大规模时序图影响力最大化的算法研究

吴安彪[1)] 袁 野[1)] 乔百友[1)] 王一舒[1)] 马玉亮[1)] 王国仁[2)]

[1)](东北大学计算机科学与工程学院 沈阳 110000)

[2)](北京理工大学计算机学院 北京 100081)

摘 要 影响力最大化问题在社交网络中有着广泛的应用,一般地可以将社交网络抽象为静态图,影响力最大化问题是指在图中找出 k 个最有影响力的顶点,使得信息最大化传播.近年来对此问题的研究主要基于静态图,但是在现实中某些特定网络不可简单地被抽象为静态图,如社交网络及路网中节点间只在某些特定时间存在联系,即节点间的联系是具有时序性的.因此,本文研究了时序图影响力最大化问题,即在时序图上寻找 k 个顶点使得信息在特定的时间段内最大化传播.传播模型的选择和节点间传播概率的计算是影响力最大化问题的基础,由于基于静态图的 IC(Independent Cascade model)传播模型无法应用于时序图,因此本文首先对 IC 模型进行改进,并提出了 ICT(Independent Cascade model on Temporal graph)传播模型,使信息可以通过 ICT 传播模型在时序图上进行传播.而后通过改进 PageRank 算法来进行计算节点间的传播概率.然后在此基础上将时序图影响力最大化问题分为两步来进行实现.第一步首先研究时序图节点影响力的计算,并提出了用来计算节点影响力的 SIC(Single Node Influence Computation)算法,然后通过对时序图中节点联系时序性这一特性的研究提出了一种改进算法 ISIC(Improved SIC).第二步是在第一步结果的基础上来寻找 k 个种子节点,首先提出了一种基本的时序图影响力最大化算法 BIMT(Basic Method for IMTG).但 BIMT 难以高效解决大规模时序图影响力最大化问题,因此通过优化节点边际效应的计算时间,提出了高效的 AIMT(Advanced Method for IMTG)算法,然后通过避免某些节点边际效应的重复计算,对 AIMT 算法进行改进,从而提出了 IMIT(Improved Method for IMTG)算法.最后通过大量实验验证了 AIMT 和 IMIT 两种算法高效性和扩展性,相比于 BIMT 算法,AIMT 和 IMIT 可以更加快速地解决大规模时序图影响力最大化问题.

关键词 时序图;影响力最大化;信息传播模型;边际效应;社交网络

中图法分类号 TP399 **DOI 号** 10.11897/SP.J.1016.2019.02647

The Influence Maximization Problem Based on Large-Scale Temporal Graph

WU An-Biao[1)] YUAN Ye[1)] QIAO Bai-You[1)] WANG Yi-Shu[1)] MA Yu-Liang[1)] WANG Guo-Ren[2)]

[1)](*School of Computer Science and Engineering*, *Northeastern University*, *Shenyang* 110000)

[2)](*School of Computer Science and Technology*, *Beijing Institute of Technology*, *Beijing* 100081)

Abstract Influence maximization problem has been widely studied in social networks, and well applied in the scenarios such as viral marketing and epidemic prevention. The influence maximization problem aims to find k influential nodes in graphs that maximize the dissemination of information. Recently, in most existing works, the social network is abstracted as static graph, then implement the influence maximization problem in the static graph, and finally return k nodes as the social network's influential nodes. However, the most former researchers ignored the truth that most

收稿日期:2018-07-16;在线出版日期:2019-04-02.本课题得到国家重点研发计划项目(2016YFC1401900)、国家自然科学基金(61572119,61622202,U1401256,61732003,61729021)、中央高校基本科研业务费专项资金(N1150402005)资助.**吴安彪**,博士研究生,主要研究方向为影响力最大化问题、时序图和图神经网络.E-mail:anbiaowu@foxmail.com.**袁 野**(通信作者),博士,教授,博士生导师,国家优秀青年基金获得者,中国计算机学会(CCF)数据库专业委员会委员,中国计算机学会(CCF)大数据专家委员会委员,中国计算机学会(CCF)高级会员,IEEE 高级会员,主要研究领域为云计算、大数据管理(包括图数据管理、不确定数据管理、数据隐私保护)、P2P 计算等.E-mail:yuanye@mail.neu.edu.cn.**乔百友**,博士,副教授,主要研究方向为图数据管理、不确定图管理.**王一舒**,博士研究生,主要研究方向为图数据管理、不确定数据管理.**马玉亮**,博士研究生,主要研究方向为社交网络、图数据管理.**王国仁**,博士,教授,长江学者特聘教授,主要研究领域为不确定数据管理、密集型数据计算等.

networks in real life cannot be simply denoted as static graphs, such as social networks, traffic networks, brain networks, and so forth, due to the connections between nodes in these networks are time related. Therefore, in this work we study the influence maximization problem on temporal graph (IMTG), that is, IMTG aims to find k influential nodes in temporal graph that maximize the number of influenced nodes. The information propagation model and propagation probability is the fundamental of influence maximization problem. So, firstly, we design a new propagation model based on IC (Independent Cascade) propagation model, named ICT (Independent Cascade model on Temporal graph) propagation model, to make information spread on temporal graphs, since the static graph-based IC propagation model cannot be directly applied to temporal graphs. Secondly, we propose a new method based on PageRank algorithm to compute the propagation probability. We improve the PageRank algorithm by taking the node's communication frequency into account, because the higher connection frequency between two nodes, the closer they are, which means they are more influenced by each other. After the issues of propagation model and propagation probability have been solved, we study the influence maximization problem on temporal graph through two steps. In the first step, we focus on the computation of single node influence probability, and propose an algorithm, named SIC (Single node Influence Computation), to implement the computation of single node influence, then we propose an improved algorithm ISIC (Improved SIC) by researching the characteristic that the connections among the nodes on temporal graph are time related. In the second step, we implement the influence maximization problem on temporal graph by using the result of the first step, and propose a basic method BIMT (Basic Method for IMTG). However, the BIMT algorithm will cost plenty of time when deal with large scale temporal graphs. So we propose an effective algorithm AIMT (Advanced Method for IMTG) by mapping each id of node's in the graph to nature numbers. Compared with IMT method, AIMT can deal with large scale graph with high efficiency. Though the research of the computation of nodes' marginal effect we find that some nodes' marginal effect is unnecessary to be recomputed during the seed node selection process, so we improve the AIMT algorithm and propose a more effective algorithm IMIT (Improved method for IMTG) to avoid multiple computation of the marginal effects of certain nodes. Finally, the experiment results verify the high efficiency and scalability of AIMT and IMIT algorithms, AIMT and IMIT algorithms can handle influence maximization on large temporal graph with less time than BIMT.

Keywords temporal graph; influence maximization; information propagation models; marginal effect; social network

1 引 言

如今,在线社交网络比如微博、微信及邮件系统等在人们的生活中扮演着重要的角色.人们在社交网络表达他们的想法、分享新闻和信息,并以此来影响网络中的其他用户,通过这种“口碑(word-of-mouth)效应”人们设计出一种新的营销技术,被称为“病毒营销”(viral marketing).为了研究病毒营销,Domingo 等人首次提出了影响力最大化问题[1-2]并将其转化为一个和图论相关的算法问题.

基于静态图的影响力最大化问题是指在网络中寻找 k 个用户作为种子节点,使得信息于特定的传播模型(如 IC 传播模型)下通过 k 个用户在网络中尽可能多的影响到其他用户,即将影响力最大化问题作为一个离散最优化问题进行研究.

但是现实生活中的很多网络图并不可以被简单地抽象为静态图,比如人与人之间的电话网络、相互之间的邮件传送、交通网络以及脑神经网络等,在这些网络中,节点之间并不会自始至终都会存在联系,而是只在某个时间段存在联系,即节点之间的联系是具有时序性的.以脑网络为例,越来越多的研究者

热衷于脑神经网络中的研究，而由于大脑中的神经元的信息传播也是具有时序性的，所以当研究有哪些神经元对脑网络中的信息传播起重要作用时，便可以通过时序图影响力最大化问题的研究解决此类问题.

由于节点间联系的时序性导致基于静态图的IC模型无法适用于时序图，因此首先需要对IC模型进行改进使得其可以适用于时序图，然后再进一步研究并解决时序图影响力最大化问题.

以图1为例来简单说明影响力最大化问题的静态图和时序图在节点影响力计算时的不同. 其中图G表示静态图，边上的权重表示节点传播概率. 图G_T表示时序图，为了简单起见，时序图G_T中各顶点间传播概率和静态图G保持一致. 边上的权重集合表示两节点在相应的时刻存在联系，以图G_T中顶点a，c为例，顶点a和顶点c边上时间权重集合为{3，6}，表示两节点在时刻3以及时刻6存在联系，其余时刻没有联系.

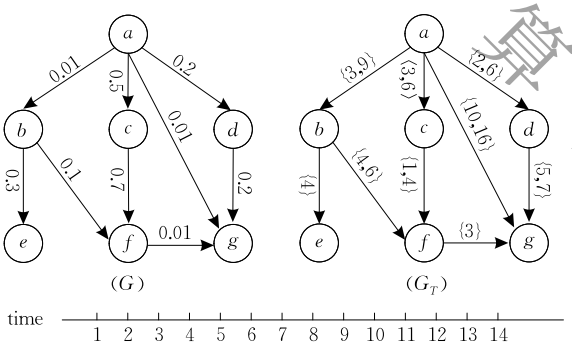


图1　静态图和时序图

接下来，分别计算顶点a在这两个图中的影响力大小. 在静态图G中，顶点a依次尝试激活其邻居点b，c和d. 假如只是成功激活了顶点c，则顶点c此时变为活跃节点，从而可以尝试去激活其邻居顶点f，如果成功激活顶点f，则顶点f尝试激活顶点g，如果激活失败则消息通过顶点a可以影响到两个顶点，所以顶点a的影响力大小为2. 在时序图G_T中，同样假设顶点a也只是激活了顶点c，但是顶点a和c边上的权值表明两个顶点只是在时刻3和6时存在联系，也就意味着顶点c最早是在时刻3的时候被激活，即顶点c在时刻3之前是处于未激活状态，然而顶点c和其邻居顶点f只在时刻1和时刻2时存在联系，所以顶点c不可能激活其邻居顶点f，所以在时序图G_T中顶点a的影响力大小为1.

由以上分析可知，如果只是单纯地将基于静态图的影响力最大化算法应用在时序图上(即忽略边上时间戳集合)，来研究时序图影响力最大化算法问题是无法得到正确结果的.

Domingo等人[1]提出了可以解决静态图影响力最大化问题的Greedy算法，Greedy算法将影响力最大化问题分为两个子问题：(1)通过蒙特卡洛模拟近似计算单节点影响力大小；(2)使用贪心近似算法选取k个最有影响力的节点. 但是Greedy算法无法有效地解决大规模静态图的影响力最大化问题，文献[3-6]都是在Greedy算法的基础上对影响力最大化问题做出优化，尽可能缩短程序的运行时间以解决大规模静态图影响力最大化问题.

导致Greedy算法效率低下主要原因是在每次迭代寻找最有影响力节点的过程中需要对所有非种子节点进行节点影响力的计算，然后这些计算中的绝大部分是没有必要的，文献[7-9]为了避免Greedy算法中的限制，使用了一种和Greedy算法完全不同的方法RIS(Reverse Influence Sampling)来解决影响力最大化问题. RIS算法通过两个步骤运行：(1)通过静态图生成一定数量的随机反向可达集合(Reverse Reachable Sets，RR sets)；(2)逐次选取覆盖RR sets最多的节点作为种子节点，直到选取k个种子节点为止.

无论是以Greedy算法为基础还是以RIS算法为基础来研究影响力最大化问题，都必须提前确定信息在社交网络中的传播模型，只有确定了信息的传播模型才可以计算节点的影响力，而节点影响力的大小也就直接决定其是否可以作为种子节点. 传统的基于静态图的传播模型无法直接应用于时序图上，所以基于静态图的影响力最大化算法无法解决时序图的影响力最大化问题.

由于在时序图中节点之间联系具有时序性，节点之间的边是随着时间动态变化的，所以时序图影响力最大化问题面临如下挑战：(1)传统的传播模型无法直接应用在时序图上；(2)节点的活跃状态需要考虑其活跃起始时间这一因素；(3)在种子节点选取的过程中节点边际效应的计算和传统的计算方式不同.

为了解决以上挑战，本文首先通过对传统独立级联模型(Independent Cascade Model，IC)做出一定的改进，使得其可以应用在时序图上从而求得时序图中各个节点的影响力，然后在基于静态图影响力最大化算法的基础上提出了一种可以解决时序图影响力最大化算法BIMT(Basic Method for IMTG). 但是由实验发现BIMT算法解决大规模时序图影

响力最大化问题需要消耗大量的时间，且大部分时间都浪费在了节点的边际效应计算上面，所以通过优化节点边际效应的计算时间，提出了更高效的AIMT(Advanced Method for IMTG)算法，最后由节点边际效应的子模性这一性质对 AIMT 算法进行改进，从而减少某些节点的边际效应的重复计算，以此提出了 IMIT(Improved method for IMTG)算法.

本文的主要贡献：

(1)本文首次以时序图为对象研究影响力最大化问题，并对 IC 传播模型进行重新设计使其可以被用来解决时序图影响力最大化问题；

(2)通过改进 PageRank 算法提出了一种新的在时序图上计算节点间传播概率的方法；

(3)由于时序图中节点间联系具有时序性，在此基础上对节点的邻居节点按时间序列进行排序，由此减少了单节点影响力计算时间；

(4)本文首先提出了基于大规模时序图影响力最大化的基本算法 BIMT，然后通过对 BIMT 算法进一步优化提出 AIMT 和 IMIT 两种可以解决大规模时序图的影响力最大化算法；

(5)通过在四种真实数据上的实验，验证了算法在大规模时序图上的高效性和扩展性.

本文第 2 节介绍相关工作；第 3 节介绍时序图影响力最大化的基础知识和相关定义；第 4 节介绍节点影响力计算的两种算法并提出时序图影响力最大化的基本算法 BIMT；第 5 节通过对 BIMT 算法进行改进提出两种可以解决大规模时序图影响力最大问题的算法；第 6 节介绍在 4 个不同数据集上进行的实验及结果分析；最后进行工作总结.

2 相关工作

信息传播模型是影响力最大化问题研究的基础，只有确定了传播模型才可以进行下一步的研究. 本节对 IC 传播模型以及在静态图和动态图上的影响力最大化算法做简要的介绍.

2.1 IC 传播模型

在 IC 传播模型中，每条有向边(u,v)设置一个实数值 $p_{u,v}\in[0,1]$，$p_{u,v}$表示节点 u 通过边(u,v)成功影响节点 v 的概率. 在初始 t 时刻，活跃节点 u 只有一次机会激活它的每个非活跃邻居节点 v，激活概率为 $p_{u,v}$，如果 v 被成功激活则其在 $t+1$ 时刻变为活跃节点. 如果节点 v 在时刻 t 有多个活跃父节点，则活跃父节点在 t 时刻以任意顺序激活节点 v. 此过程一直持续到没有新的节点被激活为止.

2.2 静态图影响力最大化算法

文献[1-2]首次提出影响力最大化问题并提出了 Greedy 算法解决此问题，然而这种算法的缺点是计算量很大，不适合在大规模社交网络上运行. 文献[3]证明了在多种传播模型下，节点影响力的数学期望的计算是一个 NP-hard 问题，并且首次将此问题转化为离散优化问题. 在 2007 年，文献[4]提出了 CELF 算法，CELF 算法利用了节点边际效应的子模性这一性质来减少计算某些节点的边际效应，从而提高算法的运行速度，虽然 CELF 算法和 Greedy 算法的时间复杂度相同，但是在运行时间上 CELF 算法比 Greedy 算法快 700 倍以上. 而后，文献[10]通过对 CELF 算法进一步优化提出了 CELF++算法，通过实验表明 CELF++算法比 CELF 算法要快 35%～55%. Borgs 等人[9]在独立级联(IC)模型下取得了理论上的突破，并提出时间复杂度为 $O(kl^2(m+n)\log^2 n/\varepsilon^3)$的算法来解决影响最大化问题，但他们的算法实际运行效率却难以令人满意. Youze 等人[7]在文献[9]的基础上提出了 TIM 算法，这种算法在确保结果在$(1-1/e-\varepsilon)$精度下的情况下使得算法运行时间接近线性最优. 文献[8]指出了文献[7]中可以做出优化的地方，并提出 IMM 算法，此算法弥补了 TIM 算法的不足之处. 在文献[5]中，开发了一种新颖的基于草图的设计(sketch-based design)来计算影响力并提出了 SKIM 算法，SKIM 算法可以在确保较高精确度的情况下在有数亿条边的大图上高效运行. 图节点的分类问题[11]已被广泛研究，在此问题的基础上对节点进行社区划分，如此便可研究面向社区的影响力最大化问题，文献[12]在 k-clique 问题的基础上提出了 kr-clique 这一新的概念，即在一个局部社区内任意节点之间在 r 步跳跃内可以任意到达，然后计算各个社区间的对社区外节点的影响力. 文献[13]一改传统寻找 k 个种子节点的做法，转而通过寻找 k 个信息标签来进行计算个人在社交网络中的最大影响力，进而提出了一种新型的影响力最大化问题. 文献[14]在传统影响力问题的基础上将社交网络中的个人地理位置这一因素考虑了进去，提出了一种面向地理位置的影响力最大化问题.

以上这些算法虽然可以解决静态图影响力最大化问题，但是由于它们都是在基于静态图的传播模型的基础来解决影响力最大化问题，该传播模型无

法适用于时序图.所以,这些算法都无法直接解决时序图影响力最大化问题.

2.3　动态图最大化算法

近年来,有一些研究者通过对基于静态图的影响力最大化算法进行改进,从而将影响力最大化问题的研究对象由静态图转移到动态图上去,并提出了可以解决动态图影响力最大化问题的算法.

文献[15-16]首次在动态图上进行了影响力最大化问题的研究.其中文献[15]是对文献[5]中的SKIM算法进行改进,并将其适用于动态图的影响力最大化问题的实现.其采用反向可达采样方法首先采样处多个采样集合,通过采样集合来找出种子节点集合,而后图中会有节点的添加或删除操作,通过计算节点的删除或添加对当前采样集合的影响来重新计算种子节点集合.由于其完全没有考虑节点间联系的因素,且本文是以全局的角度来研究时序图影响力最大化问题,其间并无节点增删的操作,所以文献[15]的研究方法无法解决本文所研究的问题.文献[16]则是使用的一种新的窗口滑动的模型来研究动态图上的实时影响力最大化问题,其研究思路为设置一个时间窗口 w,将节点间的联系看作一个 action,并将这些 action 按照时间的先后顺序存放在 w 中.窗口 w 会随着时间向下滑动,此时便涉及到新的 action 的进入和旧的 action 的退出(因为窗口的大小可以人为设定),根据节点的进入和退出,来判断是否需要对在上一个时间段所求出的窗口中的种子节点进行重新计算.而由于本文是从全局的角度在时序图上研究影响力最大化问题,所以文献[16]中的研究思路也无法解决本文所要研究的问题.

2.4　时序图

时序图和静态图的本质上的不同是时序图在边的权重上加入了时间戳这一因素.静态图上的边一旦存在便不会因时间的变化而改变,而在时序图中,边会因时间的变化在两种状态下相互转化:激活状态和非激活状态.时序图中顶点间只在边处在激活状态时是存在联系的.

在现实生活中有很多常见的网络都可以描述为时序图.(1)点对点通信网络:如电子邮件、手机短信等;(2)一对多的消息传播网络:在这种网络中注重的是单一用户对其余多个用户的信息传播;(3)生物信息网络:如代谢网络、蛋白质互作用网络等.研究表明,在生物信息网络中,各节点间的交流是时间相关的,所以 Przytycka 等人[17]认为对于生物信息网络的分析是需要借助动态网络来实现的,且现如今在对蛋白质互作用[18]和基因调控网络[19]的研究工作中已经有研究者开始分析时间对网络的影响.

一般情况下时序图可以按照顶点间联系持续时间 $\lambda(e)$ 的大小分为以下两种情况.

第一种是各顶点间联系无持续性或其持续时间 $\lambda(e)$ 可忽略不计,即 $\lambda(e)=0$.这种情况下,图中的边 e 可用三元组 (u,v,t) 进行表示,每条边 e 都存在一组时间的序列 $T_e=\{t_1,\cdots,t_n\}$.现实生活中的即时通信网络(如邮件、电话、信息网络等)和可以将持续时间 $\lambda(e)$ 忽略的网络都可以抽象表示为此种时序图.考虑到信息传播的及时性,本文是在基于 $\lambda(e)=0$ 的基础上在时序图上进行影响力最大化问题研究的.第二种是时序图中的边在一定的时间段内被激活,即 $\lambda(e)\neq 0$.在这种时序图中,持续时间是一个重要、不可忽视且必须考虑的因素.例如在物流和交通网络中[20-22],物流站点、机场和车站可抽象为顶点,顶点间的边表示站点间的物流信息,边上的时间标签表示在标签时间上有物流经过.虽然,就拓扑结构方面而言,时序图和静态图有部分的类似.但在时序图中所引入的时间标签这一性质是时序图所特有的,且时序图的拓扑关系会由时间的变化而发生改变,所以时序图上的基本拓扑性质是无法直接从静态图中进行引用的.

3　问题定义

本节将对一些基本的问题如时序图的定义、在时序图中对节点间传播概率的计算方法以及何将IC传播模型在时序图中进行应用等问题进行详细的阐述,并且对基本的概念做出定义.

首先在表1中列出了在本文中常用到的一些符号及其意义描述.

表1　符号及意义说明

符号	意义	符号	意义
G_T	时序图	$p_{u,v}$	节点 u 和 v 之间的传播概率
V	时序图顶点集合	NR_u	节点 u 的 rank 值
E	时序图边集合	Act_u	节点 u 的活跃起始时间
$\lvert V\rvert$	顶点个数	S	种子节点集合
$\lvert E\rvert$	边的个数	$\varphi(u)$	节点 u 的影响力
T_E	节点之间存在联系时刻的集合	$infs(u)$	节点 u 的边际效应
$T_{(u,v)}$	节点 u 和 v 之间存在联系的时刻集合	$\lvert infs(u)\rvert$	节点 u 的边际效应大小

3.1 基本定义

定义 1. 时序图. 给定网络 $G_T(V,E,T_E)$ 表示为节点间带有时序关系的社交网络有向时序图，V 表示节点的集合，E 表示边的集合，且 $|V|=n$，$|E|=m$. T_E 表示图中所有节点之间存在联系时刻的集合，$T_{(u,v)}$ 表示在节点 u 和 v 之间存在联系的时刻的集合，$T_{(u,v)}\in T_E$.

以图 2 为例，边上的数字表示在节点存在的联系时刻，以顶点 Cor 和 Han 为例，$T_{(\text{Cor,Han})}=\{9,11\}$，表示两顶点在时刻 9 和 11 时存在联系.

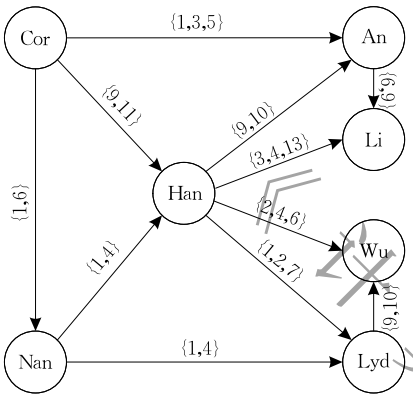


图 2 时序图

3.2 传播概率的计算

定义 2. 传播概率. 活跃节点 u 通过边 (u,v) 成功激活其邻居节点 v 的概率为传播概率，表示为 $p_{u,v}\in[0,1]$.

在传统的影响力最大化算法研究中，计算节点间的传播概率通常使用随机赋值的方法，如设定一个概率集合 $P=\{0.01,0.1,0.3,0.5\}$，然后从集合 P 中随机选取概率值作为各节点间的传播概率. 此种方式的弊端是所求取的传播概率并不符合实际情况.

为了使在算法中所用的节点间传播概率更符合实际情况，文献[23]中提出了一种基于 PageRank 算法来计算节点间传播概率的方法，首先通过 PageRank 方法计算节点 u 的 PageRank 值 PR_u，其中 PR_u 可以通过式(1)计算：

$$PR_u=\frac{1-d}{|N|}+d\times\sum_{v\in In(u)}\frac{1}{|Out(v)|}PR_v \tag{1}$$

PageRank 值越高的父节点对其子节点的影响力越大，则节点 u 和 v 的传播概率 $p_{u,v}$ 等于 u 的 PageRank 值与所有链向 v 的父节点的 PageRank 值之和的比值，如果节点 u 和 v 之间不存在边，则 $p_{u,v}=0$. 在这种方法中，$p_{u,v}$ 可以通过式(2)表示：

$$p_{u,v}=\begin{cases}\dfrac{PR_u}{\sum\limits_{v_i\in In(v)}PR_{v_i}}, & (u,v)\in E\\ 0, & (u,v)\notin E\end{cases} \tag{2}$$

这种方法相较于随机赋值法可以保证所求得的传播概率 $p_{u,v}$ 更加贴近实际情况，但是这种计算方法在时序图中有一个明显缺陷，那就是没有考虑节点间联系次数 $|T_{(u,v)}|$ 这一因素. 以图 3 为例进行简单说明. 由式(1)可知，在静态图 G 节点 C 和节点 B 的 PageRank 值是相同的，而当计算各时序图 G_T 中各个节点的 PageRank 时，如果不考虑各个节点间的联系次数则节点 E 和节点 F 的 PageRank 值也是相同的，因为两个图的结构在不考虑时间这一因素时是完全相同的.

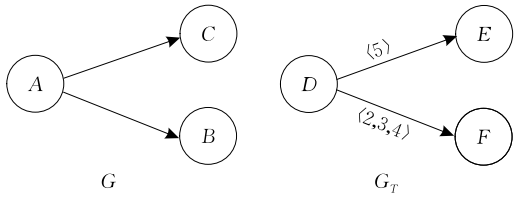


图 3 PageRank 计算示意图

但是考虑连接次数 $|T_{(u,v)}|$ 这一因素时，可以发现节点 D 和节点 F 的连接次数 $|T_{(D,E)}|$ 是小于节点 D 和节点 F 的连接次数 $|T_{(D,F)}|$ 的. 而一个节点被联系的次数越多，也就意味着其 PageRank 值越大，但是仅仅使用式(1)是无法达到这一要求的. 所以，本文对式(1)进行了改进，即将节点间的联系次数和 PageRank 算法相结合来求出节点新的 rank 值，本文命名为 NodeRank. 节点 u 的 NodeRank 值表示为 NR_u，可以通过式(3)计算：

$$NR_u=\frac{1-d}{|N|}+d\times\sum_{v\in In(u)}\frac{|T_{(u,v)}|}{\sum\limits_{v_k\in Out(v)}|T_{(v_k,v)}|}NR_v \tag{3}$$

一旦各个节点的 NodeRank 值被求得，则节点 u 和节点 v 的传播概率 $p_{u,v}$ 可以通过式(4)表示：

$$p_{u,v}=\begin{cases}\dfrac{NR_u}{\sum\limits_{v_i\in In(v)}NR_{v_i}}, & (u,v)\in E\\ 0, & (u,v)\notin E\end{cases} \tag{4}$$

3.3 对 IC 传播模型的改进

定义 3. 节点活跃起始时间. 节点 v 被其活跃父节点 u 成功激活的时刻为其活跃起始时间，表示为 Act_v，且 $Act_v=\min\{t|(t\in T_{(u,v)}\text{ and }t\geq Act_u)\}$.

以图 2 为例，若节点 Cor 为种子节点(种子节点的活跃起始时间都为 0)，其成功激活节点 Han，则

$Act_{Han}=\min\{9,11\}=9$.

静态图的影响力最大化算法无需考虑节点被激活的起始时间，而在时序图中节点被成功激活的起始时间是需要考虑的.所以本小节通过对 IC 传播模型进行改进从而得到了一种新的基于时序图的传播模型，ICT(IC model on Temporal graph)传播模型.

同样以图 2 为例，若顶点 Han 被顶点 Cor 成功激活且 $Act_{Han}=9$，则顶点 Han 变为活跃顶点且尝试激活其邻居顶点 An，Li，Wu 和顶点 Lyd.顶点 Han 只有和顶点 An 和 Li 在时刻大于等于 9 时刻有联系，而和顶点 Wu 及 Lyd 在时刻 9 之后不再联系.由于顶点 Han 在顶点 9 时刻之前处于非活跃状态，所以顶点 Wu 和顶点 Lyd 一定无法被顶点 Han 激活.

本小节基于 IC 传播模型重新设计出 ICT 传播模型，使得信息可以在时序图中进行传播.现在对信息在时序图中通过 ICT 传播模型的传播过程做详细介绍.

在最初始的网络里面，可以将所有节点 v 的活跃起始时间记为 $Act_v=-1$，表示所有节点都处于非活跃状态.当选取一个种子节点 u 之后其信息传播过程如下：

(1)种子节点 u 的活跃时间 $Act_u=0$，此时种子节点 u 以概率 $p_{u,v}$ 激活其邻居节点 v，且节点 u 有且仅有一次机会可以激活节点 v.

(2)节点 u 在尝试激活节点 v 的时候首先判断 Act_u 是否小于等于 $\max(T_{(u,v)})$，如果大于则直接跳过，开始尝试激活下一个邻居节点，如果小于等于，则节点 u 便以概率 $p_{u,v}$ 激活节点 v.

(3)无论节点 u 是否能够激活节点 v，在以后的回合中 u 都不会再去尝试着激活节点 v.

(4)一旦节点 v 被成功激活，记录其活跃起始时间 $Act_v=t_{(u,v)}$，其中 $t_{(u,v)}\in T_{(u,v)}$，且 $Act_u\leq t_{(u,v)}\leq \max(T_{(u,v)})$.

(5)信息在整个社交网络由新的活跃节点向处于非活跃状态邻居节点尝试传播出去，直到没有新的节点被激活为止.

3.4　时序图影响力最大化问题

以上的两个小节所讲述的是研究时序图影响力最大化问题前期所需要做的一些工作，本小节中将对时序图影响力最大化问题的定义进行说明.

在时序图影响力最大化问题的完整定义得出之前，需要首先对时序图中节点影响力及节点的边际效应进行定义.

定义 4.　节点影响力.节点影响力是在网络中可以被节点 u 成功激活的节点的集合，表示为 $\varphi(u)$.

定义 5.　边际效应.将种子节点集合设置为 S，非种子节点 v 的边际效应为 $infs(v)=\varphi(S\cup v)-\varphi(S)$.

其中以 $|infs(v)|$ 表示节点 v 边际效应的大小.

问题定义. 时序图影响力最大化.给定时序图 $G_T(V,E,T_E)$ 以及特定的传播模型，在时序图中找到一个节点集合 S，其中集合 S 含有节点个数 $|S|=k$，使得集合 S 的影响力 $\varphi(S)$ 最大化.集合 S 也就是 G_T 的种子节点集合.

4　时序图影响力最大化基本算法

在本节中，将给出时序图影响力最大化问题基本实现方式.

基本的时序图影响力最大化算法的思想是：将时序图影响力问题分为两步解决，首先第一步计算节点影响力，然后第二步是根据第一步所得出的实验结果使用贪心算法逐次寻找出边际效应最大的那个节点作为种子节点，直到找出 k 个种子节点为止.

在接下来的小节中首先提出了时序图节点影响力计算算法 SIC 算法以及其改进算法 ISIC 算法，然后在此基础上计算节点的边际效应，并由此提出基本的可以解决时序图影响力最大化问题的基本算法 BIMT 算法.

4.1　时序图节点影响力的计算

由于节点与节点之间都是以概率 $p_{u,v}$ 来激活邻居节点的，节点 u 影响力的数学期望可以表示为 $E[\varphi(u)]$，文献[7]指出计算 $E[\varphi(S)]$ 是一个 #P-hard 问题，为了解决这一问题，Kempe 等人提出了一种使用蒙特卡罗模拟来近似求解 $E[\varphi(S)]$ 的方法，模拟的次数越大也就越接近真实值.同样，本小节也用此方法对单节点的影响力进行计算.

在本小节首先对单节点影响力的计算算法 SIC (Single Node Influence Computation)进行说明，然后基于节点联系时序性这一特性提出一种改进算法 ISIC(Improved SIC).

4.1.1　SIC 算法

算法 SIC 的基本思想是：在时序图中，对于节点 u，将其活跃起始时间 $Act_u=0$，$\varphi(u)=\varnothing$，当其尝试激活一个邻居节点 v，便随机生成一个随机数 $p\in[0,1]$，如果 $p\geq p_{u,v}$ 且 $Act_u\leq\max(T_{(u,v)})$，则表示 v

可以被激活，$\varphi(u)=\varphi(u)\cup v$. 然后信息在激活节点中传播下去，所有被激活节点的集合就是节点 u 的影响力.

算法 1. SIC 算法.

输入：时序图 $G_T(V,E,T_E)$，节点 u，ICT 传播模型

输出：源节点 u 影响力集合 $\varphi(u)$

1. Initialize $\varphi(u)=\varnothing$; $Q.pull(u)$
2. WHILE Q is not empty
3. $u\leftarrow Q.push$
4. FOR each neighbor v of node u
5. $p\leftarrow random\_num$
6. IF $p\geq p_{u,v}$ and $Act_u\leq\max(T_{(u,v)})$
7. $Act_v\leftarrow\min(t|t\in T_{(u,v)}$ and $t\geq Act_u)$ //计算节点的活跃起始时间
8. $Q.pull(v)$
9. $\varphi(u)\leftarrow\varphi(u)\cup v$
10. END IF
11. END FOR
12. END WHILE

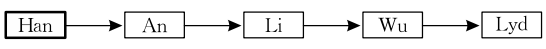
在算法 1 中，第 1 行首先将节点 u 的影响力集合设为空，并将 u 放入队列 Q 中，第 3 行表示将节点 u 从队列中取出，第 4 行表示依次访问邻居节点，第 5～9 行表示随机生成一个数值在 0～1 之间的数进行模拟，在第 6 行检测是否满足 ICT 传播模型条件，如满足则对其邻居节点进行激活并确定其活跃时间(第 7 行)，8～9 行表示节点被成功激活且被放入到队列 Q 中以及集合 $\varphi(u)$ 中去.

4.1.2 SIC 算法的改进

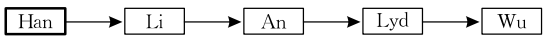
通过分析信息在时序图中的传播过程，可以发现在算法 1 中的 4～7 行有可以被优化的地方. 以图 4 为例，详细说明如何对算法 1 做出优化.

假如信息正通过顶点 Han 进行传播，且 $Act_{Han}=9$，在算法 SIC 中，顶点 Han 需要遍历所有邻居节点，但是通过之前的分析可知节点 Wu 和 Lyd 是无需遍历的，因为在时刻 9 之后它们和顶点 Han 已经不存在联系了.

所以可以对各个节点的邻居节点按照 $\max(T_{(u,v)})$ 进行由大到小的排序，以图 2 中顶点 Han 为例，如图 4 所示.



(a) 排序前



(b) 排序后

图 4 节点 Han 邻居节点

图 4 表示的是活跃顶点 Han 且 $Act_{Han}=9$ 时依次尝试激活邻居节点的顺序，在排序之前，需要对各个顶点依次访问，但是排序之后，只需要访问到顶点 Lyd 便可停止对后续节点的访问，因为包括 Lyd 在内之后的节点在时刻 9 之后都和节点 Han 不存在联系，从而不可能被激活.

基于以上分析本文给出 SIC 的改进算法 ISIC 算法. 两个算法的思想是完全一致的，通过以上分析可知，相比于 SIC 算法，ISIC 算法减少了对邻居节点的遍历.

算法 2. ISIC 算法.

输入：时序图 $G_T(V,E,T_E)$，节点 u，ICT 传播模型

输出：源节点 u 影响力集合 $\varphi(u)$

1. Initialize $\varphi(u)=\varnothing$; $Q.pull(u)$
2. WHILE Q is not empty
3. $u\leftarrow Q.push$
4. FOR each sorted neighbor v of node u
5. IF $Act_u\geq\max(T_{(u,v)})$
6. BREAK;
7. END IF
8. ELSE
9. $p\leftarrow random\_num$
10. IF $p\geq p_{u,v}$ and $Act_u\leq\max(T_{(u,v)})$
11. $Act_v\leftarrow\min(t|t\in T_{(u,v)}$ and $t\geq Act_u)$
12. $Q.pull(v)$
13. $\varphi(u)\leftarrow\varphi(u)\cup v$
14. END ELSE
15. END FOR
16. END WHILE

相比较于算法 SIC，ISIC 算法在第 4 行的循环中多了一个判定条件：第 5～6 行，表示在对邻居节点按照最大联系时间排序后，如果节点的活跃起始时间大于访问到的节点的最大联系时间，则停止对后续节点的访问，从而节省了节点影响力计算的时间. 其余部分和 SIC 算法保持一致.

4.2 种子节点的选取

上一个小节已经完成了时序图影响力最大化问题的第一步，即计算节点的影响力，且在定义 6 中给出了时序图中节点边际效应的准确定义，本小节将根据节点边际效应提出 BIMT 算法以完成第二步中的种子节点的选取.

BIMT 算法思想如下：

以 S 表示种子节点的集合，$\varphi(S)$ 表示种子节点影响节点的集合，在初始阶段 $S=\varnothing$，$\varphi(S)=\varnothing$. 此

时，由于 $S=\varnothing$ 各个节点的边际效应 $infs(u)$ 和它们的影响力 $\varphi(u)$ 相同，由上一小节所得出的各个节点影响力 $\varphi(u)$，选取 $|\varphi(u)|$ 最大的节点作为第一个种子节点，设其为 u_1，则 $S=\{u_1\}$，$\varphi(S)=\varphi(u_1)$. 然后重新计算各个节点的边际效应，$infs(u)=\varphi(S\cup u)-\varphi(S)$，选取重新计算后的边际效应最大的那个节点作为第二个种子节点，依此循环直到找出 k 个种子节点为止.

以表 2 为例描述 BIMT 算法. 首先计算出各个节点的影响力 $\varphi(u)$ 并排序，由表 2 可知由于顶点 Cor 的影响力最大，所以选取 Cor 作为第一个种子节点，即 $S=\{\text{Cor}\}$，所以 $\varphi(S)=\{\text{Cor, Han, Li, Wu, Lyd}\}$. 下一步需要重新计算其它节点边际效应 $infs(u)=\varphi(S\cup u)-\varphi(S)$. 由表 2 可知顶点 Nan 和 Han 的上一轮的边际效应都为 4，现在需要通过寻找在 $\varphi(S)$ 中不包含它们影响节点的个数来重新计算他们的边际效应，可以得到此时其边际效应大小 $|infs(\text{Nan})|=2$，$|infs(\text{Han})|=1$. 由于在 $\varphi(S)$ 中有 5 个节点，顶点 Nan 和 Han 也各有 4 个影响节点，可计算出各自新的边际效应需要 20 次的查询. 并且随着 $\varphi(S)$ 中节点的不断增加，以后的边际效应的计算也就需要更多的查询次数.

表 2　BIMT 算法过程

u	$\varphi(u)$
Cor	{Cor, Han, Li, Wu, Lyd}
Nan	{Nan, Han, An, Wu}
Han	{Han, An, Li, Wu}
…	…
Lyd	{Lyd}

算法 3.　BIMT 算法.

输入：时序图 $G_T(V,E,T_E)$，$\varphi(u)$，k，$S=\varnothing$

输出：种子节点集合 S

1. Initialize $S\leftarrow\varnothing$
2. WHILE ($|S|<k$)
3. FOR each node $u|(u\in G_T, u\notin S)$
4. $|infs(u)|=|\varphi(S\cup v)-\varphi(S)|$
5. END FOR
6. $seed\leftarrow\varnothing$; $|infs(seed)|\leftarrow 0$
7. FOR each node $u|(u\in G_T, u\notin S)$
8. IF $|infs(u)|>|infs(seed)|$
9. $seed\leftarrow u$ //选取边际效应最大的节点
10. END IF
11. END FOR
12. $S\leftarrow S\cup seed$
13. END WHILE

在算法 3 中，第 1 行首先将种子节点的集合设置为空，第 3～4 行表示计算各个非种子节点的边际效应，第 6 行表示设置一个种子 $seed$ 且初始化其边际效应为 0，第 7～11 行表示寻找最大的边际效应节点作为种子节点，第 12 行表示将种子节点并入到种子节点集合中去.

在 BIMT 算法中，计算一个非种子节点 v 的边际效应(3～4 行)的运行时间为 $O(\varphi(v)|V|)$，而由于每找一个种子节点需要对所有的非种子节点进行边际效应的计算，计算完成后寻找到边际效应最大化的节点的运行时间为 $O(|V|)$，所以寻找一个种子节点需要的运行时间为 $O(\varphi(v)|V|^2+|V|)$，因此寻找到 k 个种子节点即 BIMT 算法运行的时间为 $O(\varphi(v)|V|^2k+|V|k)=O(|V|^2k)$.

但当时序图的规模增大时，节点边际效应的运算时间会随着种子节点数量的增加而显著增加，所以 BIMT 算法无法在大规模时序图中高效寻找出 k 个种子节点，接下来章节中，将对 BIMT 算法做出优化并提出两种可以快速解决大规模时序图影响力最大化的算法.

5　对 BIMT 算法的优化

由于 BIMT 算法无法高效解决大规模时序图影响力最大化问题，所以在本节中，首先对节点边际效应的计算做出优化，从而提出了 AIMT 算法，然后根据节点边际效应的子模性这一性质，通过减少某些节点的边际效应的重复计算提出了 IMIT 算法.

5.1　AIMT 算法

上一节详细描述了 BIMT 算法的实现过程，但是通过分析发现在节点的边际效应计算的阶段会消耗大量时间，本小节便是对节点边际效应的计算进行优化从而提出 AIMT 算法.

使各个顶点和数字 1～N 之间建立一一映射，其中 N 的大小和图中节点的个数相同. 顶点 Cor，Han，…，Lyd 分别对应数字 1，2，…，7. 则其影响节点可见表 3 所示. 和上一节中的表 2 所不同的地方还有 $\varphi(S)$ 的表示方式. 首先，将起始的 $\varphi(S)$ 表示为 $\varphi(S)=\{0,0,\cdots,0\}$，其中 0 的个数等于 N. 然后选取 Cor 作为第一个种子节点，即 $S=\{\text{Cor}\}$，则通过读取 Cor 的影响节点 1、3、5、6 和 7，依次将起始 $\varphi(S)$ 中的第 1、3、5、6 和 7 中的 0 赋值为 1、3、5、6 和 7，即 $\varphi(S)=\{1,0,3,0,5,0,7\}$. 此时，为了寻找下一个种子节点，需要对其余各个节点进行新的边际效应的计算. 同样以顶点 Nan 和 Han 为例，节点 Nan

的影响节点为 2、3、4 和 6，现在无需通过到$\varphi(S)$中依次查询来计算顶点 Nan 的新的边际效应，只需要依次判断在$\varphi(S)$中的第 2、3、4 和 6 个元素是否为 0 即可. 如此，只需要 4 次计算即可得到顶点 Nan 新的边际效应，同理也可求得其与节点的新的边际效应. 此算法的优点便是无论$\varphi(S)$中的非 0 个数如何变化，每次节点新边际效应的计算不会变化.

表 3 AIMT 算法过程

u	$\varphi(u)$
Cor→1	{1, 3, 5, 6, 7}
Nan→2	{2, 3, 4, 6}
Han→3	{3, 4, 5, 6}
…	…
Lyd→7	{7}

算法 AIMT 和 BIMT 算法的思想是一样的，所不同的地方是寻找种子节点时各个节点边际效应的计算方式. 通过实验表明即使在大规模的时序图中，以 10 万节点规模的时序图为例，AIMT 算法可以在 15 s 左右的时间寻找出 50 个种子节点，而 BIMT 算法则需要耗费数十分钟.

算法 4. AIMT 算法.

输入：时序图 $G_T(V,E,T_E)$，$\varphi(u)$，k，$S=\varnothing$

输出：种子节点集合 S

1. Initialize $S\leftarrow\varnothing$, $N\leftarrow 0$
2. FOR each node $u|(u\in G_T)$
3. $u\leftarrow N$; $N{+}{+}$
4. END FOR
5. $\varphi(S)\leftarrow\{0,0,\cdots,0\}$
6. WHILE ($|S|<k$)
7. FOR each node $u|(u\in G_T, u\notin S)$
8. $|infs(u)|=|\varphi(S\cup v)-\varphi(S)|$
9. END FOR
10. $seed\leftarrow\varnothing$ and $|infs(seed)|\leftarrow 0$
11. FOR each node $u|(u\in G_T, u\notin S)$
12. IF $|infs(u)|>|infs(seed)|$
13. $seed\leftarrow u$
14. END IF
15. END FOR
16. $S\leftarrow S\cup seed$ //选取边际效应最大的节点为种子节点
17. END WHILE

在算法 4 中，第 1 行首先将种子节点的集合设置为空，第 2～4 行表示对时序图中的每一个节点建立一个和数字之间的一一映射，第 7～9 行表示计算各个非种子节点的边际效应，第 10 行表示设置一个种子 *seed* 且初始化其边际效应为 0，第 11～15 行表示寻找最大的边际效应节点作为种子节点，第 16 行表示将寻找到的种子节点并入到种子节点集合中去.

在 AIMT 算法中，计算一个非种子节点 v 的边际效应的运行时间为 $O(\varphi(v))$，而由于每找一个种子节点需要对所有的非种子节点进行边际效应的计算，计算完成后寻找到边际效应最大化的节点的运行时间为 $O(|V|)$，所以寻找一个种子节点需要的运行时间为 $O(\varphi(v)|V|+|V|)$，因此寻找到 k 个种子节点即 AIMT 算法的运行时间为 $O(\varphi(v)|V|k+|V|k)=O(|V|k)$.

但是通过对 AIMT 算法的进一步研究发现，在种子节点的选取过程中，某些节点的边际效应没有必要在每次循环中都进行计算，基于此，接下来的小节中将继续对 AIMT 算法进行优化，并提出 IMIT 算法.

5.2 IMIT 算法

在相关工作那一节本文介绍了 CELF 算法[4]并指出在寻找种子节点的过程中，各个节点的边际效应大小$|infs(u)|$是满足子模性的，即其边际效应会随着种子节点的增多而递减. 在本小节中，将 CELF 算法和 AIMT 算法相结合，设计出一种新的算法 IMIT(Improved Method for IMTG).

性质 1. 时序图中的影响力最大化函数 $f()$ 是一个单调、子模性函数. 即对于任意的两个集合 S 和 T，满足 $f(S\cup\{w\})-f(S)\geq f(T\cup\{w\})-f(T)$，当 $S\subseteq T$.

若要证明函数 $f()$ 满足子模型，即需要证明 $|infs(w)|-|infs(S)-infs(w)|\geq|infs(w)|-|infs(T)-infs(w)|$.

由于 $S\subseteq T$，首先设有集合 Z，满足 $S\cap Z=\varnothing$ 和 $S\cup Z=T$，则 $|infs(T)-infs(w)|=|infs(S\cup Z)-infs(w)|$，可知 $infs(S\cup Z)\supseteq infs(S)$，得 $|infs(S\cup Z)|>|infs(S)|$，即 $|infs(S)-infs(w)|<|infs(T)-infs(w)|$，由此得证 $|infs(w)|-|infs(S)-infs(w)|\geq|infs(w)|-|infs(T)-infs(w)|$. 所以可知时序图影响力最大化函数 $f()$ 满足子模性.

同样也可以证明函数 $f()$ 是一个单调函数，如果 $f()$ 是一个单调函数，则 $f()$ 满足 $f(S\cup X)\geq f(S)$，此时分三种情况考虑集合 S 和集合 X 的关系 (1) $X\subseteq S$，则 $S\cup X=S$，$f(S\cup X)=f(S)$，(2) $X\not\subset S$，$X\cap S=\varnothing$，则 $f(S\cup X)=f(S)+f(X)$，且 $f(X)\geq 0$，即 $f(S\cup X)\geq f(S)$，(3) $X\not\subset S$，$X\cap S\neq\varnothing$，假设 $X\cap S=$

Z，则 $Z\subset X$，$f(S\cup X)=f(S)+f(X)-f(Z)$，由于 $Z\subset X$ 则 $f(X)-f(Z)>0$，即 $f(S\cup X)>f(S)$.

IMIT 算法思想如下：首先计算所有节点的边际效应大小并排序，选取边际效应最大的节点为种子节点，然后基于边际效应 $|infs(u)|$ 的子模性可知，如果上一轮中的边际效应第二大节点 w 重新计算的边际效应大于上一轮中的边际效应第三大节点 z，则无需对其余的非种子节点重新计算边际效应，而可以直接将 w 选为种子节点即可，否则需要在剩余节点中找到第一个边际效应小于节点 w 边际效应的节点 w_i，然后重新计从节点 z 到节点 w_i 的边际效应，并按照重新计算过后的边际效应大小排序，同时找出在本轮计算中边际效应最大的节点作为下一个种子节点. 由上分析可知 IMIT 算法是可以通过减少对非必要节点边际效应的计算次数来提高算法的运行效率的.

在算法 5 中给出了 IMIT 算法的执行过程，首先在 1～3 行表示计算所有节点的边际效应并排序，且选取排在第一位节点为种子节点，5～7 行是指重新计算上次计算排在第二位节点 u_2 的边际效应大小 $|infs(u_2)|$，如果 $|infs(u_2)|\geq|infs(u_3)|$ 则直接选取节点 u_2 为新的种子节点，否则，首先在剩余节点中找到第一个边际效应小于节点 u_2 边际效应的节点 u_i，然后重新计算从节点 u_3 到节点 u_i 的边际效应(9～11 行)，并按照重新计算过后的边际效应排序，最后选取边际效应大小最大的节点为种子节点(12～14 行).

算法 5.　IMIT 算法.

输入：时序图 $G_T(V,E,T_E)$，$\varphi(u)$，k，$S=\varnothing$

输出：种子节点集合 S

1. Initialize $S=\varnothing$
2. Sort nodes $u|(u\in G_T,u\notin S)$ by $|infs(u)|$
　　//对所有节点按照边际效应大小排序
3. Result of sort：$u_1,u_2,\cdots,u_n$
4. $S\leftarrow S\cup u_1$
5. WHILE $|S|<k$
6. 　$|infs(u_2)|=|\varphi(S\cup u_2)-\varphi(S)|$
7. 　IF $|infs(u_2)|\geq|infs(u_3)|$
8. 　　$S\leftarrow S\cup u_2$
9. 　END IF
10. 　ELSE
11. 　　Find the first node $u_i(|infs(u_i)|<|infs(u_2)|)$ in the rest nodes
12. 　　Re-compute the $|infs(u_m)|,u_m\in\{u_3,\cdots,u_i\}$
13. 　　Sort nodes $u_2,u_3,\cdots,u_i$ by $|infs(u)|$
14. 　　Result of sort：$u_1,u_2,\cdots,u_{n-|S|}$
15. 　　$S\leftarrow S\cup u_1$//选取边际效应最大的节点为种子节点
16. 　END ELSE
17. END WHILE

在 IMIT 算法中，计算一个非种子节点 v 的边际效应的运行时间为 $O(\varphi(v))$，而由于每找一个种子节点需要对所有的非种子节点进行边际效应的计算，计算完成后对所有非种子节点进行排序的运行时间为 $O(|V|\log_2|V|)$，所以寻找一个种子节点需要的运行时间为 $O(\varphi(v)|V|+|V|\log_2|V|)$，因此寻找到 k 个种子节点即 AIMT 算法的运行时间为 $O(\varphi(v)|V|k+|V|\log_2|V|k)=O(|V|\log_2|V|k)$.

从时间复杂度来看 IMIT 算法是高于 AIMT 算法的，但是需要注意，在 IMIT 算法中并不是每次都需要对节点进行排序且也不是每次都需要对所有节点进行影响力计算，所以相比较于 AIMT 算法 IMIT 算法在运算速度上有了大幅度的提升.

6　实验和评估

本文选取了来自现实世界中的四种不同规模大小的真实数据集作为输入数据，实现了在时序图上节点影响力计算和种子节点选取.

6.1　实验数据和参数设置

实验数据集. 本文的实验是在四个真实的数据集上进行的，各个数据集的详细信息见表 4.

表 4　实验数据集信息

数据集	节点数	时序边数	静态边数
Email	1k	330k	25k
ClgMsg	2k	60k	20k
Mathflow	25k	500k	240k
Superuser	120k	530k	290k

其中数据集 1[24] 由在欧洲大型研究机构的电子邮件数据生成，数据集 2[25] 由在加利福尼亚大学欧文分校的在线社交网络上发送的私人消息组成，数据集 3[26] 是堆栈交换网站 Math Overflow 上的时间交互网络所生成的，数据集 4[26] 是由网站堆栈交换网站 Super User 所生成的时序网络图.

实验算法. 本文将时序图影响力最大化问题分为两步解决，第一步为通过 ICT 传播模型计算节点的影响力，提出了两种计算节点影响力的算法：SIC 和 ISIC. 第二步为选取种子节点，首先提出了基本算法 BIMT，但是 BIMT 算法无法高效解决大规模时序图影响力最大化问题，所以通过对节点边际效

应的优化提出了 IMIT 算法，以此来解决大规模时序图影响力最大化问题，最后由边际效应的子模性对 IMIT 算法进行优化提出了 AIMT 算法.

在第二步的种子节点选取过程中，除了本文实现的上述三种算法外，还对现有的一些可以快速选取种子节点的算法进行复现，在算法的运行时间和激活节点覆盖率两方面来对比分析各算法的优劣.

(1) Random. 作为一个基准比较方法，简单的从时序图中随机选取 k 个非重复节点作为种子节点.

(2) Degree. 作为一个简单的启发式方法，用来做实验对比，将节点按照出度数(out-degree)大小进行排序，选取前 k 个节点作为种子节点.

(3) DegreeSingle. 首先选取出度数(out-degree)最大的节点作为种子节点，然后判断各个非种子节点的邻居(out neighbors)是否含有种子节点，如果含有将出度数减 1，最后选取度数最大的节点为下一个种子节点，依次类推，直到选取 k 个种子节点.

(4) DegreeDiscount. 在文献[27]所提出的一种基于节点度数的算法.

实验环境. 操作系统是 Ubuntu(Linux)16.10，CPU 为 i3@3.30 GHz，内存 4 GB，硬盘 500 GB，编程环境 GUN C++.

参数设置. 在种子节点选取 BIMT，IMIT 和 AIMT 三种算法以及复现的四种算法中，选取种子节点集合大小 k 分别为 1，10，20，30，40 和 50.

6.2 节点影响力的计算时间

首先确定传播模型为 ICT 传播模型，为了验证 ISIC 算法相比于 SIC 算法更节省时间，在对网络中各节点计算其 NodeRank 值后由式(4)计算各个节点间的传播概率.

本实验如图 5 所示，表示在四种真实数据集中算法 SIC 和 ISIC 所需要的运行时间.

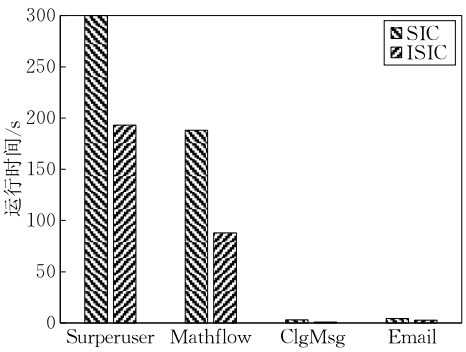


图 5 节点影响力计算时间

可以看出，相比较于 SIC 算法，算法 ISIC 在四种真实数据集中所需的运行时间都明显小于 SIC 的运行时间，实验数据表明 ISIC 算法比 SIC 算法节省 37%～57%的运行时间.

6.3 种子节点影响力变化趋势

本小节主要研究种子节点影响力大小随着种子节点规模逐渐增大的过程中的变化趋势. 为了使得实验更具有说服力，在实验的过程中对节点间的传播概率做出相应的调整，依次完成对比实验(为控制实验中变量，此次实验中的数据统一通过 BIMT 算法计算).

本实验如图 6 和图 7 所示，表示在四种真实数据集上规模大小不同的种子节点集合的影响力大小的变化趋势. 图 6 和图 7 所表示的实验不同之处在与节点之间的传播概率的大小. 图 6 中的节点间传播概率是在求出节点的 NodeRank 的基础上求取各有联系节点间的传播概率(设此时在网络中的传播概率集合为 P_1)，图 7 中的节点间传播概率是在图 6 的基础上将所求得的节点传播概率增大(变为 2 倍，设此时传播概率集合为 P_2)，依次对比在两种不同的传播概率下的种子节点影响力大小的变化趋势. 且在表 5 中给出在两种概率集合下，种子节点集合取不同规模($k=30,50$)下的各个数据集中的活跃节点个数.

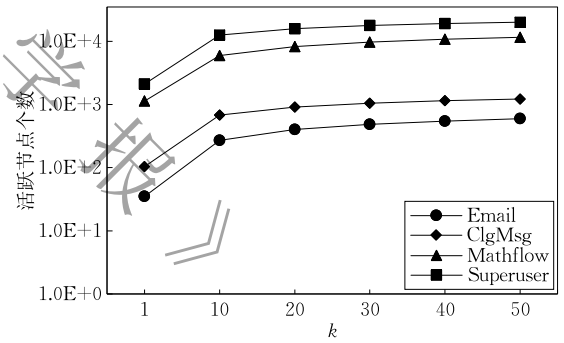


图 6 种子节点影响力变化趋势(P_1)

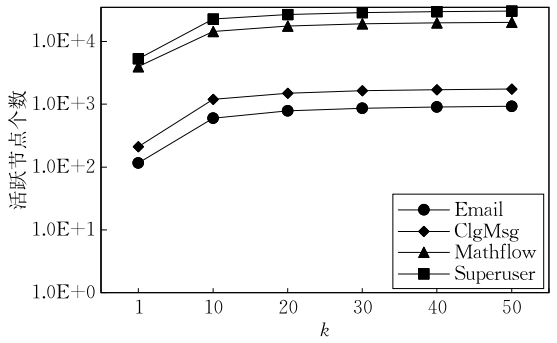


图 7 种子节点影响力变化趋势(P_2)

对比图 6 和图 7 可以看出，在 k 取值的各个阶段，图 7 中的种子节点集合的影响力都是大于图 5 的，这是因为节点间传播概率越大也就越容易激活

对方，所以在相同大小规模的种子节点集合中，传播概率整体上较大的一方也就会影响更多的节点（由表 5 可以看出）.

表 5　种子节点影响力大小

数据集	种子个数	P_1	P_2
Email	30	487	864
	50	596	930
ClgMsg	30	1050	1636
	50	1222	1749
Mathflow	30	9700	19056
	50	11587	20160
Superuser	30	17784	28947
	50	20105	30507

在表 5 中可知，在同一个数据集中，当种子节点个数相同时，在概率集合为 P_2 的情况下，网络中的活跃节点个数是明显多于 P_1 的. 同时，也能看出，相比较于图 7，图 6 中的折线在后期更趋向于水平，这说明整体概率较大的一方种子节点集合的影响力也就更容易趋于饱和.

6.4　各算法下的活跃节点覆盖率

本实验表示的是在 ICT 传播模型下，分别对 BIMT，IMIT，AIMT 三种算法以及复现的四种算法（Random，DegreeSingle，Degree 及 Degree Discount）在种子节点选取阶段活跃节点覆盖率的实验. 对于 Random 算法，考虑到其结果的随机性过大，特别是在大规模时序图上，在本次实验中对 Random 算法一共进行了 500 次实验，实验数据取其平均值.

在图 8 所表示的是各个算法在种子节点选取的各个阶段（$k=10,20,30,40,50$），活跃节点在四个数据集中的覆盖率.

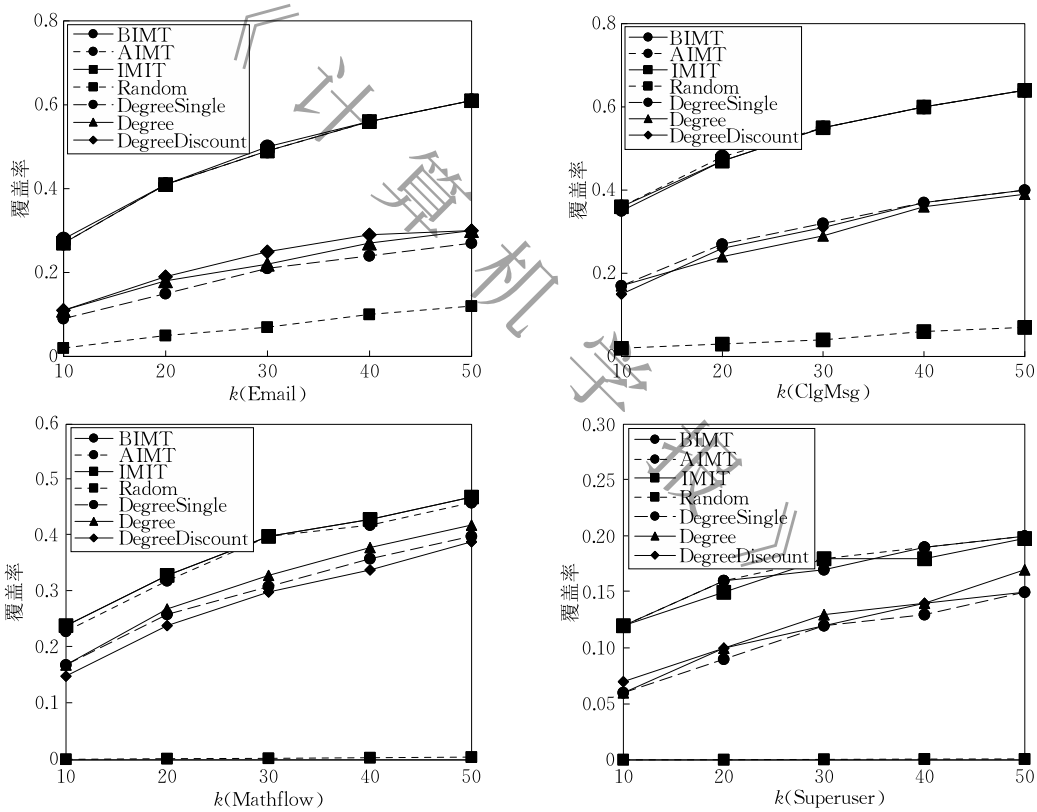


图 8　各算法在不同数据集中所求的活跃节点覆盖率

首先，可以明显地看出，在图中所示的算法中，Random 算法在四种数据集中的效果都是最差的，特别是当数据集越大时其效果也就越差. 这是因为，在正常的数据集中都存在着相当一部分的孤立节点（节点的度数为 0），随机算法在种子节点的选取过程中难免会选取到孤立节点. 通过实验统计发现，由数据集 1 到数据集 4 中，孤立节点所占比例分别为 16.43%、28.86%、33.65% 和 30.0%，在 Random 算法中，当种子节点规模增长到 50 时，在四种数据集中的活跃节点覆盖率为 12%、7%、0.5% 和 0.09%，和其余的六种算法相比，其效果是很难令人满意的.

对于三种基于节点度数的启发式算法：Degree，DegreeSingle 和 DegreeDiscount，由图 8 中的四个小图可以看出，其效果和本文所提出的三个算法（BIMT，AIMT 及 IMIT）相比，在小规模时序图上（数据集 1

和数据集 2)效果相差很大,而随着数据集规模的增大其效果也就越好.当数据集的规模增加到十万个以上节点(数据集 4)时,三种启发式算法和本文的三种算法效果已经比较接近,但是还是有一定的差距.以数据集 4 为例,当种子节点规模大小为 50 时,BIMT,AIMT 及 IMIT 三种算法选取的种子集合可以激活 20100(取平均值)个节点,而 Degree,DegreeSingle 和 DegreeDiscount 三种算法分别可激活 15503、15804 和 15510 个节点,平均 15605,和 BIMT,AIMT 及 IMIT 三种算法的平均值相差接近 4500.

三种启发式算法的不足之处在于,Degree,DegreeSingle 和 DegreeDiscount 三种算法只是简单的以节点的度数为基础来选取种子节点,而忽略了节点间会有重复邻居的这一因素.比如,有两个度数很大的节点 u 和 v,但是两个节点有很多共同的邻居节点,而一旦选取其中一个节点 u 为种子节点,另一个节点的边际效应会明显变小,但是由于三种算法在这一因素上都未对此做出相应的措施,所以在下一轮种子节点的选取中节点 v 依然很有可能会被选取为种子节点.而在 BIMT,AIMT 及 IMIT 三种算法中,这种可能性会减小很多.

6.5 算法在种子选取阶段运行时间

本实验是在 6.4 小节中实验的基础上,在 ICT 传播模型下,分别对 BIMT,IMIT,AIMT 三种算法以及复现的四种算法(Random,DegreeSingle,Degree 及 DegreeDiscount)在种子节点选取阶段运行时间的统计.在本实验中,所统计的时间为选取种子节点规模为 50 时各算法的运行时间.

对于 Random 算法,考虑到其结果的随机性过大,特别是在大规模时序图上,对于随机产生的一次种子节点集合,其实验结果没有任何代表性可言,所以在本次实验中对 Random 算法进行 500 次实验,图 9 中所表示的 Random 算法的运行时间为其 500 次实验的总和.

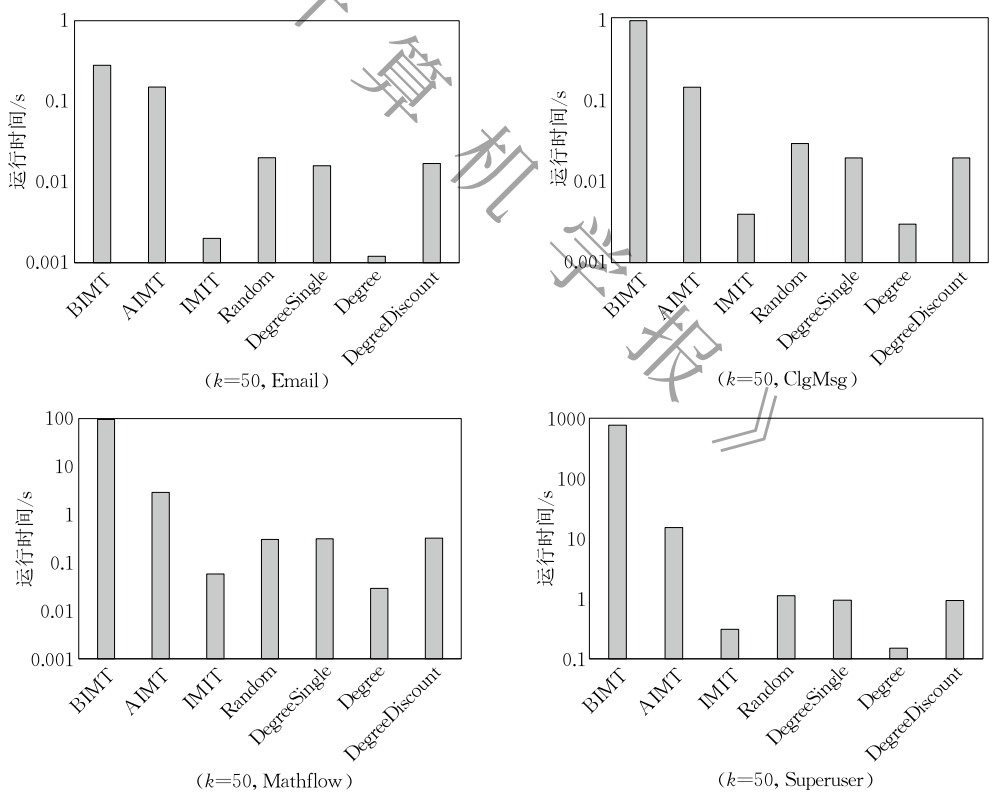


图 9 各算法在不同数据集中寻找种子集合时间

各算法在不同数据集中寻找 50 个种子节点所需时间结果如图 9 所示.通过比较可以看出,BIMT 算法在四个数据集上的运行时间最长,在本文所提出的三种算法上,IMIT 算法相比较于另外两种算法(BIMT 算法和 AIMT 算法)在运算速度上有着非常明显的优势.且如果忽略 Random 算法在此次实验中的运行次数,那么其选取种子节点所需时间应该是最短的.而忽略 Random 算法,在其余的六种算法中,可以看出,Degree 算法在四种数据集中的运行时间都是最短的,这是因为其选取种子节点的方式在这六种算法中最为简单,只是单纯的选取在数据集中节点度数的前 50 个节点即可,实验中只需

对节点按照其度数大小进行排序即可.

6.6　不同算法中的种子节点的影响力

在本实验中将对不同大小种子节点集合在三种寻找种子节点算法 BIMT、AIMT 和 IMIT 的影响力大小进行实验.

本实验的目的是为了验证和 BIMT 算法相比较，AIMT 和 IMIT 算法不仅运行时间更快，并且也可以得到和 BIMT 相同的结果.

数据集 1 实验分析

图 10 表示在数据集 1(Email)中的实验，数据集 1 的节点规模较小，只有 1k 个左右的节点. 通过三种算法(BIMT，AIMT 和 IMIT)找到的规模大小从 1 到 50 的种子集合的影响力大小. 其中横坐标表示种子节点集合规模大小，纵坐标表示时序图中节点被种子节点影响到的个数.

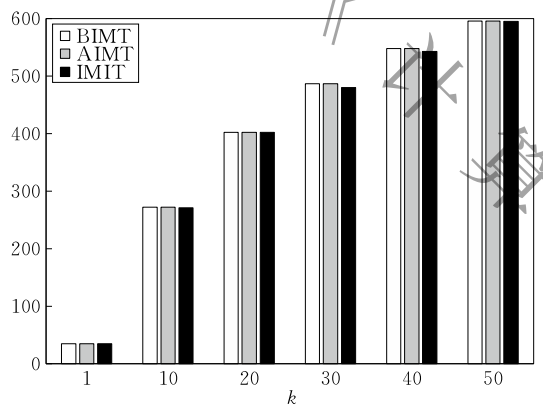


图 10　三种算法下的不同种子集合的影响力(Email)

通过图 10 可以看出，由于数据集 1 的规模较小，所以随着种子节点个数的增加，其影响力的变化趋势也就更容易趋于平稳，同样可以看出在三种算法中，各个规模的种子节点集合影响力是相差无几的，其中 BIMT 和 AIMT 算法所得出的结果是完全相同的，因为其两者的计算策略完全相同.

数据集 2 实验分析

图 11 表示在数据集 2(ClgMsg)中的实验，数据集 2 的节点规模只是略大于数据集 1，只有 2k 个节点，但是需要注意的是，虽然其节点规模大于数据集 1，但其边的规模却是小于数据集 1 的，所以相对数据集 1 其显得比较稀疏，这也就意味着其种子节点影响力在增大的过程中不会像图 12 中表示的会很快趋于平稳. 因为图越是稠密也就意味着节点有更多的邻居节点，邻居节点越多也就意味着越容易被激活. 所以由图 11 可以看出，即使当种子节点大于 10 的时候还可以看出随着种子节点集合变大时其影响力的变化趋势是很明显的.

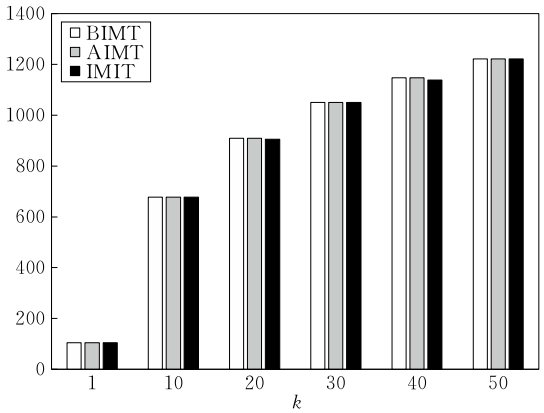


图 11　三种算法下的不同种子集合的影响力(ClgMsg)

然后通过三种算法(BIMT，AIMT 和 IMIT)找到的在数据集 2 上的规模大小从 1 到 50 的种子集合的影响力大小. 其中横坐标表示种子节点集合规模大小，纵坐标表示时序图中节点被种子节点影响到的个数. 由图 11 可以看出在三种算法中，各个规模的种子节点集合影响力几乎是相同的.

数据集 3 实验分析

图 12 表示在数据集 3(Mathflow)中的实验，数据集 3 的节点规模较大相比较于数据集 1 和数据集 2. 但是可以发现在数据集 3 中其种子节点影响力在其规模大小大于 10 的时候几乎趋于平稳. 这是因为数据集 3 是很稠密的，其有 25k 个节点以及 500k 条边，这也就使得节点很容易被激活，所以种子节点的影响力在前期增长速度很快而容易趋于稳定. 同样也可以由图 12 清楚地看出三种算法在不同规模大小的种子集合中所得出的结果也是相同的.

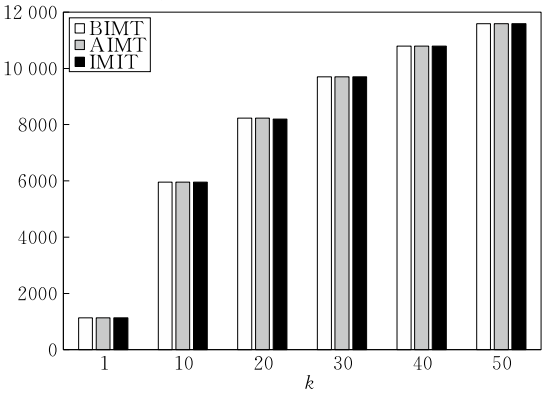


图 12　三种算法下的不同种子集合的影响力(Mathflow)

数据集 4 实验分析

图 13 表示在数据集 4(Superuser)中实验. 数据集 4 不管是从节点的规模(120k)还是从边的规模(290k)来看在四种数据集中都是最大的. 但是其种子节点影响力的增长趋势和数据 2 是很相似的. 这

是因为两者的数据疏密度是类似的，而由图 10 和图 12 可以看出，数据集 1 和数据集 3 的种子节点影响力的增长趋势是相似的，这是因为在四种数据集中，这两种数据集都是相对比较稠密的.

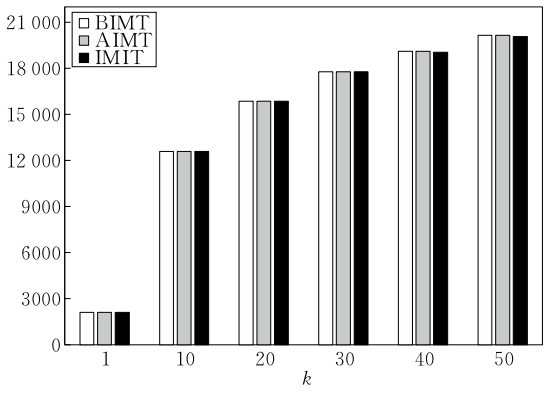


图 13 三种算法下的不同种子集合的影响力(Superuser)

所以，从以上四个图(图 10、图 11、图 12 和图 13)中可以看出，本文所给出的三种用来寻找种子节点的算法(BIMT，AIMT 和 IMIT 算法)在不同规模的大小的种子节点集合中所得出的结果都是相同的. 如此，从实验数据中便直观地证明了本文所提出的两种优化算法的准确性和高效性.

7 结 论

本文研究了时序图影响力最大化问题，即在时序图上寻找 k 个节点可以使得信息最大化传播. 首先对 IC 传播模型进行改进使得其可以应用在时序图上，在此基础上提出了一种基本的时序图影响力最大化算法 BIMT，并在 BIMT 的基础上提出了更高效的 AIMT 算法，最后由节点边际效应的子模性这一性质对 AIMT 算法进行优化提出了 IMIT 算法. 实验结果表明，AIMT 和 IMIT 两种算法可以快速、高效地解决大规模时序图影响力最大化问题.

在未来的工作中将会做如下深入的研究：(1) 基于信息类型的时序社交网络影响力最大化问题的研究，即考虑不同的信息类型对节点间传播概率的影响；(2) 近年来有很多研究者在社区影响力最大化方面取得很大成果，而在时序图上的社区发现这一领域却鲜有人研究，未来可以在时序图的基础上来研究基于时序图的社区影响力大小.

参 考 文 献

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

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

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

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

[5] Cohen E, Delling D, Pajor T, et al. Sketch-based influence maximization and computation: Scaling up with guarantees//Proceedings of the 23rd Conference on Information and Knowledge Management. Shanghai, China. 2014: 629-638

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

[7] 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

[8] Tang Y, Shi Y, Xiao X. Influence maximization in near-linear time: A martingale approach//Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. Melbourne, Australia. 2015: 1539-1554

[9] Borgs C, Brautbar M, Chayes J, et al. Maximizing social influence in nearly optimal time//Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms. Portland, USA, 2014: 946-957

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

[11] Ye Qi, Zhu Changlei, Li Gang, et al. Using node identifiers and community prior for graph-based classification. Data Science and Engineering, 2018, 3(1): 68-83

[12] Li J, Wang X, Deng K, et al. Most influential community search over large social networks//Proceedings of the International Conference on Data Engineering. San Diego, USA, 2017: 871-882

[13] Li Y, Fan J, Zhang D, et al. Discovering your selling points: Personalized social influential tags exploration//Proceedings of the Special Interest Group on Management of Data. Chicago, USA, 2017: 619-634

[14] Li J, Sellis T, Culpepper J S, et al. Geo-social influence spanning maximization. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(8): 1653-1666

[15] Kim D, Hyeon D, Oh J, et al. Influence maximization based on reachability sketches in dynamic graphs. Information Sciences, 2017, 394-395: 217-231

[16] Wang Yanhao, Fan Qi, Li Yuchen, et al. Real-time influence maximization on dynamic social streams//Proceedings of the VLDB Endow. Munich, Germany, 2017: 805-816

[17] Przytycka T M, Singh M, et al. Toward the dynamic interactome: It's about time. Briefings in Bioinformatics. 2010, 11(1): 15-29

[18] Han J D, Bertin N, Hao T, et al. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. Nature, 2004, 430(6995): 88-93

[19] Lèbre S, Becq J, Devaux F, et al. Statistical inference of the time-varying structure of gene-regulation networks. BMC Systems Biology, 2010, 4(1): 130-145

[20] Wu H, Huang Y, Cheng J, et al. Efficient processing of reachability and time-based path queries in a temporal graph// Proceedings of the 32nd International Conference on Data Engineering. Helsinki, Finland, 2016: 145-156

[21] Abdullatif A, Masulli F, Rovetta S. Tracking time evolving data streams for short-term traffic forecasting. Data Science and Engineering, 2017, 2(3): 210-223

[22] Turunen E. Using GUHA data mining method in analyzing road traffic accidents occurred in the years 2004—2008 in Finland. Data Science and Engineering, 2017, 2(3): 224-231

[23] Gong Xiu-Wen, Zhang Pei-Yun. Research on propagation model and algorithms for influence maximization in social network based on PageRank. Computer Science, 2013, 40(6A): 136-140(in Chinese)
(宫秀文，张佩云. 基于 PageRank 的社交网络影响力最大化传播模型与算法研究. 计算机科学，2013，40(6A)：136-140)

[24] Paranjape A, Benson A R, Leskovec J. Motifs in temporal networks//Proceedings of the 10th ACM International Conference on Web Search and Data Mining. Cambridge, UK, 2017: 1612-1621

[25] Panzarasa P, Opsahl T, Carley K M. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. Journal of the American Society for Information Science and Technology, 2009, 60(5): 911-932

[26] Yasseri T, Sumi R, Kertész J. Circadian patterns of Wikipedia editorial activity: A demographic analysis. PLoS One, 2012, 7(1): 1-8

[27] Chen Wei, Wang Yajun, et al. Efficient influence maximization in social networks//Proceedings of the International Conference on Knowledge Discovery and Data Mining. Paris, France, 2009: 199-2008

WU An-Biao, Ph. D. candidate. His main research interests include influence maximization problem, temporal graph and graph neural network.

YUAN Ye, Ph. D., professor, Ph. D. supervisor. His main research interests include cloud computing, big data management (including graph data management, uncertain data management, data privacy protection) and P2P computing.

QIAO Bai-You, Ph. D., associate professor. Her main research interests include graph data management and uncertain data management.

WANG Yi-Shu, Ph. D. candidate. Her main research interests include graph data management and uncertain data management.

MA Yu-Liang, Ph. D. candidate. His research interests include social networks, graph data management.

WANG Guo-Ren, Ph. D., professor. His main research interests include uncertain data management.

Background

Influence maximization problem has been widely used in social networks; in general, social networks can be denoted as static graphs, Influence maximization problem means to find top-k influential nodes in graph that maximize the dissemination of information. This problem was first proposed by Domingos P and Richardson M in 2001 for the research of viral marketing.

Recently, the researches of influence maximization are mainly based on static graphs. Therefore, we studied the influence maximization problem on temporal graph, that is, to find top-k influential nodes in temporal graph that maximize the number of influence nodes.

In this paper, firstly, we improve the IC propagation model and propose ICT propagation model to make information spread on the temporal graph based on ICT propagation model. Besides the new propagation model, we also propose a new method based on PageRank algorithm to compute the propagation probability. We improve the PageRank algorithm

by taking the node's communication frequency into account, because the higher connection frequency between two nodes, the closer they are, which means they are more influenced by each other. And then, we solve the influence maximization problem on temporal graph by dividing the problem into two steps. We study the computation of single node's influence in first step. We implement the influence maximization problem on temporal graph by using the result of the first step in second step, and propose a basic method BIMT (Basic Method for IMTG). However, the BIMT algorithm will cost plenty of time when deal with large-scale temporal graph, so we propose an effective algorithm AIMT (Advanced Method for IMTG) by optimizing the calculation of node's marginal effect. We improve the AIMT algorithm by the nature of submodular of node's marginal effect and propose a more effective algorithm IMIT (Improved method for IMTG) to avoid multiple computation of the marginal effects of certain nodes.

In the further research of the temporal graph influence maximization problem, we will focus on the relationship between the type of information and propagation probability, because users in the social networks are only interested in the information that they are interested.

This paper is supported by the National Key R&D Program of China (2016YFC1401900), the National Natural Science Foundation of China (61572119, 61622202, U1401256, 61732003, 61729021), the Fundamental Research Funds for the Central Universities (N150402005).