多社交网络的影响力最大化分析

李国良 楚娅萍 冯建华 徐尧强
(清华大学计算机科学与技术系 北京 100084)

摘 要 影响力最大化旨在从网络中识别 k 个节点,使得通过这 k 个节点产生的影响传播范围最大.该问题在病毒营销领域具有重要的应用背景,它已经引起了学术界和工业界的广泛研究.该文作者观察到已有的研究工作大多数只是针对单一网络,即在给定的一个网络上识别 k 个节点使得其在该网络上产生最大的影响范围;然而,随着社交网络的普及,丰富多样的社交平台不断涌现,以满足不同的社交需求,这使得社交人群不被局限在一个网络内,而是分布在不同的社交网络上.这种变化的一个直接影响是使得基于病毒式营销的应用,例如单一网络上的产品推广愈加不能满足推广的广度需求,很可能是单一网络上的用户量不能达到推广的目标人群数量,又或者广告商期望在多个网络平台上找到 k 个用户以最大化影响传播范围.为此,在文中,作者研究多社交网络上的影响力最大化问题.该文首先仔细地研究了影响力最大化问题在单一网络和多社交网络上的不同,并提出了实体的自传播特性以在多个网络之间建立联系.之后,作者提出了多社交网络上的影响计算模型来建模节点间的影响力,然后扩展了基于树的算法模型以适应多社交网络上的影响力最大化问题.基于所提出的影响计算模型和扩展的基于树的算法模型,作者提出了多种策略的优化算法.例如通过深层次挖掘自模特性来避免冗余计算,通过使用影响增益上界近似准确的增益来加速种子选取过程等,最后通过真实数据集上的实验表明文中所提方法在性能和影响范围上都优于已有的算法.

关键词 社交网络;影响力最大化;多社交网络;传播模型;影响力;社会媒体;数据挖掘
中图法分类号 TP393 **DOI 号** 10.11897/SP.J.1016.2016.00643

Influence Maximization on Multiple Social Networks

LI Guo-Liang CHU Ya-Ping FENG Jian-Hua XU Yao-Qiang
(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

Abstract Influence Maximization aims to identify k nodes from a network such that the influence spread invoked by the k nodes is maximized. It has various real applications in viral-marketing areas and has been extensively studied by the academic and industrial communities. We observe that existing works all focus on a single network, which identifies k nodes that has the maximal influence spread on one given network; however, with the popularization of social networks, a variety of social platforms are emerged to fulfill various social needs, which leads that social populations will not be confined to a social network, and they will be distributed in different social networks. The direct issue is that the viral-marketing based applications, such as the product promotion on a single network can't meet the breadth demands of current marketing promotion, it's probably that the whole user of the network can't reach the number of targeted population, or the advertisers expect to maximize the influence spread on multiple social networks with k users. Compared with single network, more challenges come forth. It is challengeable to find k users that have the maximal influence spread on the multiple social networks, since it has been proven that influence maximization problem is NP hard. It is more complex to evaluate the

收稿日期:2015-05-27;在线出版日期:2015-10-15.本课题得到国家自然科学基金(61373024,61422205)、国家"九七三"重点基础研究发展规划项目基金(2015CB358700)资助.**李国良**,男,1980 年生,博士,副教授,国家自然科学基金优秀青年科学基金获得者,中国计算机学会(CCF)高级会员,主要研究方向为数据库可用性、数据清洗与集成、时空文本数据处理、众包数据管理、社交网络影响分析.E-mail:liguoliang@tsinghua.edu.cn.**楚娅萍**,女,1988 年生,硕士研究生,主要研究方向为社交网络影响力分析.**冯建华**,男,1967 年生,博士,教授,中国计算机学会(CCF)杰出会员,主要研究领域为数据库、数据仓库、Web 数据管理.**徐尧强**,男,1976 年生,博士,工程师,主要研究方向为大数据分析.

influence strength between nodes, since the information propagation is more intricacy among multiple social networks. Besides, entity recognition has to be considered when analyze influence on multiple social networks, since it is normal for a person to have multiple social network accounts. With regards to these, in this paper, we study the influence maximization problem on multiple social networks. In summary, we study the differences of influence maximization between single network and multiple social networks carefully, and propose the self propagation property of entity to build relation among different networks. Later, we propose the influence calculation model to model the influence strength between nodes. We extend the tree-based algorithm model to adapt the multiple social networks situation, based on the proposed influence calculation model and the extended tree-based algorithm model, and we propose multiple optimized strategies to promote performance, such as by further exploring the submodular property to avoid redundant computation, to accelerate seed selection by using the upper influence marginal benefit to approximate the accurate benefit, etc. Finally, numerical experimental studies on real datasets demonstrate the proposed algorithms outperform existing methods significantly, and detailed experimental studies from influence spread and running time have been illustrated respectively.

Keywords social network; influence maximization; multiple social network; influence model; social media; data mining

1 引 言

互联网的快速发展,不仅带来了海量的数据,也丰富了人们之间的交流、沟通渠道.多样的社交网络在给予人们多种交流平台选择的同时,也为一些传统的生产方式提供了机会.在众多的营销方式中,口口相传①(word-of-mouth,也被称为病毒式营销②)被认为是最为有效的营销方式.不同于其他的营销策略,口口相传基于个体之间的相互信任而进行产品的传播和推广,研究[1]表明,相比之下,人们更倾向于信任从亲戚、朋友等强社交关系那里所获得的信息,基于社交网络的病毒式营销正是基于这种共识而被广泛接受.

得益于互联网和社交网络的发展,传统的社交关系正逐渐从线下转移到线上,使得对这种社交关系的跟踪和分析更加容易.所有这些使得社交网络上的病毒式营销倍受关注,作为社交网络影响力研究领域的一个重要问题,"影响力最大化"对应于病毒式营销的应用研究,更是引起了学术界和工业界的广泛关注和研究热情.影响力最大化问题最早由 Domingos 和 Richardson 等人[2-3]提出,而后 Kempe 等人[4-5]进一步提出了 top-k 的影响力最大化问题,即假设给定只能满足 k 个用户的预算费用,如何找到 k 个用户使得通过这 k 个用户所产生的影响传播范围最大.在线性阈值模型和独立级联模型下,影响力最大化问题被证明为 NP 难问题[4],而对给定 k 个用户的影响范围的准确评估被证明为#P 难问题[6].

针对影响力最大化方面的研究,已经有很多学者提出过不同方法来解决此问题.Kempe 等人提出了一个近似比为$(1-1/e)$的贪心算法来解决此问题;由于贪心算法的最坏运行时间为 $O(n^2(m+n))$,对于稍大规模的社交网络来讲伸缩性遇到挑战,计算代价高昂,为此很多学者都针对贪心算法的伸缩性问题进一步提出了很多算法以提升计算性能.

Leskovec 等人[7]通过挖掘影响函数的子模特性,提出了 CELF(Cost-Effective Lazy-Forward)算法,该算法极大地减少了评估种子影响范围的次数,他们的实验表明 CELF 算法相比贪心算法有近 700 倍的速度提升,尽管 CELF 的性能提升很明显,但是在数万节点规模的网络上寻找 top-50 的节点仍然需要数个小时的时间[8].CELF++是 Goyal 等人[9]在 CELF 算法的基础上通过进一步挖掘子模特性而优化的算法,虽然 CELF++相比 CELF 维护的信息更多,但是比 CELF 有了近 17%~61%的性能提升.

除了基于贪心算法的改进之外,还有使用启发式进行优化的算法.Kimura 和 Saito 等人[10]提出了基于最短路径的影响级联模型 SPM 和 SP1M,并据此提出算法计算该模型下的影响范围,该模型假设每个节点都只通过最短路径进行信息传播,在 SPM

① http://en.wikipedia.org/wiki/Word-of-mouth_marketing
② http://en.wikipedia.org/wiki/Viral_marketing

和 SP1M 模型下，每个节点的影响范围都可以被准确计算出来，但是这些模型忽略了用户之间的影响概率，仅仅使用最短路径而忽视了用户间的影响在传播中的重要作用.

Chen 等人[11]对 Kempe 等人所提的贪心算法进行了优化，结果表明通过对贪心算法改善而极大提升性能是非常困难的，随后提出了 degree-discount 算法，该算法虽然能提升计算性能，但是该方法假设在独立级联模型下所有边上的影响概率值都一样，很显然同现实需求不符. 另外文献[12]提出了使用社区结构来聚合具有相似特征的节点以期减少计算过程中的节点数量. Goyal 等人[13]所提的 SIMPATH 算法被证明在线性阈值模型下较为有效，他们实验表明 SIMPATH 在运行时间、内存损耗和影响范围具有很好的性能. Jiang 等人[14]还首次提出了使用模拟退火的方法来解决影响力最大化问题. Jung 等人[15]提出了 IRIE 方法以期伸缩性的解决影响力最大化问题.

在最短路径的影响级联模型下，Chen 等人[6]提出了使用最大传播路径的启发式算法来解决该问题. 即假设用户之间的信息传播按照最大传播路径而非最短路径进行传播，考虑了用户之间的影响概率. 对每个节点使用最大传播路径建立它的局部树结构，并以此评估每个节点的影响范围，通过设置最大传播路径的阈值过滤不重要的节点以减少候选节点规模. 基于最大传播路径的传播模型具有子模性，保证所求节点集合能够达到 $1-1/e$ 的近似最优比. 实验证明[6]基于最大传播路径的启发式算法能极大地提升性能. 在文献[16]中，作者使用社区结构通过聚集具有相似特性的来减少需要检验的节点数目. Kim 等人[17]提出了使用 OpenMP 元编程表达式的并行计算方法来提升计算性能. Tang 等人[18]提出了一个接近最优时间复杂度的算法和启发式算法来提升效率. Cheng 等人[19]介绍了一种基于自适应排序的算法来解决此问题.

另外还有很多针对影响力最大化方面的扩展工作. Li 等人[20]研究了基于地理位置的影响力最大化问题，即在全局网络内找到 k 个用户使得这 k 个用户对所查询的地理位置能够产生最大的影响范围. Tang 等人[21]将关系分类考虑到影响传播的过程中，通过用户之间的关系类别和待推广的产品特性，使带推广产品尽可能地在适当的关系类别中传播以减少在不必要的关系中传播. 此外 Barbieri 等人[22]还研究了话题感知的影响力最大化问题，Chen 等人[23]更进一步地研究了在线的话题感知的影响力最大化问题.

此外，还有一些其他的研究致力于通过使用机器学习或者数据挖掘的方法来提取影响传播模型下的参数问题. 例如 Tang 等人[24]提出了基于因子图模型的 TFG 模型来定量地衡量社交网络中用户间在不同话题上的影响力强度；Liu 等人[25]使用概率模型将用户节点的话题分布以及用户间的影响力强度结合的方式来学习用户间的影响力强度；Weng 等人[26]在 Twitter 上分析了不同话题上的影响力强度，首先通过 LDA 话题模型提取用户的主题分布，然后使用 TwitterRank 方法计算用户间在不同话题上的影响概率. Chen 等人[28]在复杂网络上研究了识别有影响力节点的问题，提出了一种介于中心度低相关和其他较为耗时方法折中的半局部中心度量的方法来识别有影响力的节点，最后在 SIR 传播模型上评估性能.

现有的影响力最大化研究大多针对单一网络进行分析，然而当前社交网络种类十分丰富，不同的社交网络针对不同的受众用户具有自身的特点. 例如新浪微博和 Twitter 关注信息流通，专注于共享微博客；人人网和 Facebook 注重朋友之间的互动和联络；知乎和 Quora 侧重于问答(Q&A). 而现如今的产品推广重点在于影响力最大化，即将信息推广传播到最多的用户上，为此除了找到关键节点用户之外，还需要找到更多的待推广人群，为此多个社交网络上的影响力最大化正成为营销需求. 多社交网络即在给定的多个目标网络中，找到 k 个用户，使得通过这 k 个用户的影响使得最后在多个社交网络上的传播范围最大.

相比单一网络的影响力最大化问题，多个社交网络保证待传播的用户数量更多，但同时也面临着更多挑战：(1) 如上所述，影响力最大化问题为 NP 难问题，而准确评估用户的影响范围为 #P 难问题，多网络上用户数量的增加首先带来了计算挑战：更多的候选种子节点，更广的传播范围；(2) 相比单一网络，多网络上的信息传播以及对于用户之间的影响力评估更复杂；(3) 多网络需要考虑网络间节点的实体链接①和识别问题，避免在计算节点的影响范围时产生计算错误. 因此，多社交网络上的影响力最大化研究挑战更多，复杂度更高.

针对上述挑战，本文研究了多社交网络上的影响力最大化问题，首先分析了多社交网络上信息传播的特征并对多社交网络上的影响力最大化问题进

① http://en.wikipedia.org/wiki/Entity_linking

行了定义,提出了影响计算模型解决多网络上节点间的影响概率,通过扩展树的算法模型提出了几种优化方法,并通过实验对比了所提算法的性能.

本文第 2 节介绍多社交网络信息传播的自传播特性以及阐述多社交网络上的影响力最大化问题;第 3 节介绍多社交网络上的影响计算模型以及基于树的算法模型;第 4 节介绍文章所提的几种优化算法;在第 5 节给出在真实数据集上的对比实验;最后在第 6 节总结文章工作.

2 自传播性与问题定义

2.1 自传播

给定 n 个网络 $G_1(V_1,E_1),G_2(V_2,E_2),\cdots,G_n(V_n,E_n)$,对于每个网络 $G_i(V_i,E_i)(0\leq i\leq n)$,$V_i$ 表示网络 G_i 中的节点集合,E_i 表示网络 G_i 中的边集合,ET_i 表示网络 G_i 中的实体集合. 本文使用 V 表示所有的节点集合,即 $V=\bigcup_{i=1}^{n}V_i$,使用 ET 表示所有的实体集合,即 $ET=\bigcup_{i=1}^{n}ET_i$. 由于不同网络中不同名称的节点可能指代同一个实体(entity),为了表示他们之间的关系,每个节点使用它的表面名称(surface name,该节点在网络中的名称)、网络身份(network ID,该节点所在的网络)和实体身份(entity ID,该节点实际指代的实体)3 种信息来表示. 比如 $u(e_u,G_i)$ 就表示存在网络 G_i 中,名为 u 实际指代实体 e_u 的节点.

对于任意实体 e_u,它可能存在于多个网络中,例如实体 e_u 分别在人人网和新浪微博上都有账户,分别表示为 $u(e_u,RR)$ 和 $u(e_u,WB)$. 当节点 $u(e_u,WB)$ 在微博上看到一条很有意思的文章,为了将此文章分享给其在人人网上的朋友,$u(e_u,WB)$ 将该文章转发到了人人网上. 在这个过程中,该文章被用户 $u(e_u,WB)$ 从新浪微博转发到了人人网中,本文称这种行为为自传播行为,具体定义如定义 1.

定义 1. 自传播. 给定存在网络 G_i 和网络 G_j 的实体 e_u,分别表示为 $u(e_u,G_i)$ 和 $u(e_u,G_j)$. 如果实体 e_u 将信息从网络 G_i 传播到了网络 G_j(即信息由节点 $u(e_u,G_i)$ 传播到了节点 $u(e_u,G_j)$),那么称这种行为为自传播. 由于每个实体在两两网络间的自传播频率不同,这里统一使用参数 $\delta_{e_u(G_i,G_j)}$ 表示实体 e_u 从网络 G_i 到网络 G_j 的自传播概率.

显然,正是由于实体在不同网络间的自传播,使得信息能够在不同的网络之间进行传播,而不是封闭在一个网络内.

图 1 描述了原始网络(子图 1)在自传播的基础上形成的新网络结构(子图 2). 其中子图 1 包含了人人网和新浪微博两个社交网络,子图 1 中节点间的实线表示节点之间的传播概率,不同网络间相同的节点标识表示相同的实体. 子图 2 为在自传播特性上新形成的网络结构. 其中虚线表示实体之间的自传播概率. 比如 $\delta_{a(a_i,a_j)}$ 表示实体 a 从网络 1(人人网)到网络 2(新浪微博)的自传播概率为 1. 如果一个自传播概率 $\delta_{e_u(G_i,G_j)}$ 为 1,则表示实体 e_u 总是将 G_i 中的信息传播到网络 G_j 中. 否则就以 $\delta_{e_u(G_i,G_j)}$ 的概率将网络 G_i 中的信息传播到网络 G_j 中.

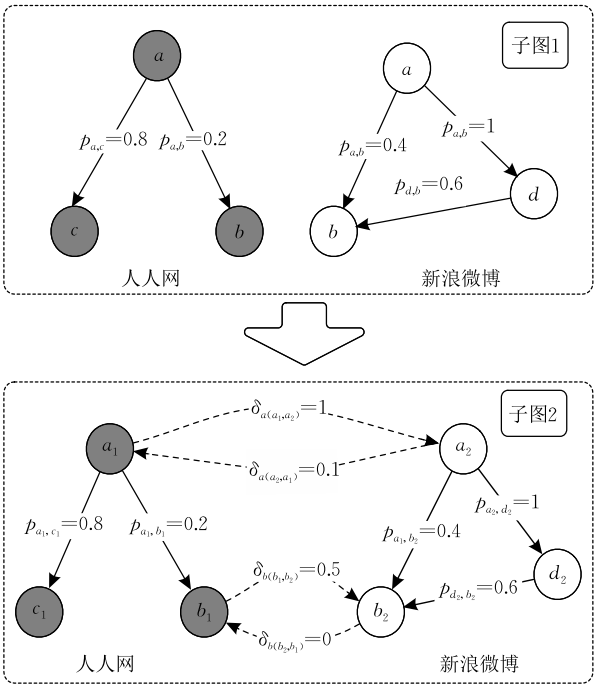

图 1 通过自传播生成的新网络结构

由于每个网络的功能不同,面向的受众人群也有所不同,同一实体在不同的网络中具有不同的影响力. 例如在新浪微博上影响力很大的人,在人人网上的影响力或许就非常小(这里简单就粉丝数量来衡量影响力的话):中国知名女演员姚晨在新浪微博上拥有粉丝数量 77 930 000 人,但在人人网上的粉丝只有 6697 人,相比之下,其微博粉丝数比人人网高出近 11 637 倍. 如果想要通过姚晨来传播信息,那么可能只需在微博上发布一次而无需在人人网上再次发布就足以覆盖其多数粉丝. 使用信息发布次数进行计费,不仅减少了冗余信息的发布,同时降低了推广费用. 为此本文在选择关键节点以最大化传播范围的时候,是按照节点而非实体进行选择,但是在计算传播范围的时候,需要按照实体进行度量,避免计算结果误差过大.

2.2 传播模型及问题定义

本文使用传播模型中使用最为广泛和最具代表

性的独立级联模型[4,6,27]来模拟传播过程.在独立级联模型中,每个节点有两种状态:激活和未激活,其中激活表示该节点接受或者传播某种信息(例如微博上的转发、点赞等行为).形式上,独立级联模型可以归纳如下:在初始阶段,初始激活节点集合(也称为种子集合)$S\in V$ 被选中在这 n 个网络中进行传播,这里使用 A_t 和 E_t 分别表示在阶段 t 后被激活的节点集合和实体集合,因此有 $A_0=S$,$E_0=\bigcup_{u\in A_0}e_u$,在阶段 $t+1$,每个在集合 A_t 中的节点 u 都只有一次机会来激活它的处于非激活状态下的外向邻居节点,此过程一直持续到 $A_t=\varnothing$,此时 $E_t=\varnothing$.在这个过程中所有被 S 所激活的实体集合记为 $\sigma(S)=\bigcup_{i=0}^{i=t}E_i$,其中 $E_{t+1}=\varnothing$.本文称 $\sigma(S)$ 为集合 S 的影响范围(influence spread).在此基础上,多网络上的影响力最大化问题的定义如定义 2.

定义 2.　多社交网络的影响力最大化.给定 n 个社交网络 $G_1(V_1,E_1),G_2(V_2,E_2),\cdots,G_n(V_n,E_n)$,这 n 个社交网络的所有节点表示为 $V=\bigcup_{i=1}^{n}V_i$,给定一个整数 k,多社交网络上的影响力最大化旨在找到一个包含 k 个节点的集合 $S(|S|=k)$,使得对于任何包含 k 个节点的集合 $K\in V$,都有 $\sigma(S)\geq\sigma(K)$.这里称 S 为种子集合,S 中的每个节点为种子节点.

显然,多网络上的影响力最大化问题仍然满足单调性和子模性(submodular),即使用贪心算法可以找到一个近似比为 $\left(1-\frac{1}{e}\right)$ 的近似最优解.Kempe 等人[4]已经证明了影响力最大化问题为 NP 难问题,同时,准确评估影响范围也被证明为 #P 难问题[6].

贪心算法面临的最大挑战是伸缩性问题,随着网络规模的增加,使得伸缩性问题更加突出.在真实数据集上的实验显示,贪心算法在百万规模的数据集上需要数天时间才能选出种子集合,计算时间较长.下面我们先介绍多社交网络上的影响计算模型,然后扩展基于树的算法模型[6]给出有效的解决措施,从而解决多社交网络下数据规模扩展带来的伸缩性挑战.

3　影响计算模型和基于树的算法模型

3.1　节点间的近似影响计算模型

在多社交网络环境下,考虑实体间的自传播特性,节点对之间的传播路径不仅有多条,还会跨越网络.下面以图 2 为例说明在多网络下,给出本文所提的多网络下的影响计算模型.

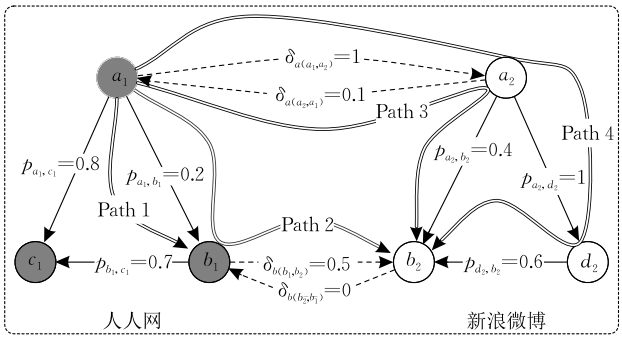

图 2　多网络下的影响计算模型示意图

图 2 所示的示意图包含两个网络:人人网和新浪微博.网络之间的虚线如同图 1 一样,用来表示实体间存在自传播性.其中灰色的节点属于人人网,白色的节点属于新浪微博.图 2 所示的示意图中,具有相同名字,不同下标(下标表示网络 ID)的节点指代同一个实体,例如 a_1 和 a_2 都指代实体 a.这里以节点 a_1 对实体 b 为例来说明如何计算一个节点对实体的影响强度.不考虑自传播性,从节点 a_1 到实体 b 只有一条路径:Path1($a_1\rightarrow b_1$);如果考虑实体 a 的自传播性($a_1\rightarrow a_2$),经由节点 a_1 到实体 b 还有:Path2($a_1\rightarrow b_1\rightarrow b_2$)、Path3($a_1\rightarrow a_2\rightarrow b_2$)和 Path4($a_1\rightarrow a_2\rightarrow d_2\rightarrow b_2$) 3 条传播路径.这里需要注意路径 Path2,可以看出对节点 b_2 的影响是在 b_1 的基础上产生的,而 b_1 和 b_2 指代相同的实体,在计算过程中该条路径不被计算在内.如此,节点 a_1 对实体 b 的影响路径有 Path1、Path3 和 Path4.

路径 $P=(u=n_1,n_2,\cdots,v=n_m)$ 表示经由节点 u 到节点 v.在独立级联模型下,点 u 沿着路径 P 对点 v 所产生的影响传播强度记为 $pp(P)$,其中 $pp(P)$ 使用式(1)计算:

$$pp(P)=\prod_{i=1}^{m-1}p(n_i,n_{i+1}) \tag{1}$$

由于节点对之间的可达路径有很多条,在网络规模较大的情况下更是如此,逐一计算所需的计算量十分大.本文使用节点间具有最大传播概率的路径来近似衡量节点对之间的传播概率以减少计算规模,这里使用 $MPP(u(e_u,G_i),v(e_v,G_j))$ 来表示由点 $u(e_u,G_i)$ 到点 $v(e_v,G_j)$ 的所有路径中具有最大传播概率的路径,并以该最大传播路径上的值近似作为他们之间的传播概率.则节点 $u(e_u,G_i)$ 到节点 $v(e_v,G_j)$ 的传播概率可近似表示如式(2):

$$pp(u(e_u,G_i),v(e_v,G_j))\approx pp(MPP(u(e_u,G_i),v(e_v,G_j))) \tag{2}$$

在多网络情况下,需要衡量节点对实体的影响强度,这里假设在 n 个网络上有 $m(m\leq n)$ 个表面名

称指代实体 e_v(即实体 e_v存在 m 个网络上).这里需要注意在使用最大传播路径近似节点间的影响力时,如果目标节点与它的前驱指代相同实体,则该路径不被计算在内.如此由节点 $u(e_u,G_i)$到实体 e_v使用最大传播路径近似之后,可以得到一个以节点 $u(e_u,G_i)$为根节点,叶节点为指代实体 e_v的多叉树.节点 $u(e_u,G_i)$对实体 e_v的影响可使用式(3)计算,其中 G_{jm}表示第 m 个指代实体 e_v的节点所在的网络 ID.

$$pp(u(e_u,G_i),e_v)=1-\prod_{i=1}^{m}(1-pp(u(e_u,G_i),v(e_v,G_{jm}))) \quad (3)$$

示例. 以图 2 为例解释如何使用近似的影响计算模型来计算节点 a_1对实体 b 的影响传播强度.由图可知,实体 b 存在人人网和新浪微博,分别表现为 b_1和 b_2,经由 a_1到节点 b_1只有一条路径 Path1($a_1\rightarrow b_1$),相应的,该路径也是具有最大传播的路径,$pp(a_1,b_1)=1-(1-pp(a_1\rightarrow b_1))=0.2$;从节点 a_1到节点 b_2有 3 条路径,由于路径 Path2($a_1\rightarrow b_1\rightarrow b_2$)中节点 b_2的前驱 b_1指代相同实体,该条路径不在计算之内.则剩下的两条路径的传播概率分别为 Path3:$pp(a_1\rightarrow a_2\rightarrow b_2)=1\times 0.4=0.4$,Path4:$pp(a_1\rightarrow a_2\rightarrow d_2\rightarrow b_2)=1\times 1\times 0.6=0.6$,可知具有最大传播概率的路径为 Path4$\langle a_1\rightarrow a_2\rightarrow d_2\rightarrow b_2\rangle$,则 $pp(a_1,b_2)=pp(a_1\rightarrow a_2\rightarrow d_2\rightarrow b_2)=0.6$,根据式(3)可知,节点 a_1对实体 b 的近似传播概率为 $1-(1-0.2)\times(1-0.6)=0.67$.

3.2 基于树的算法模型

根据式(3),可以计算出每个节点的影响传播范围,即将该节点对每个实体的影响范围加权,具体计算使用式(4):

$$\sigma(u(e_u,G_i))=\sum_{e_v\in E}pp(u(e_u,G_i),e_v) \quad (4)$$

式(4)中,E 表示 n 个网络中的所有实体集合,根据式(4)对节点集合 V 中的每个节点计算它的影响传播范围,其中具有最大影响传播范围的节点为第 1 个种子节点 s_1.当选择第 2 个种子节点时,应是同第 1 个种子节点 s_1一起能够产生最大影响范围的节点,换句话说,第 2 个种子节点应该是在已有的种子集合基础之上能够产生最大影响范围增益的节点.假设第 1 个种子为 u,第 2 个候选种子为 v,节点 v 在节点 u 上的影响范围增益表示在加入节点 v 之后所增加的影响范围.节点 v 在当前种子集合 S 上的影响范围增益表示为 $gain(v|S)$,计算方式如式(5)所示.

$$gain(v|S)=\sigma(S\bigcup v)-\sigma(S) \quad (5)$$

式(5)中 $\sigma(S)$表示节点集合 S 所能产生的影响范围.显然,$\sigma(S)$并不等于 S 中的每个节点所能产生的影响范围加权,因为 S 中的节点到某一目的点 v 的最大化传播路径可能产生交叉,加权求解会造成所求值比真实值大的假象.为此,需要计算集合 S 对每个实体的影响概率,然后加权求解,具体计算如式(6).

$$\sigma(S)=\sum_{e_v\in E}pp(S,e_v) \quad (6)$$

式(6)中,$pp(S,e_v)$表示节点集合 S 对实体 e_v的影响传播概率.因此,可以使用式(3)进行计算,即 $pp(S,e_v)=1-\prod_{i=1}^{m}(1-pp(S,v(e_v,G_{jm})))$.其中 $pp(S,v(e_v,G_{jm}))$表示集合 S 对节点 $v(e_v,G_{jm})$的影响强度,同样的,对于节点 $v(e_v,G_{jm})$的影响不能通过集合 S 中的每个点对 $v(e_v,G_{jm})$的影响加权计算,因为集合 S 中的点到 $v(e_v,G_{jm})$的最大传播路径会有重叠.为此,我们扩展基于树的算法模型[6]来解决该计算问题.

基于树的算法模型,将节点 v 视为树根,所有能够抵达点 v 的起始节点作为该树的叶子节点,通过叶子节点沿着抵达点 v 的最大传播路径逆向构建一棵树.可知,所有能够影响到点 v 的节点一定在以点 v 为树根的最大逆向传播树里.如此,可以根据式(7)计算 $pp(S,v(e_v,G_{jm}))$的值:

$$pp(S,v)=\begin{cases}1, & v\in S\\ 1-\prod_{c\in C_v}(1-pp(S,c)\cdot pp(c,v)), & v\notin S\end{cases} \quad (7)$$

由于空间有限,式(7)中使用 v 来简单表示 $v(e_v,G_{jm})$,其中 C_v表示以点 $v(e_v,G_{jm})$构成的最大逆向传播树中点 $v(e_v,G_{jm})$的儿子节点集合.这里需要注意所有的传播只在以点 $v(e_v,G_{jm})$构成的最大逆向传播树里进行.对于每个节点 $u(e_u,G_i)$,本文使用 $O(u(e_u,G_i))$和 $I(u(e_u,G_i))$分别表示所有被点 $u(e_u,G_i)$影响的节点集合和所有能够影响点 $u(e_u,G_i)$的节点集合.因此可知,在计算最大传播路径的时候可以获得 $O(u(e_u,G_i))$,而在构建最大逆向传播树的时候可以获得 $I(u(e_u,G_i))$,其中 O 和 I 表示方向,O 表示外向邻居(Outgoing),I 表示内向邻居(Ingoing).

4 基于上界的算法优化

由前文讨论可知,当选择下一个种子节点时,在

已有的种子集合的基础上能够产生最大影响范围增益的节点即为下一个种子节点. 当 $O(u(e_u,G_i))\cap O(v(e_v,G_j))=\varnothing$ 时表示点 $u(e_u,G_i)$ 和点 $v(e_v,G_j)$ 没有共同影响的节点，否则需要重新计算被他们共同影响的节点的激活概率. 多网络上的影响范围增益是针对所能影响到的实体的增益. 需要先计算每个节点的影响概率，然后针对每个实体计算影响增益，因此针对每一个候选种子都需要计算每个节点和每个实体在当前种子集合下的影响概率. 为了避免计算冗余，本文使用 EM_S（EM 表示 entity map）和 NM_S（NM 表示 node map）分别缓存种子节点对每个实体和每个节点的影响概率. 算法 1 为候选种子 u 在现有种子集合 S 上的影响增益计算过程（其中 u 和 v 为形如 $u(e_u,G_i)$ 的简写形式）.

算法 1.　$gain(u,S,EM_S,NM_S)$.

输入：候选种子节点 u，种子集合 S，EM_S，NM_S

输出：候选种子节点 u 的影响增益 $gain(u|S)$

初始化 $EM_{S\cup u}=EM_S$；

初始化 $gain(u|S)=0.0$；

FOR(EACH v IN $O(u)$){

IF(v NOTIN NM_S.keyset()){

IF(v.entity NOTIN EM_S.keyset()){

$EM_{S\cup u}[v.\text{entity}]=pp(u,v)$；

}

ELSE{

$EM_{S\cup u}[v.\text{entity}]=1-(1-EM_S[v.\text{entity}])\times(1-pp(u,v))$；

}

}

ELSE{

$npp(u,v)=pp(S\cup u,v)$；

$nval=1-(1-EM_S[v.\text{entity}])\times(1-npp(u,v))/(1-NM_S[v])$；

$EM_{S\cup u}[v.\text{entity}]=nval$；

}

}

FOR(e IN $EM_{S\cup u}$.keyset()){

IF(e IN EM_S) THEN{

$gain(u|S)+=EM_{S\cup u}[e]-EM_S[e]$；

}

ELSE{

$gain(u|S)+=EM_{S\cup u}[e]$；

}

}

RETURN $gain(u|S)$；

算法 1 描述了候选种子节点 u 在已有种子集合 S 上影响增益的计算过程. 具体包含两个部分，首先需要计算集合 $(S\cup u)$ 对每个实体的影响概率，然后加权每个实体的增益值作为最后 u 在集合 S 上的增益值. 根据算法 1 可知，每一轮选择具有最大影响范围增益的节点，直至选择 k 个种子为止. 经过我们的实验发现，算法 1 所耗时间比较长，因为它需要准确计算每个节点的影响范围，然后需要计算每个实体的增益值才能求解. 如果节点 v 被集合 S 和 u 共同影响，需要根据式(7)求解节点的影响范围 $npp(u,v)$ 时. 如果节点 u 和 s 分别独立的影响节点 v，则 u 和 s 对 v 的共同影响概率为 $1-pp(u,v)\times pp(s,v)$，否则需要根据式(7)计算. 由此可知，影响独立假设下的影响概率 $pp(u\cup s,v)$ 要大于根据式(7)所计算值，因为式(7)减去了由 $u\cup s$ 共同影响的那部分值. 如果假设集合 S 中的每个点对节点 $v(e_v,G_i)$ 都独立影响，则集合 S 对 $v(e_v,G_i)$ 的影响概率 $pp(S,v(e_v,G_i))$ 使用式(8)计算.

$$\hat{pp}(S,v)=\begin{cases}1, & v\in S\\ 1-\prod\limits_{u\in S}(1-pp(u,v)), & v\notin S\end{cases}\tag{8}$$

因为对于实体的影响概率是根据其多个被指代的节点计算，根据式(8)可以得出集合 S 对实体 e_v 的影响概率计算如式(9).

$$\hat{pp}(S,e_v)=1-\prod_{v.\text{entity}=e_v}(1-pp(S,v))\tag{9}$$

假设实体 e_v 存在 n 个表面形式，记节点 u 在集合 S 的基础上对实体 e_v 的影响增益为 $\hat{g}ain(u|S,e_v)$，则 $\hat{g}ain(u|S,e_v)=\hat{pp}(S\cup u,e_v)-pp(S,e_v)$. 根据式(9)，对增益值 $\hat{g}ain(u|S,e_v)$ 有如下推导：

$$\begin{aligned}&\hat{g}ain(u|S,e_v)=\hat{pp}(S\cup u,e_v)-pp(S,e_v)\\&=1-\prod_{i=1}^{n}(1-pp(S,v_i))\times(1-pp(u,v_i))-pp(S,e_v)\\&=1-\prod_{i=1}^{n}(1-pp(S,v_i))\times\prod_{i=1}^{n}(1-pp(u,v_i))-pp(S,e_v)\\&=1-(1-pp(S,e_v))\times\prod_{i=1}^{n}(1-pp(u,v_i))-pp(S,e_v)\\&=(1-pp(S,e_v))\times\Big(1-\prod_{i=1}^{n}(1-pp(u,v_i))\Big)\\&=(1-pp(S,e_v))\times pp(u,e_v).\end{aligned}$$

由此，得出节点 u 在集合 S 上的影响增益上界 $\hat{g}ain(u|S)$ 为式(10)：

$$\hat{g}ain(u|S)=\sum_{e_v\in E}\hat{g}ain(u|S,e_v)\tag{10}$$

对节点集合 V 中的每个节点 u，我们都预先计算它对每个实体的影响概率并缓存在 map 里，记为 EM_u，那么在已有种子集合 S 的基础上，节点 u 的影响增益上界可归纳为算法 2.

算法 2. Uppergain(u,EM_u,EM_S).

输入:候选种子 u,EM_u 和 EM_S

输出:候选种子 u 的影响增益上界 $\hat{g}ain(u|S)$

初始化 $\hat{g}ain(u|S)=0.0$;

FOR (e_v IN EM_u.keyset()){

IF(e_v IN EMS.keyset()){

$\hat{g}ain(u|S)+=\hat{g}ain(u|S,e_v)$;

}

ELSE{

$\hat{g}ain(u|S)+=EM_u[e_v]$}

}

RETURN $\hat{g}ain(u|S)$;

显然,如果一个节点的影响增益均大于其他节点的影响增益上界,那么该节点就是下一个种子节点,而无需重新计算其他节点的影响增益.本文通过借助大顶堆 H 来维护每个节点的相关信息,同时辅助种子选取.大顶堆中的每个元素包含 3 个信息:节点信息(称为 node)、节点的影响范围(称为 spread)和该节点影响范围计算时的状态值(称为 status).初始阶段,堆节点中的影响范围 spread 为每个节点的初始影响范围(使用式(4)计算),状态 status 初始化为 0.算法 3(BlendedIMMS)描述了多网络下寻找 top-k 个影响力范围最大的种子节点的计算过程.

算法 3. BlendedIMMS.

输入:n 个社交网络 $G_1(V_1,E_1),\cdots,G_n(V_n,E_n)$;种子个数 k;影响增益阈值 φ;最大传播路径阈值 θ

输出:大小为 k 的种子集合 S;

对集合V中的每个节点u预计算$O(u)$,$I(u)$,$\sigma(u)$和EM_u;

建立并初始化大顶堆 H;

初始化 $S=\varnothing$,$EM_S=\varnothing$,$NM_S=\varnothing$,$UM=\varnothing$;

FOR($i=0$;$i<k$;$i++$){

$UM=\varnothing$;

Node $tnode=H$.top();

WHILE($tnode$.status!=i){

$gain=0.0$;

IF($tnode$.node NOTIN UM){

$gain=$Uppergain($tnode$.node,EM_u,EMS);

$UM[tnode.node]=gain$;

SET $tnode$.spread=$gain$;

}

ELSE{

$gain=$gain($tnode$.node,S,EM_S,NM_S);

IF($UM[tnode.node]>0$ && $(gain/UM[tnode.node]>\varphi)$){

***tnode*.spread=*gain*;**

BREAK;

}

$tnode$.spread=$gain$;

$tnode$.status=i;

}

H.sort();

$tnode=H$.top();

}

$tnode=H$.pop();

$S=S\cup tnode$.node;

UPDATE EM_S 和 NM_S;

H.sort();

}

RETURN S;

算法 3 中初始输入为 n 个社交网络和 k 值.期望得到大小为 k 的种子集合 S 使得这 k 个种子节点的影响传播范围最大化.刚开始,算法需要预先计算每个节点 u 的 $O(u)$、$I(u)$、EM_u 和初始影响范围 $\sigma(u)$.并根据 $\sigma(u)$ 建立并初始化大顶堆 H.之后,初始化种子集合 S、map EM_S、map NM_S 为空集,同时使用 map UM 缓存每轮候选种子节点的影响增益上界值.整个过程总共 k 轮,使用 i 记录当前轮次.每轮先初始化 UM 为空集,之后查看堆顶元素 $tnode$,如果堆顶元素 $tnode$ 的状态值为 i,则弹出堆顶元素并添加种子集合 S 中.因为 i 值用来表征当前堆节点中的影响范围增益值是在哪一轮计算的,如果是在当前轮计算,并且经过堆排序之后仍然处于堆顶,则说明该节点为能够产生最大影响范围增益的点.否则,先判断当前节点是否在 map UM 里:如果不在,使用算法 2 计算堆顶节点 $tnode$ 的影响范围增益上界,分别缓存在 UM,同时设置堆顶元素 $tnode$ 的 spread 为增益上界值;如果 $tnode$ 在 map UM 里,则使用算法 1 计算它准确的影响增益值,之后设置堆顶节点的 spread 值为准确增益值,并设置它的状态 status 为 i 值.最后堆排序并获得堆顶元素,重新开始判断.整个过程持续直到选取了 k 个种子.

影响增益上界比准确的影响增益要高出一些,为此可以通过设置阈值,即如果通过算法 1 获得的准确影响增益与使用算法 2 计算出的影响增益上界值的比值在一定阈值内,就认定当前候选节点为下一个种子节点,本文使用变量 φ 表示比值阈值.为了节省空间,具体的代码为算法 3 中添加下划线和加粗部分的内容.每次计算过当前节点的准确增益值之后,如果它的增益上界值大于 0,并且准确增益值同增益上界比值大于阈值 φ,就终止 WHILE 循环,弹出当前节点作为下一个种子节点.

由前文推导知,影响增益上界值要比真实的增

益值大，通过使用推导出的公式可以比较方便的计算出增益上界值，对精度要求不高时，可以避免大量运算. 为此我们考虑直接使用增益上界值作为评估种子节点增益的方法. 并通过实验判断与准确计算下的增益值相比相差多少. 算法 4 为完全使用影响增益上界值的计算过程.

算法 4.　BoundBasedIMMS.

输入：n 个社交网络 $G_1(V_1,E_1),\cdots,G_n(V_n,E_n)$；种子个数 k；

输出：大小为 k 的种子集合 S；

对集合V中的每个节点 u 预计算$O(u)$，$I(u)$，$\sigma(u)$和EM_u；

建立并初始化大顶堆 H；

初始化 $S=\varnothing$，$EM_S=\varnothing$；

```
FOR(i=0; i<k; i++){
  Node tnode=H.top();
    WHILE (tnode.status!=i){
      gain=0.0;
      gain=Uppergain(tnode.node,EM_u,EMS);
      gain=gain(tnode.node,S,EM_S,NM_S);
      tnode.spread=gain;
      tnode.status=i;
      H.sort();
      tnode=H.top();
    }
  tnode=H.pop();
  S=S∪tnode.node;
  UPDATE EM_S;
  H.sort();
}
RETURN S;
```

算法 4 为完全使用影响上界的方式选择 top-k 个种子的过程. 每次使用算法 2 计算候选节点的影响增益上界，并更新堆节点的 status 为 i，spread 值为增益上界值，直到堆顶节点的 status 为 i 时弹出该节点并添加到种子集合 S 中，最后经过 k 轮迭代，返回种子集合 S.

5　实　验

本文分别实现了使用影响增益上界优化的算法 BlendedIMMS 以及使用完全影响增益上界的算法 BoundBasedIMMS 两种方法，并实现了使用准确增益的算法（这里表示为 IMMS，即完全使用影响增益来评估候选种子，具体将算法 4 中的 $gain=$ Uppergain($tnode$.node，EM_u，EMS) 替换为 $gain=$ gain($tnode$.node，S，EM_S，NM_S），由于篇幅有限不再赘述）. 最后本文分别在 DBLP、Citeseer、Aminer 和 Linkedin 等数据集上进行实验，将所提算法同当前较为先进的两种算法 PMIA[6] 和 IRIE[15] 进行比较，分别从运行时间和影响范围两个方面评估算法的性能差异. 由于 PMIA 和 IRIE 适用于单网络的影响力最大化问题，为此，针对输入的多网络通过预处理将其整合成一个网络，指代相同实体的节点重新编号，节点间的关系同多网络相同. 最后根据不同算法所选取的种子集合在原始的多网络上通过蒙特卡洛模拟计算平均的影响范围.

5.1　数据集

多网络上的影响力分析需要考虑不同网络上节点间的实体链接与识别问题，虽然已有研究成果去解决多网络上节点的实体链接，但是考虑到准确度问题，我们使用实体能够一一对应的数据集 DBLP① 和 Citeseer② 以及 Linkedin③[29] 和 Aminer④[29].

DBLP 和 Citeseer 为科研作者网络，网络上的节点都是实名制，本文将具有相同名字的节点指代同一个实体，不同网络上的节点信息使用节点名称与网络 ID 进行标识，Linkedin 和 Aminer 为文章[29] 所用，我们使用作者提供的真实节点的实体映射作为两个网络上的实体映射集合.

对于 DBLP 和 Citeseer 数据集，同一个网络内节点间的影响概率使用两个作者之间的合作关系计算得到，使用式(11)计算：

$$p(a\to b)=\frac{N(a,b)}{N(a)} \tag{11}$$

式(12)中 $p(a\to b)$表示节点 a 对节点 b 的影响概率，其中 $N(a,b)$表示作者 a 和 b 的合作次数，$N(a)$表示作者 a 总共的著作次数. 显然，如果两个作者之间的合作次数越多，那么他们之间相互的影响概率就越强. 对于 Aminer 和 Linkedin，我们使用常用的 $1/d(v)$作为节点同其他关联节点之间的影响强度，其中 $d(v)$为节点 v 的入度.

多个网络上实体间的自传播概率可以根据每个实体在多个网络上的行为计算得到，例如用户 A 在新浪微博上发布了 100 条微博，其中有 10 条都转发到了人人网上，则 $\delta_{A(\text{Weibo},\text{Renren})}=0.1$. 由于 DBLP 和 Citeseer、Aminer 和 Linkedin 捕捉同一实体的自传播概率比较困难，这里使用随机生成数作为实体上的自传播概率. 首先，需要找到两个网络上的共同节点作为能够产生自传播概率的实体，然后针对每个

① http://en.wikipedia.org/wiki/DBLP
② http://en.wikipedia.org/wiki/CiteSeer
③ https://www.linkedin.com/
④ http://cs.aminer.org/network_integration#b2855

实体生成[0-1]的实数作为自传播概率,由于所有的数据均提前生成并固定,每个算法使用相同的数据信息,对实验结果不会造成影响.

DBLP 和 Citeseer、Aminer 和 Linkedin 均为有向网络,具体的数据集统计信息如表 1 所示.

表 1 DBLP 和 Citeseer 数据集概要信息

数据集	点数	边数	平均出度	节点总数	共同点数
DBLP	1436596	12311706	8.57	1574809	73131
Citeseer	138368	754316	5.45		
Aminer	1056941	7859752	7.44	7783231	3041
Linkedin	6726290	38721380	5.76		

5.2 实验效果

本文分别在 DBLP 和 Citeseer、Aminer 和 Linkedin 这 4 个数据集所构成的两组网络上分别比较了 IMMS、BlendedIMMS、BoundBasedIMMS 和 PMIA、IRIE、SPM(由 CELF 算法改进)和 SP1M(由 CELF 算法改进)这 5 种算法的实验效果,其中 PMIA、IRIE、SPM 和 SP1M 由 Chen 等人[6,15]在文章中所用源代码,由于 SPM、SP1M 这两种贪心算法所耗时间过长,在 DBLP、Citeseer 两个网络上寻找 20 个节点耗时约 53 h,而在影响范围上同 IRIE、PMIA 和本文所提算法相差无几,由于传统贪心算法耗时过长,实验部分只对 IRIE 和 PMIA 进行对比.

对比实验从运行时间和影响范围(Influence Spread)两个方面衡量方法的性能.对于已选取的种子集合,在独立级联模型基础上使用的蒙特卡洛模拟来评估节点集合的影响范围,多网络下指代同一个实体的节点的激活只能算作一次,但可以多次被激活,每个种子集合使用 20000 次蒙特卡洛模拟传播过程,最后使用均值作为每个种子集合的影响范围.下面分别从影响范围和运行时间对比不同算法在多网络下的影响力最大化的性能表现.

5.2.1 影响范围(Influence Spread)

实验分别比较了在固定种子数量的情况下,最大传播路径的阈值 θ 分别为 0.01、0.02、0.03、0.04 和 0.05 时 IMMS、BlendedIMMS、BoundBasedIMMS 和 PMIA 这 4 种方法的影响范围.图 3 和图 4 为在 DBLP 和 Citeseer 数据集上分别固定种子数目为 50 和 100 的情况下,4 种方法在不同阈值下的影响传播范围,图 5 为在 Aminer 和 Linkedin 数据集上种子个数为 100 时不同阈值范围下 4 种算法的结果.

从图 3、图 4 和图 5 看出,随着阈值 θ 增大,上述几种方法的影响范围均呈现下降趋势,这是因为 θ 为最大传播路径的阈值,用来控制每个节点的局部传播范围.最大传播路径上的概率值小于 θ,该路径就会被过滤掉,使得计算结果近似于真实结果,但也小于真实的影响范围.同时也总结出,在相同阈值下,随着种子数量的增加,影响范围也会增加.并且 4 种方法都能够达到一定的传播范围,从整体来看本文所提算法 IMMS、BlendedIMMS 和 BoundBasedIMMS 大体相当,并都要略优于 PMIA.这是

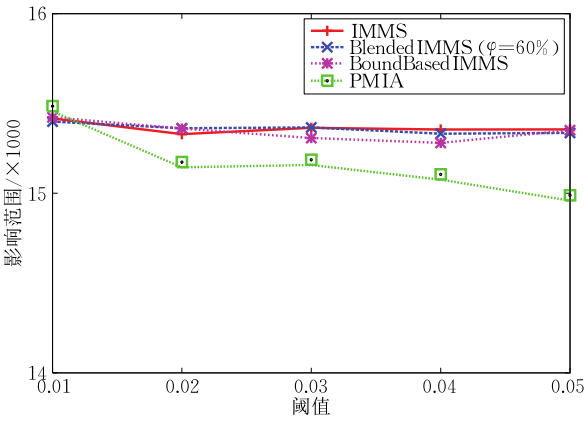

图 3 DBLP&Citeseer 数据集上的影响范围结果
($k=50$ 时,影响范围与阈值 θ 的关系)

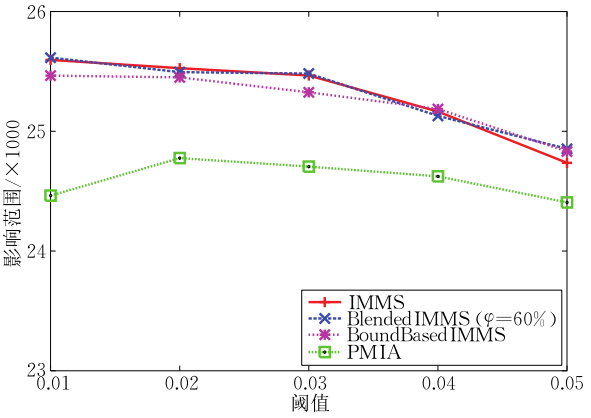

图 4 DBLP&Citeseer 数据集上的影响范围结果
($k=100$ 时,影响范围与阈值 θ 的关系)

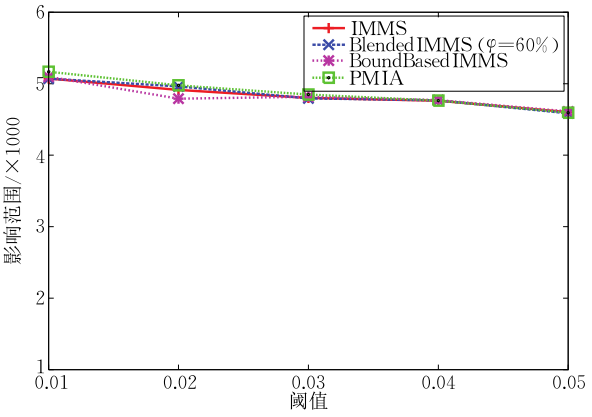

图 5 Aminer&Linkedin 数据集上的影响范围结果
($k=100$ 时,影响范围与阈值 θ 的关系)

因为 IMMS、BlendedIMMS 和 BoundBasedIMMS 在计算影响范围增益的时候按照实体的增益计算，这也是多网络同单一网络的不同之处，一个实体可能存在于多个网络，增益的计算需要按照实体而非节点进行计算.

由于 IRIE 同 θ 不相关，图 6 和图 7 比较了在最大传播路径阈值 θ 为 0.01（对 PMIA、IMMS、BlendedIMMS 和 BoundBasedIMMS 这 4 种方法而言）时，种子数目 k 分别为 20、40、60、80 和 100 时 5 种算法的影响范围. 其中图 6 为 DBLP 和 Citeseer 两个网络上的结果，图 7 为 Aminer 和 Linkedin 两个网络上的比较结果. 可以看出，随着种子数目 k 的增加，5 种方法的影响范围都会增长，并且差异不大，由此得知，本文所提方法在结果上能够同 PMIA、IRIE 等方法取得一致性.

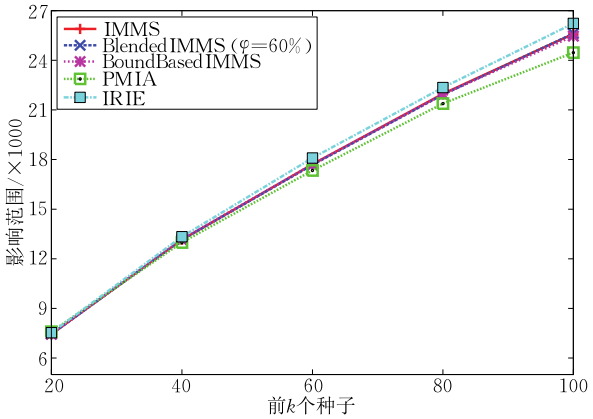

图 6　DBLP&Citeseer 数据集上的影响范围结果（固定 $\theta=0.01$，种子数目 k 同影响范围的关系）

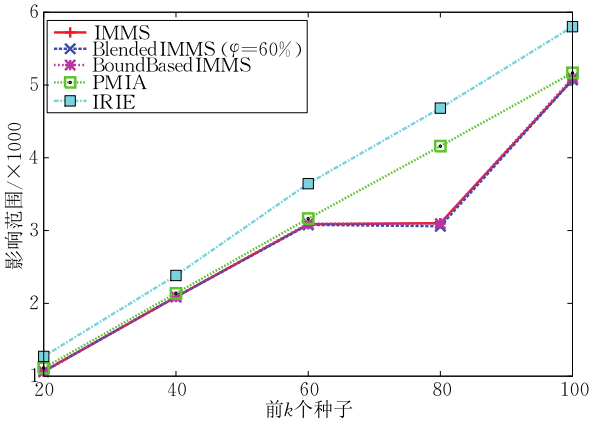

图 7　Aminer&Linkedin 数据集上的影响范围结果（固定 $\theta=0.01$，种子数目 k 同影响范围的关系）

5.2.2　运行时间(Running Time)

此外，实验分别比较了在固定种子数目 k 时，不同阈值 θ 对时间的影响（针对 PMIA、IMMS、BlendedIMMS 和 BoundBasedIMMS 这 4 种方法）.

图 8 和图 9 为在数据集 DBLP 和 Citeseer 上，在固定种子数目 k 为 50 和 100 时，不同阈值 θ 下 4 种方法的运行时间；图 10 为在数据集 Aminer 和 Linkedin 数据集上，种子数目 k 为 100 时，阈值范围在 0.01～0.05 上 4 种不同算法的影响范围结果.

由图 8、图 9 和图 10 看出，随着阈值 θ 的增加，

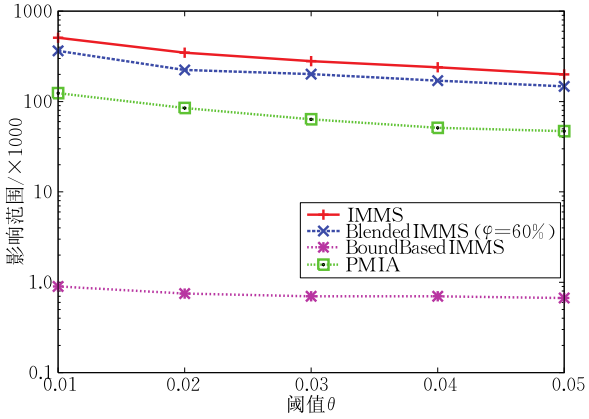

图 8　DBLP&Citeseer 数据集上运行时间结果（$k=50$ 时，运行时间与阈值 θ 的关系）

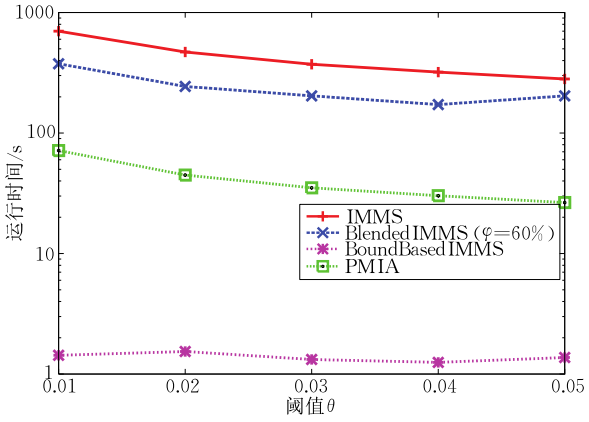

图 9　DBLP&Citeseer 数据集上运行时间结果（$k=100$ 时，运行时间与阈值 θ 的关系）

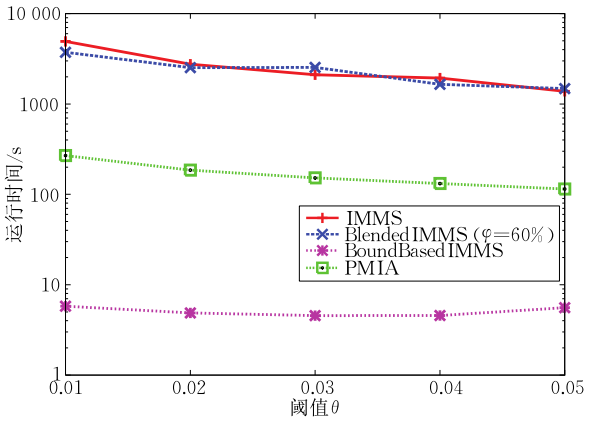

图 10　Aminer&Linkedin 数据集上运行时间结果（$k=100$ 时，运行时间与阈值 θ 的关系）

运行时间呈下降趋势，这是因为随着阈值 θ 的增大过滤了一些不重要的节点，减少了候选种子的规模；此外，运行时间最长的为使用准确影响增益计算的IMMS，其次为使用 BlendedIMMS 的方法，之后为PMIA，运行时间最短的为完全基于影响上界的BoundBasedIMMS，运行时间很稳定并且一直维持在 1 秒左右，比 PMIA 高出 70 倍的数量级，比IMMS 和 BlendedIMMS 快 2 个数量级.

图 11 和图 12 显示了在固定阈值 θ 为 0.01 时，不同种子数目 k 下，5 种算法的运行时间结果比较. 其中图 11 为在数据集 DBLP 和 Citeseer 两个网络上的结果，图 12 为数据集 Aminer 和 Linkedin 两个网络上的对比结果. 从图 11、图 12 中可以看出，随着种子数量的增加，运行时间也会增长，出乎意料的是运行时间最长的反而是 IRIE 算法，PMIA 表现一直很稳定，这其中性能表现最优的为 BoundBasedIMMS，运行时间一直维持在秒级，表现了极强的性能. 这是因为 BoundBasedIMMS 算法使用节点间完全独立的假设进行近似计算，免去了诸多中间计算过程.

从运行时间和影响范围的实验结果来看，无论是运行时间还是影响范围，完全基于影响增益上界的方法 BoundBasedIMMS 均有着较高的性能，虽然使用影响独立假设会损失部分计算精度，但是相比之下运行效率提高了近 2 个数量级. 基于准确计算的IMMS 和使用 Bound 作为过滤的 BlendedIMMS 运行时间上比较高，但却有着最好的影响范围.

图 3 至图 12 中的 BlendedIMMS 我们设置的阈值 φ 为 60%. 为了验证阈值 φ 对 BlendedIMMS 方法的影响，实验还比较了不同阈值 φ 下，5 种方法在运行时间和影响范围的对照结果. 图 13 为固定种子数目 k 为 50，阈值 θ 为 0.01 时，阈值 φ 分别为 20%、40%、60% 和 100% 时分别对影响范围的影响. 图 14 为在上述配置下对运行时间的影响. 从图 13 和图 14 看出，阈值 φ 对影响范围和运行时间的影响不大，这其中主要因为网络是稀疏的. 但总体来看完全基于影响上界的方法 BoundBasedIMMS 表现出了较高的性能，从运行时间和影响范围来看

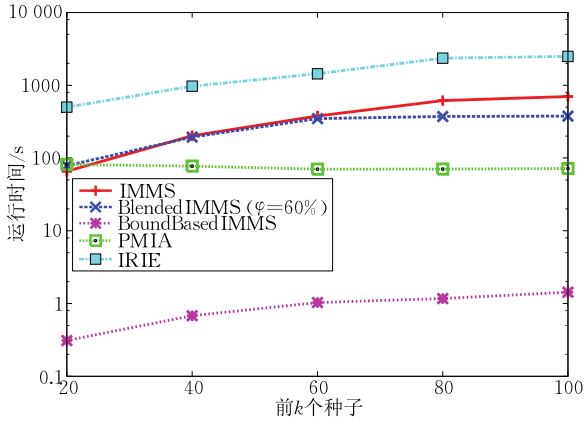

图 11 DBLP&Citeseer 数据集上运行时间结果
（θ=0.01 时，运行时间与种子数目 k 的关系）

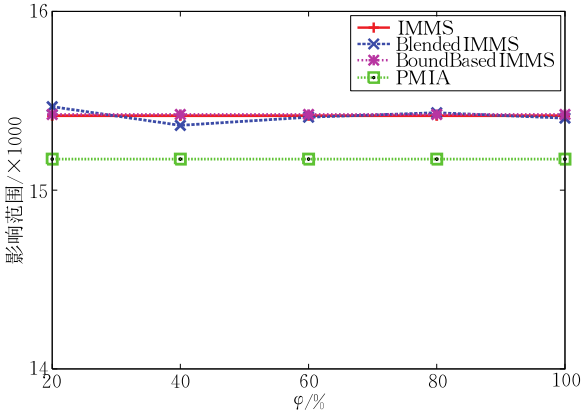

图 13 DBLP&Citeseer 数据集上影响范围结果
（k=50，θ=0.01 时，影响范围与阈值 φ 的关系）

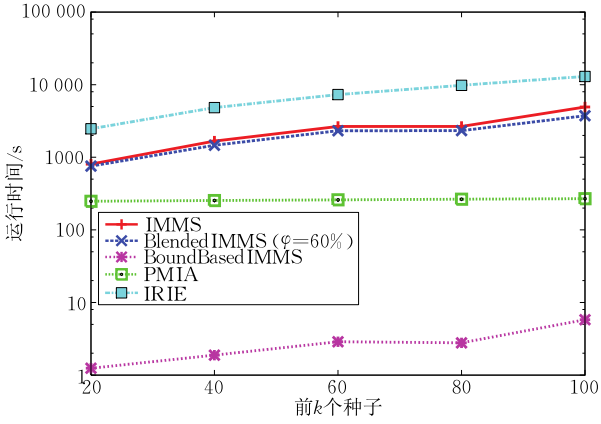

图 12 Aminer&Linkedin 数据集上运行时间结果
（θ=0.01 时，运行时间与种子数目 k 的关系）

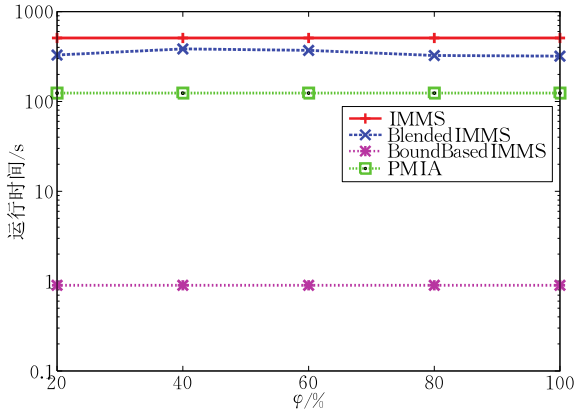

图 14 DBLP&Citeseer 数据集上运行时间结果
（k=50，θ=0.01 时，运行时间与阈值 φ 的关系）

都优于其他方法.而当前最为先进的方法 PMIA 从影响范围上来看比不过 IMMS 和 BlendedIMMS，而从运行时间上又高出 BoundBasedIMMS 方法.

6　结束语

本文研究了多社交网络上的影响力最大化问题.首先与单一网络相比，多社交网络具有自传播性，通过该特性可以将多个网络建立联系；其次提出了针对多社交网络上节点对实体的影响计算模型来评估多网络下节点间的影响计算问题；并在独立级联影响模型下提出多种解决方案来解决多网络下的影响力最大化问题；最后通过真实数据集上进行实验，对比了本文所提方法与当前最先进的方法在影响范围和运行时间上的实验效果.最后的实验表明所提算法在影响范围和运行时间都能够达到满意的效果，并明显优于现有方法.

传统的贪心算法通过多次蒙特卡洛模拟来计算给定节点集合在给定网络上的影响范围，计算量比较大，本文通过扩展基于树的算法模型，并在此基础上通过进一步挖掘影响函数的子模性(submodular)(如算法 3 所示)来避免冗余计算，最后通过假设节点间的影响相互独立，使用节点的影响增益上界近似节点的增益值来提升计算性能，而最后的实验结果表明这种近似的方法并未明显减少所选种子节点的影响范围.传统贪心算法在大型网络上伸缩性较差，而实验所选数据集的节点多为百万规模的大型网络，根据实验效果来看，所提算法的伸缩性较好，运行时间并未随网络规模有明显异动.

参　考　文　献

[1] Misner I R. The Word's Best Known Marketing Secret: Building Your Business with Word-of-Mouth Marketing. San Jose, California, USA: Bard Press, 1999

[2] Domingos P, Richardson M. Mining the network value of customers//Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, USA, 2001: 57-66

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

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

[5] Kempe D, Kleinberg J, Tardos É. Influential nodes in a diffusion model for social networks//Caires L, Italiano G F, Monteiro L, et al, eds. Automata, Languages and Programming. Libson, Portugal, 2005: 1127-1138

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

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

[8] Chen W, Wang Y, Yang S. Efficient influence maximization in social networks//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France, 2009: 199-208

[9] Goyal A, Lu W, Lakshmanan L V. CELF++: Optimizing the greedy algorithm for influence maximization in social networks//Proceedings of the 20th International Conference Companion on World Wide Web. Hyderabad, India, 2011: 47-48

[10] Kimura M, Saito K. Approximate solutions for the influence maximization problem in a social network//Gabrys B, Howlett R J, Jain L C eds. Knowledge-Based Intelligent Information and Engineering Systems. Bournemouth, UK, 2006: 937-944

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

[12] Chen Y C, Peng W C, Lee S Y. Efficient algorithms for influence maximization in social networks. Knowledge and Information Systems, 2012, 33(3): 577-601

[13] Goyal A, Lu W, Lakshmanan L V. SIMPATH: An efficient algorithm for influence maximization under the linear threshold model//Proceedings of the 2011 IEEE 11th International Conference on Data Mining (ICDM). Vancouver, Canada, 2011: 211-220

[14] Jiang Q, Song G, Gao C, et al. Simulated annealing based influence maximization in social networks//Proceedings of the 25th AAAI Conference on Artificial Intelligence. California, USA, 2011: 127-132

[15] Jung K, Heo W, Chen W. IRIE: Scalable and robust influence maximization in social networks//Proceedings of the 2012 IEEE 12th International Conference on Data Mining. Brussels, Belgium, 2012: 918-923

[16] Chen Yi-Cheng, Peng Wen-Chih, Lee Suh-Yin. Efficient algorithms for influence maximization in social networks. Knowledge and Information Systems, 2012, 33(3): 577-601

[17] Kim J, Kim S K, Yu H. Scalable and parallelizable processing of influence maximization for large-scale social networks?//

Proceedings of the 29th International Conference on Data Engineering (ICDE). Brisbane, Australia, 2013: 266-277

[18] Tang Y, Xiao X, Shi Y. Influence maximization: Near-optimal time complexity meets practical efficiency//Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. Snowbird, USA, 2014: 75-86

[19] Cheng Suqi, Shen Huawei, Huang Junming, et al. IMRank: Influence maximization via finding self-consistent ranking//Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval. Gold Coast, Australia, 2014: 475-484

[20] Li G, Chen S, Feng J, et al. Efficient location-aware influence maximization//Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. Snowbird, USA, 2014: 87-98

[21] Tang S, Yuan J, Mao X, et al. Relationship classification in large scale online social networks and its impact on information propagation//Proceedings of the 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies. Shanghai, China, 2011: 2291-2299

[22] Barbieri N, Bonchi F, Manco G. Topic-aware social influence propagation models. Knowledge and Information Systems, 2013, 37(3): 555-584

[23] Chen S, Fan J, Li G, et al. Online topic-aware influence maximization. Proceedings of the VLDB Endowment, 2015, 8(6): 666-677

[24] Tang J, Sun J, Wang C, et al. Social influence analysis in large-scale networks//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France, 2009: 807-816

[25] Liu L, Tang J, Han J, et al. Mining topic-level influence in heterogeneous networks//Proceedings of the 19th ACM International Conference on Information and Knowledge Management. Toronto, Canada, 2010: 199-208

[26] Weng J, Lim E P, Jiang J, et al. TwitterRank: Finding topic-sensitive influential twitterers//Proceedings of the 3rd ACM International Conference on Web Search and Data Mining. New York, USA, 2010: 261-270

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

[28] Chen Duanbing, et al. Identifying influential nodes in complex networks. Physica A: Statistical Mechanics and Its Applications, 2012, 391(4): 1777-1787

[29] Zhang Yutao, et al. COSNET: Connecting heterogeneous social networks with local and global consistency//Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Sydney, Australia, 2015: 1485-1494

LI Guo-Liang, born in 1980, Ph.D., associate professor. His current research interests include database usability, data cleaning and integration, spatio-textual data processing, crowdsourcing and social influence.

CHU Ya-Ping, born in 1988, M. S. candidate. Her current research interest is social influence.

FENG Jian-Hua, born in 1967, Ph. D., professor. His current research interests include database, data warehouse, and web data management.

XU Yao-Qiang, born in 1977, Ph. D., engineer. His current research interest is big data analysis.

Background

This work is supported by the National Natural Science Foundation of China project "Location-Aware Influence Maximization" and the National Basic Research Program of China project "Crowd Computing in Big Data".

Influence Maximization has many applications in viral marketing, which is very important in current environment thanks to the development of Internet and social network. Influence maximization has been extensively studied by the industrial and academic communities. Existing related works mainly study the problem on single network, while ignored the importance and influence of multiple social networks. This paper aims at solving influence maximization problem on multiple social networks, which selects k nodes on the given multiple social networks to maximize the expected influence spread on the multiple social network. It is rather challengeable and necessary to solve it on multiple social networks.

We have extensively studied the influence maximization problem, and several works has been published. Such as the location-aware influence maximization and the topic-aware influence maximization.