

影响最大化问题中基于 K-truss 的投票改进算法

孙飞翔, 陈卫东, 林天森

(华南师范大学 计算机学院, 广州 510631)

摘要: 在社交网络的影响最大化(IM)问题中, 近似算法通过大量的 Monte-Carlo 模拟计算节点集的影响范围, 导致时间复杂度提高, 而多数启发式算法在具有不同拓扑结构的图上存在稳定性较差的问题。提出基于 K-truss 的改进投票算法 TrussVote。在投票阶段, 通过引入 K-truss 的相关理论及算法定义节点的有效投票能力, 用于表示节点对其不同邻居的投票倾向, 同时在计算得票分数时考虑边的传播概率, 提高解决 IM 问题的效率。在每轮投票结束后, 将得票分数最高的节点选为种子节点。在更新阶段, 结合节点间的相似性指标定义衰减因子, 以有效区分邻居节点投票能力的弱化程度。此外, 基于 IC 模型下的原始传播结果, 提出传播差异作为传播范围的等价分析指标。在不同规模真实网络数据集上的实验结果表明, 相比 RNR、VoteRank++ 等算法, 该算法不仅能有效降低时间复杂度, 而且可在最短的时间内感染更多的节点, 具有广泛的影响范围。

关键词: 社交网络; 影响最大化; 投票算法; K-truss 分解; IC 模型; SIR 模型

开放科学(资源服务)标志码(OSID):



中文引用格式: 孙飞翔, 陈卫东, 林天森. 影响最大化问题中基于 K-truss 的投票改进算法[J]. 计算机工程, 2022, 48(11): 291-298.

英文引用格式: SUN F X, CHEN W D, LIN T S. Improved voting algorithm based on K-truss for influence maximization problem[J]. Computer Engineering, 2022, 48(11): 291-298.

Improved Voting Algorithm Based on K-truss for Influence Maximization Problem

SUN Feixiang, CHEN Weidong, LIN Tiansen

(School of Computer Science, South China Normal University, Guangzhou 510631, China)

[Abstract] In the Influence Maximization(IM) problem of social networks, approximation algorithms are used to calculate the impact range of node sets via a significant amount of Monte-Carlo simulations, thus resulting in an increase in time complexity. Most heuristic algorithms exhibit unsatisfactory stability on graphs with different topological structures. This study proposes an improved voting algorithm, TrussVote, which is based on K-truss. In the voting stage, the effective voting ability of a node is defined by introducing the relevant theory and algorithm of K-truss, which is used to represent the voting tendency of a node to its different neighbors. Additionally, the propagation probability of edges is considered when calculating the voting score to improve the efficiency of the algorithm in solving IM problems. After each round of voting, the node with the highest voting score is selected as the seed node. In the update phase, the attenuation factor is defined based on the similarity index between nodes to effectively distinguish the weakening degree of neighboring nodes' voting ability. In addition, based on the original propagation results under the IC model, the Diffusion Difference(DD) is proposed as an analysis index equivalent to the propagation range. Experimental results on real network datasets of different scales show that compared with RNR, VoteRank++, and other algorithms, the proposed algorithm not only effectively reduces the time complexity, but also infects more nodes in the shortest duration and has a wider range of influence.

[Key words] social network; Influence Maximization(IM); voting algorithm; K-truss decomposition; IC model; SIR model

DOI: 10. 19678/j. issn. 1000-3428. 0063033

0 概述

随着微信、Facebook 和 Twitter 等在线社交网络的兴起, 越来越多的人使用不同的社交网络进行交流

和分享, 使得网络中信息扩散研究成为社交网络分析的热点。影响最大化(Influence Maximization, IM)作为信息扩散研究中的关键问题, 具有较广泛的应用场景。文献[1-3]分别把 IM 问题应用于病毒式营销、谣

基金项目: 国家自然科学基金(61370003)。

作者简介: 孙飞翔(1995—), 男, 硕士研究生, 主研方向为复杂网络分析; 陈卫东, 教授、博士; 林天森, 硕士研究生。

收稿日期: 2021-10-25 修回日期: 2022-01-03 E-mail: 1258305030@qq.com

言控制和社会计算领域,将社交网络建模为一个图 G , 并给图 G 的每条边分配一个传播概率。IM 问题是给定一个传播模型及正整数 k , 从节点集 V 中选择一个不多于 k 个节点的种子集合 S , 使得 S 的影响范围 $\sigma(S)$ 达到最大。

文献[4]证明 IM 问题也是 NP-hard 问题, 表明 IM 问题在多项式时间内不可能得到最优解的精确算法, 其求解方法主要是近似算法和启发式算法。文献[5-6]针对 IM 问题, 提出基于次模函数的贪心近似算法, 具有较大的时间复杂度, 因此, 该算法在规模较大的图上难以得到实际有效的应用。以上贪心近似算法均具有较大的时间复杂度, 其原因为在节点选择过程中, 每个种子节点的选择均使用大量的 Monte-Carlo 模拟来计算节点集的影响范围。为了更快地求解 IM 问题, 启发式算法利用不同的策略来代替模拟过程。此类算法分为基于中心性的算法、基于社区的算法和基于排序的算法。启发式算法与近似算法相比具有较低的时间复杂度, 但是在具有不同拓扑结构的图上, 启发式算法和近似算法都缺乏稳定性。文献[7]提出基于 VoteRank 的投票算法, 在求解 IM 问题时具有较强的稳定性。在该类算法中, 节点 v 具有投票能力 va_v 和得票分数 vs_v 两种属性, 其中 va_v 表示节点 v 能够给邻居节点投票的票数, vs_v 表示节点 v 的邻居给其投票的票数总和。此外, 在每轮投票中得票分数最高的节点将不参与后续的投票, 即投票能力为 0, 其邻居节点的投票能力也会随之得到一定程度的弱化。投票算法通常定义一个衰减因子 f 来表示投票能力的弱化程度。然而该算法并未有效地区分节点对其不同邻居的投票能力, 种子节点所有的邻居均使用相同的衰减因子, 不能有效地表示出它们投票能力的弱化程度。

本文基于 K-truss 提出一种改进的投票算法 TrussVote, 用于求解 IM 问题。基于边的 truss 值设计节点有效投票能力的计算方法, 该方法不仅考虑网络拓扑结构对投票能力的影响, 同时也有效地区分节点对其不同邻居的投票能力。基于节点的相似性提出动态的衰减因子, 表示节点邻居的投票能力具有不同程度的弱化。此外, 本文结合 IC 模型下的原始传播结果提出影响范围的等价分析指标——传播差异 (Diffusion Difference, DD), 以直观地区分各个算法的实验效果。

1 相关工作

为了表述方便, 表 1 所示为本文所需主要变量的说明。

表 1 重要变量的说明

Table 1 The description of important variables

变量	描述	变量	描述
n	节点数	d_v	节点 v 的度
m	边数	\bar{d}	图 G 的平均度
N_v	节点 v 的邻居	(v,u)	图 $G=(V,E)$ 中边 $(v,u) \in E$
V'	最大度的节点集	T'	社区合并的迭代次数
q	排序稳定的迭代次数	h	得票分数需更新的个数

网络中节点的中心性是衡量节点重要性的关键指标, 中心性较高的节点在网络中通常占据着信息传播的关键位置。节点的中心性分为局部中心性和全局中心性。文献[8-10]和文献[11-12]分别基于局部中心性和全局中心性提出不同的启发式算法, 以求解 IM 问题。此类算法的主要原理是在计算每个节点的中心性之后, 选择中心性最高的前 k 个节点作为种子节点集。相比基于局部中心性的启发式算法, 基于全局中心性的启发式算法在获得较优实验效果的同时提高了算法的时间复杂度。

影响力的扩散与社区结构有着密切关联, 并且在社区内节点之间联系紧密, 更容易传播影响力。文献[13-14]基于图的社区提出用于求解 IM 问题的方法, 其原理是将网络中的节点划分为不同的社区, 并在每个社区中选取影响力较大的节点作为种子节点。文献[15-16]提出基于排序的算法, 采用不同方式定义节点的影响力, 通过迭代方式使节点按照影响力的排序达到一个稳定状态。该类算法与基于中心性的算法相比能够获得更广泛的影响力覆盖范围, 但迭代次数的增加也提升了算法的时间复杂度。

文献[7]提出一种新的解决方案用来求解 IM 问题, 它最早基于投票机制提出 VoteRank 算法。在初始化阶段, 每个节点的投票能力均被设置为 1, 在投票过程中, 每轮选出一个得票分数最高的节点作为种子节点, 在此基础上, 使用固定的衰减因子弱化该种子节点一跳邻居的投票能力。当权重表示节点间联系紧密性的加权网络时, 文献[17]基于 VoteRank 提出 WVoteRank 算法, 在计算节点的得票分数时, 权重的差异使得节点对其不同邻居的投票能力产生差异。文献[18]设计的 NCVoteRank 算法认为节点的投票能力取决于其在网络中的拓扑结构, 并提出邻居核心值 (Neighborhood Coreness, NC) 的概念, 当计算节点的得票分数时, 综合考虑了邻居节点的 NC 值及投票能力。文献[19]提出 EnRenew 算法, 采用信息熵的方式来表示节点的重要性及投票能力。文献[20]不仅引入投票比例的概念以区分节点对其不同邻居的投票能力, 而且在更新种子节点邻居的投票能力时, 采用不同程度的衰减因子分别弱化其一跳和二跳邻居的投票能力。

2 本文算法

本文基于第 1 节中的投票机制提出 TrussVote 算法, 该算法在投票阶段使用 K-truss 的相关理论及算法区分节点对其不同邻居的投票能力, 在更新阶段通过节点相似性指标定义一个动态的衰减因子, 用于表示种子节点邻居的投票能力具有不同程度的衰减。

2.1 K-truss 分解

K-core^[21]揭示了网络的结构性质和层次性质, 而 K-truss 则是 K-core 基于三角形的一种扩展, 它所推导的子图在结构上更接近一个团^[22]。

定义 1(边的支持度 $\text{sup}(e, G)$) 边的支持度等于图 $G = (V, E)$ 中边 $e \in E$ 所在三角形的个数。

定义 2(K-truss 子图 T_K) 若 $T_K = (V_{T_K}, E_{T_K})$ 满足 $K \geq 2, \forall e \in E_{T_K}$ 且 $\text{sup}(e, G) \geq K-2$, 则称 T_K 为 G 的 K-truss 子图。

定义 3(边 $e=(v, u)$ 的 truss 值 t_{vu}) $t_{vu} = \max \{K: e \in E_{T_K}, T_K \text{ 为图 } G \text{ 的 K-truss 子图}\}$ 。

定义 4(边集 K-class, 记为 Φ_K) 图 $G = (V, E)$ 的 $\Phi_K = \{e=(v, u) | e \in E, t_{vu} = K\}$ 。

本文的投票算法通过引入 K-truss 相关理论, 不仅考虑网络的拓扑结构对投票能力产生的影响, 同时有效地区分节点对其不同邻居拥有不同的投票倾向。对于边 $e=(v, u)$, t_{vu} 的计算与 v 和 u 的公共邻居相关, 因此边的 truss 值在一定程度上反映了节点间联系的紧密程度。对于同一节点的不同邻居, 邻边的 truss 值越大, 其两端节点的联系越紧密。无向图示例如图 1 所示。

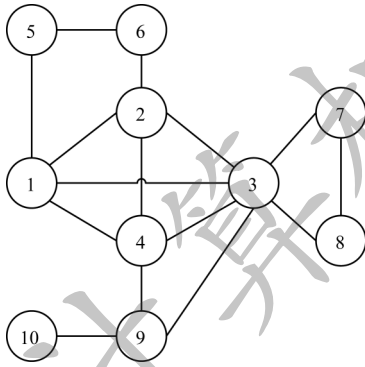


图 1 无向图示例

Fig.1 The example of undirected graph

在节点 3 的邻边中, 边 $(3, 4)$ 具有最大的 truss 值, 它们的联系也更加紧密。基于网络的核分解理论^[23], K-truss 分解与 K-core 分解相似, K 值越大的子图在网络中的相对位置越趋于核心。对于网络中不同位置的节点, truss 值可以对节点的投票能力进行有针对性和不同程度的放大。从图 1 可以看出, Φ_3 和 Φ_1 中的边对它们两端节点投票能力的放大程度不同, 在同一轮投票中趋于中心位置的节点将会获得更高的得票分数。

2.2 有效投票能力

节点间相似性差异会导致节点对其邻居的投票能力不同。为了更准确地表示这种差异, 本文基于边的 truss 值提出有效投票能力 $e_v^{\text{eva}}(u)$, 用于计算节点 v 对其邻居节点 $u \in N_v$ 的投票能力。当不考虑节点间的相似性时, 节点 v 给其每个邻居投票的概率均为 $1/d_v$; 当考虑节点间的相似性时, 本文基于边的 truss 值定义了节点 v 对其邻居节点 $u \in N_v$ 的有效投票能力, 其计算过程如式(1)所示:

$$e_v^{\text{eva}}(u) = \frac{v_v^{\text{va}} \cdot t_{vu}}{d_v} \quad (1)$$

2.3 得票分数

本文在计算节点的得票分数时, 利用有效投票能力来表示节点对其邻居拥有不同的投票票数, 而这种有效的投票能力也会受到边的传播概率的影响。在实际传播过程中, 节点 v 通过边 (v, u) 将 u 激活, 其激活概率大于该边的传播概率。因此, 基于传播模型的随机性特点, 本文将投票算法中能够得到的有效投票能力乘以传播概率, 用来表示 u 能够获得 v 投票的票数期望值。对于节点 v , $e_v^{\text{eva}}(u) \cdot p_{uv}$ 表示能够得到其邻居节点 $u \in N_v$ 的实际票数, 其中 p_{uv} 为边 (u, v) 的传播概率。与其他投票算法相似, 本文节点 v 的得票分数来源于邻居节点的投票, 其计算过程如式(2)所示:

$$v_v^{\text{vs}} = \sum_{u \in N_v} e_v^{\text{eva}}(u) \cdot p_{uv} \quad (2)$$

2.4 衰减因子

本文基于 ra 指标^[24]提出一种动态的衰减因子, 以区分种子节点不同邻居的衰减情况。ra 指标是一种基于资源分配策略的节点相似性指标, $\text{ra}(v, u)$ 表示节点 v, u 之间的相似度, 也表示 v 接收到 u 资源的能力。其计算过程如式(3)所示:

$$r_{\text{ra}}(v, u) = \sum_{w \in N_v \cap N_u} \frac{1}{d_w} \quad (3)$$

当节点 v 被选为种子节点后, 其邻居节点 $u \in N_v$ 需去除来自 v 的资源, 本文将减去的这部分视为节点 u 投票能力的衰减程度 f_u , 其计算过程如式(4)所示:

$$f_u = 1 - r_{\text{ra}}(v, u) \quad (4)$$

当节点 v 被选为种子节点后, 其邻居节点 $u \in N_v$ 的投票能力可根据式(5)计算:

$$v_v^{\text{va}} = f_u \cdot v_v^{\text{va}} \quad (5)$$

2.5 复杂度分析

TrussVote 的算法过程: 首先, 在初始化阶段, 计算所有边的 truss 值, 并且给每个节点都赋予相同的初始投票能力; 然后, 重复 k 轮投票, 每次均将投票分数最高的节点选为种子节点, 并使用动态的衰减因子更新其邻居节点的投票能力。TrussVote 如算法 1 所示。

算法 1 TrussVote 算法

输入 图 $G(V, E)$, 初始集合大小 k

输出 初始传播集合 S

1. $S \leftarrow \emptyset$
2. 计算边的 truss 值
3. for $v \in V$ do
4. $va_v \leftarrow 1$
5. end for
6. for $v \in V$ do
7. $vs_v \leftarrow 0$
8. for $u \in N_v$ do
9. $vs_v \leftarrow vs_v + e_v^{\text{eva}}(u) \cdot p_{uv}$
10. end for
11. end for
12. 将得票分数最高的节点 v 加入到 S


```

13. while |S| < k do
14.   移除节点 v 的得票分数
15.   H ← H ∪ Nv
16.   for u ∈ Nv do
17.     vu ← vu · fu
18.     H ← H ∪ Nu
19.   end for
20.   va ← 0
21.   for u ∈ H do
22.     vs ← 0
23.     for w ∈ Nu do
24.       vs ← vs + evaw(u) · pwu
25.     end for
26.   end for
27.   将得票分数最高的节点 v 加入到 S
28. end while
29. return S

```

在算法 1 中,步骤 2 是边 truss 值的计算阶段,计算每个节点邻边 truss 值的时间复杂度为 $O(\bar{d}^2)$,因此本阶段的时间复杂度为 $O(n\bar{d}^2)$ 。步骤 3~步骤 11 是投票能力与得票分数的初始化阶段,投票能力初始化的时间复杂度为 $O(n)$,得票分数初始化的时间复杂度为 $O(n\bar{d}^2)$,因此,该阶段的时间复杂度为 $O(n\bar{d}^2)$ 。步骤 12~步骤 28 是投票能力与得票分数的更新阶段:首先,选出得票分数最高的节点 v,并更新 N_v 的投票能力,时间复杂度为 $O(\bar{d})$;其次,更新 N_v 及其邻居的得票分数,时间复杂度为 $O(\bar{d}^2)$ 。因此,投票能力与得票分数更新阶段的时间复杂度为 $O(k\bar{d}^2)$ 。由此可得,算法总的时间复杂度为 $O(n\bar{d}^2 + n\bar{d}^2 + k\bar{d}^2) = O(n\bar{d}^2)$ 。

3 实验设计

本文所有的算法均使用 Python 语言来实现。运行环境为 Windows10 操作系统,内存 8 GB,处理器 Intel® Core™ i7-7700 CPU @3.60 GHz。

3.1 数据集

本文实验选用 4 个不同规模的真实网络数据集,数据集的统计特征如表 2 所示,其中 p_c 表示网络的传播阈值^[25]。

表 2 数据集的统计特征

Table 2 Statistics characteristic of datasets

数据集	节点数	边数	p_c
Email	1 133	5 451	0.056
Hamster	1 858	12 534	0.025
CacondMat	23 133	93 497	0.036
Brightkite	58 228	214 078	0.015

3.2 对比算法

为验证本文算法的性能,本文将其与当前具有代表性的启发式算法进行对比。不同算法的时间复杂度对比如表 3 所示。

表 3 不同算法的时间复杂度对比

Table 3 Time complexity comparison among different algorithms

算法	时间复杂度
DegreeDistance ^[9]	$O(n\log_a n + k^2 \bar{d})$
CCA ^[11]	$O(km)$
MCDE ^[12]	$O(k^2 n \log_a n)$
IM-ELPR ^[14]	$O(T'm)$
RNR ^[15]	$O(kq(m + n \log_a n))$
VoteRank++ ^[20]	$O(n\bar{d}^2 + kh\bar{d}^2 + kn)$
TrussVote	$O(n\bar{d}^2)$

3.3 实验设置与评价指标

本文在 4 个不同规模的真实网络数据集上,使用 IC 模型^[4]和 SIR 模型^[26]对不同算法进行对比实验。

3.3.1 IC 模型

在 IM 问题中,IC 模型是一种应用最为广泛的影响力传播模型。在 IC 模拟传播过程中,每个数据集使用 p_c 作为传播概率,并且每次选择 50 个节点作为初始传播源的种子集合。在 IC 模型下的影响范围即种子节点影响扩散的个数,并将其作为算法的主要评价指标。由于 IC 模型的传播具有随机性,因此本文使用 10 000 次的 Monte-Carlo 模拟来保证实验能够得到较为精确的影响范围。但是多数算法在 IC 模型下的影响范围比较接近,并且当 k 值不同时,不同算法的表现也会上下浮动。为了便于分析,本文给出一个等价的分析指标——传播差异。算法 a_1 和 a_2 在 $k=x$ 时的传播差异如式(6)所示:

$$D_{DDx}(a_1, a_2) = \sigma_{a_1}(S_x) - \sigma_{a_2}(S_x) \quad (6)$$

其中: $\sigma_a(S_k)$ 表示算法 a 选取大小为 k 的种子集 S_k 所达到的影响范围。

3.3.2 SIR 模型

SIR 模型是一种经典的病毒传播模型,近年来在 IM 问题中得到了广泛的应用。在 SIR 模型中,每个节点具有易感 S、感染 I 和治愈 R 这 3 种状态。节点由易感 S 到感染 I 的状态转换称为感染,感染率为 μ ;节点由感染 I 到治愈 R 的状态转换称为治愈,治愈率为 β 。该模型下算法的评价指标有 2 个,第 1 个是当给定初始感染比例时对比算法在 t 时的感染规模 $F(t)$,计算过程如式(7)所示:

$$F(t) = \frac{n_{I(t)} + n_{R(t)}}{n} \quad (7)$$

其中: $n_{I(t)}$ 表示 t 时感染状态的节点个数; $n_{R(t)}$ 表示 t 时恢复状态的节点个数。第 2 个评价指标是当初始感染比例不同时,对比算法的最终感染规模 $F(t_c)$,计算过程如式(8)所示:

$$F(t_c) = \frac{n_{R(t_c)}}{n} \quad (8)$$

其中: $n_{R(t_c)}$ 表示当传播过程达到稳定时处于恢复状态的总节点数。以下实验均采用 100 次的传播来得到 $F(t)$

及 $F(t_c)$ 。因为在本文实验中, $F(t)$ 在 $t=20$ 之前均已达到峰值,所以在实验中只展示了 $t \leq 20$ 的感染规模。另外,网络初始感染比例 ρ 根据网络规模及传播阈值来选取,本文在Email、Hamster、CacondMat、Brightkite数据集上最大感染比例下的感染个数分别为50、50、150、200。

在SIR模型中,感染比例 $\lambda = \mu/\beta$ 对模型的传播速度具有至关重要的作用,本文将其设置为1.5。感染率的选择没有固定标准^[19],一些算法根据边的传播概率来选取种子节点,而本文为了保证算法的有效性 & 统一性,将每个数据集的传播阈值 p_c 设置为感染率。

4 实验结果分析

在解决IM问题的投票算法中,节点向其邻居的投票可以被看作是一个局部信息的扩散过程,当每轮投票结束后均选择信息量最高的节点作为种子节点。投票算法通过信息扩散最大化的方法来识别网络中最有影响力的节点。因此,IM问题作为信息扩散的核心及关键问题,使用投票机制获得较优的

实验效果。

4.1 IC模型下的结果分析

本文将CCA算法作为基准算法来计算不同算法在各个数据集上的传播差异。当种子节点数 k 分别为10、20、30、40、50时,CCA算法在不同数据集上的传播差异如表4所示。

表4 CCA算法的传播差异

Table 4 Diffusion difference of CCA algorithm

数据集	传播差异				
	$k=10$	$k=20$	$k=30$	$k=40$	$k=50$
Email	112.87	147.15	170.81	189.57	207.19
Hamster	111.98	144.12	164.36	182.64	197.73
CacondMat	373.17	452.62	505.34	549.80	581.52
Brightkite	423.72	473.08	507.34	553.31	629.57

在不同数据集上各个算法使用IC模型产生的传播差异如图2所示。从图2可以看出,大多数算法在开始时选择的节点拥有类似的影响范围,但随着种子节点数的增大,算法的表现呈现出不同的特点。

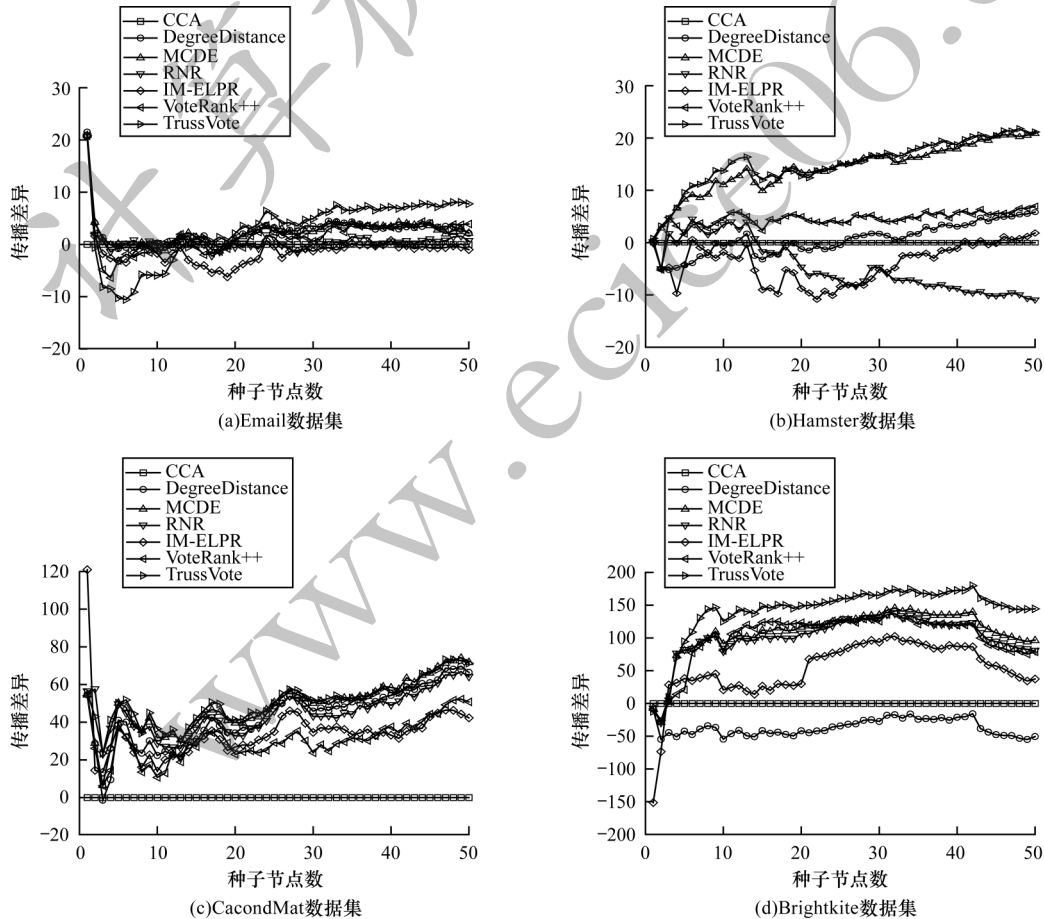


图2 IC模型下不同算法的运行结果

Fig.2 The running results of different algorithms under IC model

在Email数据集上,TrussVote和VoteRank++具有类似的传播特点,当种子节点数较小时影响的范围相

较于其他算法小,但随着种子节点数增大,影响范围逐步增大并趋于稳定。相比VoteRank++,TrussVote在种

子节点数较大时影响范围最大。这种变化趋势也说明了 TrussVote 在消除影响力重合时的作用更加显著。在 Hamster 和 CacondMat 数据集上, TrussVote 与 MCDE 总体的变化趋势接近, 但 TrussVote 的表现更好, 在 Email 和 Brightkite 数据集上, TrussVote 具有较优的实验结果。在 CacondMat 数据集上, 除 CCA 算法以外, 其他算法具有相似的传播特点, 影响范围的变化情况都基本类似, 而 TrussVote 算法在 $k \in [1, 50]$ 的多数情况下能获得最广泛的影响范围。

在 Brightkite 数据集上, CCA 与 DegreeDistance 算法的表现较差, 在其他数据集上的表现也有波动。这说明基于中心性的算法, 在网络特点不同的数据集中难以产生持续有效的实验效果。在 Hamster 数据集上 RNR 算法的表现说明它在度数较大的数据

集中, 也不能有效解决富人俱乐部效应。与投票类算法相似, 随着种子节点数增大, IM-ELPR 算法的影响范围逐渐优于其他算法, 但 TrussVote 与其相比, 不仅时间复杂度更低, 而且传播效果也更好。在 IC 模型下不同算法的实验结果表明, TrussVote 算法具有更广泛的影响范围。

4.2 SIR 模型下的结果与分析

当给定初始感染比例时, SIR 模型下不同算法的感染规模 $F(t)$ 随迭代轮次变化的结果如图 3 所示。当初始感染节点数相同时, TrussVote 只在 Hamster 数据集上的峰值略低于 MCDE, 而在其他数据集上均达到峰值。TrussVote 选取的种子集能够在最短的时间内感染更多的节点, 说明 TrussVote 具有更快的感染速度与更大的感染规模。

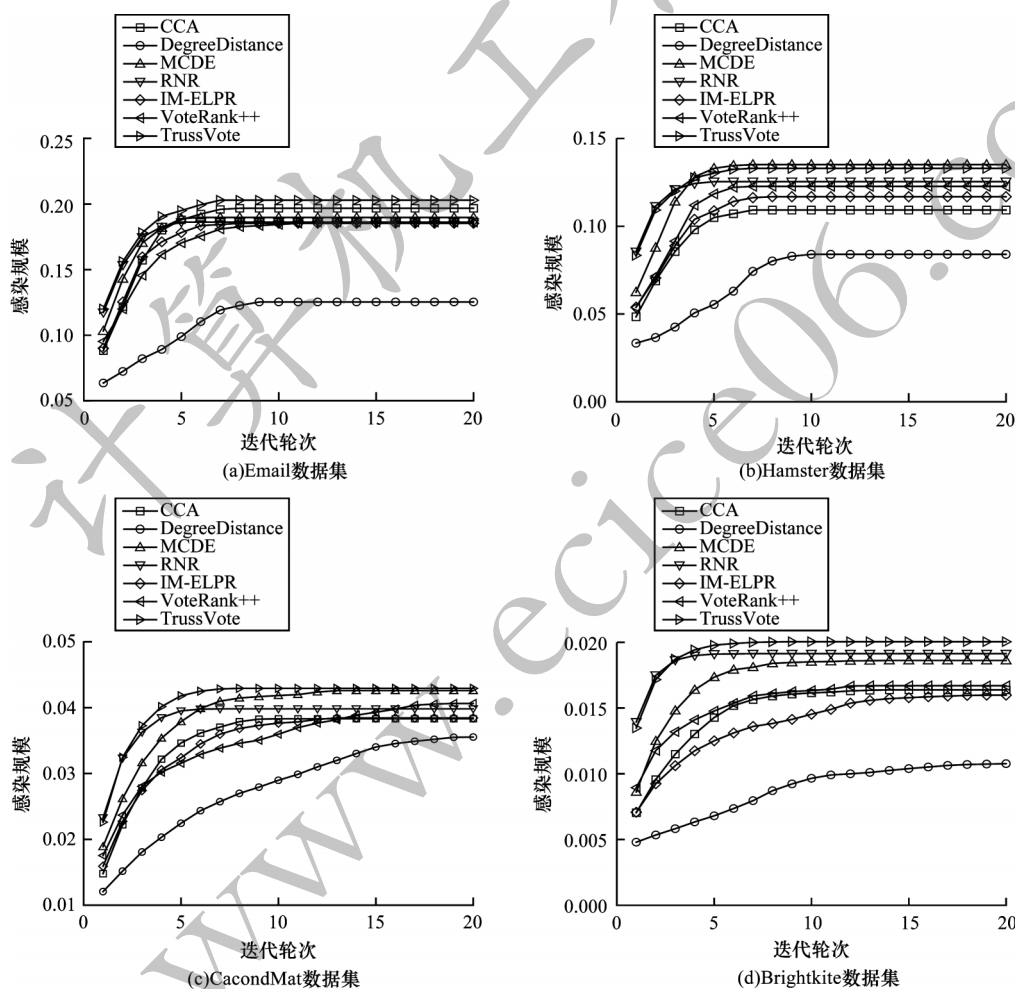


图3 不同算法的感染规模随迭代轮次的变化曲线

Fig.3 Change curves of infection scale of different algorithms with iterations

各个算法的最终感染规模 $F(t_c)$ 随初始感染比例增大的变化曲线如图 4 所示。在 Email 和 CacondMat 数据集上, TrussVote 在初始感染比例较小的情况下最终感染规模略低于 VoteRank++ 及 IM-ELPR 算法, 但随着初始感染比例的增大, TrussVote 拥有更大的感染规模。TrussVote 在 Hamster 数据集上初始感染比例为 2.65%

时表现相对较差, 但在 Hamster 和 Brightkite 数据集上, TrussVote 的最终感染规模总体最优。这说明 TrussVote 选取节点的策略及算法更新的过程都较为合理, 具有较优的稳定性。在 SIR 模型下不同算法的实验结果表明, TrussVote 不仅具有更快的感染速度及更大的感染规模, 在不同初始感染比例下也拥有最好的表现。

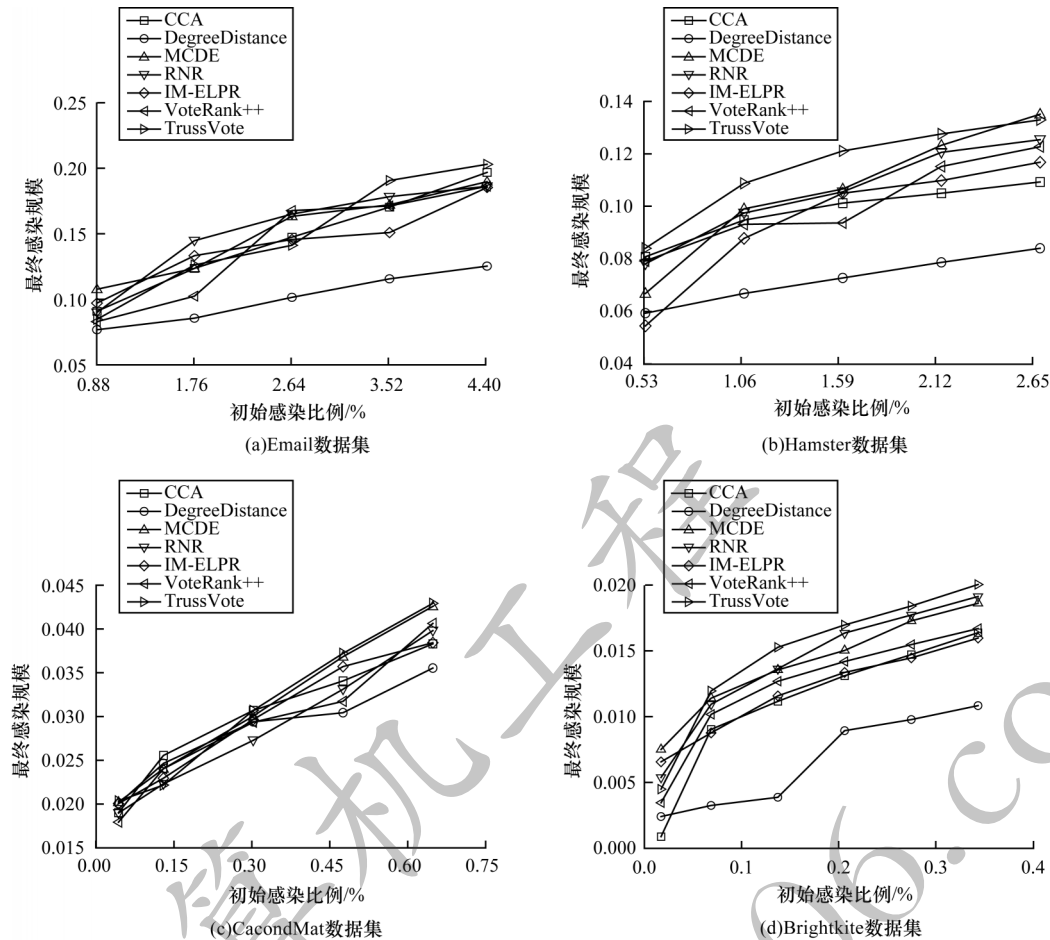


图4 不同算法的最终感染规模随初始感染比例的变化曲线

Fig.4 Change curves of final infection scale of different algorithms with initial infection ratio

因此,相比基于中心的算法,TrussVote在不增大时间复杂度的同时取得了更稳定且有效的实验效果。当迭代轮次较多时,RNR和IM-ELPR在较大网络规模中的时间和影响范围相比TrussVote较差。另外,与VoteRank++更新节点两跳之内的邻居投票能力不同,TrussVote在更新阶段对节点一跳之内的邻居投票能力赋予了不同程度的更新,不仅具有更低的时间复杂度,同时也能够获得拥有更大影响范围的种子集合。以上实验结果也验证了TrussVote算法的正确性及有效性。

5 结束语

针对社交网络中的影响最大化问题,本文提出基于K-truss的投票改进算法。在投票阶段和更新阶段分别引入边的truss值及节点相似性指标对投票类算法进行改进,以区分更新节点邻居的投票能力。在不同规模的真实网络数据集上的实验结果表明,本文算法在具有较低时间复杂度的同时,给IM问题提供了较有效的解决方案。下一步考虑将本文TrussVote投票算法应用到IM的拓展问题中,如利润最大化问题,使其具有更优的鲁棒性。

参考文献

- [1] WU J, HU B, ZHANG Y. Maximizing the performance of advertisements diffusion: a simulation study of the dynamics of viral advertising in social networks[J]. Simulation, 2013, 89(8): 921-934.
- [2] WU P, PAN L. Scalable influence blocking maximization in social networks under competitive independent cascade models[J]. Computer Networks, 2017, 123: 38-50.
- [3] FAN C J, ZENG L, SUN Y Z, et al. Finding key players in complex networks through deep reinforcement learning[J]. Nature Machine Intelligence, 2020, 2(6): 317-324.
- [4] 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, USA: ACM Press, 2003: 137-146.
- [5] 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, USA: ACM Press, 2007: 420-429.
- [6] GOYAL A, LU W, LAKSHMANAN L V S. CELF++: optimizing the greedy algorithm for influence maximization in social networks[C]//Proceedings of the 20th International Conference Companion on World Wide Web. New York, USA: ACM Press, 2011: 47-48.

- [7] ZHANG J X, CHEN D B, DONG Q, et al. Identifying a set of influential spreaders in complex networks[J]. Scientific Reports, 2016, 6: 27823.
- [8] 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, USA: ACM Press, 2009: 199-208.
- [9] SHEIKHAHMADI A, NEMATBAKHSI M A, SHOKROLLAHI A. Improving detection of influential nodes in complex networks[J]. Physica A: Statistical Mechanics and Its Applications, 2015, 436: 833-845.
- [10] WANG X J, SU Y Y, ZHAO C L, et al. Effective identification of multiple influential spreaders by degree punishment[J]. Physica A: Statistical Mechanics and Its Applications, 2016, 461: 238-247.
- [11] 曹玖新, 董丹, 徐顺, 等. 一种基于k-核的社会网络影响最大化算法[J]. 计算机学报, 2015, 38(2): 238-248.
- CAO J X, DONG D, XU S, et al. A k-core based algorithm for influence maximization in social networks[J]. Chinese Journal of Computers, 2015, 38(2): 238-248. (in Chinese)
- [12] SHEIKHAHMADI A, NEMATBAKHSI M A. Identification of multi-spreader users in social networks for viral marketing[J]. Journal of Information Science, 2017, 43(3): 412-423.
- [13] ZHAO Y X, LI S H, JIN F. Identification of influential nodes in social networks with community structure based on label propagation[J]. Neurocomputing, 2016, 210: 34-44.
- [14] KUMAR S, SINGHLA L, JINDAL K, et al. IM-ELPR: influence maximization in social networks using label propagation based community structure[J]. Applied Intelligence, 2021, 51(11): 7647-7665.
- [15] RUI X B, MENG F R, WANG Z X, et al. A reversed node ranking approach for influence maximization in social networks[J]. Applied Intelligence, 2019, 49(7): 2684-2698.
- [16] HE Q, WANG X W, LEI Z C, et al. TIFIM: a two-stage iterative framework for influence maximization in social networks[J]. Applied Mathematics and Computation, 2019, 354: 338-352.
- [17] SUN H L, CHEN D B, HE J L, et al. A voting approach to uncover multiple influential spreaders on weighted networks[J]. Physica A: Statistical Mechanics and Its Applications, 2019, 519: 303-312.
- [18] KUMAR S, PANDA B S. Identifying influential nodes in social networks: neighborhood coreness based voting approach[J]. Physica A: Statistical Mechanics and Its Applications, 2020, 553: 1-12.
- [19] GUO C, YANG L, CHEN X, et al. Influential nodes identification in complex networks via information entropy[J]. Entropy, 2020, 22(2): 1-9.
- [20] LIU P F, LI L J, FANG S Y, et al. Identifying influential nodes in social networks: a voting approach[J]. Chaos, Solitons & Fractals, 2021, 152: 1-9.
- [21] KITSAK M, GALLOS L K, HAVLIN S, et al. Identification of influential spreaders in complex networks[J]. Nature Physics, 2010, 6(11): 888-893.
- [22] MALLIAROS F D, ROSSI M E G, VAZIRGIANNIS M. Locating influential nodes in complex networks[EB/OL]. [2021-09-20]. <https://www.nature.com/articles/srep19307.pdf>.
- [23] MALLIAROS F D, GIATSIDIS C, PAPADOPOULOS A N, et al. The core decomposition of networks: theory, algorithms and applications[J]. The VLDB Journal, 2020, 29(1): 61-92.
- [24] ZHOU T, LÜ L, ZHANG Y C. Predicting missing links via local information[J]. The European Physical Journal B, 2009, 71(4): 623-630.
- [25] RADICCHI F, CASTELLANO C. Fundamental difference between superblockers and superspreaders in networks[EB/OL]. [2021-09-20]. <https://arxiv.org/pdf/1610.02908.pdf>.
- [26] PASTOR-SATORRAS R, VESPIGNANI A. Epidemic dynamics and endemic states in complex networks[J]. Physical Review E, 2001, 63(6): 1-10.

编辑 薛晋栋