大规模时序图中种子节点挖掘算法研究

邹晓红[1,2]，许成伟[1]，陈晶[1,2]，宋彪[1]，王明月[1]

（1. 燕山大学信息科学与工程学院，河北 秦皇岛 066004；
2. 河北省计算机虚拟技术与系统集成重点实验室，河北 秦皇岛 066004）

摘 要：针对现有基于时序图的影响力最大化算法多因时间效率低或影响范围窄，不适用于大规模网络的问题，提出了一种融合启发式算法和贪心策略的种子节点挖掘算法（CHG）。首先，基于时序图中信息传播的时序性，给出了节点二阶度概念，并以此对节点影响力进行启发式评估；其次，根据影响力评估结果对节点进行初步过滤筛选，构建候选种子节点集；最后，通过计算候选种子节点的边际效应，解决节点间影响范围重叠问题，保证获取最优种子节点组合。在 3 个不同规模的时序网络数据集上进行了实验，实验结果表明，所提算法在相对较短的运行时间下，仍能够保证所得种子节点集具有较高的网络全局影响力，在时间效率与种子节点集影响范围 2 个方面取得了更好的平衡。

关键词：时序图；影响力最大化；种子节点挖掘；信息传播；边际效应

中图分类号：TP399

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2022170

Research on seed node mining algorithm in large-scale temporal graph

ZOU Xiaohong[1,2], XU Chengwei[1], CHEN Jing[1,2], SONG Biao[1], WANG Mingyue[1]

1. School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China
2. Hebei Key Laboratory of Computer Virtual Technology and System Integration, Qinhuangdao 066004, China

Abstract: Most of the existing maximizing influence algorithms based on temporal graph were not applicable for large-scale networks due to the low time efficiency or narrow influence range. Therefore, the seed node mining algorithm named CHG combining heuristic algorithm and greedy strategy was proposed. Firstly, based on the time sequence characteristics of information diffusion in temporal graph, the concept of two-order degree of nodes was given, and the influence of nodes was heuristically evaluated. Secondly, the nodes were filtered according to the influence evaluation results, and the candidate seed node set was constructed. Finally, the marginal effect of candidate seed nodes was calculated to solve the overlap of influence ranges between nodes to ensure the optimal combination of seed nodes. The experiments were carried out on three different scale data sets, and the results show that the proposed algorithm can ensure the high influence of the seed node set even though its running time is relatively shorter. And it can achieve a better trade-off between the time efficiency and the influence range of the seed node set.

Keywords: temporal graph, influence maximization, seed node mining, information diffusion, marginal effect

---

收稿日期：2022−07−08；**投稿日期：**2022−08−29

基金项目：国家自然科学基金资助项目（No.62172352, No.61871465）；河北省自然科学基金资助项目（No.F2019203157）；河北省高等学校科学技术研究资助项目（No.ZD2019004）；河北省创新能力提升计划项目（No.22567626H）

Foundation Items: The National Natural Science Foundation of China (No.62172352, No.61871465), The Natural Science Foundation of Hebei Province (No.F2019203157), Science and Technology Research Project of Hebei Province Higher Education (No.ZD2019004), Innovation Capability Improvement Plan Project of Hebei Province (No.22567626H)

0 引言

近些年，随着信息技术的快速发展，在线社交媒体层出不穷，用户通过社交媒体相互沟通，分享观点，并借助社交网络完成信息的快速传播，以此来影响更多的用户，这种影响力的扩散与传播成为社交网络研究领域热点问题之一。Domingos 和 Richardsom[1-2]首次提出有关社交网络影响力最大化（IM, influence maximization）问题，IM 问题是指在网络图中选定 k 个种子节点作为信息源点，使其在特定信息传播模型下进行信息传播，目的是以最小的代价尽可能多地影响其他节点，其研究成果广泛应用于市场营销、水质检测等方面[3]。

目前，大多数 IM 问题是基于静态图网络模型展开的，但针对具有动态性特征的社交网络，不能简单地将其表示为静态图网络模型进行处理，如生物信息网络、通信网络、道路交通网络等网络结构[4-6]，节点间只在某一时间或某段时间发生交互作用，存在一定的时间关系特性且具有明显的交互顺序，具有这种特征的网络被称为时序网络[7-8]，一般将其抽象为时序图进行表示。时序网络模型涵盖了节点间联系的时间属性信息，是对现实社交情况的真实反映，以其为对象进行 IM 问题的研究具有现实意义。

现有基于时序网络的影响力最大化算法较少，文献[9]中所给出的 IMIT（improved method for the influence maximization problem on temporal graph）算法需要采用数万次蒙特卡罗模拟方法计算单一节点影响力，因此耗费了大量的运行时间，虽然最终所得种子节点集影响力较高，但算法时间效率过低而不能高效地完成大规模网络中种子节点的挖掘。与文献[9]相比，文献[10]中所给出的种子节点挖掘算法在运行时间上至少降低了一个数量级，但面对大规模网络数据时，该算法无法保证种子节点的质量，会出现种子节点集影响范围损失较大的情况。算法时间效率是处理大规模数据首要考虑的问题，脱离算法精度只谈时间效率也不符合实际，如何做到种子节点集影响范围与算法时间效率之间的平衡，从而高效地完成大规模时序网络中种子节点挖掘这一问题尚待解决。鉴于此，本文提出了一种将启发式算法和贪心策略相融合的种子节点挖掘算法，该算法集中了启发式算法速度快与贪心策略精度高这 2 个优势，首先，结合时序图中信息传播的时序特性，对单一节点影响力进行启发式评估，从而筛选出一定数量有影响力的候选节点；其次，基于贪心策略，通过计算候选节点的边际效应，避免由于节点间影响范围重叠导致种子节点集影响力损失较大的情况发生；最后，力求在时间效率和影响范围 2 个方面均取得良好效果。

本文主要贡献如下。

1) 结合时序图中节点间联系次数这一因素，在传统方法的基础上，提出了一种新的时序图节点间传播概率计算方法，使之更加符合真实社交情况。

2) 给出了一种基于时序图的节点影响力评估方法，并根据节点影响力的评估结果，对节点进行初步筛选，极大地缩小了节点边际效应的计算范围，降低了算法时间复杂度。

3) 在贪心式选取种子节点阶段，进一步优化了候选种子节点边际效应的计算方法以及利用边际效应选取种子节点的策略。

4) 在 3 个真实的时序网络数据集上进行了实验，实验结果表明，所提 CHG（combining heuristic algorithm and greedy strategy）算法在保证较高影响力的同时，具有相对较低的时间消耗，具备一定的准确性、高效性和可扩展性。

1 相关工作

影响力最大化问题自提出以来，就受到国内外学者的广泛关注。Kempe 等[11]首先将影响力最大化定义为在特定信息传播模型下挖掘影响力最大的 k 个节点的离散优化问题，并证明其是一个 NP 难问题，同时给出了经典的贪心算法，可以达到近似 63%的最优解。针对贪心算法运行时间过高的问题，Leskovec 等[12]利用边际效应子模函数的特性提出了 CELF（cost-effective lazy forward selection）算法，实验结果显示该算法比贪心算法的效率提高了近 700 倍。Chen 等[13]针对传统度估计重叠问题，提出了启发式算法 DegreeDiscount，实验结果表明该算法在时间效率上得到了有效的提高。之后，影响力最大化问题被进一步研究，Bagheri 等[14]考虑社交网络拓扑结构的特点，利用社区检测算法对网络进行划分，并根据社区的结构确定每个社区的影响节点配额。王帅等[15]考虑到外界因素对网络系统的影响，如网络结构损坏导致网络连通性被破坏或信息传播过程中受到干扰，从而定义了稳健影响力最大化问题，其目的是寻找具有稳健信息传播能力的种子节点。Tong 等[16]考虑到信息传播的时间代价，在

种子选择的每个步骤均加入时间因素，通过在种子选择的每个阶段施加预算限制，在给定的时间范围内实现影响力最大化的目标。

以上 IM 算法均是基于静态图模型提出的，基于动态图的 IM 问题也受到众多学者的广泛讨论。Yang 等[17]提出通过时间快照将动态图转换成多个按时间片离散分布的静态图进行处理，从而完成时态子图（temporal subgraph）的挖掘。Sheng 等[18]针对现有基于网络拓扑结构数据存在高维、低效率的局限性，通过网络表征学习将网络中的每个节点转换为低维向量表示，然后在低维潜在空间中解决动态影响力最大化问题。针对动态图中节点的添加或删除操作，Wang 等[19]提出一种滑动窗口模型，窗口随着时间向下滑动来探测节点的更新，以此完成实时的种子节点挑选，该算法主要用于解决动态更新的网络中种子节点的挖掘问题。吴安彪等[9]首次以时序图为研究对象，从网络全局角度出发对基于时序图的影响力最大化问题展开研究，并给出了基于时序图的影响力最大化算法 IMIT，该算法首先利用蒙特卡罗模拟方法计算单个节点的影响力，然后选定影响力最大的节点为第一个种子节点，最后通过计算网络中剩余节点边际效应，并利用边际效应计算过程中所具有的子模性，得到最终的种子节点集，实验结果显示该算法所挖掘出的种子节点集具有极高的影响范围，但是也付出了较高的时间代价，不能高效地完成大规模网络中种子节点的挖掘工作。在此基础上，陈晶等[10]更注重算法的时间效率，给出了时序社交网络两阶段影响力最大化（TIM, two-stage impact maximization）算法，该算法考虑时序图中节点间联系次数对节点重要性的影响，提出以节点的活跃程度来衡量节点影响力，并从节点间联系次数最多的前 100 个节点中确定最终的种子节点集，实验表明该算法在时间效率上具有一定优势，但面对大规模网络数据时，精度略显不足。

2　基本概念

2.1　时序图

定义 1　时序图（temporal graph）[20-21]。给定网络 $G_{\mathrm{T}}(V,E,T_{\mathrm{E}})$ 表示节点间具有时序关系的社交网络有向时序图[18]。其中，V 表示节点的集合，E 表示边的集合，T_{E} 表示图中所有节点之间存在联系时刻的集合，$T_{(u,v)}$表示节点 u 和 v 之间存在联系时刻的集合，$T_{(u,v)}\in T_{\mathrm{E}}$，$u,v\in V$。

时序图是表示在边上带有时间戳属性的动态有向图，且节点间的边会随着边上时间戳的到来而被激活，它表示两节点在此刻存在联系。图 1 为一个简单的时序图，其中边上的数字表示时间戳。以节点 A、B 为例，$T_{(A,B)}$={1,3,5}表示节点 A 分别在 1,3,5 时刻与节点 B 存在联系并以传播概率 $P_{(\mathrm{A,B})}$尝试将其激活。

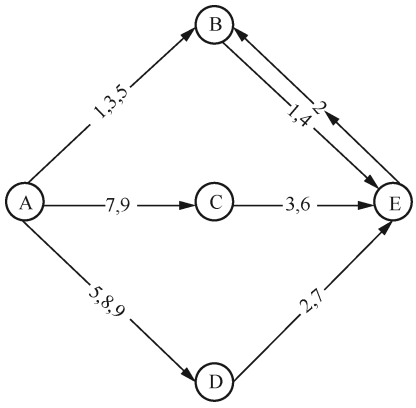

图 1　时序图

时序图与普通静态图相比，最大的不同是节点之间的联系存在时间上的先后顺序，即信息传播具有时序性。例如，图 1 中节点 A 与节点 C 分别在 7,9 时刻联系，节点 C 与节点 E 已经在更早的 3,6 时刻进行联系，在这种情况下，节点 A 便不能把信息通过节点 C 传递到节点 E。

2.2　节点间传播概率及信息传播模型

社交网络中节点间传播概率与信息传播模型是研究 IM 问题最基本的 2 个要素。

2.2.1　时序图节点间传播概率

定义 2　传播概率。活跃节点 u 通过有向边 $E_{(u,v)}$ 成功激活其邻居节点 v 的概率称为节点间的传播概率，表示为 $P_{(u,v)}\in[0,1]$。

传统计算节点间传播概率有以下 2 种方法。1) 随机赋值方法，例如，设定概率集合 P={0.01,0.03, 0.05,0.1}，从集合 P 中随机选取概率值作为各节点间的传播概率，此方法的弊端是所求取的传播概率并不符合实际情况。2) 度估计方法，即用节点入度（In-Degree）的倒数来估计该节点被上一级节点激活的概率，如式(1)所示。

$$P_{(u,v)}=\frac{1}{\text{In-Degree}(v)} \tag{1}$$

时序图中记录了节点间的联系次数，节点与其邻居节点间联系次数越多，激活概率越大，可见节

点的所有入边并非等值权重，故该方法并不适用于时序图中节点间传播概率的计算。除父节点对子节点的主动激活次数外，子节点对父节点的反向作用也会影响两节点间传播概率的大小，例如，图 1 中节点 B 与节点 E 产生了交互，意味着两节点的亲密度进一步增大，表现为子节点更易受父节点的影响，所以相比于节点 C、D，节点 B 对节点 E 的激活概率更大，因此结合时序图中节点间联系的时序信息，对传统度估计计算传播概率的方法进行改进，将节点间联系次数 $\left|T_{(u,v)}\right|+\left|T_{(v,u)}\right|$ 设定为时序边 (u,v)的权重，则时序边(u,v)占节点 v 所有入边的比重为 $\dfrac{\left|T_{(u,v)}\right|+\left|T_{(v,u)}\right|}{\sum\limits_{v'\in \mathrm{In}(v)}\left(\left|T_{(v',v)}\right|+\left|T_{(v,v')}\right|\right)}$，则节点 v 被节点 u 激活的概率可以通过式(2)进行计算。

$$P_{(u,v)}=\frac{\left|T_{(u,v)}\right|+\left|T_{(v,u)}\right|}{\sum\limits_{v'\in \mathrm{In}(v)}\left(\left|T_{(v',v)}\right|+\left|T_{(v,v')}\right|\right)} \tag{2}$$

其中，In(v)表示节点 v 的入度节点集，v' 表示入度节点，$\left|T_{(u,v)}\right|+\left|T_{(v,u)}\right|$ 表示节点 u,v 间的联系次数。

2.2.2 基于时序图的信息传播模型

定义 3 节点活跃起始时间。节点 v 被其活跃父节点 u 成功激活的时刻称为节点的活跃起始时间，表示为 Act_v，$\mathrm{Act}_v=\min\{t|(t\in T_{(u,v)}\ \&\ t\geqslant \mathrm{Act}_u)\}$。

以图 1 为例，若节点 A 为种子节点（设种子节点的活跃起始时间为 0），且其成功激活节点 C，则 $\mathrm{Act}_C=\min\{7,9\}=7$。

定义 4 节点状态（state）。节点状态是指节点在网络中对信息传播所能做出的反应，分为活跃状态（active）和非活跃状态（inactive）。

在初始网络中，所有节点 v 的活跃起始时间均为 $\mathrm{Act}_v=-1$，节点状态为非活跃状态，当选定种子节点集后，设置种子节点 $\mathrm{Act}_v=0$，并将其状态修改为活跃状态。下面以种子节点 u 为例，具体描述信息传播过程。

1) 种子节点 u 以概率 $P_{(u,v_i)}$ 尝试激活其邻居节点 v_i，有且仅有一次激活机会，若成功激活节点 v_i，则记录 v_i 的最早激活时间 Act_{v_i}，并修改节点状态为活跃状态；若没有激活节点 v_i，则继续尝试激活下一邻居节点 v_{i+1}。

2) 若种子节点的某一邻居节点 v 被成功激活，则节点 v 会将信息继续传播下去。节点 v 在尝试激活其邻居节点 w_i 时，首先判断节点 w_i 是否处于非活跃状态，若是则继续判断 Act_{v_i} 是否小于或等于 $\max T_{(v,w_i)}$，若满足则以概率 $P_{(v,w_i)}$ 尝试激活其邻居节点 w_i；否则跳过该节点尝试激活下一邻居节点 w_{i+1}。

3) 信息在整个网络中由最新的活跃节点向处于非活跃状态的邻居节点进行传播扩散，直到整个网络没有新的节点被激活为止。

3 算法设计

CHG 算法将种子节点的挖掘分为节点影响力启发式评估、构建候选种子节点集以及种子节点贪心式选取这 3 个步骤。

3.1 节点影响力启发式评估

3.1.1 节点影响力评估方法的定义

传统启发式度中心性算法是指用节点度的大小衡量节点的影响力，节点度越大，该节点潜在的影响力就越大，但由于时序图中信息传播存在时序性，单一参考节点度数尚不能准确判定该节点在时序图中作为种子节点信息传播效果的好坏。例如，图 2 中节点 a 在 5 时刻分别将信息传播至其后继一阶节点(b,c,d)，但此时其后继一阶节点已在更早的时刻与其后继节点完成了联系，故此种情况下，节点 a 的信息只能传播至其后继一阶节点，而无法影响到更高阶节点，这种节点在全局网络中的影响力较小。

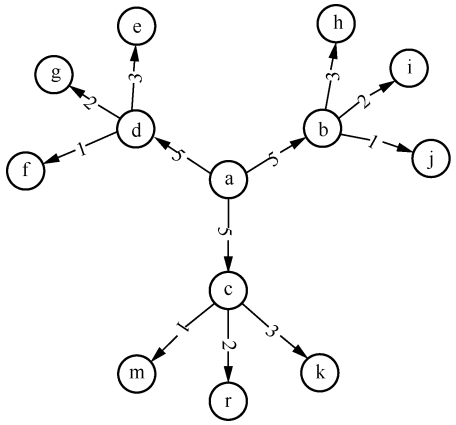

图 2 时序图中信息传播的时序性

时序图中潜在影响力较大的节点除具有一定的社交广度外，还要保证信息的可扩散性，即信息可传播至更高阶节点。基于以上 2 个因素，给出定义 5。

定义 5　二阶度（two-order degree）。时序图中节点 u 在满足信息传播的时序关系下，在其后继二阶范围内能够潜在影响到的最多节点数，表示为

$$D(u)=\sum_{v\in O(u)}\left|E_{(u,v)}\right|+\sum_{v\in O(u)}\left(\sum_{w\in O(v)}\left|E_{(v,w)}\right|\min\left(T_{(u,v)}\right)\leqslant \max(T_{(v,w)})\right) \tag{3}$$

其中，$O(u)$ 表示节点 u 的后继一阶节点集，$|E_{(v,w)}|\min(T_{(u,v)})\leqslant\max(T_{(v,w)})$ 表示在满足信息传播时序性的条件下，即后继一阶节点的活跃起始时间早于其继续尝试激活二阶节点的时间，节点在其后继二阶范围内能够潜在成功激活的节点。

二阶度大的节点可以保证信息能够由中心节点以更好的效果向外扩散传播，基于以上分析，本文为时序图中节点影响力定义了一种启发式计算方法，即以节点的二阶度对其影响力大小进行评估，表示为

$$\mathrm{Inf}(u)=\left|D(u)\right| \tag{4}$$

其中，Inf (u)表示节点 u 的影响力大小。

3.1.2　二阶度评估方法可靠性理论依据

网络中节点影响力的传播是一个离散扩散过程，其围绕种子节点向周边邻居节点扩散，并由邻居节点继续向下后继一阶节点传播影响，子节点被激活的概率总是受到父节点激活概率的影响，邻居节点的阶数越大，节点被激活的概率越小。文献[22]证明了节点 u 的第 i 阶邻居节点 v 被其成功激活的概率的计算满足容斥定理。文献[23]在此基础上进一步分析得出，在相对较小的传播概率下，节点在网络中的影响力随着邻居节点阶数的增大而逐渐收敛，节点对其后继三阶邻居节点的激活概率会降到较低水平，即使进行数万次的蒙特卡罗模拟，信息扩散过程中三阶及三阶以外的节点被成功激活的次数也同样较少[23]。同时，这也符合 Walkers 等[24]提出的三度影响力原则，即信息传播过程中存在内部衰减，个人的影响力局限于三阶邻居节点之内，故本文根据节点在其后继二阶范围内所能激活的最多节点数，即用节点的二阶度来评估其影响力的方法是可靠的，本文方法的误差率与时间效率通过 4.3.3 节实验进行了检验。

3.2　构建候选种子节点集

定义 6　边际效应 φ。节点 u 的边际效应是指将其加入种子节点集 S 中，所能带来的影响力增量 $\varphi_s(u)$。

$$\varphi_s(u)=\mathrm{Inf}(S\cup\{u\})-\mathrm{Inf}(S) \tag{5}$$

其中，Inf(S)表示种子集 S 的影响力大小。

由于节点与节点之间的影响范围会产生重叠，种子节点集最终的影响范围并非每个节点影响力的简单相加，故影响力最大的前 k 个节点未必是最好的种子节点组合。针对节点间影响范围重叠问题，最有效的解决方法是采用贪心算法思想，每次将当前能够产生最好影响效果的节点加入种子节点集中，直到找到 k 个种子节点为止，但此方法要求计算当前网络中所有节点的边际效应，从而延长了算法的运行时间。

文献[25]表明，大部分社交网络都具有无标度的特性，即大量节点拥有少量连接，少量节点拥有大量连接。网络中影响力较小的节点对种子节点的选取不构成影响，却增加了算法的遍历范围与次数，提升了算法的时间复杂度。因此，CHG 算法通过设置节点过滤因子 r 对时序图节点进行初步筛选，利用二阶度评估方法对节点影响力进行启发式评估，选取影响力评估值较大的前 rN 个节点构建候选种子节点集 L，这样可极大地缩小节点边际效应的计算范围，进一步提升算法效率，其中，$|L|=rN$，N 表示网络节点总数，r 的取值通过 4.3.2 节相关实验获取。

3.3　种子节点贪心式选取

此阶段采用贪心算法思想，通过计算候选节点的边际效应，解决节点间影响力重叠问题，以保证最终获取最优的种子节点组合。

3.3.1　节点边际效应的计算

计算一个节点边际效应的传统方法是将该节点加入已有种子节点集中，然后计算新种子集的影响节点增量，此增量即该节点边际效应大小。由于每计算一个节点的边际效应都要进行 R 次信息扩散模拟实验才可得出最终结果，因此该方法具有极高的复杂度。针对此问题，本文采用以空间换时间的策略，给出了一种新的节点边际效应计算方法，将所有候选种子节点在时序图上进行信息模拟扩散传播，并读取每个节点的影响范围，例如，当前种子集 S 的影响节点集为 S_1={a,b,c,d,e}，节点 u 的影响节点集为 U={a,b,f,h}，则只需要计算差集 $U-S_1$={f,h}，并统计差集中元素个数即可得出节点 u 的边际效应，这样便有效地将节点边际效应计算的时间复杂度降低到 $O(Rrn)+O(1)$，其中 R 表示共进行 R 次节点影响范围模拟实验。

3.3.2 利用节点边际效应确定最优种子集

性质 1 子模性。设集合 $S \subseteq T$，任意元素 x 添加到集合 S 中获得的函数收益大于或等于添加到集合 T 中所获得的收益，即满足收益递减特性，表示为

$$f(S \cup \{x\}) - f(S) \geqslant f(T \cup \{x\}) - f(T) \quad (6)$$

文献[12]论证了各个节点的边际效应大小是满足子模性的，即节点的边际效应会随着种子节点的增多而递减。IMIT 算法同样利用了节点边际效应计算过程中所具有的子模性，其主要思想如下：1) 如果上一轮中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，则不需要对其余的非种子节点重新计算边际效应，直接将 w 选为种子节点即可；2) 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 y，然后重新计算从节点 z 到节点 y 这段区间中所有节点的边际效应。

本文针对 IMIT 算法进一步优化，在算法中设置变量 Max_Inf 与 Max_node 以实时更新保存当前边际效应的最大值及其节点，在计算下一个节点之前，用当前边际效应最大值 Max_Inf 与其上轮该节点的边际效应进行比较，这样便可避免 IMIT 算法对所截取的一段节点进行边际效应的冗余计算问题，CHG 算法贪心选取种子节点流程如表 1 所示。其中，数值表示节点边际效应值，√表示选定为种子节点。

表 1 CHG 算法贪心选取种子节点流程

节点	第 1 轮	第 2 轮	Max_Inf	Max_node
a	100 √	—	—	—
b	80	30	30	b
c	60	50	50	c √
d	40	—	—	—
e	30	—	—	—

首先，选定候选种子节点中影响力最大的节点 a 为第一个种子节点。其次，从节点 b 开始进行第 2 轮节点边际效应计算，并将其写入变量 Max_Inf 和 Max_node 中，在计算下一个节点 c 之前，先用当前边际效应的最大值 Max_Inf 与上一轮中节点 c 的边际效应进行比较（30<60），判断是否需要继续计算节点 c 的边际效应，并对变量 Max_Inf 与 Max_node 进行更新，同样地，在计算下一个节点 d 之前，将 Max_Inf 与上一轮中节点 d 的边际效应进行比较（50>40），根据子模性可知，节点 c 即此轮边际效应的最大节点，直接将其选出即可。在 IMIT 算法中至少要计算 b,c,d,e 这 4 个节点的边际效应，而对其优化后只需要计算 b,c 这 2 个节点。CHG 算法如下所示。

算法 1 CHG 算法

输入 社交网络 $G_{\mathrm{T}}(V,E,T_{\mathrm{E}})$，$k$

输出 种子节点集 S

1) 初始化 $S=\phi$ ， $L=\phi$
2) for 图 G_{T} 中任意节点 u do
3) $\mathrm{Inf}(u)=|D(u)|$
4) end for
5) for i=1 to rN do
6) $v=\mathrm{argmax}_u \left\{\mathrm{Inf}(u)\middle|u \in \frac{V}{L}\right\}$
7) $L=L \cup \{v\}$
8) end for
9) sort node by Inf(u) in L
10) return result : Q $[u_1,u_2,\cdots,u_{rn}]$ /*算法前 2 个步骤完成*/
11) $S=S \cup \{u_1\}$ and 删除 u_1 in Q
12) while $|S|<k-1$
13) for u_i in Q do
14) 计算 $\varphi_s(u_i)$
15) 更新 Max_Inf , Max_node
16) if Max_Inf > $\varphi_{s'(u_i+1)}$
17) break /*子模性提前结束遍历*/
18) end if
19) end for
20) v = Max_node
21) $S=S \cup \{v\}$
22) 删除 v in Q
23)end while

步骤 1)表示将种子节点集 S 与候选种子节点集 L 初始化为空集；步骤 2)～步骤 4)表示对时序图中节点影响力进行评估；步骤 5)～步骤 8)表示选取前 rN 个影响力较大的节点构建候选种子节点集 L；步骤 9)～步骤 10)表示对候选种子节点按影响力大小排序，并装入队列 Q 中。步骤 11)～步骤 22)表示贪心式计算节点边际效应，并选取出最终的 k 个种子节点，其中步骤 16)～步骤 18)表示利用子模性减少对非必要节点的计算，提前结束算法程序遍历（伪代码中 s'表示上一轮种子节点集）。计算时序图中所有节点二阶度时间复杂度为 $O(n^2)$，按节点影响力大小排序并构建候选种子节点的时间复杂度为 $O(n\log(n))$，在贪心选取种子节点阶段，计算所有候选种子节点的影响范围时间复杂度为 $O(Rrn)$，计算节点边际效应的时间复杂度为 $O(1)$，利用子模性性质选取 k 个种子节

点的时间复杂度为 $O(k)$，故 CHG 算法的时间复杂度为 $O(n^2)+O(n\log(n))+O(Rrn)+O(k),n=N$。

4　实验与分析

4.1　实验设置

4.1.1　实验环境

操作系统为 Windows 10，英特尔处理器 Core i5-6300HQ 四核，内存为 8 GB，编程语言为 Python3.0，编程环境为 Pycharm。

4.1.2　实验数据

本节实验选取 3 个不同规模的时序社交网络数据集，具体参数如表 2 所示。

表 2　　实验数据集参数

数据集	节点数/个	边数/条	时间跨度/天
Digg	7 786	132 506	56
Mathoverflow	21 688	107 581	2 350
Superuser	101 052	356 822	2 735

Digg 数据集[26]记录了社交新闻网站 Digg 某段时间内用户间相互评论的信息；Mathoverflow 数据集[27]是根据 Mathoverflow 网站上的问答信息生成的社交网络；Superuser 数据集[27]是由堆栈交换网站 Superuser 所生成的时序网络图。

4.2　实验内容

本节实验从种子节点集的影响范围与运行时间 2 个方面对算法进行综合评价，除本文所提 CHG 算法外，分别复现了以下 5 种算法，作为实验对比算法。

IMIT 算法[9]。该算法采用 10 000 次蒙特卡罗模拟计算单个节点的影响力，并基于贪心思想利用节点边际效应完成种子节点的最终选取。

TIM 算法[10]。该算法以节点间联系次数评价节点影响力，并从评价值较大的前 100 个节点中筛选出最终的种子节点集。

Degree 算法。该算法是经典度启发式算法，将节点按照出度（Out-Degree）大小进行排序，直接选取前 k 个节点作为种子节点。

DegreeDiscount 算法[13]。该算法是启发式算法的代表，选取度数最大的节点作为种子节点，然后将所选节点邻居的度数进行折扣，直到选出 k 个节点。

Random 算法。该算法是基准比较算法，从时序图中随机选取 k 个非重复节点作为种子节点，该算法随机性较大，共对其做 500 次实验，实验结果取平均值。

4.3　实验结果及分析

4.3.1　复杂网络无标度性的验证

为了验证复杂网络的无标度特性，对 3 个数据集中的节点度分布情况进行了统计，结果如图 3 所示。

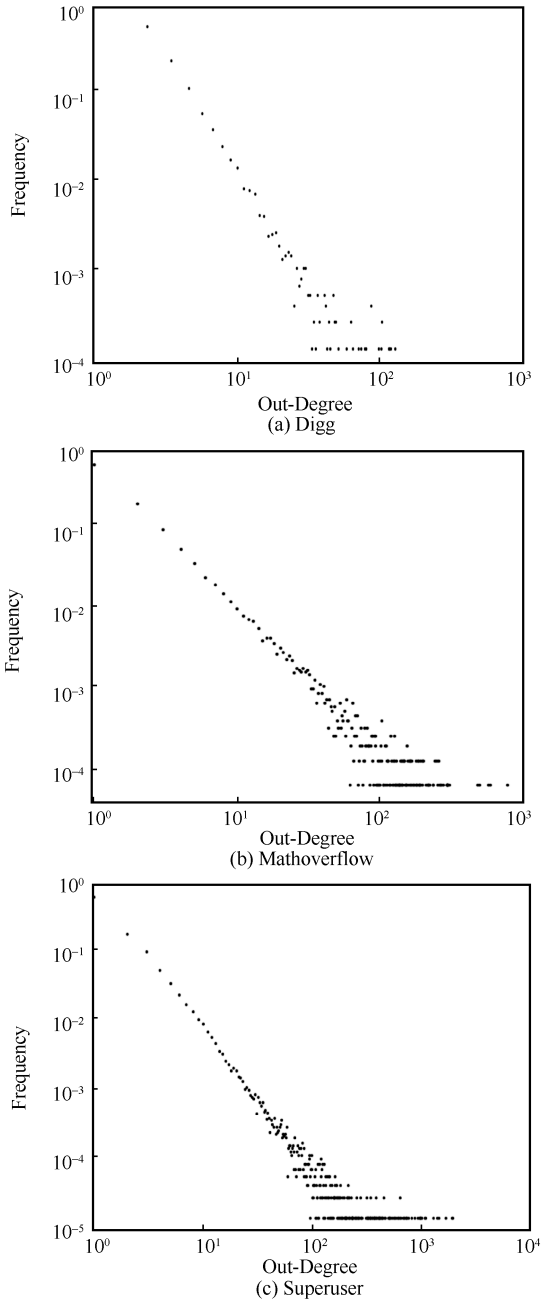

图 3　数据集节点度分布情况

图 3 中横坐标代表节点出度，纵坐标代表该值所对应节点的个数与总节点个数的比值，即出现频率（Frequency），这里以对数形式表示。从图 3 中可以看出，3 个数据集中节点的度均服从幂律分布，

说明具有较大影响力的节点分布较稀疏，大部分节点都拥有少量连接，影响力较小，这为 CHG 算法构建候选种子节点集的方法提供了有力依据。

4.3.2 节点过滤因子 *r* 值的设定

为了修剪网络中大部分影响力较小的节点，算法通过设定节点过滤因子 *r*，依据 3.1 节中获得的节点影响力评估结果对其进行过滤筛选，*r* 值的大小将直接影响算法的精度，因此，本文通过实验检验了不同 *r* 值对算法精度的影响，并确定了一个最合适的 *r* 值，从而使算法获得更高的准确性和更低的时间复杂度。由于时序信息传播模型是以经典独立级联模型为基础修改得到的，依旧存在独立级联模型中节点激活过程不确定性的特点。为了让实验更具说服力，在实验中，对节点间的传播概率进行相应的调整，将由式(2)计算出的传播概率 *P* 分别缩小 50%、增大一倍，观察在 3 种不同的传播概率下，不同 *r* 值时 CHG 算法所得种子节点集的影响范围，实验结果如图 4 所示。

从图 4 中可以看到，当 *r* 值为 0.1～0.2 时，种子节点集影响范围随着 *r* 值的增大而增大；当 *r* 值为 0.2～0.3 时，种子节点集影响范围略有上升，幅度较小；当 *r* 值大于 0.3 时，种子节点集影响范围趋于收敛。产生以上结果是因为随着 *r* 值的增大，越来越多影响力较高的节点进入候选种子节点集，可供算法挑选的节点增多，故在一定范围内，种子节点集的影响力不断升高。随着 *r* 值的增大，一些影响力相对较低的节点进入了候选种子节点集，但其对最终种子节点的挑选不构成影响，不会改变实验结果，所以最终种子节点集的影响力趋于平稳。由此可见，*r* 值过小，可能导致一些有影响力的节点被过滤掉，从而影响算法的精度；*r* 值过大，则会出现过滤效果不明显，造成候选种子节点集中存在冗余节点。当 *r* 取值为 0.2 时，既有效控制了候选种子集的规模，又能保证最终种子节点集具有较高的影响力。

4.3.3 节点影响力二阶度评估方法性能实验

为了验证节点影响力二阶度评估方法的准确性，本文实验采用 10 000 次蒙特卡罗模拟方法计算节点的精确影响力，分析节点影响力二阶度评估方法误差率（ER, error rate），实验结果如图 5 所示。其中，横坐标表示采用 2 种方法所得影响力排名靠前的 *x* 个节点（*N* 表示节点总数），纵坐标表示与蒙特卡罗模拟方法相比，二阶度评估方法所得结果的误差率，计算式为

$$\mathrm{ER}=\frac{|A-B|}{x} \tag{7}$$

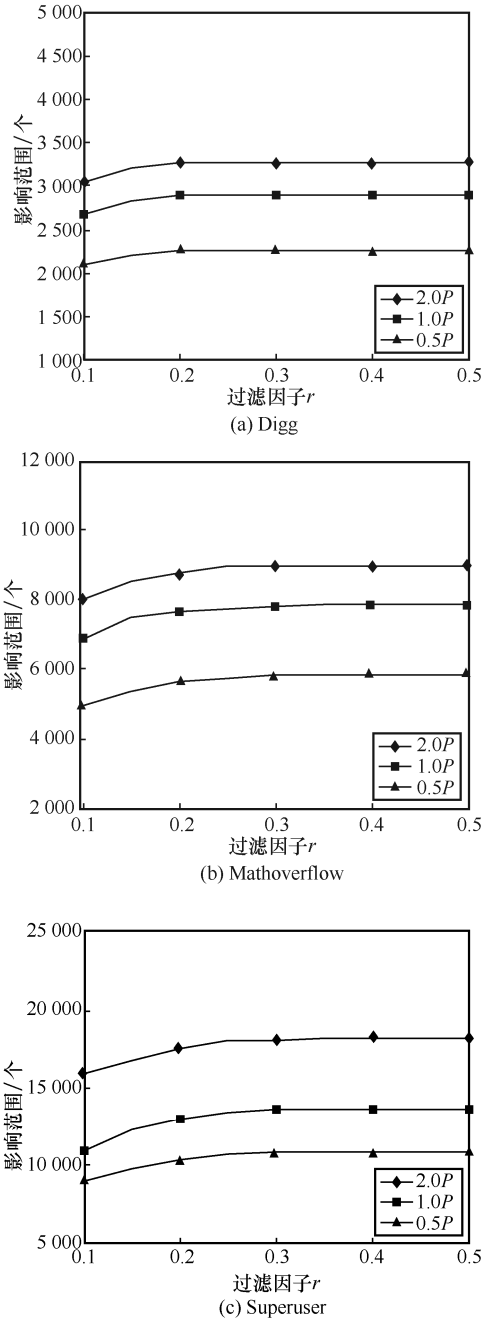

图 4 不同 *r* 值时算法 CHG 所得种子节点集影响范围

其中，集合 *A*、*B* 分别表示采用蒙特卡罗模拟方法与二阶度评估方法所计算出的影响力排名靠前的 *x* 个节点的集合。由图 5 可知，随着节点影响力计算范围的增大，二阶度评估方法的误差率也逐渐下降，相比于更精确的蒙特卡罗模拟方法，其获取影响力排名前 0.20*N* 个节点的计算误差率降到了 0.05 左右，由此可见，利用节点二阶度评估节点影响力，

并根据评估结果构建候选种子节点集的方法具有较高准确性。

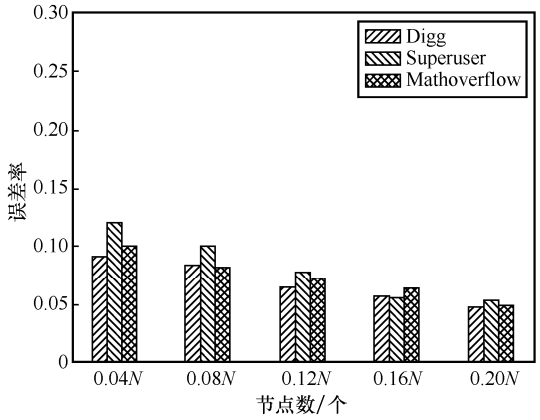

图 5　节点影响力二阶度评估方法误差率

本节采用 2 种方法计算网络所有节点影响力的运行时间，结果如图 6 所示。从图 6 可知，蒙特卡罗模拟方法时间复杂度较高，尤其处理较大规模网络时，运行时间过长，在大规模数据集 Superuser 下，其运行时间已超过 140 s，相比之下，二阶度评估方法具有更高的时间效率。

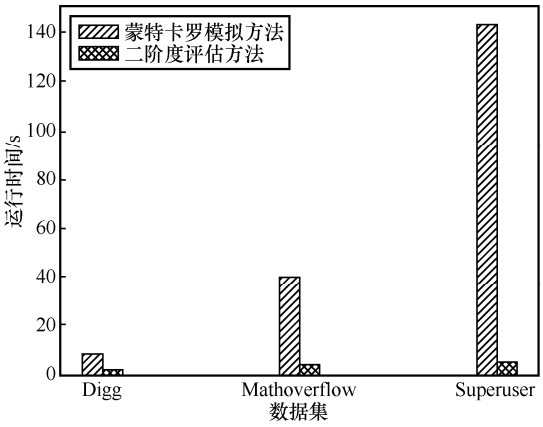

图 6　运行时间对比

4.3.4　不同算法所得种子节点集影响力对比

为了验证算法在影响范围指标上的效果，在特定时序信息传播模型下，对本文所提 CHG 算法以及复现的 5 种对比算法进行了种子节点集的影响力对比，种子节点的数量 k 分别设为 10,20,30,40,50，最终实验结果如图 7～图 9 所示。

在 Digg 数据集下，6 种算法所得种子节点集的影响范围均随着种子节点数的增多而增大，IMIT 算法的影响范围最大，与之相比，在 5 个不同数量的种子节点集下，CHG 算法的影响范围分别下降了 4.8%、5.2%、5.3%、7.2%、9.3%。当 k<30 时，TIM 影响范围与 IMIT、CHG 接近；当 k>30 时，TIM 的影响范围开始下降，与 IMIT、CHG 之间的差距逐渐增大。

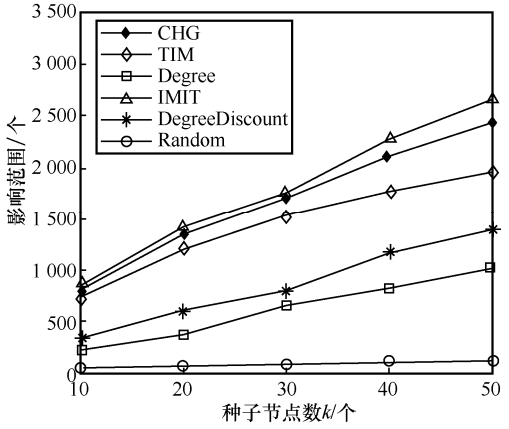

图 7　不同算法所得种子节点集影响范围（Digg）

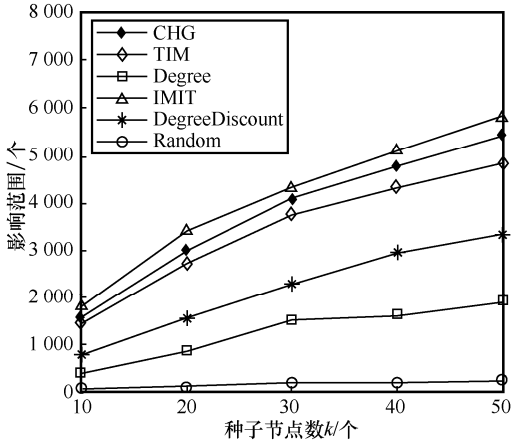

图 8　不同算法所得种子节点集影响范围（Mathoverflow）

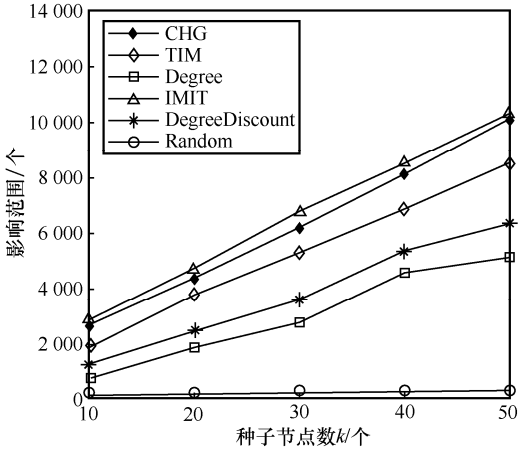

图 9　不同算法所得种子节点集影响范围（Superuser）

在 Mathoverflow 数据集下，CHG 算法影响范围相比于 IMIT 算法在 5 个种子节点集下分别下降

了 11.1%、8.9%、6.4%、7.6%、7.9%。当 k<20 时，TIM 影响范围与 CHG 接近；当 k>20 时，2 种算法之间的差距逐渐增大。在更大规模的数据集 Superuser 下，TIM 的影响效果与 IMIT、CHG 的差距更为明显，其影响范围相比于 IMIT 分别下降了 33.5%、20.9%、19.5%、19.1%、18.2%。而 CHG 曲线与 IMIT 曲线接近，5 个种子节点集下影响范围分别下降了 4.2%、8.1%、7.5%、5.4%、3.3%。DegreeDiscount 与 Degree 在时序图上表现出的影响效果一般，2 个启发式算法影响范围相比于 IMIT 分别下降了 53.2%与 68.8%。

产生上述实验结果的原因如下。Degree 算法不足之处在于其只关注了单一节点影响力大小，却忽略了节点间的相互作用，同时，由于时序图上信息传播所具有的时序性，该算法对单一节点影响力的评估也不够准确，这 2 个因素导致 Degree 算法所得种子节点质量下降，DegreeDiscount 算法针对此问题进行了改进，但是相对于贪心算法来说，其所挖掘出的种子节点集的影响力仍然有一定差距。TIM 算法以节点间联系次数来简单评价节点的重要程度，然后截取前 100 个节点进行最终种子节点的挑选，节点间联系次数更多反映的是传播概率的大小，用来衡量节点影响力尚存在一定误差，故导致一些真实影响力较大的节点，或者更好的节点组合并不存在于前 100 个节点之中，则所选出的种子集也并非最优的节点组合，这是造成 TIM 算法影响力损失的主要原因。另外，TIM 算法在面对小规模数据集或种子节点数较小时，尚可以获得相对较好的效果；当面对大规模数据集时，TIM 算法误差性开始显现，其影响力将会大幅下降，正如文献[10]所述，TIM 算法更适用于中小规模网络上的种子节点挖掘。IMIT 算法采用蒙特卡罗模拟方法计算单一节点影响力，再通过计算节点边际效应而得出最终种子节点集，获得了极高的影响力。CHG 算法首先对单一节点影响力进行启发式评估，并根据评估结果选取影响力最大的前 rN 个节点构建候选种子节点集，然后贪心选取种子节点，以保证最优的种子节点组合，故最终达到了与 IMIT 影响力非常接近的理想效果。由于 CHG 算法将种子节点的筛选范围缩小至候选种子节点集，该集合基本涵盖了对最终实验结果产生影响的所有节点，但不完全排除被过滤掉的节点是某轮边际效应计算中最大值节点的可能性，这是造成 CHG 算法所得种子节点集影响范围略低于 IMIT 算法的主要原因。

4.3.5 不同算法运行时间对比

1)为了验证 CHG 算法在利用节点边际效应贪心式选取种子阶段对 IMIT 算法的优化，对 2 种算法在该阶段选取 50 个种子节点的运行时间进行了实验对比，结果如图 10 所示。从图 10 中可以看到，CHG 算法相比 IMIT 算法在 3 个数据集下挖掘 50 个种子节点运行时间均更小，这是因为随着数据集规模的增大，IMIT 算法将会面临较为复杂的节点边际效应计算，而 CHG 算法通过对节点边际效应的计算方法进行改进，同时又优化了利用边际效应选点的策略，极大限度地降低了计算的复杂程度，实验数据显示，此阶段 CHG 算法比 IMIT 算法节省了 59%～71%的运行时间。

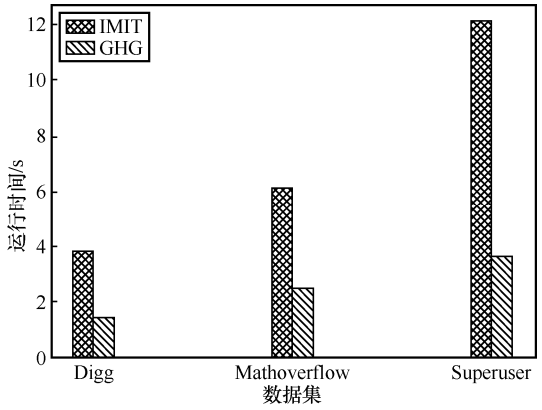

图 10 贪心式选取种子节点阶段运行时间

2) 6 种算法在 3 个数据集下分别挖掘 50 个种子节点的运行时间如表 3 所示。

表 3 算法运行时间

算法	Digg/s	Mathoverflow/s	Superuser/s
CHG	4.1	6.9	9.2
IMIT	12.9	46.4	154.2
TIM	3.9	5.4	8.6
Degree	1.7	2.2	3.2
DegreeDiscount	2.6	3.8	5.5
Random	0.8	0.9	1.2

从表 3 中可以看到，在 6 种算法中，IMIT 算法运行时间最长，尽管该算法在种子节点选取的过程中利用了节点边际效应的子模性，但是在对单一节点影响力计算上耗费了大量时间，导致算法最终运行时间较高。相比之下，CHG 算法对节点影响力进

行启发式评估，根据评估结果构建候选种子节点集，并对节点边际效应的计算方法进行了优化，减少了对非必要节点的遍历计算，极大限度地降低了算法时间复杂度，最终的运行时间相比于 IMIT 算法在 3 个数据集下分别减少了 68.2%、85.1%、94.0%。TIM 在时间效率方面表现良好是因为该算法极大限度上缩减了种子节点的筛选范围，将最终的种子节点局限于影响力最大的前 100 个节点之中，直接导致了算法最终所得种子节点集影响力的下降，可见该算法在追求时间高效的同时，也牺牲了一定的准确性。

综上所述，与 IMIT 算法相比，CHG 算法在运行时间上平均缩短了 82.4%，而影响范围仅平均降低了 6.9%，算法展现出了一定的高效性与可扩展性，更好地做到了影响范围与时间效率 2 个方面之间的平衡，即使在大规模网络中，CHG 算法也能避免 TIM 算法所出现准确率低的问题，并能够高效地完成大规模时序图中种子节点的挖掘。

5　结束语

为了解决时序图中种子节点集影响范围与算法时间效率两者间如何取得平衡，从而能够高效地完成大规模数据集下种子节点挖掘这一问题，本文给出了一种将启发式算法和贪心策略相结合的种子节点挖掘算法 CHG，首先对节点影响力进行启发式评估，并以此构建候种子节点集，最后对候选节点进行边际效应的计算，从而得到最终的种子节点集。实验结果表明，CHG 在时间效率与影响范围 2 方面均取得了理想效果，为大规模时序网络中种子节点的挖掘提供了更高效的策略。

在未来的工作中将会考虑进行如下深入研究。1) 针对不同类型的时序社交网络，进一步结合节点激活成本、耗费时间、不同信息类型等更多实际因素，研究如何进行种子节点的挖掘；2) 本文从全局的角度出发研究时序社交网络的影响力最大化问题，下一步可以尝试对时序图进行时间切片，以动态的视角研究实时影响力最大化问题。

参考文献：

[1] DOMINGOS P, RICHARDSON M. Mining the network value of customers[C]//Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. NewYork: ACM Press, 2001: 57-66.

[2] RICHARDSON M, DOMINGOS P. Mining knowledge-sharing sites for viral marketing[C]//Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2002: 61-70.

[3] WU J, CHEN Z G, ZHAO M. Information cache management and data transmission algorithm in opportunistic social networks[J]. Wireless Networks, 2019, 25(6): 2977-2988.

[4] HAN J D J, BERTIN N, HAO T, et al. Evidence for dynamically organized modularity in the yeast protein-protein interaction network[J]. Nature, 2004, 430(6995): 88-93.

[5] MIRITELLO G, MORO E, LARA R. Dynamical strength of social ties in information spreading[J]. Physical Review E, Statistical, Nonlinear, and Soft Matter Physics, 2011: doi.org/10.1103/PhysRev E.83.045102.

[6] WU H H, HUANG YU Z, CHENG J, et al. Efficient processing of reachability and time-based path queries in a temporal graph[J]. arXiv Preprint, arXiv: 1601.05909, 2016.

[7] REDMOND U, CUNNINGHAM P. Subgraph isomorphism in temporal networks[J]. arXiv Preprint, arXiv: 1605.02174, 2016.

[8] TAKAGUCHI T, YANO Y, YOSHIDA Y. Coverage centralities for temporal networks[J]. The European Physical Journal B, 2016, 89(2): 35.

[9] 吴安彪, 袁野, 乔百友, 等. 大规模时序图影响力最大化的算法研究[J]. 计算机学报, 2019, 42(12): 2647-2664.
WU A B, YUAN Y, QIAO B Y, et al. The influence maximization problem based on large-scale temporal graph[J]. Chinese Journal of Computers, 2019, 42(12): 2647-2664.

[10] 陈晶, 祁子怡. 基于时序关系的社交网络影响最大化算法研究[J]. 通信学报, 2020, 41(10): 211-221.
CHEN J, QI Z Y. Research on social network influence maximization algorithm based on time sequential relationship[J]. Journal on Communications, 2020, 41(10): 211-221.

[11] KEMPE D, KLEINBERG J, TARDOS É. Maximizing the spread of influence through a social network[C]//Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2003: 137-146.

[12] LESKOVEC J, KRAUSE A, GUESTRIN C, et al. Cost-effective outbreak detection in networks[C]//Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2007: 420-429.

[13] CHEN W, WANG Y J, YANG S Y. Efficient influence maximization in social networks[C]//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2009: 199-208.

[14] BAGHERI E, DASTGHAIBYFARD G, HAMZEHA. FAIMCS: a fast and accurate influence maximization algorithm in social networks based on community structures[J]. Computational Intelligence, 2021, 37(4): 1779-1802.

[15] 王帅, 刘静. 一种针对网络结构破坏下鲁棒影响力最大化问题的 Memetic 算法[J]. 计算机学报, 2021, 44(6): 1153-1167.
WANG S, LIU J. A memetic algorithm for solving the robust influence maximization problem towards network structural perturbances[J]. Chinese Journal of Computers, 2021, 44(6): 1153-1167.

[16] TONG G M, WANG R Q, DONG Z, et al. Time-constrained adaptive influence maximization[J]. IEEE Transactions on Computational Social Systems, 2021, 8(1): 33-44.

[17] YANG Y, YAN D, WU H H, et al. Diversified temporal subgraph pattern mining[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2016: 1965-1974.

[18] SHENG W, SONG W B, LI D, et al. Dynamic influence maximization via network representation learning[J]. Frontiers in Physics, 2022, 9: 827468.

[19] WANG Y H, FAN Q, LI Y C, et al. Real-time influence maximization on dynamic social streams[J]. Proceedings of the VLDB Endowment, 2017, 10(7): 805-816.

[20] ROSSI I, MUSOLESIM. TORSELLO A. On the k-anonymization of time-varying and multi-layer social graphs[C]//Proceedings of the 9th Internation AAAI Conference on Web and Social Media. Palo Alto: AAAI Press, 2015: 377-386.

[21] 王一舒, 袁野, 刘萌, 等. 大规模时序图数据的查询处理与挖掘技术综述[J]. 计算机研究与发展, 2018, 55(9): 1889-1902.
WANG Y S, YUAN Y, LIU M, et al. Survey of query processing and mining techniques over large temporal graph database[J]. Journal of Computer Research and Development, 2018, 55(9): 1889-1902.

[22] ZHANG M, DAI C N, DING C, et al. Probabilistic solutions of influence propagation on social networks[C]//Proceedings of the 22nd ACM International Conference on Information & Knowledge Management. New York: ACM Press, 2013: 429-438.

[23] 李美玲, 钱付兰, 徐涛, 等. 基于种子候选的贪心策略影响力最大化算法[J]. 模式识别与人工智能, 2020, 33(11): 1033-1042.
LI M L, QIAN F L, XU T, et al. Greedy strategy influence maximization algorithm based on seed candidates[J]. Pattern Recognition and Artificial Intelligence, 2020, 33(11): 1033-1042.

[24] WALKERS K. Connected: the surprising power of our social networks and how they shape our lives[J]. Journal of Family Theory & Review, 2011, 3(3): 220-224.

[25] BARABASIAL, ALBERT R. Emergence of scaling in random networks[J]. Science, 1999, 286(5439): 509-512.

[26] ROSSI R, AHMED N. The network data repository with interactive graph analytics and visualization[C]//Proceedings of the 29th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2015: 4292-4293.

[27] LESKOVEC J, KREVL A. SNAP datasets: Stanford large network dataset collection[EB]. 2014

[作者简介]

邹晓红（1967− ），女，吉林省吉林市人，博士，燕山大学教授、硕士生导师，主要研究方向为图挖掘、社会网络、复杂网络分析等。

许成伟（1997− ），男，黑龙江齐齐哈尔人，燕山大学硕士生，主要研究方向为社交网络。

陈晶（1976− ），女，河北秦皇岛人，博士，燕山大学教授、硕士生导师，主要研究方向为对等网络、社会计算、Web 服务等。

宋彪（1996− ），男，河北张家口人，燕山大学硕士生，主要研究方向为不确定图挖掘。

王明月（1996− ），女，河北秦皇岛人，燕山大学硕士生，主要研究方向为社交网络。