基于社交网络的影响力最大化算法

王璿[1]，张瑜[1]，周军锋[1]，陈子阳[1,2]

（1. 东华大学计算机科学与技术学院，上海 201620；2. 上海立信会计金融学院信息管理学院，上海 201620）

摘 要：影响力最大化问题研究在给定传播模型下如何选取社交网络中的一组种子用户，使信息通过这些用户实现最大范围的传播。现有算法主要存在 2 个问题：一是由于影响范围有限、时间复杂度高，难以适用于大规模社交网络；二是仅局限于特定传播模型，只能解决单一类型社交网络下的影响力最大化问题，当使用在不同类型社交网络上时效果较差。对此，基于 2 个经典影响力传播模型，结合反向影响采样技术，提出一种高效的影响力最大化（MTIM）算法。为验证 MTIM 算法的高效性，将其与 IMM、TIM 和 PMC 等贪心算法，以及 OneHop 和 Degree Discount 等启发式算法在 4 个真实社交网络上进行对比实验，结果表明 MTIM 算法能够提供 $\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$ 近似保证，显著扩大影响范围，并有效提高运行效率。

关键词：社交网络；影响力最大化；种子集；传播模型

中图分类号：TP393

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2022152

Influence maximization algorithm based on social network

WANG Xuan[1], ZHANG Yu[1], ZHOU Junfeng[1], CHEN Ziyang[1,2]

1. School of Computer Science and Technology, Donghua University, Shanghai 201620, China

2. School of Information Management, Shanghai Lixin University of Accounting and Finance, Shanghai 201620, China

Abstract: The influence maximization (IM) problem asks for a group of seed users in a social network under a given propagation model, so that the information spread is maximized through these users. Existing algorithms have two main problems. Firstly, these algorithms were difficult to be applied in large-scale social networks due to limited expected influence and high time complexity. Secondly, these algorithms were limited to specific propagation models and could only solve the IM problem under a single type of social network. When they were used in different types of networks, the effect was poor. In this regard, an efficient algorithm (MTIM) based on two classic propagation models and reverse influence sampling (RIS) was proposed. To verify the effectiveness of MTIM, experiments were conducted to compare MTIM with greedy algorithms such as IMM, TIM and PMC, and heuristic algorithms such as OneHop and Degree Discount on four real social networks. The results show that MTIM can return a $\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$ approximate solution, effectively expand the expected influence and significantly improve the efficiency.

Keywords: social network, influence maximization, seed set, propagation model

0 引言

影响力最大化（IM, influence maximization）[1-2]研究如何从社交网络中选择一组最具影响力的种子节点，基于这些节点发起信息传播，使最终的传播范围最大化。该问题广泛应用在产品营销[3]、疾

收稿日期：2022−05−23；**修回日期：**2022−07−27

通信作者：陈子阳，zychen@ysu.edu.cn

基金项目：国家自然科学基金资助项目（No.61873337）；上海自然科学基金资助项目（No.20ZR1402700）

Foundation Items: The National Natural Science Foundation of China (No.61873337), The Natural Science Foundation of Shanghai (No.20ZR1402700)

病控制[4]和个性化推荐[5]等方面。例如，商家会从社交网络中选择最具影响力的部分用户，基于这些用户对产品进行推广和营销，使更多的用户了解并最终转化为潜在顾客。

影响力最大化问题需要基于特定传播模型来描述信息在网络中的传播过程。目前使用最为广泛的是独立级联（IC, independent cascade）模型[6]和线性阈值（LT, linear threshold）模型[7]。不同的传播模型适用于不同类型的社交网络。社交网络可以分为个体网络和群体网络[8]。个体网络主要考虑单个节点和单个节点之间的影响关系，适用于独立级联模型。群体网络主要考虑单个节点和多个节点之间、多个节点和多个节点之间的影响关系，适用于线性阈值模型。基于选定的传播模型，影响力最大化问题等价于选择影响力尽可能大的种子集。

为了得到合适的种子集，Kempe 等[1]首先基于 IC 模型和 LT 模型提出了一个贪心算法。该算法可以同时在这 2 个模型上提供$\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$近似保证（$\varepsilon$ 为误差参数），但是时间复杂度过高，难以适用于大规模社交网络。后续的研究者陆续提出基于某种特定传播模型的高效算法[10-12]。这些算法虽然在大规模社交网络上的运行效率得到提升，但是仅局限于特定传播模型中，只能解决单一类型社交网络下的影响力最大化问题，当使用在不同类型社交网络上时效果较差[13]。为解决该问题，本文提出一种可以同时支持 IC 模型和 LT 模型的高效种子集求解算法，该算法包括 3 个阶段。1) 预处理阶段，基于节点度筛选策略，筛选出有效节点集；2) 采样阶段，基于边界约束策略，确定采样次数并从有效节点集中采样；3) 种子选择阶段，应用贪心策略选择种子节点，并基于影响力增量剪枝策略，剪枝种子选择时的部分无效排序。具体来说，本文的贡献如下。

1) 提出边界约束策略，以快速确定估计最优采样次数。提出基于节点度筛选策略，以提升种子集质量。提出基于影响力增量剪枝策略，以提高算法运行效率。

2) 结合这 3 个策略，提出一种三阶段的影响力最大化（MTIM, mixed three-stage influence maximization）算法，该算法不但能够同时支持 IC 模型和 LT 模型，而且具备优越的近似保证和期望时间复杂度。

3) 将 MTIM 与 IMM、TIM、PMC 等贪心算法，以及 OneHop、DegreeDiscount 等启发式算法在 4 个真实社交网络上对比实验。结果表明，MTIM 算法能够适用于大规模社交网络，提供$\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$近似保证，并有效提升影响范围和效率。

1 背景知识和相关工作

1.1 背景知识

本文使用加权有向图 $G=(V,E,W)$表示社交网络，其中，V 表示节点集（用户），E 表示有向边集（用户间关系），W 表示每条有向边对应权值的集合。$W(u,v)\in[0,1]$表示有向边(u,v)的权值，代表传播过程中节点 u 把信息传递给节点 v 的概率，即 u 激活 v 的概率。为表述方便，使用 n 和 m 分别表示节点集 V 和有向边集 E 的大小，$\mathrm{In}(v)$和 $\mathrm{Out}(v)$分别表示节点 v 的入邻居集合和出邻居集合。

影响力最大化问题旨在通过种子集（信息传播的源头节点）进行信息传播，实现传播范围的最大化。而信息如何在网络中传播是由传播模型确定的。在信息传播过程中，如果节点 v 接受某种信息，则称该节点为激活节点，否则称其为未激活节点。目前主流的 2 种影响力传播模型的主要区别在于节点的激活方式。

1) IC 模型。假设节点 u 在 i 时刻被激活，则节点 u 在 i+1 时刻只有一次机会以传播概率 $W(u,v)$激活其尚未被激活的出邻居 $v\in\mathrm{Out}(u)$。

2) LT 模型。每个节点有概率阈值 $\varphi\in[0,1]$，该阈值表示节点被激活的难易程度。如果节点 v 在 i 时刻未被激活，且满足$\phi_v\leqslant\sum_{u\in A}W(u,v)$，则节点 v 在 i+1 时刻被激活，其中 A 是节点 v 在前 i 时刻被激活的入邻居集合。

给定社交网络 G、种子集$S\subseteq V$ 以及传播模型 M，传播过程如下。1) 在第 0 时刻，S 中的所有节点被激活，其他节点未被激活。2) 如果一个节点在 i 时刻被激活，则该节点在 i+1 时刻只有一次机会激活其尚未被激活的出邻居（激活方式取决于传播模型 M），之后它就不能再激活任何节点。3) 重复步骤 2)，直至不再有节点被激活，传播结束。种子集 S 在传播模型 M 下激活节点的总数表示为$\sigma(S)$，代表种子集 S 的预期影响范围。表 1 给出了本文的常用符号。

问题定义（影响力最大化问题）给定社交网络 G、参数 k 以及传播模型 M，影响力最大化问题旨在找出种子集 $S \subseteq V$ 且 $|S| = k$，使种子集预期影响范围 $\sigma(S)$ 最大化。

表 1　常用符号设置

符号	含义
$G=(V,E,W)$	社交网络
n,m	$n=\|V\|, m=\|E\|$
R	反向可达集
$\mathcal{R}$	反向可达集的集合，例如 $\mathcal{R}=\{R_1,R_2,\cdots\}$
$\Lambda_{\mathcal{R}}(S)$	集合 S 在 $\mathcal{R}$ 中的覆盖范围
$\sigma(S)$	集合 S 的预期影响范围
$F_{\mathcal{R}}(S)$	集合 S 对 $\mathcal{R}$ 的覆盖率
OPT	任意 k 个种子的最大影响范围
$\mathbb{E}[\cdot]$	随机变量的期望值
lv	被链接程度
In(·)	入邻居集合
Out(·)	出邻居集合

1.2　相关工作

现有影响力最大化算法大多基于反向影响采样（RIS, reverse influence sampling）技术[9]选取种子节点。反向影响采样就是先随机选一个节点，从该节点出发，沿着该节点所有入边的相反方向模拟传播，这样反向可达的节点集合就称为反向可达集（RR set, reverse reachable set）；再生成足够多的反向可达集，从中找出影响力最大的种子集（即能够覆盖最多反向可达集的节点集合）。这里，如果节点 v 在集合 R 中出现，则称节点 v 覆盖集合 R；如果集合 S 中至少有一个节点在集合 R 中出现，则称集合 S 覆盖集合 R。根据文献[9]可知，对于从节点 v 反向采样得到的反向可达集 R，节点 $u \in V \backslash \{v\}$ 覆盖集合 R 的概率等于节点 u 激活节点 v 的概率。利用 RIS 技术，可以使反向可达集更容易包含极具影响力的节点，从而提高算法效率。

例 1　（反向可达集构建示例）以 IC 模型为例，反向可达集构造如下。首先在图 1(a)所示的社交网络 g 中随机选择节点 b，此时反向可达集 $R=\{b\}$。接着沿节点 b 的所有入边反向执行广度优先遍历，对入边(c,b)生成随机数 $r_1=0.3$。由于 $r_1 \leqslant 0.6$，节点 c 被节点 b 激活，将节点 c 加入 R 中，即 $R=\{b,c\}$。同理，为节点 c 的入边(a,c)生成随机数 $r_2=0.9$。由于 $r_2>0.8$，节点 a 没有被节点 c 激活，不将节点 a 加入 R 中，图 1(b)中的虚线表示遍历失败，灰线表示遍历成功，黑线表示社交网络 g 中的有向边。此时不再有节点能被激活，最终反向可达集 $R=\{b,c\}$。

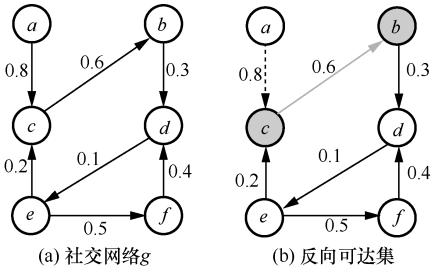


图 1　反向可达集构建示例

Borgs 等[9]在种子集预期影响范围与反向可达集之间建立了以下联系。

引理 1　设 $S \subseteq V$ 为种子集，R 为传播模型 M 下生成的反向可达集，则

$$\sigma(S) = n \Pr[S \bigcap R \neq \varnothing] \tag{1}$$

引理 1 表明，可以使用反向可达集来估计任意种子集的预期影响范围。假设生成一个反向可达集集合 $\mathcal{R}=\{R_1,R_2,\cdots\}$，令 $\Lambda_{\mathcal{R}}(S)$ 表示种子集 S 覆盖的反向可达集数量，代表种子集 S 在集合 $\mathcal{R}$ 中的覆盖范围，则 $\frac{n\Lambda_{\mathcal{R}}(S)}{|\mathcal{R}|}$ 是对种子集预期影响范围的无偏估计，即 $\mathbb{E}[\sigma(S)] = \frac{n\Lambda_{\mathcal{R}}(S)}{|\mathcal{R}|}$。

Borgs 等的解决方案。利用引理 1，Borgs 等[9]提出一种影响力最大化算法，该算法分为两步：1) 采样，即生成足够多的反向可达集；2) 种子选择，即贪心地选择对反向可达集覆盖率最大的种子集。Borgs 等证明，如果检验了 $O\left(\frac{k\ell^2(m+n)\log^2 n}{\varepsilon^3}\right)$ 条边，则该算法可以提供 $\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$ 近似保证。

TIM 和 IMM。Tang 等[14-15]基于 RIS 技术提出了 TIM 算法和 IMM 算法，这些算法的时间复杂度均为 $O\left(\frac{(k+\ell)(m+n)\log n}{\varepsilon^2}\right)$，明显优于 Borgs 等的算法，但是仍存在大量冗余的计算开销。TIM 算法利用切尔诺夫边界（Chernoff bound）来确定采样次数，而不再通过检验遍历的边数来判断是否满足近似保证，该算法适用于 LT 模型。而 IMM 算法利用鞅技术改进，进一步减少采样次数，能够同时适用 IC 模型和 LT 模型，算法性能稳定高效。

其他基于特定传播模型的解决方案。Sun 等[10]基于多轮扩散（MRT, multi-round triggering）模型

提出的 MRIM 算法，解决了多轮扩散下的影响力最大化问题。Guo 等[11]基于触发（TR, triggering）模型提出的 IMCB 算法，从社区结构角度解决了影响分布不平衡的问题。Guo 等[12]基于加权级联（WC, weighted cascade）模型提出 SUBSIM 算法，对反向可达集生成过程进行了优化。

2 MTIM 算法

本文针对现有算法效率低、适用传播模型单一的问题，提出一种三阶段影响力最大化算法——MTIM 算法。MTIM 算法包括 3 个阶段。

1) 预处理阶段。基于节点度筛选策略，根据节点的出度和被链接程度，筛选出有效节点集 $\mathcal{C}$ 。

2) 采样阶段。基于边界约束策略，先迭代地确定近似最优采样次数 θ^*，再从 $\mathcal{C}$ 中随机选点采样 θ^* 次，得到反向可达集集合 $\mathcal{R}=\{R_1,R_2,\cdots,R_{\theta^*}\}$ 。

3) 种子选择阶段。应用贪心策略找出对集合 $\mathcal{R}$ 覆盖率最大的种子集；同时，基于影响力增量剪枝策略，剪枝部分种子选择时的无效排序。

MTIM 算法的具体流程如算法 1 所示。

算法 1 MTIM 算法

输入 社交网络 G，种子个数 k

输出 种子集 S

1) $\mathcal{C}\leftarrow$ Filtering(G) //预处理

2) $\mathcal{R}\leftarrow$ Sampling(G,$\mathcal{C}$) //采样

3) $S\leftarrow$ NodeSelection($\mathcal{R}$,k) //种子选择

4) return S

2.1 基于节点度筛选策略的预处理

RIS 方法每次从整个社交网络中随机选点采样，由于所选节点质量参差不齐，导致求解出的种子集对反向可达集的覆盖率偏低，使种子集影响范围有限。针对该问题，本文提出基于节点度筛选策略。该策略的基本思想为根据节点的出度和被链接程度（lv, linked value）[16]，筛选出潜在影响力较大的节点集合（即有效节点集）。利用该策略，不仅可以缩小采样时的选点范围，而且可以提高种子集对反向可达集的覆盖率，从而有效改善种子质量。

预处理阶段的主要工作如下。1)根据节点的出度和被链接程度，从社交网络 G 中筛选出节点集合 $\mathcal{A}$ 和集合 $\mathcal{B}$，且 $|\mathcal{A}|=|\mathcal{B}|=r\%|V|$；2) 取这 2 个集合的并集作为有效节点集 $\mathcal{C}$ 。算法 2 为预处理阶段的伪代码。

算法 2 Filtering 算法（预处理阶段）

输入 社交网络 G

输出 有效节点集 $\mathcal{C}$

1) 初始化 $n\leftarrow r\%|V|,\mathcal{A}\leftarrow\varnothing,\mathcal{B}\leftarrow\varnothing$

2) 计算节点的出度

3) for $i=1$ to n do //获取 $\mathcal{A}$

4) 选择当前出度最大的节点 $v\in V$，并将其插入 $\mathcal{A}$

5) end for

6) 根据式(2)计算节点的被链接程度

7) for $i=1$ to n do //获取 $\mathcal{B}$

8) 选择当前被链接程度最大的节点 $v\in V$，并将其插入 $\mathcal{B}$

9) end for

10) $\mathcal{C}\leftarrow\mathcal{A}\bigcup\mathcal{B}$ //获取有效节点集 $\mathcal{C}$

11) return $\mathcal{C}$

对于有向图中的节点，其度包含出度和入度这 2 个含义，因而从这 2 个方面考虑。1) 出度大的节点，影响其出邻居的潜在可能性较大，则其影响力往往较大。因而，可以根据节点的出度降序排序，获得出度前 r% 大的节点集合 $\mathcal{A}$（算法 2 的步骤 2)～步骤 5)）。2) 由于传播在多个节点间进行，考虑单个节点的入度并无意义，可求单个节点被其入邻居链接的程度。被链接程度大的节点，其被入邻居影响的潜在可能性大，则影响力大的节点往往存在于被链接程度大的节点的反向可达集中。因而，可以先根据式(2)迭代地计算节点的被链接程度（d 为平衡因子），再根据节点的被链接程度降序排序，获得被链接程度前 r% 大的节点集合 $\mathcal{B}$（算法 2 的步骤 6)～步骤 9)）。

$$u.\mathrm{lv}=1-d+d\sum_{v\in\mathrm{ln}(u)}\frac{v.\mathrm{lv}}{v.\mathrm{outdegree}} \tag{2}$$

此时，潜在影响力大的节点多在集合 $\mathcal{A}$ 和集合 $\mathcal{B}$ 中，因而，可以取两集合的并集作为有效节点集 $\mathcal{C}$ 输出（算法 2 的步骤 10)～步骤 11)）。

例 2 （预处理示例）对图 1(a)社交网络 g 预处理流程如下。假设筛选比例为 70%。首先，计算每个节点的出度，并根据出度降序排序，取出度前 70% 大的节点集 $\mathcal{A}=\{a,c,d,e\}$；然后，计算每个节点的被链接程度，并根据被链接程度降序排序，取被链接程度前 70% 大的节点集 $\mathcal{B}=\{b,c,d,e\}$；最后，取 $\mathcal{A}$ 和 $\mathcal{B}$ 的并集 $\mathcal{C}=\{a,b,c,d,e\}$ 作为有效节点集。

2.2 基于边界约束策略的采样

针对 RIS 方法难以确定采样次数的问题，提出边界约束策略。该策略的基本思想如下，首先估计出近似最优采样次数的取值区间；然后根据该区间依次计

算不同采样次数下影响范围近似解下界和最优解上界之比，不断计算直至该比值达到给定要求时停止。利用该策略，可以快速确定近似最优采样次数。

采样阶段的主要工作如下，首先估计出近似最优采样次数 θ^*；然后从预处理阶段获得的有效节点集 $\mathcal{C}$ 中随机选点采样 θ^* 次，从而得到反向可达集集合 $\mathcal{R}=\{R_1,R_2,\cdots,R_{\theta^*}\}$。

2.2.1 最优采样次数的估计

为估计采样次数 θ，需进一步分析。假设 S_k^O 是期望影响力最大的 k 大种子集，即 $\mathrm{OPT}=\mathbb{E}[\sigma(S_k^O)]$，根据引理 1，如果 θ 大小适当，则 $nF_{\mathcal{R}}(S_k^O)\approx\mathrm{OPT}$。给定 $\varepsilon\in[0,1]$，根据文献[15]可推导出近似最优采样次数 θ^*，如式(3)所示。

$$\theta^*=\frac{2n\left(\left(1-\frac{1}{\mathrm{e}}\right)\alpha+\beta\right)^2}{\mathrm{OPT}\,\varepsilon^2}=\frac{\frac{\lambda^*}{\varepsilon^2}}{\mathrm{OPT}} \tag{3}$$

其中，$\lambda^*=2n\left(\left(1-\frac{1}{\mathrm{e}}\right)\alpha+\beta\right)^2$，$\alpha=\sqrt{\ln 2n}$，$\beta=\sqrt{\left(1-\frac{1}{\mathrm{e}}\right)\left(\ln\binom{n}{k}+\ln 2n\right)}$。

根据式(3)，如果 OPT 已知，则可以计算出近似最优采样次数。但是 OPT 实际未知，因而考虑根据 OPT 的取值范围估计出 θ^* 的取值区间。

已知 $\mathrm{OPT}\in[1,n]$，由于 S_k^O 中包含 k 个节点，至少能够影响到这 k 个节点，则可知 $\mathrm{OPT}\in[k,n]$。为估计出 θ^* 的取值区间，给出引理 2[15]与定理 1。

引理 2　给定 $\varepsilon\in(0,1)$，$\ell\geqslant 1$，令 θ^O 表示理论最优采样次数，θ^* 表示根据式(3)计算出的近似最优采样次数，则满足

$$\theta^O\leqslant\theta^*\leqslant\theta^O\left(1+\frac{\log 2}{\log n^{\ell}}\right) \tag{4}$$

定理 1　给定 $x\in[k,n]$，令 $\theta_{\max}=2\lambda^*\varepsilon^{-2}k^{-1}$，$\theta_{\min}=\lambda^*n^{-1}$，根据引理 2，可知

$$\theta_{\min}\leqslant\frac{\lambda^*}{\varepsilon^2 x}\leqslant\theta_{\max} \tag{5}$$

证明　构造函数 $\theta(x)=\lambda^*\varepsilon^{-2}x^{-1}$，可知该函数随着 x 的增大而单调递减。已知 $x\in[k,n]$，则函数值域为 $[\theta(n),\theta(k)]$。根据引理 2，$\theta_{\min}\leqslant\theta(n)\leqslant\theta(x)$；同时根据式(3)计算的近似最优值恰在 $\theta(x)$ 的值域内，则必然满足

$$\theta_{\min}\leqslant\theta^*\leqslant\theta(k)\left(1+\frac{\log 2}{\log n^{\ell}}\right)\leqslant\theta_{\max} \tag{6}$$

证毕。

根据定理 1，可以估计出近似最优采样次数的取值区间为 $[\theta_{\min},\theta_{\max}]$。

2.2.2 影响范围边界的估计

为了从取值区间 $[\theta_{\min},\theta_{\max}]$ 中找出近似最优采样次数 θ^*，可以根据该区间依次计算出不同采样次数 θ 下影响范围的近似比，即当前解下界和最优解上界之比，如果该比值大于或等于 $\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$，则立刻停止并返回当前采样次数 θ，否则将当前采样次数 θ 乘以 2 并重复上述步骤。为估计影响范围的边界，给出引理 3[17]。

引理 3　给定种子集 S 和由 θ 个反向可达集构成的集合 $\mathcal{R}=\{R_1,R_2,\cdots,R_\theta\}$，则对于 $\forall\lambda>0$，满足

$$\sigma^{\mathrm{lower}}(S)=\left(\frac{\ln 2n}{18}-\left(\sqrt{\Lambda(S)+\frac{2\ln 2n}{9}}-\sqrt{\frac{\ln 2n}{2}}\right)^2\right)\frac{n}{\theta} \tag{7}$$

$$\sigma^{\mathrm{upper}}(S_k^O)=\left(\sqrt{\frac{\Lambda(S)}{1-\frac{1}{\mathrm{e}}}+\frac{\ln 2n}{2}}+\sqrt{\frac{\ln 2n}{2}}\right)^2\frac{n}{\theta} \tag{8}$$

根据引理 3，当前种子集 S 影响范围的近似比可以用 $\frac{\sigma^{\mathrm{lower}}(S)}{\sigma^{\mathrm{upper}}(S_k^O)}$ 表示。

2.2.3 算法描述

根据 2.2.1 节估计出的近似最优采样次数 θ^* 的取值区间 $[\theta_{\min},\theta_{\max}]$，以及 2.2.2 节判断是否为近似最优采样次数的边界约束条件，将其组合，即可构成基于边界约束策略的采样阶段。

算法 3 为采样阶段的伪代码。具体地，首先，根据式(5)初始化 $\theta_{\min}$ 和 $\theta_{\max}$（算法 3 的步骤 1)）；接着，至多执行 $i_{\max}$ 次 for 循环（算法 3 的步骤 3)～步骤 13)），在第 i 次时，先从有效节点集 $\mathcal{C}$ 中随机选择节点采样，生成 θ_i 个反向可达集，再根据式(7)和式(8)计算出当前采样次数下种子集 S_i 影响范围下界与上界之比，如果该比值大于或等于 $\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$ 或者当前执行第 $i_{\max}$ 次循环，则停止并返回当前种子集，否则将 $\theta_{\min}$ 乘以 2 并重复计算。组

合后算法的近似比为$\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$，原因在于：1) 如果循环次数少于$i_{\max}$时就停止，则必能够提供$\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$近似保证；2) 如果循环次数为$i_{\max}$时才停止，此时的采样次数必大于或等于$\theta^*$，则必能采样足够多的反向可达集，即必能提供$\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$近似保证。

算法 3 Sampling 算法（采样阶段）

输入 社交网络G，有效节点集$\mathcal{C}$

输出 反向可达集集合$\mathcal{R}$和种子集S

1) 根据式(5)初始化$\theta_{\min}, \theta_{\max}$
2) $i_{\max} \leftarrow \left\lceil \mathrm{lb}\frac{\theta_{\max}}{\theta_{\min}} \right\rceil$
3) for $i=1$ to $i_{\max}$ do
4) $\quad\theta_i \leftarrow \theta_{\min}$
5) $\quad\mathcal{R} \leftarrow \mathrm{BuildHyperGraph}(G,\mathcal{C},\theta_i)$
6) $\quad S_i \leftarrow \mathrm{NodeSelection}(\mathcal{R})$
7) $\quad$根据式(7)和式(8)分别计算$\sigma^{\mathrm{lower}}(S_i)$，$\sigma^{\mathrm{upper}}(S_k^O)$
8) $\quad$if $\frac{\sigma^{\mathrm{lower}}(S_i)}{\sigma^{\mathrm{upper}}(S_k^O)} \geqslant \left(1-\frac{1}{\mathrm{e}}-\varepsilon\right) \| i==i_{\max})$ then
9) $\quad\quad$return $\mathcal{R}$ and S_i
10) $\quad$else
11) $\quad\quad\theta_{\min} \leftarrow 2\theta_{\min}$
12) $\quad$end if
13) end for

算法 4 为生成反向可达集集合的伪代码。具体流程如下。首先，将集合$\mathcal{H}$初始化为空集（算法 4 的步骤 1)）；接着，执行θ次 for 循环（算法 4 的步骤 2)～步骤 6)），每次随机选择节点$v\in\mathcal{C}$，沿该节点所有入边的相反方向进行广度优先遍历（即基于传播模型发起信息传播），将激活的节点依次插入反向可达集R_i，并将R_i插入$\mathcal{H}$；最后，输出$\mathcal{H}$（算法 4 的步骤 7)）。

算法 4 BuildHyperGraph 算法

输入 社交网络G，节点集$\mathcal{C}$，采样次数θ

输出 一个由θ个反向可达集构成的集合$\mathcal{H}$

1) 初始化$\mathcal{H} \leftarrow \varnothing$
2) for $i=1$ to θ do
3) $\quad$随机选择节点$v\in\mathcal{C}$
4) $\quad$沿节点v入边的相反方向执行广度优选遍历，并将激活的节点依次插入反向可达集R_i
5) $\quad$将R_i插入$\mathcal{H}$
6) end for
7) return $\mathcal{H}$

例 3（采样示例）IC 模型下采样阶段流程如下。假设采样次数$\theta=3$，参数$k=1$。预处理阶段从图 1(a)中筛选出有效节点集$\mathcal{C}=\{a,b,c,d,e\}$（详见例 2）。如图 2 所示，从$\mathcal{C}$中随机选点$b$、$c$、$e$，生成反向可达集集合$\mathcal{R}=\{R_1,R_2,R_3\}$，其中$R_1=\{b,c\}$，$R_2=\{a,c\}$，$R_3=\{b,c,d,e\}$。图 2 中的虚线表示遍历失败，灰线表示遍历成功，黑线表示社交网络g中的有向边。因为节点c对$\mathcal{R}$覆盖率最大，即节点c的影响力最大，所以将其加入种子集。最后，返回种子集$\{c\}$。

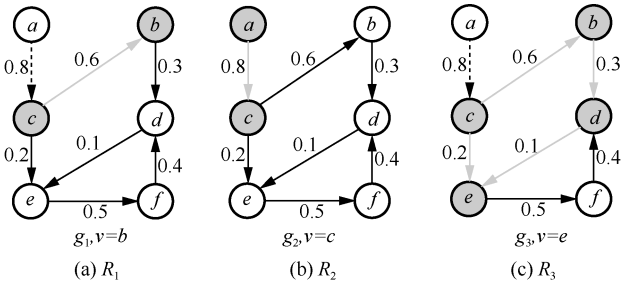


图 2 采样示例

2.3 基于影响力增量剪枝策略的种子选择

由于 RIS 方法每次找出对反向可达集集合$\mathcal{R}$覆盖率最高的种子节点后，需要更新其他节点对$\mathcal{R}$的覆盖率，并根据各节点对$\mathcal{R}$的覆盖率重新排序，导致存在节点对$\mathcal{R}$的覆盖率更新后相对排名不变时的无效排序，影响算法运行效率。针对该问题，本文提出基于影响力增量的剪枝策略。该策略的基本思想为保存前一轮排序后的数据，在删除该节点及其覆盖的反向可达集后，比较前一轮中原次大节点更新后和第三大节点更新前对$\mathcal{R}$的覆盖率，若前者大于等于后者，则直接选择原次大节点作为新一轮中的种子，而无须重新排序。利用该策略，可以剪枝部分种子选择时的无效排序，从而降低算法时耗。

种子选择阶段的主要工作如下。应用贪心策略找出对$\mathcal{R}$覆盖率最高的种子集S；同时，剪枝节点对$\mathcal{R}$的覆盖率更新后相对排名不变时的无效排序。

算法 5 为种子选择阶段的伪代码。具体地，首先，初始化种子集为空集（算法 5 的步骤 1)），计算出每个节点 $v \in V$ 对集合 $\mathcal{R}$ 的覆盖率 $F_{\mathcal{R}}(\{v\})$，保存至 pairs $< v, F_{\mathcal{R}}(\{v\}) >$（算法 5 的步骤 2)～步骤 5)）；接着，执行 k 次 for 循环，每次根据 $F_{\mathcal{R}}(\{v\})$ 对 pairs 降序排序，选择对 $\mathcal{R}$ 的覆盖率最高的节点，将其加入种子集，并删除该节点及其覆盖的反向可达集（算法 5 的步骤 6)～步骤 9)和步骤 12)）。同时，至多执行 $k-i$ 次 while 循环，每次保存前一轮排序结果，比较前一轮中次大节点更新后以及第三大节点更新前对 $\mathcal{R}$ 的覆盖率，若前者大于等于后者，则直接选择原次大节点作为新一轮的种子；重复比较，直至不满足该关系或者种子个数达 k 时结束循环（算法 5 的步骤 10)～步骤 21)）；最后，将种子集 S 输出（算法 5 的步骤 22)）。

算法 5　NodeSelection 算法（种子选择阶段）

输入　反向可达集集合 $\mathcal{R}$，种子个数 k

输出　种子集 S

1) 初始化　$S \leftarrow \varnothing$, pairs $\leftarrow \varnothing$
2) foreach $(u \in V)$ do //计算节点影响力
3)　　u.inf $= F_{\mathcal{R}}(\{u\})$
4)　　将 $< v, F_{\mathcal{R}}(\{v\}) >$ 插入 pairs
5) end for
6) for $i = 1$ to k do　//寻找种子
7)　　根据 pairs[i].second 对 pairs 降序排序
8)　　$u \leftarrow$ pairs[0].first
9) 将 u 插入 S
10)　　while $(i < k$ && pairs.size() $\geqslant 3)$ do
11)　　pairs' $\leftarrow$ pairs
12)　　删除 u 及其覆盖的反向可达集，并更新 pairs
13)　　if (pairs'[2].second $\leqslant$ pairs[0].second) then
14)　　将 pairs[0].first 插入 S
15)　　$u \leftarrow$ pairs[0].first
16)　　i++
17)　　else
18)　　break
19)　　end if
20)　　end while
21) end for
22) return S

例 4（种子选择示例）令 $F_{\mathcal{R}}^{(i)}(\{v\})$ 表示第 i 轮中节点 v 对集合 $\mathcal{R}$ 的覆盖率。如图 3 所示，第 1 轮已计算出每个节点对 $\mathcal{R}$ 的覆盖率，降序排序后选择当前对 $\mathcal{R}$ 覆盖率最大的节点 b 作为种子，并更新其他节点影响力。进入第 2 轮，先比较前一轮中原次大节点 c 更新后和原第三大节点 a 更新前对 $\mathcal{R}$ 的覆盖率，发现 $F_{\mathcal{R}}^{(2)}(\{c\}) \geqslant F_{\mathcal{R}}^{(1)}(\{a\})$，则节点 c 必为新一轮的最优种子，直接选择节点 c 而无须重新排序。

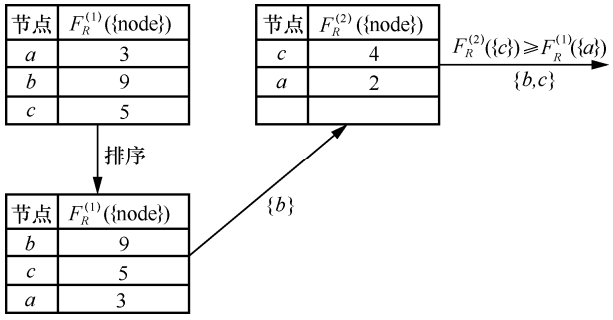


图 3　种子选择示例

2.4　MTIM 算法时间复杂度分析

预处理阶段（算法 2）主要用于筛选有效节点，其时间复杂度为 $O(n(n\text{cnt}+\log n))$，其中 cnt 为迭代次数。采样阶段（算法 3）主要用于生成反向可达集，其时间复杂度为 $O\left(\frac{k(n+m)\log n}{\varepsilon^2}\right)$。种子选择阶段（算法 5），所花时间主要与反向可达集集合 $\mathcal{R}$ 和采样次数 θ 相关，其时间复杂度为 $O\left(\sum_{i=1}^{\theta}|R_i|\right)$。组合 3 个阶段后得到的 MTIM 算法（算法 1），其时间复杂度为 $O\left(n(n\text{cnt}+\log n)+\frac{k(n+m)\log n}{\varepsilon^2}\right)$。

3　实验与分析

3.1　实验设置

实验的硬件配置 Intel(R) Xeon(R) Silver 4208 CPU @ 2.10 GHz，运行内存 64 GB，操作系统 Ubuntu 20.04（64 位），所有算法均采用 C++实现并使用 G++4.8.5 编译。

实验采用 4 个真实的社交网络数据集。其中，Slashdot 是提供技术资讯服务的社交网络；soc-LiveMocha 是提供外语学习服务的社交网络；Web-BerkStan 是 BerkStan 的社交网络；soc-pokec 是斯洛伐克的社交网络。

表 2 给出了数据集的统计信息。其中，|*V*|表示图中的节点个数，|*E*|表示图中的边个数，Avg.deg 表示图的平均度。

表 2 数据集的统计信息

数据集	|*V*|	|*E*|	Avg.deg
Slashdot	77 360	905 468	11.7
soc-LiveMocha	104 103	2 193 083	21.7
Web-BerkStan	685 230	7 600 595	11.1
soc-pokec	1 632 804	22 301 964	13.7

3.2 算法性能比较分析

算法评价标准包括：①运行时间，即求解出种子集的时间；②预期影响范围，即求解出的种子集能够影响到的节点个数。

实验使用 IC 模型和 LT 模型，基于 4 个社交网络数据集分别在 $k\in\{1,10,20,50,100\}5$ 种规模下实验。根据之前的工作[10-15, 17]，设置误差参数 $\varepsilon=0.5$。根据表 3，不同数据集单位时间筛选节点总数在 $r=70$ 时最大，因而预处理阶段的筛选比例 r% 设置为 70%。为避免误差，本文所涉及的算法都运行 30 次，各算法评价指标取其均值。

为验证算法高效性，设置对比算法，具体如下。

1) TIM 算法[14]，为贪心算法。TIM 算法基于反向影响采样技术，利用切尔诺夫边界确定采样次数，支持 LT 模型，可应用于大规模社交网络。本文将该算法用于 LT 模型下对比实验。

2) IMM 算法[15]，为贪心算法。IMM 算法是 TIM 算法的改进算法，利用鞅技术确定采样次数，同时支持 IC 模型和 LT 模型。本文以 IMM 算法为基准，并同时用于 IC 模型和 LT 模型下对比实验。

3) OneHop 算法[18]，为启发式算法。OneHop 算法基于跳步思想选取种子，支持 IC 模型，是目前精确度最高的启发式算法。本文将该算法用于 IC 模型下对比实验。

4) PMC 算法[19]，为贪心算法。PMC 算法基于蒙特卡罗模拟技术，支持 IC 模型。该算法将原图随机切割为 τ 个子图，在子图中进行 $\mathcal{T}$ 次传播模拟来估计节点影响力，选择前 k 大的节点作为种子。本文设置 $\tau=200$，$\mathcal{T}=10\,000$，并将该算法用于 IC 模型下对比实验。

5) DegreeDiscount 算法[20]，为经典启发式算法。DegreeDiscount 算法基于折扣度思想选取种子，支持 LT 模型。本文将该算法用于 LT 模型下对比实验。

6) MTIM 算法，为本文算法。

3.2.1 IC 模型下的结果

第一组实验。基于 IC 模型比较了 MTIM、IMM、OneHop、PMC 这 4 种算法在不同数据集上的预期影响范围，结果如图 4 所示。根据图 4 可以发现，1) 随着种子集规模 k 的增大，4 种算法的预期影响范围总体均呈上升趋势，并且种子集预期影响范围的增幅随种子个数的增加而递减。2) MTIM 的预期影响范围最广，IMM 和 PMC 次之，而 OneHop 表现最差。具体而言，MTIM 算法的预期影响范围较 IMM 算法提高了约 20%，IMM 算法的预期影响范围较 PMC 算法提高了 10%～20%，而 OneHop 算法的预期影响范围为 IMM 算法的 50%左右。

其原因主要是节点度筛选策略的应用。该策略通过约束采样范围，提高种子集 S 对反向可达集集合 $\mathcal{R}$ 的覆盖率 $F_{\mathcal{R}}(S)$，从而扩大影响范围。根据式(1)，$\mathbb{E}[\sigma(S)]=\dfrac{n\Lambda(S)}{|\mathcal{R}|}=nF_{\mathcal{R}}(S)$，即种子集预期影响范围 $\mathbb{E}[\sigma(S)]$ 与覆盖率 $F_{\mathcal{R}}(S)$ 正相关。图 5 统计了 $k=100$ 时 IMM 算法和 MTIM 算法在各数据集上的覆盖率，可以发现 MTIM 算法较 IMM 算法覆盖率提高约 20%。因而，MTIM 算法较 IMM 算法预期影响范围提高约 20%。而 PMC 算法由于需要对网络中的所有节点进行多次传播模拟，取预期影响范围平均值来估计节点影响力，导致实际结果不够精确。OneHop 启发式算法基于跳步思想选择种子，并未考虑复杂网络结构，导致所选节点质量不高。因而，IMM 算法、PMC 算法和 OneHop 算法的预期影响范围均不如 MTIM 算法。

表 3 不同筛选比例 *r*%下的筛选节点个数和筛选时间比较

数据集	$r=60$		$r=70$		$r=80$		$r=90$	
	$\lvert\mathcal{C}\rvert$	时间/s	$\lvert\mathcal{C}\rvert$	时间/s	$\lvert\mathcal{C}\rvert$	时间/s	$\lvert\mathcal{C}\rvert$	时间/s
Slashdot	59 755	4.029	69 105	4.404	75 860	5.097	77 315	10.751
soc-LiveMocha	81 506	5.299	89 459	5.549	96 683	6.253	103 164	9.368
Web-BerkStan	590 801	14.121	632 038	14.642	650 509	15.757	668 825	20.728
soc-pokec	1 178 735	30.886	1 301 817	32.184	1 476 290	37.013	1 632 804	49.413

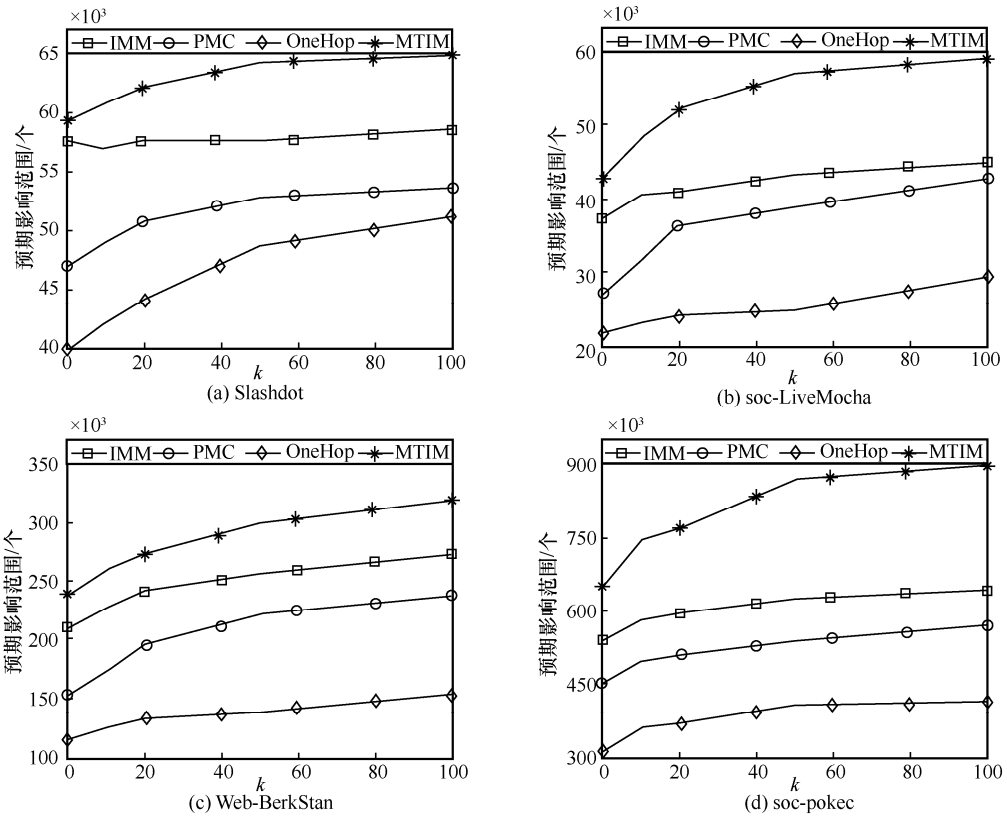


图 4　IC 模型下的预期影响范围比较

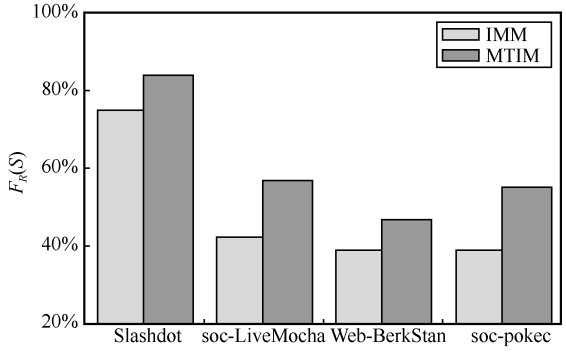


图 5　覆盖率统计信息

第二组实验。基于 IC 模型比较了 MTIM、IMM、OneHop、PMC 这 4 种算法在不同数据集上的运行时间，结果如图 6 所示。根据图 6 可以发现，1) 随着种子集规模 k 的增大，MTIM 算法、IMM 算法和 PMC 算法的运行时间及其差距倍增，而 OneHop 算法运行时间趋于平稳。2) OneHop 的运行时间最短，MTIM 次之，PMC 和 IMM 的运行时间较长。具体而言，当 $k < 20$ 时，MTIM 算法与 OneHop 算法的性能差异不大；当 $k > 20$ 时，MTIM 算法略慢于 OneHop 算法。MTIM 算法较 IMM 算法快了 4～9 倍，且数据集规模越大，提升效果越明显。PMC 算法较 IMM 算法减少运行时间约 50%。

其原因主要有两点。1) 边界约束策略的应用。该策略利用更高精度的边界约束来估计最优采样次数，降低确定采样次数时间，从而提升算法的运行效率。图 7(a)统计了 $k=100$ 时 IMM 算法和 MTIM 算法在各数据集上的确定采样次数时间，由于各数据集的运行时间相差较大，因此将时间设置为 $T=\lg y$，其中 y 表示实际运行时间，可以发现：MTIM 算法较 IMM 算法减少确定采样次数时间 40%～50%。图 7(b)统计了 $k=100$ 时 IMM 算法和 MTIM 算法生成的反向可达集个数（即采样次数），可以发现，MTIM 算法所确定的采样次数约为 IMM 算法的 70%。2)影响力增量剪枝策略的应用。该策略避免了部分种子选择时的无效排序。图 7(c)统计了 $k=100$ 时 IMM 算法和 MTIM 算法在各数据集上种子选择时的有效排序次数，可以发现：MTIM 算法较之 IMM 算法剪枝了约 15%的无效排序。因而，MTIM 算法较之 IMM 算法快了 4～9 倍。而 PMC 算法将原社交网络随机分割为多个较小规模的子图网络，从子图中选择种子。因而，PMC 算法快于 IMM 算法。OneHop 算法基于启发式规则粗略估计节点影响力，直接选择前 k 大的节点作为种子。因而，OneHop 算法的运行速度总体优于 MTIM 算法、IMM 算法和 PMC 算法。

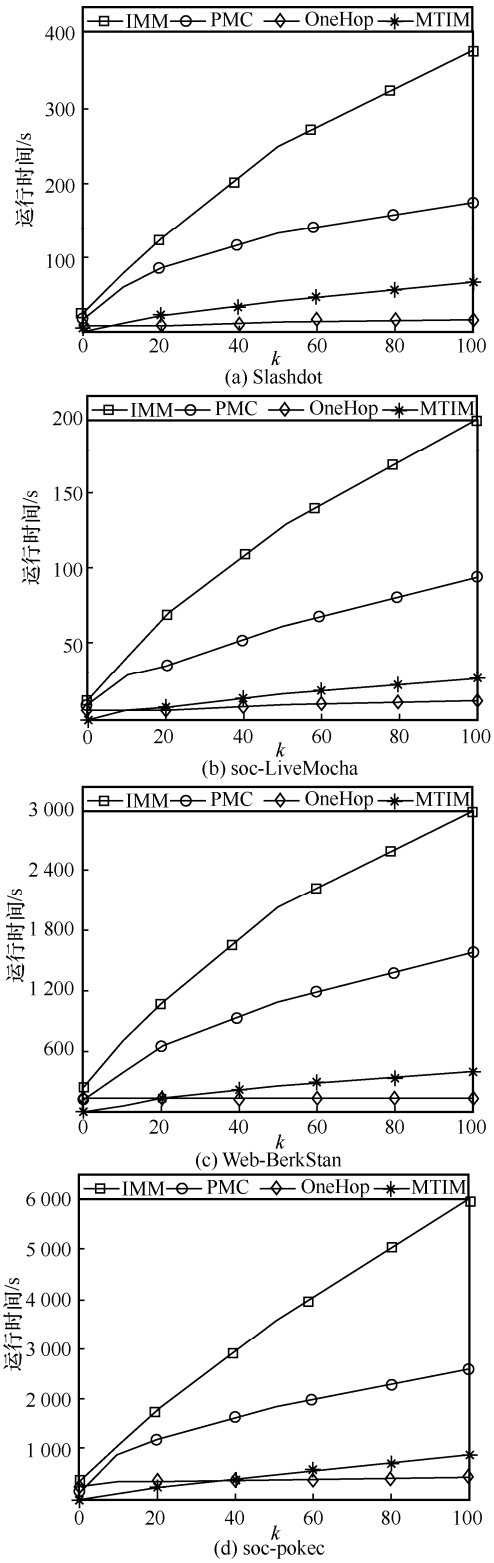


图 6 IC 模型下的运行时间比较

3.2.2 LT 模型下的结果

第一组实验。基于 LT 模型比较了 MTIM、IMM、TIM、DegreeDiscount 这 4 种算法在不同数据集上的预期影响范围，结果如图 8 所示。根据图 8 可以发现，1) 随着种子集规模 k 的增大，4 种算法的预期影响范围总体呈上升趋势，且预期影响范围的增幅随种子个数的增加而递减。2) MTIM 的预期影响范围最广，IMM 和 TIM 次之，而 DegreeDiscount 表现最差。具体而言，MTIM 算法较 IMM 算法扩大预期影响范围约 30%，且数据集规模越大，提升效果越明显；IMM 算法和 TIM 算法折线几乎重合，性能差异不大；而 DegreeDiscount 算法的预期影响范围约为 IMM 算法的 50%。

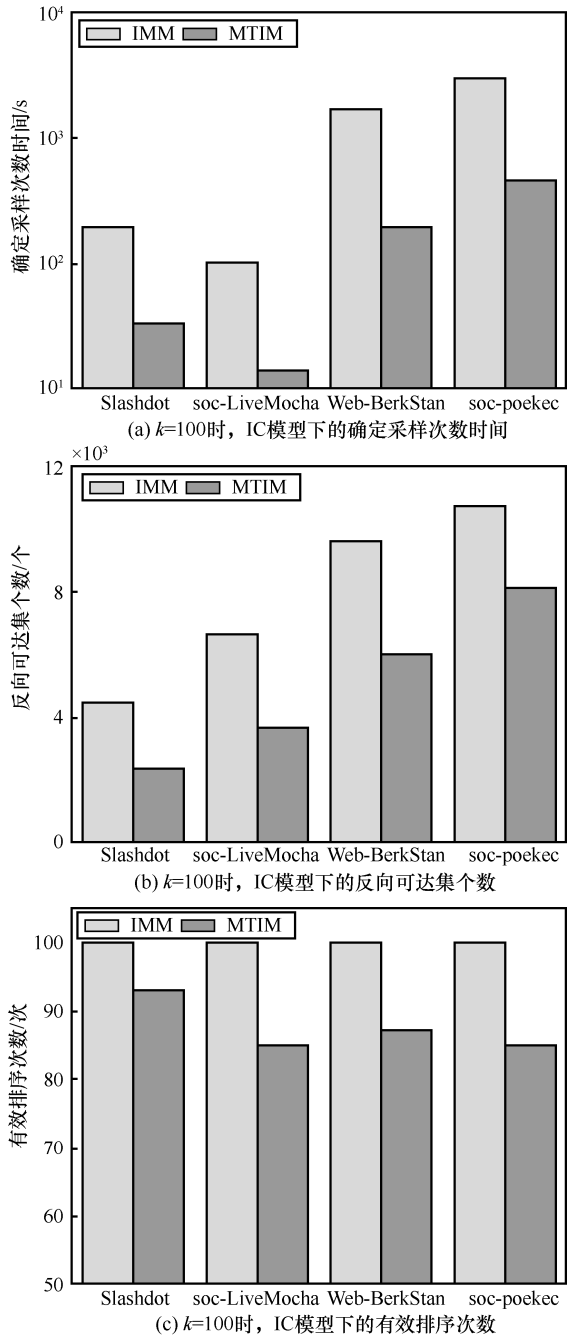


图 7 统计信息

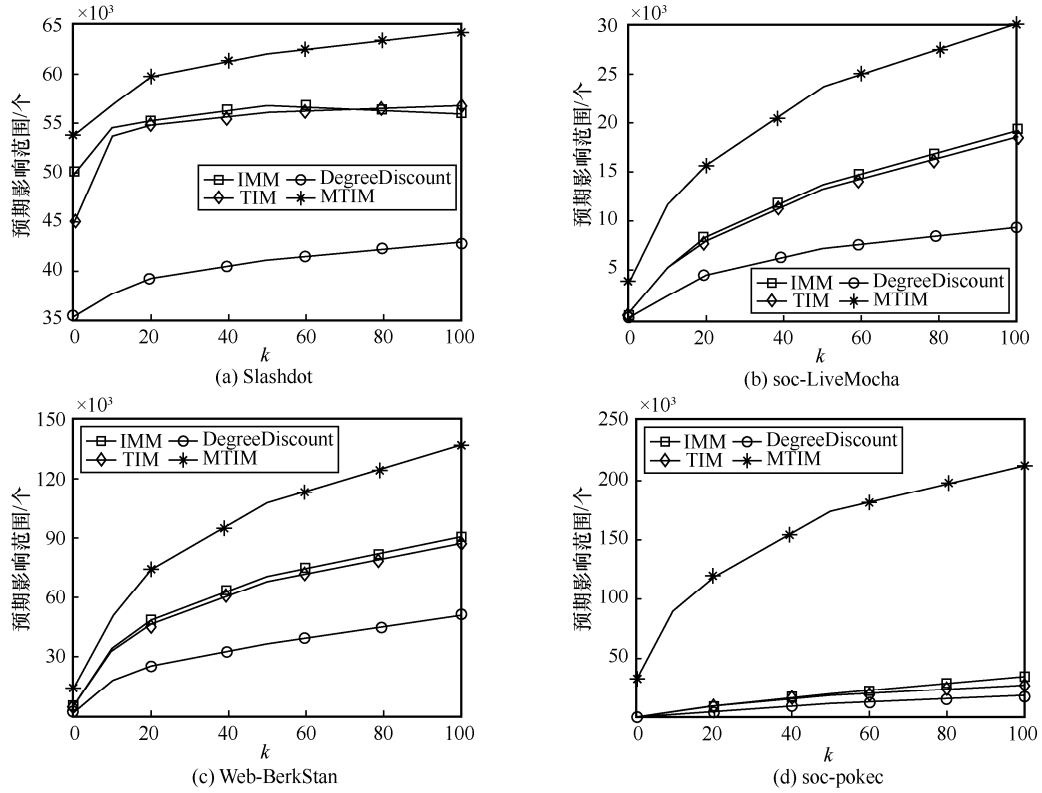


图 8　LT 模型下的预期影响范围比较

其原因与 IC 模型类似，主要是节点度筛选策略的应用。利用该策略，可以提高种子集对反向可达集的覆盖率，从而扩大种子集预期影响范围。因而，MTIM 算法较 IMM 算法扩大预期影响范围约 30%。IMM 算法是 TIM 算法的改进算法，在确保获得相同近似保证的同时，主要研究如何提升算法的运行速度。因而，IMM 算法和 TIM 算法的预期影响范围极为接近。DegreeDiscount 算法是经典启发式算法，基于折扣度思想来选择种子节点，并未考虑网络结构的复杂性。因而，该算法的预期影响范围远不如 MTIM 算法、IMM 算法和 TIM 算法。

第二组实验。基于 LT 模型比较了 MTIM、IMM、TIM、DegreeDiscount 这 4 种算法在不同数据集下的运行时间，结果如图 9 所示。根据图 9 可以发现，1) 随着种子集规模 k 的增大，DegreeDiscount、MTIM 和 IMM 的运行时间递增，并且增幅逐渐减小；而 TIM 在 $k<10$ 时运行时间递减，在 $k\geqslant 10$ 时运行时间递增，并且增幅逐渐减小。2) MTIM 运行时间最短，DegreeDiscount 和 IMM 次之，TIM 运行时间最长。具体而言，MTIM 算法略快于 DegreeDiscount 算法，MTIM 算法较 IMM 算法快了 1.5～2.3 倍，而 TIM 算法的运行时间为 IMM 算法的 2 倍多。

其原因与 IC 模型类似，是因为边界约束策略和影响力增量剪枝策略的应用。利用这 2 个策略，不仅可以快速确定近似最优采样次数，提升算法效率，而且可以避免部分种子选择时的无效排序，降低算法时耗。因而，MTIM 算法优于 TIM 算法和 IMM 算法，略快于 DegreeDiscount 启发式算法。但是，相比于 IC 模型，LT 模型下的优化效果并不显著，主要是因为节点信息的传播方式不同，IC 模型下反映的是单个用户与单个用户之间的影响关系，而 LT 模型下反映的是多个用户与单个用户之间的影响关系。

综合 2 个传播模型下的实验结果可知，1) 相比于 IMM、TIM 和 PMC 等贪心算法，以及 OneHop 和 DegreeDiscount 等启发式算法，MTIM 算法均获得最大预期影响范围，提供 $\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$ 近似保证，且种子集规模越大，优势越明显。2) 与 IMM、TIM 和 PMC 等贪心算法相比，MTIM 算法运行时间最短。而与启发式算法相比，在 IC 模型上，MTIM 算法略慢于 OneHop 算法；在 LT 模型上，MTIM 算

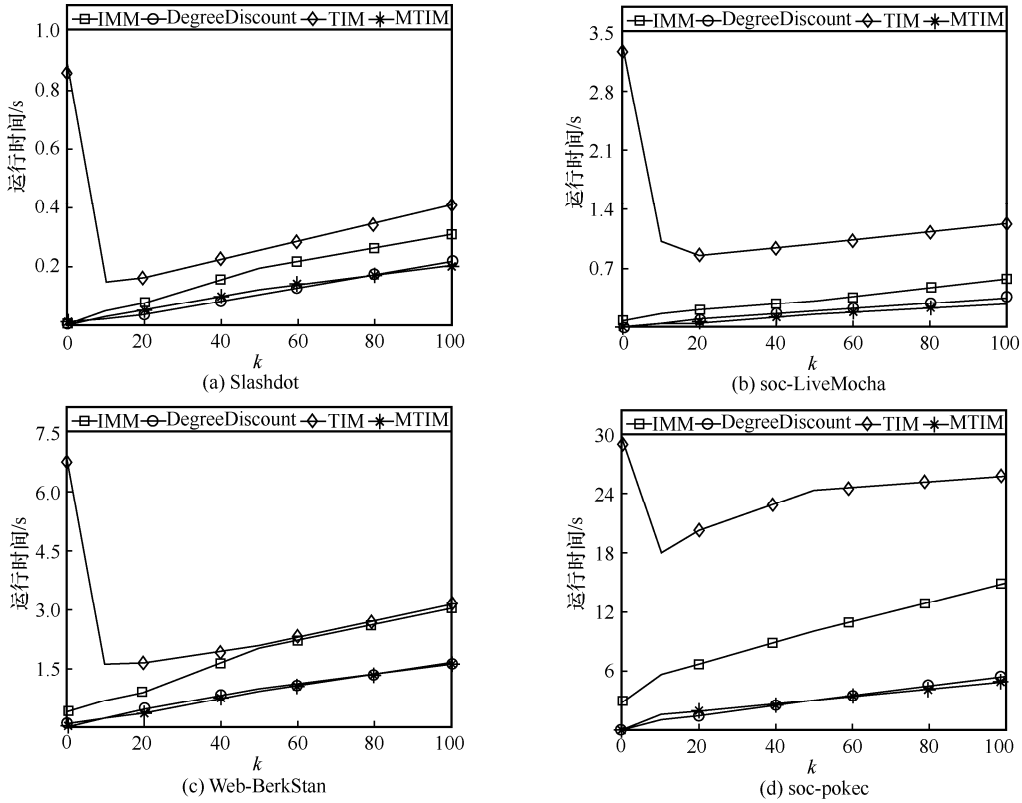


图 9　LT 模型下的运行时间比较

法运行速度与 DegreeDiscount 算法极为接近，总体略快于 DegreeDiscount 算法。3) MTIM 算法不仅预期影响范围更优、精确度更高，而且运行速度优于大多贪心算法，略快于部分启发式算法。因而，MTIM 算法能够更好地适用于大规模社交网络。

4　结束语

针对现有影响力最大化算法效率低、适用模型单一的问题，本文基于 2 个基础影响力传播模型，结合反向影响采样技术，提出了 MTIM 算法，该算法包括 3 个阶段。1) 预处理阶段：基于节点度筛选策略，筛选出有效节点集。2) 采样阶段：基于边界约束策略，确定近似最优采样次数并从有效节点集中选点采样。3) 种子选择阶段：应用贪心策略选择种子节点，并基于影响力增量剪枝策略，剪枝种子选择时的无效排序。基于 4 个真实社交网络的实验结果表明，MTIM 算法不仅可以提供$\left(1-\frac{1}{\mathrm{e}}-\varepsilon\right)$近似保证，而且其预期影响范围远高于 DegreeDiscount 和 OneHop 等启发式算法，优于 IMM、TIM、PMC 等贪心算法；在运行时间方面，MTIM 算法显著快于 IMM、TIM、PMC 等贪心算法，总体上略慢于 OneHop，略快于 DegreeDiscount。因而，MTIM 算法在拥有较快运行速度的同时，保证了较大预期影响范围、较高近似保证，能够更好地适用于大规模社交网络。

后续工作中将会进行如下深入研究。1) 影响力传播模型的扩展。进一步考虑特定的、复杂多变的应用场景下，如何解决影响力最大化问题。2) 动态图下的研究。实际情况下，社交网络的结构以及用户间的关系往往会随着消息的传播而发生一定变化，未来可以尝试在动态图上进行研究。

参考文献：

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

[2] LI Y C, FAN J, WANG Y H, et al. Influence maximization on social graphs: a survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(10): 1852-1872.

[3] PENG H, HUANG K F, YANG L X, et al. Dynamic maintenance strategy for word-of-mouth marketing[J]. IEEE Access, 2020, 8:

126496-126503.

[4] REN J F, LIU M T, LIU Y, et al. Optimal resource allocation with spatiotemporal transmission discovery for effective disease control[J]. Infectious Diseases of Poverty, 2022, 11(1): 34.

[5] PAUL A, WU Z F, LIU K, et al. Personalized recommendation: from clothing to academic[J]. Multimedia Tools and Applications, 2022, 81(10): 14573-14588.

[6] BINESH N, GHATEE M. Distance-aware optimization model for influential nodes identification in social networks with independent cascade diffusion[J]. Information Sciences, 2021, 581: 88-105.

[7] BLESA M J, GARCÍA-RODRÍGUEZ P, SERNA M. Forward and backward linear threshold ranks[C]//Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. New York: ACM Press, 2021: 265-269.

[8] ALI A, WANG H, KHAN A N. Mechanism to enhance team creative performance through social media: a transactive memory system approach[J]. Computers in Human Behavior, 2018, 91: 115-126.

[9] BORGS C, BRAUTBAR M, CHAYES J, et al. Maximizing social influence in nearly optimal time[C]//Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia: Society for Industrial and Applied Mathematics, 2014: 946-957.

[10] SUN L C, HUANG W R, YU P S, et al. Multi-round influence maximization[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM Press, 2018: 2249-2258.

[11] GUO J X, WU W L. Influence maximization: seeding based on community structure[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2020, 14(6): 1-22.

[12] GUO Q T, WEI Z W, et al. Influence maximization revisited: efficient reverse reachable set generation with bound tightened[C]//Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2020: 2167-2181.

[13] ARORA A, GALHOTRA S, et al. Debunking the myths of influence maximization: an in-depth benchmarking study[C]//Proceedings of the 2017 ACM International Conference on Management of Data. New York: ACM Press, 2017: 651-666.

[14] TANG Y Z, XIAO X K, et al. Influence maximization: near-optimal time complexity meets practical efficiency[C]//Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2014: 75-86.

[15] TANG Y Z, SHI Y C, XIAO X K. Influence maximization in near-linear time: a martingale approach[C]//Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2015: 1539-1554.

[16] TAKAI Y, MIYAUCHI A, IKEDA M, et al. Hypergraph clustering based on PageRank[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM Press, 2020: 1970-1978.

[17] TANG J, TANG X Y, et al. Online processing algorithms for influence maximization[C]//Proceedings of the 2018 International Conference on Management of Data. New York: ACM Press, 2018: 991-1005.

[18] TANG J, TANG X Y, YUAN J S. An efficient and effective hop-based approach for influence maximization in social networks[J]. Social Network Analysis and Mining, 2018, 8(1): 1-19.

[19] OHSAKA N, AKIBA T, YOSHIDA Y, et al. Fast and accurate influence maximization on large networks with pruned Monte-Carlo simulations[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2014, 28(1): 138-144.

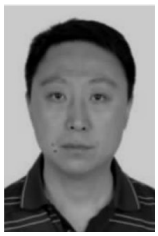
[20] CHEN W, WANG Y J, YANG S Y. Efficient influence maximization in social networks[C]//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2009: 199-208.

[作者简介]

王璿（1977− ），女，黑龙江齐齐哈尔人，博士，东华大学副教授，主要研究方向为数据查询、生物信息处理、分布式并行计算。

张瑜（1997− ），男，江苏泰州人，东华大学硕士生，主要研究方向为社交网络。

周军锋（1977− ），男，陕西西安人，博士，东华大学教授，主要研究方向为图数据的查询处理技术、推荐系统关键技术。



陈子阳（1973− ），男，黑龙江五常人，博士，上海立信会计金融学院教授，主要研究方向为数据库理论与技术。