CN 43-1258/TP
ISSN 1007-130X

计算机工程与科学
Computer Engineering & Science

第 38 卷第 6 期 2016 年 6 月
Vol. 38, No. 6, Jun. 2016

文章编号:1007-130X(2016)06-1128-07

基于影响路径的个性化影响最大化算法*

杨书新,王　希,彭秋英
(江西理工大学信息工程大学,江西 赣州 341000)

摘　要:个性化影响最大化问题是近年来社交网络影响最大化问题研究领域一个较新的分支,其现有解决方案普遍建立在网络边影响传播强度一致的假设下,该假设对于真实社交网络缺乏普遍适用性。为此基于独立级联模型,提出最大影响路径算法(MIPA)。该算法通过三个阶段来求解个性化影响最大化问题,首先将边影响强度作对数转换以获得最大影响路径,从而计算网络节点对目标节点的邻居节点的影响;然后利用多条经过目标节点邻居的最大影响路径联合计算目标节点受到的影响强度;最后选择 Top-k 节点作为种子节点,从而摆脱边影响强度的一致性约束,获取高质量的种子集。在不同的真实社交网络数据集上进行的对比实验验证了算法的有效性。

关键词:社交网络;个性化;影响最大化;特定用户
中图分类号:TP393　　**文献标志码**:A
doi:10.3969/j.issn.1007-130X.2016.06.010

A personalized influence maximization algorithm based on influence path

YANG Shu-xin, WANG Xi, PENG Qiu-ying
(School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China)

Abstract: Personalized influence maximization in social network has become a new branch of influence maximization study in recent years. Different from existing research that assumes equal propagating strengths of social network edges, our work aims to find out the top-k most influential nodes for the target user without inappropriate assumption. We propose a maximized-influence-path algorithm (MIPA) based on the independent cascade model. It solves the problem through three stages. Firstly, to compute the propagating strengths from the nodes of social network to the neighbors of the target node, the strengths of edges are transformed into its logarithmic form for getting the maximized influence paths. Secondly, the strength of maximized influence paths which pass through different neighbors with the same source nodes are consolidated to calculate the node's propagating strength on the target node. Finally, the seed set with high propagating strength is found out by selecting the top-k nodes. We testify the algorithm on several real-world social networks. Experimental results validate the proposed algorithm.

Key words: social network; personalization; influence maximization; target user

* **收稿日期**:2015-07-13;**修回日期**:2015-08-21
基金项目:国家自然科学基金(41362015);江西省科技厅青年科学基金(20122BAB211035);江西省教育厅科技项目(GJJ14431, GJJ14432, GJJ14458)
通信地址:341000 江西省赣州市江西理工大学信息工程学院
Address: School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, Jiangxi, P. R. China

1 引言

近年来,随着互联网应用技术的高速发展及移动终端的大量普及,社交网络服务 SNS(Social Network Services)在全世界范围内风靡流行,将人们在现实生活中形成的相对稳定的社交圈以一种更加直观、形象的形式体现出来。与此同时,由于互联网用户在线活动不受地域及时间约束,拥有共同兴趣爱好的两个人,即使生活在不同地区甚至不同国家,也能通过社交网络应用轻松快捷地交流。这些微妙的差异促使在线社交网络在信息传播结构上具有区别于现实生活社交圈的新特性,也使对社交网络进行进一步深入研究具有重要意义,而社交网络影响最大化 IM(Influence Maximization)问题更是因其潜在的巨大商业价值和坚实的数据基础成为企业商家及学术界研究的焦点。

2002 年,Domingos 和 Richardson[1] 提出利用马尔可夫随机场来计算社交网络影响传播概率的方法,将影响最大化问题的研究引入算法领域[2]。Kempe 等人[3] 证明这个最优化问题为 NP-hard,并给出一个贪心解决策略,在此基础上研究人员不断改进,提出了 CELF[4] 及其优化算法 CELF++[5]、MixGreedy[6] 等算法。考虑到贪心策略求解效率一直为人诟病,DegreeDiscountIC[6]、LDAG[7]、模拟退火算法[8] 等一系列结合启发策略的求解算法也被提出并用以解决影响最大化问题。在影响最大化的发展过程中逐渐延伸出一个分支——节点代价各异的影响最大化问题[4],对其求解基本沿用了传统影响最大化的解决方案[9,10]。分析现有影响最大化研究工作不难发现,随着学术界大量精力的投入,对传统影响最大化问题算法效率和求解质量的提升逐渐进入平台期,要想再寻求突破,必须有更加全面的视野和创新的思维。

2013 年,Guo 等人[11] 将视线回归市场营销,深入分析市场规律后发现:通过长期的经营活动,商家不但累积了营销经验,同时也发现了一些"重要用户",这些"重要用户"因其远高于一般用户的潜在商业价值而倍受重视。在推出新产品时,若无法直接联络"重要用户"、营销预算又不足以完全覆盖"重要用户"的所有朋友,如何找到在影响"重要用户"过程中起关键作用的用户,从而展开有效的个性化营销是商家亟待解决的问题,由此催生出影响最大化问题的一个新分支——个性化影响最大化。个性化影响最大化旨在以社交网络为背景,以特定网络用户为对象,挖掘使其受影响程度达到最大的初始影响传播用户集合。目前,Klout、PeerIndex 等社交数据分析公司正利用影响最大化算法为大量广告商提供营销参考,而随着个人价值在商业营销中的作用逐渐显现,寻求高效的个性化影响最大化解决方案无疑是对社交数据分析人员提出的新要求。为此,本文针对个性化影响最大化问题,基于独立级联模型提出最大影响路径算法 MIPA(Maximized-Influence-Path Algorithm),在不同数据集上进行的实验,分析并验证了所提出方法的有效性。

2 相关工作

现有个性化影响最大化问题的解决方案主要是针对线性阈值模型及独立级联模型这两种基础影响传播模型展开的。Guo 等人[11] 中提出了三种不同的算法:局部贪心算法 LGA(Local Greedy Algorithm)、高效局部贪心算法 ELGA(Efficient Local Greedy Algorithm)、局部级联算法 LCA(Local Cascade Algorithm)。这三种算法均是针对独立级联模型设计的,其中 LGA 算法基本沿用了传统影响最大化的思路,利用蒙特卡洛模拟预测节点影响,进行模拟的次数直接影响了算法执行效率;为达到提高求解效率的目的,ELGA 算法将蒙特卡洛模拟的对象由每条网络边改为整个网络图,大幅减少了蒙特卡洛模拟的次数,但仿真实验显示:仅处理节点数为 602、有 17 595 条有向边社交网络图时,该算法耗时已经达到了 10^6 s;LCA 算法摒弃了蒙特卡洛模拟,转而采用概率来估算影响强度,它通过最短路径建立节点与目标节点的局部级联社区(Local Cascade Community),将局部级联社区的联合影响强度作为对目标节点的影响强度。LCA 在运行时间上相比前述二者可谓表现出色,但它所计算的目标节点影响强度是建立在节点间边影响传播强度一致的假设下,当节点间边影响强度不同时,用节点间最短路径的影响强度来反映实际影响强度存在较大误差,因而算法所求得的初始用户集合无法达到最佳影响传播效果。

随后,郭静等人[12] 的研究填补了线性阈值模型下的个性化影响最大化的研究空白,他们提出的目标贪婪算法 TGA(Target-based Greedy Algorithm)将一般网络节点对目标节点的影响分为两段来计算:一段为一般网络节点到目标节点的邻居节点;另一段为目标节点的邻居到目标节点。这种

计算方法同样是基于节点间边影响传播强度一致的假设,经理论证明其具有较小方差保证,在求解精度上也比较接近传统贪心算法。令人略感惋惜的是相比文献[12]中提到的基于目标节点邻居影响的LND算法,目标贪婪算法虽然在目标节点影响强度上多0.15,付出的代价却是四个数量级的运行时间差。

张伯雷等人[13]的研究是面向多个目标节点的影响最大化,为避免目标节点被重复影响,他们先使用K-medoids聚类算法将社交网络节点进行聚类处理,然后不断更新聚类中心试图获得更广泛的信息覆盖。事实上,这种有多个目标节点的影响最大化问题介于传统网络全局影响最大化和个性化影响最大化之间,如果存在求解效率高的个性化影响最大化求解方法,可以通过设置虚拟节点连接所有目标节点的方法,策略性地将其转化为个性化影响最大化问题,从而获得问题最优解。

通过分析上述工作可以看出:个性化影响最大化问题作为算法领域一个较新的问题,现有的解决方案思路普遍局限于边影响强度一致的约束,求解算法效率也有一定的提升空间。为此,本文基于影响传播路径,提出最大影响路径算法克服边强度约束求解该问题。

3 预备知识

为便于理解后续算法,本节介绍个性化影响最大化的一些预备知识,内容包括影响传播模型、个性化影响最大化问题。

影响传播建模是研究社交网络影响扩散的基本问题之一。在社交网络影响传播过程中,对于一个确定的网络节点v,在某个固定时刻的状态为"未被影响"或"已被影响",只取其一。若v在某一时刻的状态转换为"已被影响",那么在其后的时间里,v将可能影响其"未被影响"的邻居节点。需要说明的是:在整个社交网络影响传播的过程中,节点状态只能由"未被影响"转移为"已被影响"。以上关于节点状态切换的约定对线性阈值模型和独立级联模型均成立。

不同于线性阈值模型的网络影响积累模式,社交影响在独立级联模型中是依概率在节点间传播的。若节点v在第i轮被影响,那么在第$i+1$轮,v将依概率将影响传播到其状态为"未被影响"的邻居节点,无论此次影响结果如何,在此后的影响传播过程中,v将不再有机会对其所有邻居节点造成影响。这样的影响扩散过程依次迭代于每一轮状态被切换为"已被影响"的节点上,直至不再有新的节点被影响。

现有对社交网络影响最大化问题的研究中,普遍采用有向图DAG(Directed Acyclic Graph)来表示一个社交网络$G(V,E)$,其中V和E分别表示用户集合及他们之间的关注关系的集合,n和m则分别表示集合V和E中元素的个数。社交网络分析人员对用户信息及其网络社交行为特征、相邻用户的同质性及影响网络事件传播的时间、空间等客观因素进行分析,将这些反映用户间影响传播特征的因素综合量化为节点间通过网络边e_i传播影响的强度p_{e_i}($0\leqslant p_{e_i}<1$)。p_{e_i}值越大,代表用户间关系越紧密,越容易产生影响;相反,p_{e_i}值越小,代表用户间虽建立关系却不常分享信息,彼此间影响微乎其微。

对于给定一个网络$G(V,E)$及其每条边对应的影响强度p_{e_i}、目标节点$v_t\in G$,$P_{v_t}(\cdot)$为对v_t影响强度函数,个性化影响最大化问题需要解决的是:从网络中挑选不超过k个节点的种子集S,使得目标节点v_t受S影响的强度达到最大。其形式化地描述如下(S_i为任意元素不超过k的节点集合):

$$S=\arg\max_{S_i\subseteq V\setminus v_t,\ |S_i|\leqslant k}P_{v_t}(S_i) \tag{1}$$

通过定义可知,个性化影响最大化问题是影响最大化问题的一个特殊形式,其影响传播的目的不再面向整个网络,试图尽可能多地影响网络节点,而是面向网络中特定的节点v_t,尽可能使v_t受到的影响强度最大。

4 基于影响路径的个性化社交网络影响最大化算法

在社交网络中,对于给定的一个种子集,计算其影响传播范围是NP-hard问题[3]。此外,由式(1)可知,对于个性化影响最大化求得的种子集S及任意节点数不超过k的集合S_i,均有$P_{v_t}(S_i)\leqslant P_{v_t}(S)$,因此,个性化影响最大化的重点在于计算节点对目标节点的影响强度。

对于网络中任意节点v_1、v_2,若网络图中存在一条路径$path(v_1,v_2)$,即:v_1的网络影响有可能通过用户间的信息传播过程蔓延至v_2,其路径影响强度记为$P_{path(v_1,v_2)}$。

$$P_{path(v_1,v_2)}=\prod_{e_i\in path(v_2,v_2)}p_{e_i} \tag{2}$$

社交网络中节点关系错综复杂,很可能存在 v_1 能够影响到 v_2 的路径不止一条,这些影响传播路径共同组成一个以 v_1 为源点、v_2 为汇点的网络 $N_{(v_1,v_2)}$,记 v_1 的网络影响通过 $N_{(v_1,v_2)}$ 能够到达 v_2 的强度为 $P_{v_2}(v_1)$。尽管 $N_{(v_1,v_2)}$ 缩小了计算 $P_{v_2}(v_1)$ 需遍历的范围,但遍历 $N_{(v_1,v_2)}$ 中包含的每一条影响路径来计算网络节点期望影响力仍为 NP-hard[6],因此本文提出以多条最大影响路径来联合估算 $P_{v_2}(v_1)$ 的算法。

定义 1(最大影响路径) 在社交影响传播网络中,若节点 v_1 到 v_2 存在的 l 条不同路径 $path(v_1,v_2)$,则指定其中一条使得路径影响强度 $P_{path(v_1,v_2)}$ 达到最大的路径,称为 v_1 到 v_2 的最大影响路径,其影响强度记为 $P_{\max(v_1,v_2)}$。

最大影响路径算法将个性化影响最大化问题的求解分为三个步骤:首先,求解网络节点到目标节点邻居的最大影响路径;然后,利用节点经过不同目标节点邻居到达目标节点的最大影响路径联合计算节点对目标节点的影响强度;最后,根据网络节点对目标节点的影响强度选取种子节点,形成种子集。

(1)求解最大影响路径。

考虑到目标节点 v_t 的邻居是其受网络影响的最直接来源,最大影响路径算法首先求解网络节点 v_i 到 v_t 的邻居集合 $A=\{v_j \mid v_j = neighbor(v_t)\}$ 中每个目标节点邻居的最大影响路径。然而,从最大影响路径定义及式(2)可知,穷举所有连接 v_i 到 v_j 的路径并计算其路径影响强度,再从结果中取最大值将耗费大量时间。为了更加快捷有效地获得最大影响路径并计算 $P_{\max(v_i,v_j)}$,将边影响强度 p_{e_i} 作如下对数函数转换:

$$a_{e_i}=\begin{cases}-\ln p_{e_i}, 0<p_{e_i}<1\\ \infty, p_{e_i}=0\end{cases}$$

则:

$$P_{\max(v_1,v_2)}=\max\Big(\prod_{e_i\in path_l(v_i,v_j)} p_{e_i}\Big)=$$

$$\max\Big(\prod_{e_i\in path_l(v_i,v_j)} e^{-a_{e_i}}\Big)=e^{-\min\left(\sum_{e_i\in path_l(v_i,v_j)} a_{e_i}\right)}$$

通过将 p_{e_i} 到 a_{e_i} 的转换,将求解 v_i 到 v_j 最大影响路径问题 $\max\Big(\prod_{e_i\in path_l(v_1,v_2)} p_{e_i}\Big)$ 转换为求解由 v_i 到 v_j、以 a_{e_i} 为网络边权值的最短路径问题 $\min\Big(\sum_{e_i\in path_l(v_1,v_2)} a_{e_i}\Big)$,从而提高算法求解效率。在算法实现中,本文采用加入优先队列的 Dijkstra 算法求解最大影响路径。

(2)计算路径联合影响 $P_{v_t}(v_i)$。

在分别求得 v_i 到集合 A 中每个节点的最大影响路径后,算法依式(3)用多条以 v_i 为起点的最大影响路径联合计算 v_i 对 v_t 的影响强度 $P_{v_t}(v_i)$。对于集合 A 中的节点 v_j,其目标节点影响强度等于连接 v_j 与 v_t 的网络边的影响传播强度,即:$P_{v_t}(v_j)=p_{v_jv_t}$。

$$P_{v_t}(v_i)=1-\prod_{v_j\in A}(1-P_{\max(v_i,v_j)}\cdot p_{v_jv_t}) \quad (3)$$

(3)选择种子节点。

对第(2)阶段计算所得的每个节点的目标节点影响强度排序,选取 Top-k 高影响强度的节点形成种子集。

图 1 为利用最大影响路径算法以节点 v_7 为目标节点求解的一个简单示例。最大影响路径首先利用 Dijkstra 算法获得每个非关键节点(v_1、v_2、v_3)对目标节点邻居(v_4、v_5、v_6)的最大影响路径,图中虚线分别表示 v_1 到 v_4、v_5、v_6 的最大影响路径、路径影响强度分别为 $P_{\max\langle v_1,v_4\rangle}$、$P_{\max\langle v_1,v_5\rangle}$、$P_{\max\langle v_1,v_6\rangle}$,然后计算 v_1 对目标节点 v_7 的影响强度 $P_{v_7}(v_1)$:

$$P_{v_7}(v_1)=1-\prod_{v_j\in\{v_4,v_5,v_6\}}(1-P_{\max\langle v_1,v_j\rangle}\cdot p_{v_jv_7})$$

同理可以计算出 $P_{v_7}(v_2)$、$P_{v_7}(v_3)$;目标节点邻居对目标节点的影响强度为连接两者的边影响强度,因此可得 $P_{v_7}(v_4)$、$P_{v_7}(v_5)$、$P_{v_7}(v_6)$。最后将 $P_{v_7}(v_1)$、$P_{v_7}(v_2)$,…,$P_{v_7}(v_6)$ 按降序排序,假设 $k=2$,则选择目标节点影响强度排在前两位的节点形成种子节点集。

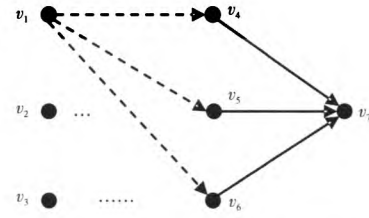

Figure 1 An example of MIPA

图 1 最大影响路径算法示例

具体伪代码如算法 1:

算法 1 最大影响路径算法

输入:社交网络图 $G(V,E)$,社交网络每条边的影响强度 $\{p_{e_i} \mid e_i\in E, 0\leqslant p_{e_i}<1\}$,目标节点 v_t,种子集大小 k;

输出:种子节点集 S。

Begin

1. $S\leftarrow\varnothing$; $A\leftarrow neighbor(v_t)$;
2. $V'=A\cup v_t$;$G'=G\backslash\{e_{uv} \mid u\in V', v\in V\}$;
3. for each $v_j\in A$

4. $P_{v_t}(v_j) = p_{v_j v_t}$;
5. 利用 Dijkstra 算法在 G' 中求解所有 $v_i \in V\backslash V'$ 到达 v_j 的最大影响路径,计算其影响强度 $P_{\max(v_i,v_j)}$;
6. end for
7. for each $v_i \in V\backslash V'$
8. $P_{v_t}(v_i) = 1 - \prod_{v_j \in A}(1 - P_{\max(v_i,v_j)} \cdot p_{v_j v_t})$;
/ * 多条路径计算联合影响强度 * /
9. end for
10. 对所有 $v_i \in V\backslash v_t$ 依 $P_{v_t}(v_i)$ 排序;
11. $S = \{v_i \mid v_i$ 为取得 $Top\text{-}k$ $P_{v_t}(v_i)$ 的节点$\}$;
12. End

算法首先将目标节点 v_t 及目标节点的邻居集合 A 视为关键节点 V',删除网络中起点为关键节点的所有网络边生成图 G'(行 2),以避免单条影响路径包含多个邻居节点而导致的最大影响路径重复计算;然后基于图 G' 求解每一个非关键节点 $v_i \in V\backslash V'$ 到每一个目标节点邻居 $v_j \in A$ 的最大影响路径并计算其影响强度 $P_{\max(v_i,v_j)}$(行 3～行 6);接着根据式(4)计算 v_i 通过这些经过目标节点邻居的最大影响路径对 v_t 的联合影响强度 $P_{v_t}(v_i)$(行 7～行 9);最后,根据所有网络节点对 v_t 的影响强度选择 Top-k 节点形成种子集,算法结束(行 10～行 12)。

5 实验结果及分析

本文实验是在主频为 3.70 GHz 的 Intel Xeon E5-1620 v2 CPU、运行内存为 16 GB 的计算机上进行,所有算法代码均使用 C++编写。实验用到的社交网络数据集全部选自 Stanford Network Analysis Platform(http://snap.stanford.edu/index.html)提供的开放数据集,统计特性如表 1 所示。

Table 1 Description for testing datasets

表 1 实验数据集描述

	数据集名	数据来源	节点数	边数	平均聚类系数
数据集 1	p2p-Gnutella08	P2P network	6 301	20 777	0.010 9
数据集 2	wiki-Vote	Wikipedia vote	7 115	103 689	0.140 9
数据集 3	soc-Epinions1	Epinions.com	75 879	508 837	0.137 8

本文采用局部级联算法(LCA)、LND 算法[12]、随机(Random)算法[11]三种个性化影响最大化算法作为基准算法,与最大影响路径算法(MIPA)进行实验对比。

实验参数主要有种子节点数 k 以及网络边影响传播强度 p_{e_i},为使实验结果能更加全面地反映算法性能,从三个数据集中随机抽取网络节点分别作为目标节点。在实验参数设置上:种子节点个数 k 取值由 1 到 10;每条网络边的影响强度 p_{e_i} 于实验前期在 0 到 0.5 之间随机分配。

图 2～图 4 为不同算法求得的种子集对目标节点的影响强度对比,实验中经随机抽取作为目标节点的分别为:数据集 1 中 ID 号分别为 583、1592 的节点;数据集 2 中 ID 号分别为 363、1389 的节点;数据集 3 中 ID 号分别为 31、1084 的节点。

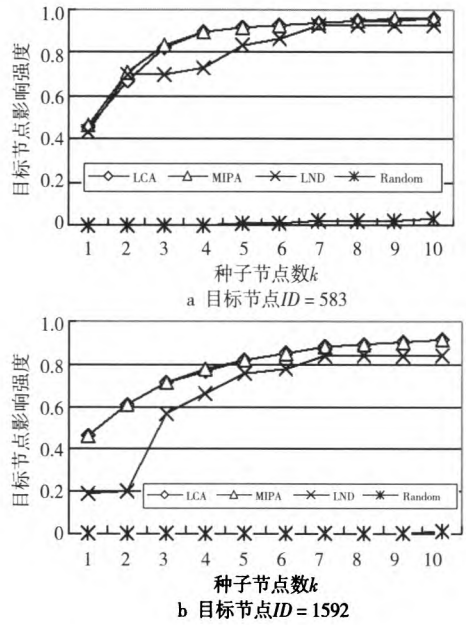

Figure 2 Personalized influence maximization results on dataset 1

图 2 对数据集 1 的个性化影响最大化结果

从图 2～图 4 可以看出:在三个不同的数据集上,MIPA 所得的种子集均能取得比 LCA、LND、Random 更好的目标节点影响强度。随着种子节点数 k 的增加,在目标节点影响强度上,MIPA、LCA、LND 取得的提高相比 Random 更加明显;当 k 增加到一定程度时,目标节点影响强度趋于饱和则不再增加,MIPA 总是率先达到饱和点,LCA 次之。这是因为:在解决个性化影响最大化问题上,Random 完全随机地从网络中选取种子节点,方法太过简单、缺乏策略,增加种子节点数量不能保证提高目标节点影响强度;LND 则只考虑了目标节点邻居对目标节点的直接影响,虽然在某种程度上能够取得较好的目标影响强度,但忽略了其他网络节点的作用;LCA 和 MIPA 算法虽然都在算法求解过程中增加了相关路径策略,但在网络中每条边

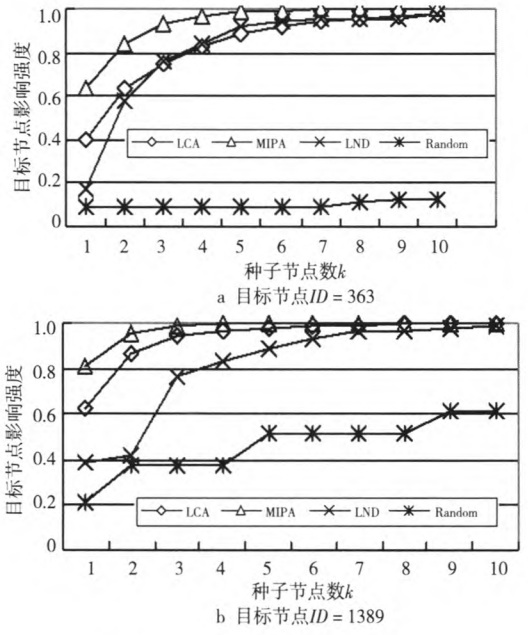

Figure 3 Personalized influence maximization results on dataset 2

图 3 对数据集 2 的个性化影响最大化结果

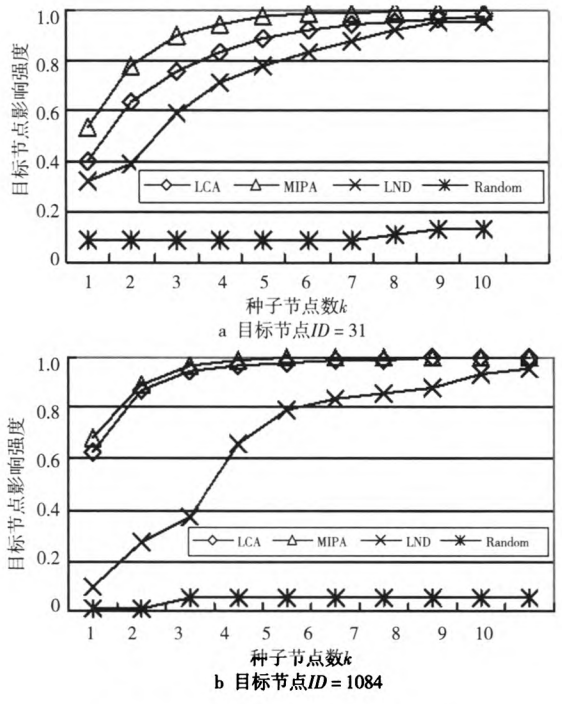

Figure 4 Personalized influence maximization results on dataset 3

图 4 对数据集 3 的个性化影响最大化结果

的影响强度不一致的情况下,采用节点间最大影响路径强度来代表节点间影响而导致的误差 E_{MIPA} 显然小于 LCA 中最短路径导致的误差 E_{LCA}(式(4))。正因为 MIPA 算法采用的影响强度计算方法贴近实际、更加合理,克服了 LCA 中最短路径在影响强度估算时缺乏代表性这一弱点,在影响强度上取得了一定的优势。

$$E_{\mathrm{MIPA}}-E_{\mathrm{LCA}}=(P_{v_t}(v)-P_{\max(v,v_t)})-(P_{v_t}(v)-P_{shortest(v,v_t)})=P_{shortest(v,v_t)}-P_{\max(v,v_t)}\leqslant 0 \tag{4}$$

图 5 为不同算法在实验数据集上运行时间对比,由于三个数据集的规模及统计特性有所不同,各算法在求解时间上也有一定区别。从单个数据集分析,在算法运行时间方面,MIPA 耗时最长,LCA 次之,LND、Random 两种算法耗时相对较短,其原因在于:MIPA 考虑到在现实网络中边影响传播强度的不一致性,利用加入优先队列的 Dijkstra 算法获得最大影响路径,时间复杂度为 $O(n'\cdot(2m+n\cdot\lg n)+n+n\cdot\log n)$,其中 Dijkstra 算法时间复杂度为 $O(2m+n\cdot\lg n)$;LCA 将所有网络边影响强度视为一致,计算最短路径仅需广度优先遍历网络图,时间复杂度仅为 $O(m+n(\bar{L}+k))$;LND、Random 求解策略比较粗放,时间复杂度分别为 $O(k\cdot n')$、$O(m)$。其中:n、m 分别为网络节点数和边数,k 为种子集大小,n' 为目标节点邻居数量,$\bar{L}$ 为 LCA 算法计算单个节点到目标节点最短路径集合时涉及到的网络边的平均值。

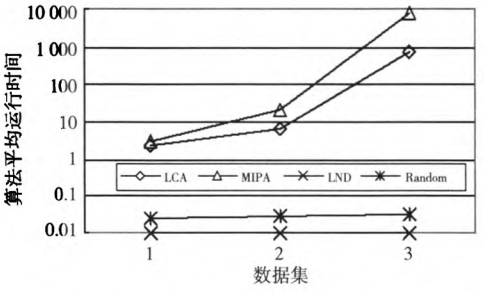

Figure 5 Running time of the four algorithms

图 5 运行时间对比

综合图 2~图 5 可以看出,在三组数据规模及统计特性各不相同的数据集上,本文提出的最大影响路径算法均能取得较好的目标节点影响强度,这说明了 MIPA 的普遍适用性。MIPA 首先计算网络中节点对目标节点的影响强度,然后依贪心策略取 Top-k 节点形成的种子集。实验结果表明,在处理节点数较少的数据集 1 时,MIPA 和 LCA 二者所获得的目标影响强度相当,LCA 算法在运行时间上占有一定优势;而对于节点数较多的数据集 3,MIPA 虽然耗时较大,但在目标影响强度上取得的优势也比较明显。因此,在实际运用中,可根据对效率和目标影响强度的不同需求在二者中选择更加适合的求解手段。此外,由于个性化影响最大化目标函数 $P_{v_t}(\cdot)$ 的单调性及其子模特性,根据文献[14]中所得出的结论:"如果一个子模函数 f 的一个贪心最大化算法返回的节点集为 S_{greedy},则

相较于该问题的最优解 S,$f(S_{greedy}) \geqslant (1-1/e)\max_{|A|\leqslant k} f(S)$(其中 e 为自然常数)",引入贪心策略的最大影响路径算法求得的种子节点集具有最优解 63%的理论精度保证。

6 结束语

本文在现有的社交网络个性化影响最大化问题研究的背景下,更加清晰地认识到真实网络中边影响传播强度不一致的实际情况,为在此基础上获得更高目标节点影响强度,结合影响传播路径策略提出了最大影响路径算法 MIPA。与现有的个性化影响最大化解决方案不同之处在于:MIPA 将网络边强度进行对数转换,使算法能够快速获得节点间最大影响路径并以此估算节点对目标节点的影响强度。在三组规模及统计特性各不相同的真实数据集上进行测试,实验结果证实了该算法的有效性,同时验证了其在目标节点影响强度上所取得的优势的普遍适用性。

在未来的工作中,一方面我们将进一步探索个性化影响最大化求解策略,以寻求在求解效率方面有所突破;另一方面,还将研究个性化影响最大化问题求解的附加价值,即:种子节点集在保证对目标节点影响强度的同时对一般网络节点形成的影响程度及其范围,争取从多角度满足多元化的个性化影响最大化需求。

参考文献:

[1] Domingos P, Richardson M. Mining the network value of customers[C]// Proc of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001: 57-66.

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

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

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

[5] Goyal A, Lu W, Lakshmanan L. CELF++: optimizing the greedy algorithm for influence maximization in social networks[C]// Proc of the 20th International Conference Companion on World Wide Web, WWW 2011, 2011: 47-48.

[6] Chen Wei, Wang Ya-jun, Yang Si-yu. Efficient influence maximization in social networks[C]// Proc of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009: 199-208.

[7] Wang Chi, Chen Wei, Wang Ya-jun. Scalable influence maximization for prevalent viral marketing in large-scale social networks[C]// Proc of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010: 1029-1038.

[8] Jiang Qing-ye, Song Guo-jie, Cong Gao, et al. Simulated annealing based influence maximization in social networks[C]// Proc of the National Conference on Artificial Intelligence, El Segundo: AI Access Foundation, 2011: 127-132.

[9] Nguyen H, Zheng Rong. On budgeted influence maximization in social networks [J]. IEEE Journal on Selected Areas in Communications, 2013, 31(6): 1084-1094.

[10] Han Shuo, Zhuang Fu-zhen, He Qing, et al. Balanced seed selection for budgeted influence maximization in social networks[C]// Proc of the 18th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, 2014: 65-77.

[11] Guo Jing, Zhang Peng, Zhou Chuan, et al. Personalized influence maximization on social networks[C]// Proc of the 22nd ACM International Conference on Conference on Information & Knowledge Management, 2013: 199-208.

[12] Guo Jing, Zhang Peng, Fang Bin-xing, et al. Personalized key propagating users mining based on LT model[J]. Chinese Journal of Computers, 2014, 37(4): 809-818. (in Chinese)

[13] Zhang Bo-lei, Qian Zhu-zhong, Wang Qin-hui, et al. Maximize information coverage algorithm for target market[J]. Chinese Journal of Computers, 2014, 37(4): 894-904. (in Chinese)

[14] Nemhauser G, Wolsey L, Fisher M. An analysis of approximations for maximizing submodular set functions[J]. Mathematical Programming, 1978, 3(14): 265-294.

附中文参考文献:

[12] 郭静,张鹏,方滨兴,等. 基于 LT 模型的个性化关键传播用户挖掘[J]. 计算机学报,2014,37(4): 809-818.

[13] 张伯雷,钱柱中,王钦辉,等. 面向目标市场的信息最大覆盖算法[J]. 计算机学报,2014,37(4): 894-904.

作者简介:

杨书新(1978-),男,江西九江人,博士,副教授,CCF 会员(11120M),研究方向为社交网络、个性化推荐系统和图数据管理。**E-mail**:yimuyunlang@sina.com

YANG Shu-xin, born in 1978, PhD, associate professor, CCF member (11120M), his research interests include social network, personalized recommendation system, and graph data management.