为止。

为了说明上述步骤的执行过程，本文给出了如下实例。设图 3 为一个有向的社交网络图，每条边的概率设为 0.06，设所需种子节点数量 $k=2$，$\lambda = 0.8$，由外层到里层 ks 值分别为 0、1、3，初始阶段所有节点都未被标记。Fg 用来存储被标记为覆盖状态的节点。

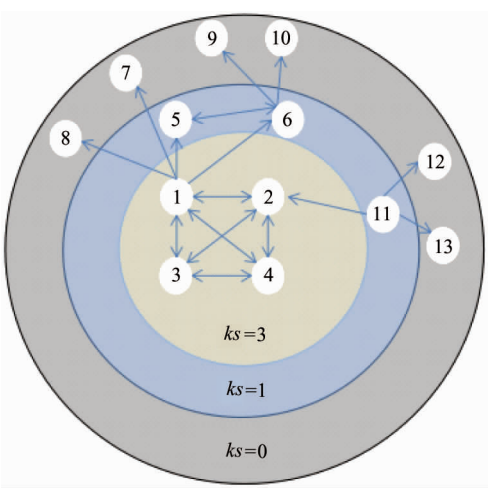


图 3　CTMD 算法种子节点选择模拟图

第 1 轮　计算所有节点的出度值（$D(i)$ 表示节点 i 的出度）：

$D(1)=7$；$D(2)=D(3)=D(4)=D(6)=D(11)=3$；$D(5)=1$；$D(7)=D(8)=D(9)=D(10)=D(12)=D(13)=0$。

利用式（1）和（2）计算所有节点的影响力得到：

$infs(1)=3.336$；$infs(2)=infs(3)=infs(4)=3.144$；$infs(5)=1.048$；$infs(6)=infs(11)=1.144$；$infs(7)=infs(8)=infs(9)=infs(10)=infs(12)=infs(13)=0$。

因此选择节点 1 作为种子节点，将节点 1 标记为覆盖状态并添加到 Fg 中，然后根据式（3）计算与节点 1 距离为 2 以内的所有节点（{2,3,4,5,6,7,8,9,10}）被激活的概率：

$pact(2)=pact(3)=pact(4)=0.067$；$pact(5)=pact(6)=0.063$；$pact(7)=pact(8)=0.06$；$pact(9)=pact(10)=0.0038$。

如果设置 $\theta=0.065$，那么节点集合{2, 3, 4}将被标记为覆盖状态。将这些节点添加进覆盖集合 Fg 中。

第 2 轮　网络 G 中除去 Fg 中所有节点，更新剩余节点的出度值，设节点 v 更新后的出度值为 $d(v) = \text{len}(N(v) - N(v) \cap \text{Fg})$，因此有：

$d(5)=1$；$d(11)=2$；$d(6)=3$；$d(7)=d(8)=d(9)=d(10)=d(12)=d(13)=0$。

选择更新后具有最大出度值的节点 6 作为第 2 个种子节点，然后计算与节点 1、6 距离为 2 以内的且未被覆盖所有节点（{5,7,8,9,10}）被激活的概率：

$pact(5)=0.116$；$pact(7)=pact(8)=pact(9)=pact(10)=0.06$。

所以节点 5 被标记未覆盖状态并添加进 Fg 中，选择过程结束。

3.4　CTMD 算法伪代码

输入：社交网络 $G(V, E)$，种子节点规模 K，平衡参数 λ，覆盖阈值 θ

输出：种子节点集合 S

BEGIN：

（1）dictionary $H = \phi$ $H1 = \phi$，set $D1 = \phi$，$S = \phi$

(2)　for each node i in V:

(3)　　$H[i] = infs(i)$　　//将每个节点的影响力值存入字典 H 中

(4)　end for

(5)　For k in range(K):

(6)　　$v = \{i \mid \max(H[i])\}$

(7)　　D1.add(v) and $H1[v] = 1$　　//将影响力值最大的节点 v 添加进 D1,并且标记节点 v 为覆盖状态

(8)　　S.add(v)

(9)　　D2 = d2(S) - D1　//找到与 S 中节点距离为 2 以内的并且不在 D1 中的所有节点添加进 D2 中,d2(S)

　　//表示与 S 中节点距离为 2 以内的节点的集合

(10)　　For each node u in D2:

(11)　　calculate pact(u)

(12)　　If(pact(u) > = θ):

(13)　　D1.add(u) and $H1[u]$ = true

(14)　end for

(15)　H = ϕ

(16)　For each i in V-D1:

(17)　　$d(i) = D(i) - \text{len}(N(i) \cap D1)$

// $D(i)$ 表示节点的初始出度, $d(i)$ 表示节点 i 更新后的出度

(18)　　$H[i] = d(i)$　　//用更新后的出度值代表节点影响力值

(19)　end for

(20)　end for

上述算法伪代码中,第(1)行表示程序所需要的数据结构,第(2)、(3)行计算网络中每个节点的影响力值并存入字典 H 中;第(5)、(6)行选择影响力值最大的节点 v 作为第一个种子节点,并把它标记为覆盖状态;第(8)~(10)行计算与种子节点距离为 2 以内且没有被覆盖的所有节点被激活的概率;第(11)、(12)行判断节点被激活的概率是否大于覆盖阈值,如果大于覆盖阈值则标记为覆盖状态;第(14)~(16)行更新未被覆盖节点的出度值,以更新后节点出度的大小作为选取种子节点的标准。

CTMD 算法的复杂度分析过程如下:设网络 $G(V,E)$ 的节点数为 n, 边数为 m, 种子集规模为 K, $Y_2(S)$ 为种子集 S 的 2 级邻居集合, $|Path_2(u,S)|$ S 到节点 u 的路径数量, $u \in Y_2(S)$, $R = \max_{|S| \leqslant K}\{|Path_2(S)|\}$, $|path_2(S)|$ 为从 S 到达 S 的所有 2 级邻居节点的路径数量,注意到 R 随 n 呈指数级增长。

算法的时间复杂度主要有 4 个部分构成,伪代码中(2)~(4)行求解节点影响力时间复杂度与网络规模成线性关系,时间复杂度为 $O(n)$; (9)行可用 DFS 算法进行搜索,时间复杂度为 $O(R)$; (11)行计算节点的激活概率的时间复杂度为 $O(|Path_2(u,S)|)$, 由于 $\sum_{u \in Y_2(S)} |Path(u,S)| = |Path(S)| \leqslant R$, 故(10)~(14)行的时间复杂度为 $O(R)$; (17)~(19)行更新节点的度值时间复杂度为 $O(n)$; 计算得总时间复杂度为 $O(K(R+n))$。

算法的空间复杂度为 $O(n+m+R)$, 其中 $n+m$ 为输入网络规模,R 为需要存储的所有路径。

4　实验设计与结果分析

4.1　实验数据集

本文实验的 3 个真实数据集取自网站 http://snap.stanford.edu/data/。第一个网络是 Wiki-Vote,该网络包含从维基百科开始到 2008 年 1 月的所有维基百科投票数据。网络中的节点代表维基百科用户,从节点 i 到节点 j 的有向边表示用户 i 投票给用户 j;第二个网络 cit-HepTh 是一个高能物理论文引用网络,如果一篇论文 i 引用了论文 j,则该图中包含从 i 到 j 的有向边;第三个网络是 web-NotreDame,节点表示来自圣母大学(Domain nd.edu)的页面,有向边表示它们之间的超链接;注意在计算影响力时,所有数据集的边都需要反向处理,另外,由于 web-NotreDame 网络含有自循环的边,因此需要提前删除这些边。数据集的基本信息如表 1 所示。

为了验证 CTMD 算法的性能,本文将其与 Degree 算法、CCA 算法和 IRIE 算法进行实验对比,3 种算法描述如表 2 所示。

表 1　数据集

名称	节点数	边数	平均聚集系数	网络直径
Wiki-Vote	7115	103689	0.1409	7
cit-HepTh	27770	352807	0.3120	13
web-NotreDame	325729	1497134	0.2346	46

表 2　算法描述

算法名称	算法描述
Degree	选取网络中度最大的 k 个节点
CCA	基于网络层次结构和影响半径 d 的启发式算法
CTMD 算法	基于覆盖阈值 θ 的度最大启发式算法
IRIE	影响力估计影响力排名

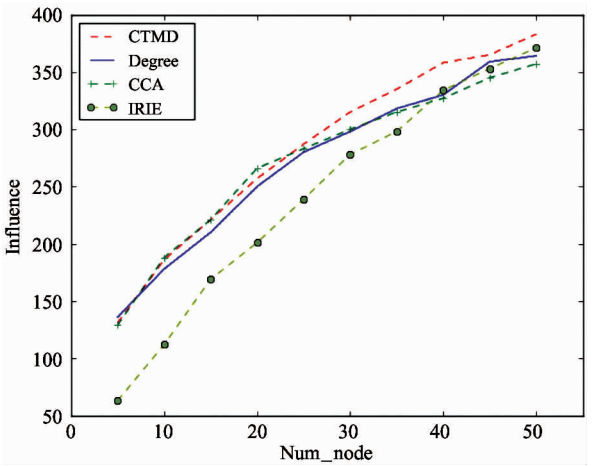
对于 CCA 算法，在所有实验中 d 设置为 1，对于 CTMD 算法在所有实验中平衡参数 λ 设置为 0.8，覆盖阈值 θ 依据网络和传播概率而定。实验中种子节点数量取值范围为 1 ~ 50，使用的传播模型是独立级联模型（IC 模型）和权重级联模型（WC 模型）。IC 模型假设网络中每条边的传播概率相同，本文在不同传播概率（$p \in \{0.02, 0.04, 0.06\}$）下分析对比各算法性能。WC 模型设网络中每条边的概率等于节点入度的倒数。本文通过蒙特卡洛模拟传播过程 1000 次取平均值来精确估计节点集合的影响力。

4.2　IC 模型下的实验结果与分析

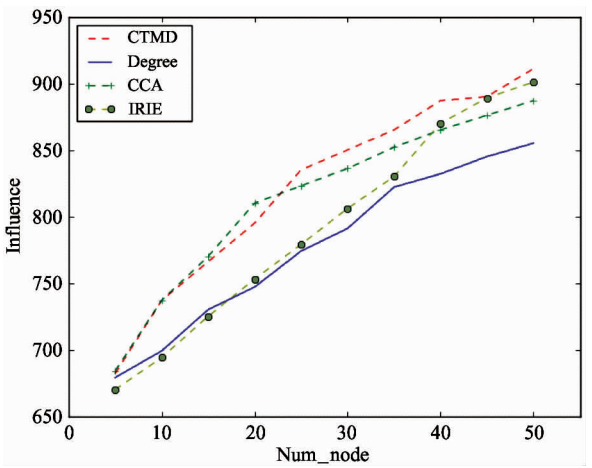
本文对社交网络影响最大化算法的影响范围评价指进行测试。影响范围是指利用算法给出一个初始种子节点集合，传播结束后使得受影响节点的数量越多，则该算法效果越好；图 4 是在小规模的社交网络 Wiki-Vote 上的实验结果，对于 CTMD 算法，当传播概率 $p = 0.04$，近似最佳覆盖阈值 $\theta = 0.08$，因此，当 p 的取值不同时，令 $\theta = 0.08$。

图 4（a）、（b）和（c）显示了在不同传播概率下，选择 50 个种子节点的传播过程中 3 种算法的影响范围，其中，横坐标为种子节点个数，纵坐标为激活节点个数。

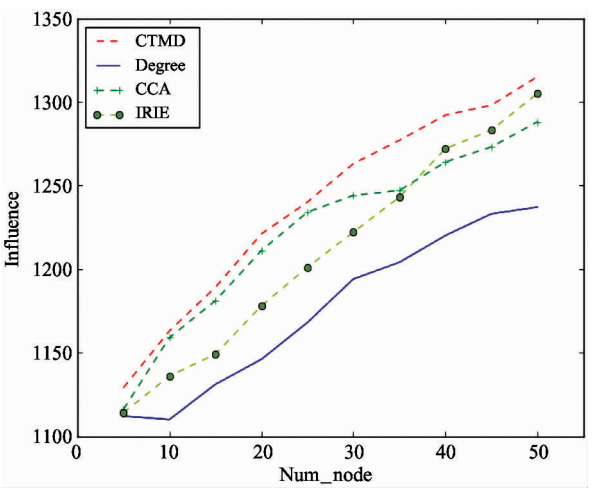
如图 4（a），当 $p = 0.02$ 时 CTMD 算法的影响范围好于其它 3 种算法，CTMD 算法的影响效果比 IRIE 和 Degree 算法分别提高了 3.23% 和 5.22%，



（a）$p = 0.02, \theta = 0.08, \lambda = 0.8$



（b）$p = 0.04, \theta = 0.08, \lambda = 0.8$



（c）$p = 0.06, \theta = 0.08, \lambda = 0.8$

图 4　IC 模型下 Wiki – Voted 网络中不同传播概率的影响范围比较

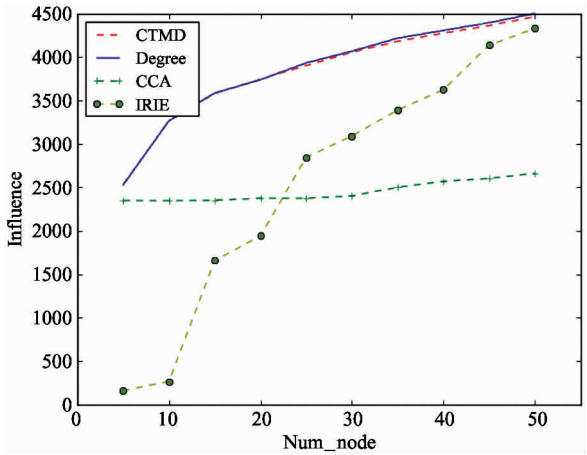
CCA 算法效果最差;如图 4(b),当 $p=0.04$ 时,CTMD 算法的影响效果比 IRIE 和 CCA 算法分别提高了1.11% 和 2.71%,degree 算法效果最差;如图 4(c),当 $p=0.06$ 时,CTMD 算法的影响效果稍好于 IRIE 算法,比 CCA 算法提高了 2.10%,Degree 算法效果最差。从图 4 中还可以看出,随着传播概率变大,CTMD 算法的效果越来越好,这是因为覆盖阈值 θ 取的是在传播概率 $p=0.04$ 时的近似最佳值,当 $p=0.04$ 时,被标记未覆盖状态的节点是固定的,在传播过程中这些节点的绝大多数被激活,但还是有少数激活概率接近 0.04 的节点没被激活,当传播概率变大时,这些少数的节点将被激活,覆盖状态的节点几乎等于激活的节点,故影响效果更好。

图 5(a)、(b)和(c)是在一个较大规模社交网络 web-NotreDame 上的实验结果,对于 CTMD 算法,当传播概率 $p=0.04$ 时,最近似最佳覆盖阈值 $\theta=0.14$,因此,当 p 取值不同时,也令 $\theta=0.14$,从图 5 中可以看出,当 $p=0.04$、0.06 时,CTMD 算法都明显好于其余 3 种算法。

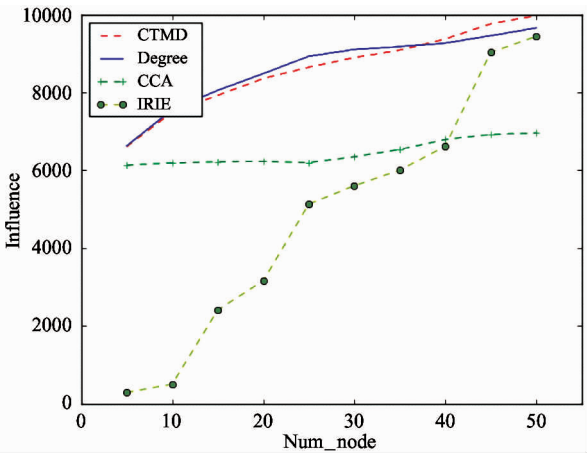
如图 5(a),当 $p=0.02$ 时 Degree 算法的影响效果最好,稍高于 CTMD 算法,比 IRIE 算法提高了 4.00% 左右,比 CCA 算法提高 35% 左右;如图 5(b),当 $p=0.04$ 时,CTMD 算法效果最好,比 Degree 算法和 IRIE 算法分别提高了 3.28%、5.67%,CCA 算法效果最差;如图 5(c),当 $p=0.06$ 时,CTMD 算法的影响效果比 IRIE 和 Degree 算法分别提高了 5.49%、8.20%,所以在 IC 模型下本文的算法具有明显的优势。

4.3　WC 模型下的实验结果与分析

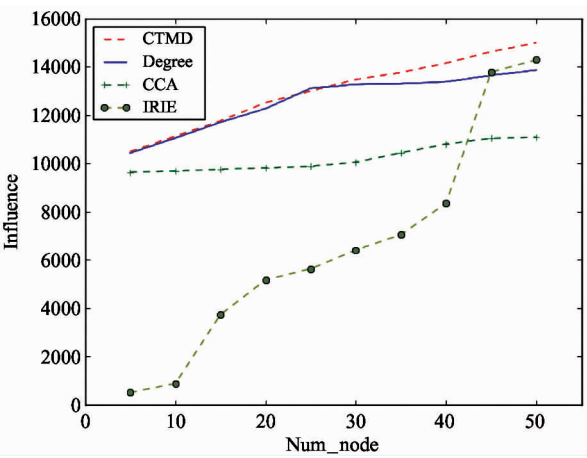
在实际的社交网络中边的传播概率很有可能各不相同,图 6 是 WC 模型下在 web-NotreDame 网络上的实验结果,CTMD 算法的覆盖阈值 $\theta=0.14$,显示了在不同传播概率下选择 50 个种子节点过程中 3 种算法的影响范围。从图 6 中可以看出,在 WC 模型下,IRIE 算法的影响效果最好,比 CTMD 算法和 Degree 算法分别提高了 4.29%、16.84%,CCA 算法效果最差。图 7 是 WC 模型下在 cit-HepTh 网络上的实验结果,CTMD 算法的覆盖阈值 $\theta=0.16$,从图 7 中可以看出 IRIE 算法的影响效果最好,比 CTMD



(a) $p=0.02,\theta=0.14,\lambda=0.8$



(b) $p=0.04,\theta=0.14,\lambda=0.8$



(c) $p=0.06,\theta=0.14,\lambda=0.8$

图 5　IC 模型下 Web－NotreDame 网络中不同传播概率的影响范围比较

和 CCA 算法提高了 3.81%、16.17%,Degree 算法的影响效果最差。因为 θ 不是近似最佳覆盖阈值,因

此,CTMD 算法的影响效果还可以进一步提升。

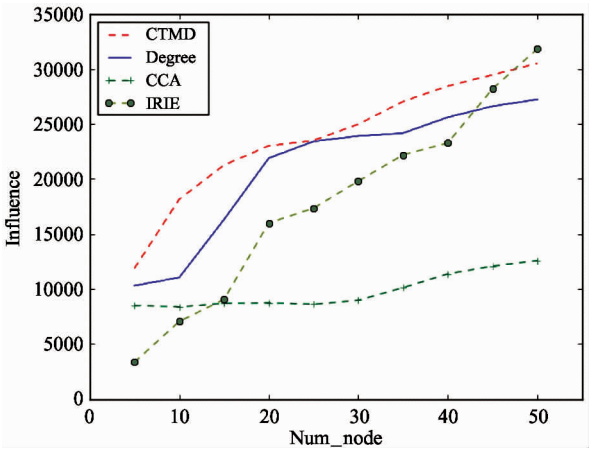


图 6　WC 模型下 web – NotreDame 网络($\theta=0.14$, $\lambda=0.8$)种子节点影响效果比较

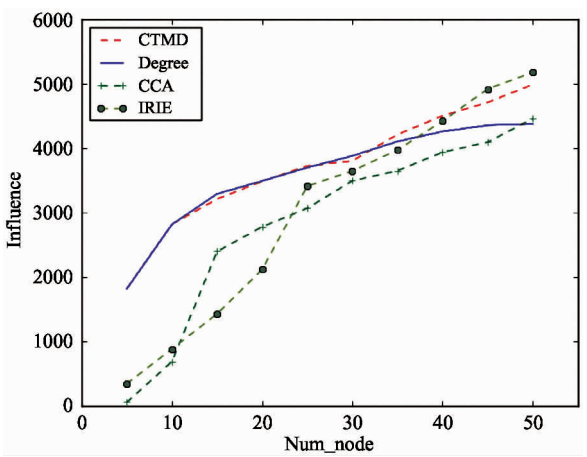


图 7　WC 模型下 cit – HepTh 网络($\theta=0.16$, $\lambda=0.8$)种子节点影响效果比较

通过分析 WC 模型下的实验结果可知,CTMD 算法的影响效果低于 IRIE 的原因是由于 CTMD 算法在每次标记节点后,使用节点新的度值(需减去已标记节点的数量)来排名节点的影响力。由于 WC 模型中每条边的传播概率不同,因此没有达到预期的效果,为此,本文采用覆盖策略中的式(4)方法来解决。

应用该策略后,在 WC 模型下 cit-HepTh 网络上的实验结果如图 8 所示;在 web-NotreDame 网络上的实验结果如图 9 所示。从图 8 中可以看出 CTMD 算法的影响效果比 IRIE 算法提高了 3.56%。从图 9中可以看出 CTMD 算法的影响效果比 IRIE 算法提高了 1.24%。

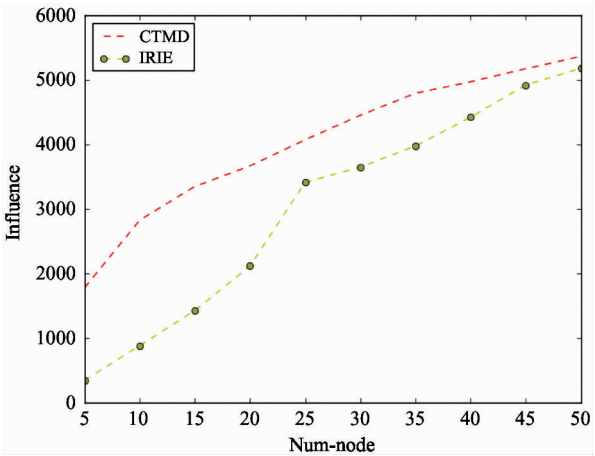


图 8　WC 模型下 cit – HepTh 网络($\theta=0.16$, $\lambda=0.8$)改进后的种子节点影响效果

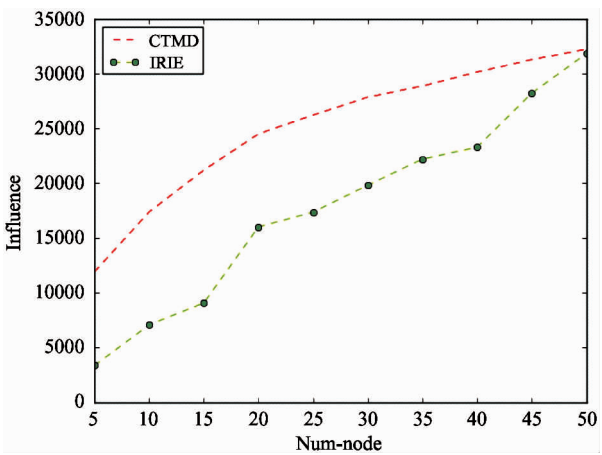


图 9　WC 模型下 web – NotreDame 网络($\theta=0.14$, $\lambda=0.8$)改进后的种子节点影响效果

此外,本文比较了在 IC 模型下,从不同的数据集中选择 50 个种子节点,进行多次测量去平均值的方法比较了 CTMD 算法与其余算法的运行时间,如表 3 所示。

表 3　运行时间(s)

名称	Wiki-Vote	cit-HepTh	web-NotreDame
Degree	0.57	1.05	3.57
CCA	0.71	1.48	69.47
IRIE	8.38	79.26	612.24
CTMD	3.21	9.34	122.15

从表中可以看出,在较大规模的 web-NotreDame 网络上,算法 CTMD 的运行时间比 IRIE 算法少,其

原因是 CTMD 算法只考虑两度以内的节点被激活的概率,随着迭代的增加,有大量节点被标记为覆盖,因此每一次迭代后需要计算的节点数量迅速减少,而 IRIE 算法要考虑所有节点的最短路径;CTMD 算法的时间复杂度比 Degree 和 CCA 算法高,但影响效果却是最好的。

5　结　论

为了解决目前的影响力最大化算法存在的影响力重叠、时间复杂度高等问题,本文提出了 CTMD 算法。首先,该算法利用改进的 k-shell 算法计算节点影响力以选取第一个种子节点;其次,计算与种子节点距离为 2 以内的节点被激活的概率,基于覆盖阈值 θ,把易激活的节点标记为覆盖状态;最后,更新节点影响力,选择影响力最大的节点作为种子节点,CTMD 算法重复执行上述过程直到选择到指定数量的种子节点。实验结果表明 CTMD 算法表现出较好的适应性。在实际网络中,每条边的传播概率有可能不相同, CTMD 算法在 IC 和 WC 模型下比 Degree 算法、CCA 算法和 IRIE 算法有更好的影响效果;从运行时间上来看,CTMD 算法运行时间适中,但在影响效果上是最好的。因此,该算法确保了在较低时间复杂度的同时,影响力传播效果最优,且具有较高的实用价值。CTMD 算法核心的内容是覆盖阈值的选取,不同的覆盖阈值 θ, 将对算法产生不同的影响效果。从实验中可知,θ 与转播概率和网络结构有关,本文下一步的工作是探索 θ 值与传播概率和网络结构是否存在定量关系。

参考文献

[1] Richardson M, Domingos P. Mining knowledge-sharing sites for viral marketing[C]. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton Alta, Canada, 2002. 61-70

[2] Kempe D, Kleinberg J, Ardos é. Maximizing the spread of influence through a social network[C]. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, USA, 2003. 137-146

[3] Chen W, Wang Y, Yang S. Efficient influence maximization in social networks[C]. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 2009. 199-207

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

[5] 田家堂, 王轶彤, 冯小军. 一种新型的社会网络影响最大化算法[J]. 计算机学报, 2011, 34(10):1956-1965

[6] 曹玖新, 董丹, 徐顺, 等. 一种基于 k-核的社会网络影响最大化算法[J]. 计算机学报, 2015, 38(2):238-248

[7] Aybike Ş, Resul K. Using swarm intelligence algorithms to detect influential individuals for influence maximization in social networks[J]. *Expert Systems with Applications*, 2018,114:224-236

[8] Tang J, Tang X, Yuan J. An efficient and effective hop-based approach for influence maximization in social networks[J]. *Social Network Analysis & Mining*, 2018, 8(1):10

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

[10] Vichaya A, Joseph K, Fredrickson P A, et al. IRIE: a scalable influence maximization algorithm for independent cascade model and its extensions[J]. *Rev Crim*, 2011, 56(10):1451-1455

[11] 刘晓东. 大规模社会网络中影响最大化问题高效处理技术研究[D]. 湖南:国防科学技术大学计算机学院, 2013. 75-92

[12] Litou I, Kalogeraki V, Gunopulos D. Influence maximization in a many cascades world[C]. In: Proceedings of the 37th IEEE International Conference on Distributed Computing Systems, Atlanta, USA, 2017. 911-921

[13] Zareie A, Sheikhahmadi A, Khamforoosh K. Influence maximization in social networks based on TOPSIS[J].

Expert Systems with Applications, 2018, 108: 96-107

[14] 宋永浩, 史骁, 胡斌, 等. 基于多目标贪心策略的增益最大化团队构建算法[J]. 高技术通讯, 2018, 28(4):279-290

[15] Yang W S, Weng S X. Application of the ant colony optimization algorithm to competitive viral marketing[C]. In: Proceedings of the 7th Hellenic Conference on Artificial Intelligence, Lamia, Greece, 2012. 1-8

[16] Bozorgi A, Samet S, Kwisthout J, et al. Community-based influence maximization in social networks under a competitive linear threshold model[J]. *Knowledge-Based Systems*, 2017,134:149-158

[17] 刘院英, 郭景峰, 魏立东, 等. 成本控制下的快速影响最大化算法[J]. 计算机应用, 2017, 37(2):367-372

[18] Bucur D, Iacca G, Marcelli A, et al. Improving multi-objective evolutionary influence maximization in social networks[C]. In: Proceedings of the 21st International Conference on Applications of Evolutionary Computation, Parma, Italy, 2018. 117-124

[19] 李小康, 张茜, 孙昊, 等. 社交网络中多渠道影响最大化方法[J]. 计算机研究与发展, 2016, 53(8): 1709-1718

[20] 张平, 王黎维, 彭智勇, 等. 一种面向团体的影响最大化方法[J]. 软件学报, 2017, 28(8):2161-2174

[21] Wang Y, Vasilakos A V, Jin Q, et al. PPRank: Economically selecting initial users for influence maximization in social networks[J]. *IEEE Systems Journal*, 2017, 11(4):2279-2290

Research on influence maximization algorithm based on coverage threshold

Chen Jing[*,**,***], Liu Xian[*]

([*] College of Information Science and Engineering, Yanshan University, Qinhuangdao 066004)

([**] Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao 066004)

([***] Software Engineering Laboratory in Hebei Province, Qinhuangdao 066004)

Abstract

The existing influence maximization algorithm has problems, such as overlapping seed node influence and high time complexity, when the edges have different propagation probabilities in the network, and the node influence cannot be estimated accurately. Therefore, A new heuristic algorithm called coverage threshold maximum degree (CTMD) is proposed to solve the existing problems. The main idea of the algorithm is to use the improved k-shell algorithm to calculate the influence of the node to select the first seed node. Calculating the activation probability of nodes within two degrees, and on the basis of the coverage threshold θ, the easily-activated node is marked as an overlay state. The node degree value is updated until a specified number of seed nodes is selected. The experiments are performed to compare and analyze the performance of the core coverage algorithms (CCA), Degree (MaxDegree), IRIE (influence ranking influence estimation) and CTMD. The experimental results show that in the IC model and the WC model, the influence range of the CTMD algorithm has a good advantage. In addition, by testing the running time of the CTMD algorithm, it can be seen that with the gradual increase of the network size the algorithm has a low time complexity.

Key words: social network, node influence, influence maximization, coverage threshold, k-shell