Journal of Computer Applications ISSN 1001-9081 2018-02-10
计算机应用,2018,38(2):464-470 CODEN JYIIDU http://www.joca.cn

文章编号:1001-9081(2018)02-0464-07 DOI:10.11772/j.issn.1001-9081.2017071820

基于 k-核过滤的社交网络影响最大化算法

李阅志[1,2],祝园园[1,2*],钟 鸣[1,2]

(1.软件工程国家重点实验室(武汉大学),武汉 430072; 2.武汉大学 计算机学院,武汉 430072)

(* 通信作者电子邮箱 yyzhu@whu.edu.cn)

摘 要:针对现有社交网络影响最大化算法影响范围小和时间复杂度高的问题,提出一种基于独立级联模型的 k-核过滤算法。首先,介绍了一种节点影响力排名不依赖于整个网络的现有影响力最大化算法;然后,通过预训练 k,找到对现有算法具有最佳优化效果且与选择种子数无关的 k 值;最后,通过计算图的 k-核过滤不属于 k-核子图的节点和边,在 k-核子图上执行现有影响最大化算法,达到降低计算复杂度的目的。为验证 k-核过滤算法对不同算法有不同的优化效果,在不同规模数据集上进行了实验。结果显示,应用 k-核过滤算法后:与原 PMIA 算法相比,影响范围最多扩大 13.89%,执行时间最多缩短 8.34%;与原核覆盖算法(CCA)相比,影响范围没有太大差异,但执行时间最多缩短 28.5%;与 OutDegree 算法相比,影响范围最多扩大 21.81%,执行时间最多缩短 26.96%;与 Random 算法相比,影响范围最多扩大 71.99%,执行时间最多缩短 24.21%。进一步提出了一种新的影响最大化算法 GIMS,它比 PMIA 和 IRIE 的影响范围更大,执行时间保持在秒级别,而且 GIMS 算法的 k-核过滤算法与原 GIMS 算法的影响范围和执行时间差异不大。实验结果表明,k-核过滤算法能够增大现有算法选择种子节点集合的影响范围,并且减少执行时间;GIMS 算法具有更好的影响范围效果和执行效率,并且更加鲁棒。

关键词:社交网络;影响最大化;k-核;独立级联模型;传播树

中图分类号:TP312 **文献标志码**:A

k-core filtered influence maximization algorithms in social networks

LI Yuezhi[1,2], ZHU Yuanyuan[1,2*], ZHONG Ming[1,2]

(1. *State Key Laboratory of Software Engineering (Wuhan University), Wuhan Hubei* 430072, *China*;
2. *School of Computer, Wuhan University, Wuhan Hubei* 430072, *China*)

Abstract: Concerning the limited influence scope and high time complexity of existing influence maximization algorithms in social networks, a k-core filtered algorithm based on independent cascade model was proposed. Firstly, an existing influence maximization algorithm was introduced, its rank of nodes does not depend on the entire network. Secondly, pre-training was carried out to find the value of k which has the best optimization effect on existing algorithms but has no relation with the number of selected seeds. Finally, the nodes and edges that do not belong to the k-core subgraph were filtered by computing the k-core of the graph, then the existing influence maximization algorithms were applied on the k-core subgraph, thus reducing computational complexity. Several experiments were conducted on datasets with different scale to prove that the k-core filtered algorithm has different optimization effects on different influence maximization algorithms. After combined with k-core filtered algorithm, compared with the original Prefix excluding Maximum Influence Arborescence (PMIA) algorithm, the influence range is increased by 13.89% and the execution time is reduced by as much as 8.34%; compared with the original Core Covering Algorithm (CCA), the influence range has no obvious difference and the execution time is reduced by as much as 28.5%; compared with the original OutDegree algorithm, the influence range is increased by 21.81% and the execution time is reduced by as much as 26.96%; compared with the original Random algorithm, the influence range is increased by 71.99% and the execution time is reduced by as much as 24.21%. Furthermore, a new influence maximization algorithm named GIMS (General Influence Maximization in Social network) was proposed. Compared with PIMA and Influence Rank Influence Estimation (IRIE), it has wider influence range while still keeping execution time at second level. When it was combined with k-core filtered algorithm, the influence range and execution time do not have significant change. The experimental results show that k-core filtered algorithm can effectively increase the influence ranges of existing algorithms and reduce their execution times; in addition, the proposed GIMS algorithm has wider influence range and better efficiency, and it is more robust.

Key words: social network; Influence Maximization (IM); k-core; independent cascade model; diffusion tree

收稿日期:2017-07-25;**修回日期**:2017-09-10。

基金项目:国家自然科学基金资助项目(61502349);湖北省自然科学基金资助项目(2015CFB339);苏州市科技发展项目(SYG201442)。

作者简介:李阅志(1993—),男,湖北黄冈人,硕士研究生,主要研究方向:社交网络、数据挖掘; 祝园园(1984—),女,河南周口人,副教授,博士,CCF 会员,主要研究方向:图数据库、大规模数据分析、数据挖掘; 钟鸣(1982—),男,湖北孝感人,副教授,博士,CCF 会员,主要研究方向:大规模数据分析、图查询与处理、数据挖掘。

0　引言

随着 Facebook 和微信微博等社交网络的流行,越来越多的用户喜欢在社交网络中分享自己的观点和其他信息,使得网络影响力传播的研究成为社交网络分析的热点[1]。影响力最大化问题作为病毒式营销和广告投放等社交网络推荐研究领域的一个重要问题,引起了广泛的关注和研究热情。

影响最大化问题最早由 Domingos 等[2]定义为如何寻找 t 个初始节点,使得信息的最终传播范围最广。Kempe 等[3]将影响最大化定义为一个离散优化问题,证实了这个问题在独立级联模型下是 NP 难问题,提出了对后续研究影响极大的两种传播模型: 独立级联(Independent Cascade,IC) 模型和线性阈值(Linear Threshhold,LT) 模型,并基于子模性质(sub-modularity) 提出了一个基本的贪心算法 Greedy。为了降低 Greedy 算法的时间复杂度,后来的研究者又提出了若干改进算法,如 CELF (Cost-Effective Lazy Forward) 算法[4]、DegreeDiscount 算法[5]、核覆盖算法(Core Covering Algorithm,CCA)[6]、PMIA (Prefix excluding Maximum Influence Arborescence) 算法[7]、IRIE (Influence Rank Influence Estimation) 算法[8]等。其中 PMIA 算法和 IRIE 算法被认为是现有影响力最大化算法中影响范围和时间效率排名靠前的算法。这些算法基本都使用了子模性质,虽然时间效率有了很大的提升,但是影响范围效果仍然比不上 Greedy 算法[3]。

Kempe 等[3]已经证明具有子模性的影响范围函数使用 Greedy 算法求解,能取得最优解的 63%[9]。该结论表明当前取得最大影响范围的 Greedy 算法和最优解仍有差距。为了缩小现有影响最大化算法和最优解的差距,本文提出一种采用 k-核过滤的算法,可以应用于现有的大多数基于子模性质的影响最大化算法,扩大它们的影响范围,降低它们的时间复杂度。基于该 k-核过滤算法,本文对 PMIA 算法、CCA、OutDegree 算法、Random 算法分别进行优化,扩大了影响范围,降低了执行时间,尤其对于 CCA,在不减小影响范围的情况下执行时间缩短了 28.5%。并且通过预训练 k 首次发现同一算法在同一数据集上取得最佳优化效果的 k 是固定值。李国良等[10]提出了一种针对多网络实体影响最大化的算法,本文将 k-核过滤算法结合该算法思想应用到单个社交网络上,提出了一种新的影响最大化算法 GIMS(General Influence Maximization in Social networks) ,该算法比 PMIA 和 IRIE 的影响范围更大,执行时间更短。

1　传播模型和问题定义

设有向图 $G=(V,E)$ 表示一个社交网络,其中: V 表示节点集合,n 表示节点总数; E 表示边集,m 表示边的总数。$\forall v \in V$ 表示节点,$\forall e(u,v) \in E$ 表示一条 u 指向 v 的有向边。

1.1　独立级联模型

独立级联模型是一个概率模型。对于网络 G 中的每个节点都有两个状态: 激活和未激活。每个节点只能从未激活状态转变为激活状态,每个节点可以被它的相邻节点激活。对于每条边 $e(u,v) \in E$,需指定一个影响概率 $p(u,v) \in [0,1]$,$p(u,v)$ 表示节点 u 通过边 $e(u,v)$ 影响节点 v 的概率。给定初始激活节点集合 S_0,传播过程以如下方式进行: 当传播至第 $t+1$ 步时,利用在第 t 步中被激活的节点,根据成功概率 $p(u,v)$ 试图去激活它们的邻居节点,并将在这一步中被激活的节点加入到 S_t 形成 S_{t+1}; 重复这一过程,直至不再有新的节点被激活。整个过程中 u 只尝试激活 v 一次,激活成功则 v 由未激活变为激活状态,激活失败则不再尝试激活。

1.2　问题定义

定义 1　有向图 $G=(V,E)$ 中,给定一个输入 t,独立级联模型下的影响最大化(Influence Maximization, IM) 问题是找到 G 一个节点子集 $S^* \in V$,S^* 的最终影响范围 $\sigma(S^*)$ 满足:

$$\sigma(S^*) = \max\{\sigma(S) \mid |S| = t, S \subseteq V\}$$

$$\text{s.t.} \quad |S^*| = t$$

定义 2　一个集合函数 $f: 2^V \to \mathbf{R}$ 是单调的,如果 $f(S) \leqslant f(T)$ 对所有 $S \subseteq T$ 都成立。

定义 3　一个集合函数 $f: 2^V \to \mathbf{R}$ 是子模的,如果 $f(S \cup \{u\}) - f(S) \geqslant f(T \cup \{u\}) - f(T)$ 对所有 $S \subseteq T \subseteq V$ 和 $u \in V$ 都成立。

Kempe 等[3]证明了影响最大化问题是 NP 难问题,并且证明影响范围函数满足单调性和子模性,由此提出了可获得 $(1-1/e)$ 的近似最优解 Greedy 算法。Greedy 算法的影响范围是现有影响最大化算法中效果最好的,能取得最优解的 63%[3],但时间效率很低,执行一次需要几个小时甚至几天。本文提出一种 k-核过滤算法,可以扩大现有算法的影响范围,并且降低算法的执行时间复杂度。

2　基于 k-核过滤的影响最大化算法

本文提出了一种 k-核过滤算法,可以应用于很多影响最大化算法,扩大它们的影响范围,缩短其执行时间。同时,本文还提出了一种新的单社交网络影响最大化算法 GIMS,其影响范围比广泛认可的 PMIA 算法和 IRIE 算法效果更好,执行时间比 PMIA 算法更少。

2.1　k-核

定义 4　集合 $V_k \subseteq V$ 的任一节点 v 的度数不少于 k,由 V_k 所推导出的最大诱导子图 $G_k(V_k,E_k)$ 称为 k-核。

k-核概念由 Seidman[11]于 1983 年提出,可用来描述度分布所不能描述的网络特征,揭示源于系统特殊结构的结构性质和层次性质。虽然节点度在影响最大化问题中被研究者广泛关注,但是度较大的节点可能较为分散,而 k-核使得度较大的节点聚集起来,网络分布越集中,就越能受到彼此影响,从而产生更大的网络影响力。因此本文采用 k-核来优化现有影响最大化算法。

2.2　k-核过滤算法

本文提出的 k-核过滤算法不是一个独立的影响最大化算法,而是通过与现有影响最大化算法相结合来扩大现有算法的影响范围,并缩短其执行时间。k-核过滤算法的基本思想是通过预训练 k,找到对现有算法具有最佳优化效果、并且与选择种子节点数 t 无关的固定 k 值,在给定需要选择的节点数 t 时,通过计算图的 k-核过滤不属于 k-核子图的节点和边,在 k-核子图上执行现有影响最大化算法,从而达到减少计算量的目的。

k-核过滤算法涉及两个步骤: 1) 通过预先训练找到能产生优化效果最好的参数 k,如算法 1 所示; 2) 计算网络的 k-核,并应用在现有算法中,如算法 2 所示。

算法 1 预训练 k。

输入 有向图 $G=(V,E)$，迭代次数 $iter$;

输出 最佳优化效果 k。

```
optk = 0, k = 0, t = rand(0, 10), σ_opt = 0;
while k < iter
    compute G_k, the k-core subgraph of G;
    imply existing IM algorithm on G_k;
    if σ_k > σ_opt
    optk = k; σ_opt = σ_k;
    k ++;
return optk;
```

算法 2 k-核优化现有算法。

输入 有向图 $G=(V,E)$，最佳优化效果 k，种子节点数 t;

输出 种子集合 S。

```
compute G_k, the k-core subgraph of G;
S = existing IM algorithm on G_k;
return S;
```

算法 1 的预训练过程需要提前完成以选出最佳优化效果 k，由本文实验可知，在选取不同的种子个数 t 时，取得最佳优化效果的 k 是固定值。接着采用算法 2 计算 G 的种子集合 S，此时的执行时间会减少很多，影响范围效果也会更好。本文利用 Batagelj 等[12]提出的算法计算 k-核子图 G_k，通过迭代删除度小于 k 的所有顶点和与之相连的边，剩下的子图就是 k-核，时间复杂度为 $O(m)$。针对不同的现有算法，k-核过滤算法有不同的结合方式，算法的时间复杂度取决于现有算法的时间复杂度，但通常比结合前的原算法效率更高。假设现有算法的时间复杂度为 $f(n,m)$，那么 k-核过滤算法的时间复杂度为 $\max\{O(m), f(n_k,m_k)\}$，其中 n_k 和 m_k 表示 k-核子图的节点数和边数。通常 $f(n,m)$ 都是大于 $O(m)$ 的复杂度，通过 k-核过滤之后算法时间复杂度不超过 $O(m)$ 和 $f(n_k,m_k)$ 的最大值，但一定小于 $f(n,m)$。算法 2 也可以脱离算法 1 直接执行，可以任意选择 k，对现有算法的影响范围和执行时间都会有改进，只是没有预训练 k 的优化效果好。

2.3 k-核优化现有算法

现有影响最大化算法中，PMIA 算法和 IRIE 算法传播影响效果较好，其中 PMIA 的内存耗费较大，IRIE 算法执行时间较长。本文将 k-核过滤算法分别与 CCA[6]、PMIA 算法[7]、OutDegree 算法[1]和 Random 算法相结合，得到 KCoreCCA 算法、KCorePMIA 算法、KCoreOutDegree 算法和 KCoreRandom 算法，以扩大原算法的影响范围，缩短其执行时间。IRIE 算法[8]和 PageRank 算法[13]是基于节点排名的算法，节点排名依赖于整个网络所有节点的收敛，k-核过滤算法原理是先剪枝不重要的节点，因此无法应用在这两个算法中。

2.3.1 KCoreCCA

核覆盖算法(CCA)是基于覆盖距离选择核数最大的 t 个节点作为种子节点。核数 k 表示节点属于 k-核，而不属于 $(k+1)$-核。CCA 和 k-核过滤算法都是基于 k-核的概念提出的，较小核数的节点对 CCA 毫无贡献，剪枝核数较小的节点虽然对 CCA 的影响范围没有改变，但是可以大大缩短 CCA 的执行时间。

算法 3 KCoreCCA。

输入 有向图 $G(V,E)$，种子节点数 t，覆盖距离 d;

输出 种子集合 S。

```
initialize S = ∅;
compute G_k(V_k, E_k), the k-core subgraph of G;
computeCores(G_k);
foreach vertex v ∈ V_k do
    CO_v = false;
for i = 1 to t do
    w = arg max_v{C_v | v ∈ V_k\S, CO_v = false};
    u = arg max_v{d_v | v ∈ V_k\S, C_v = C_w, CO_v = false};
    S = S ∪ {u};
    foreach vertex v in {v | d_{u,v} ≤ d, v ∈ V_k} do
        CO_v = true;
return S;
```

其中：C_v 为节点的核数，CO_v 表示节点覆盖属性，$d_{u,v}$ 表示节点 u 和 v 之间的距离。算法 3 是将 k-核过滤算法应用于 CCA，CCA 的时间复杂度是 $O(tm)$[6]。KCoreCCA 先计算出 k-核子图(第 2)行)，时间复杂度是 $O(m)$；第 3) ~12)行遵循 CCA 的基本思路，但只针对 k-核计算，而不是整个网络，时间复杂度是 $O(tm_k)$。所以 KCoreCCA 的时间复杂度是 $\max\{O(m), O(tm_k)\}$，因为剪枝了很多节点，减少了计算量。

KCoreCCA 是以核数考量选择种子节点，而以出度考量选取种子节点的算法，称为 OutDegree 算法，类似优化 CCA 的方式，应用 k-核过滤算法于 OutDegree 算法，得到 KCoreOutDegree 算法，OutDegree 算法时间复杂度为 $O(n+m)$，KCoreOutDegree 算法时间复杂度为 $\max\{O(m), O(n_k,m_k)\}$。k-核过滤算法应用于随机选取种子节点的 Random 算法，得到 KCoreRandom 算法，Random 算法时间复杂度为 $O(t)$，KCoreRandom 算法时间复杂度为 $\max\{O(m), O(t)\} = O(m)$。

2.3.2 KCorePMIA 算法

PMIA 算法先计算每个节点 $\forall v \in V$ 的本地树结构 PMIIA 和 PMIOA，基于本地树结构 PMIIA 计算当前种子集合 S 对每个节点 u 的影响 $ap(u,S,PMIIA(v,\theta))$，并根据影响线性(Influence Linearity)性质计算影响系数 $\alpha(v,u)$，然后根据本地树结构结合子模性选出影响最大的 t 个节点。PMIA 算法需计算每个节点的本地树结构，虽然加快了影响传播的计算和更新，但保存每个节点的本地树结构需要耗费大量内存，计算全部节点本地树结构效率也较低。本文采用 k-核过滤算法筛选出可能是最具影响力的节点，只需计算这些节点的本地树结构，从而减少了大量计算。

算法 4 KCorePMIA 算法。

输入 有向图 $G(V,E)$，种子节点数 t，路径传播阈值 θ;

输出 种子集合 S。

```
set S = ∅;
set IncInf(v) = 0 for each node v ∈ V
compute G_k(V_k, E_k), the k-core subgraph of G;
foreach vertex v ∈ G_k do
    compute PMIIA(v, θ, S);
    set ap(u, S, PMIIA(v, θ, S)) = 0, ∀u ∈ PMIIA(v, θ, S);
    compute α(v, u), ∀u ∈ PMIIA(v, θ, S)
    foreach vertex u ∈ PMIIA(v, θ, S) do
        IncInf(u) += α(v, u) * (1 - ap(u, S, PMIIA(v, θ, S)));
for i = 1 to t do
    pick u = arg max_{u ∈ V_k\S}{IncInf(u)};
    compute PMIOA(u, θ, S);
    foreach v ∈ PMIOA(u, θ, S) ∩ V_k do
```

14) for $w \in$ PMIIA(v,θ,S) $\setminus S$ do
15) *IncInf*(w) $-=$ $\alpha(v,w) * (1 - ap(w,S,\text{PMIIA}(v,\theta,S)))$;
16) $S = S \cup \{u\}$;
17) for $v \in$ PMIOA($u,\theta,S\setminus\{u\}$) $\setminus\{u\} \cap V_k$ do
18) compute PMIIA (v,θ,S);
19) compute $ap(w,S,\text{PMIIA}(v,\theta,S))$, $\forall w \in \text{PMIIA}(v,\theta,S)$
20) compute $\alpha(v,w)$, $\forall w \in \text{PMIIA}(v,\theta,S)$
21) for $w \in$ PMIIA(v,θ,S) $\setminus S$ do
22) *IncInf*(w) $-=$ $\alpha(v,w) * (1 - ap(w,S,\text{PMIIA}(v,\theta,S)))$;
23) return S;

PMIA 算法的时间复杂度是 $O(nt_{i\theta} + tn_{o\theta}n_{i\theta}\log n)$，其中 $n_{i\theta}$ 和 $n_{o\theta}$ 分别是所有节点的 PMIIA 本地树结构和 PMIOA 本地树结构的最大节点数，$t_{i\theta}$ 是计算所有节点的 PMIIA 本地树结构的最长时间[7]，由于每个节点都需要计算 PMIIA 和 PMIOA 子树结构，PMIA 算法的时间复杂度一定大于 $O(m)$。而 KCorePMIA 算法先计算 k-核子图 G_k（第 3) 行)，时间复杂度是 $O(m)$，过滤掉部分节点后，只需在 G_k 上再执行 PMIA 算法(第 4) ~23) 行)，时间复杂度是 $O(n_k t_{i\theta k} + tn_{o\theta k}n_{i\theta k}\log n_k)$，因此 KCorePMIA 算法的时间复杂度为 $\max\{O(m), O(n_k t_{i\theta k} + tn_{o\theta k}n_{i\theta k}\log n_k)\}$。当 k 较大时，n_k 远小于 n，KCorePMIA 算法的时间复杂度不大于 $O(m)$。

2.4 GIMS 算法

李国良等[10]提出的 BoundBasedIMMS (BoundBased Influence Maximization in Multiple Social networks) 算法可以有效解决存在实体对应关系的多网络影响最大化问题，本文对该算法进行扩展，以设计一种可以有效解决单个社交网络上的影响最大化问题的算法 GIMS。GIMS 扩展了基于树的算法模型[7]，结合独立级联模型的单调性和子模性，降低了时间复杂度，同时也保证了影响范围效果。

路径 $P_{u,v} = (n_1 = u, n_2, \cdots, n_m = v)$ 表示经由节点 u 到节点 v 的一条路径。在独立级联模型下，节点 u 沿路径 P 激活节点 v 的概率为:

$$pp(P_{u,v}) = \prod_{i=1}^{m-1} p(n_i, n_{i+1})$$

节点 u 可以通过多条路径影响到节点 v，为减少计算量，采用最大影响路径 $MPP(u,v)$ 来近似节点 u 对节点 v 的影响概率:

$$pp(u,v) \approx pp(MPP(u,v)) = pp(\arg\max(pp(P_{u,v})))$$

通过计算所有节点到节点 u 的影响概率，可以构建最大逆向传播树 $ITree(u)$，考虑到当 $pp(u,v)$ 小于某一个阈值 θ 时，节点 v 被 u 激活的概率较小，则忽略此节点。同理，计算节点 u 到所有其他节点的影响概率，可以构建最大传播树 $Otree(u)$。

给定激活的种子集合 S，假设 S 中的每个节点独立地影响其他未激活的节点，并且只在已构建的传播树中传播影响，则 S 对未激活节点 v 的影响概率为:

$$pp(S,v) = \begin{cases} 1, & v \in S \\ 1 - \prod_{u \in S}(1 - pp(u,v)), & v \notin S \end{cases}$$

在种子集合 S 中加入一个新激活的节点 u 后，种子集合对节点 v 的影响增益 $gain(u \mid S, v)$ 为:

$$gain(u \mid S,v) = pp(S \cup \{u\}, v) - pp(S,v) = 1 - \prod_{t \in S \cup \{u\}}\left(1 - pp(t,v)\right) - \left(1 - \prod_{t \in S}(1 - pp(t,v))\right) = pp(u,v)\prod_{t \in S}(1 - pp(t,v)) = pp(u,v)(1 - pp(S,v))$$

假设每个节点都是独立影响其他节点，那么种子集合 S 中新增加一个激活节点 u 后总的影响增益 $gain(u \mid S)$ 为:

$$gain(u \mid S) = \sum_{v \in V} gain(u \mid S,v) = \sum_{v \in V} pp(u,v)(1 - pp(S,v))$$

根据影响最大化问题的子模性，每次选取影响增益 $gain(u \mid S)$ 最大的节点加入种子集合 S 中，直到选取 t 个节点。

算法 5　GIMS 算法。

输入　有向图 $G(V,E)$，种子节点数 t，路径传播阈值 θ;

输出　种子集合 S。

1) precompute $ITree(v)$ and $OTree(v)$ with θ, for $\forall v \in V$;
2) initialize big heap H of Node {*v. id*, *v. spread*, *v. status*}, for $\forall v \in V$;
3) set $S = \varnothing$;
4) initialize $pp(S, v) = 0$, for $\forall v \in V$;
5) for $i = 1$ to t do
6) Node *tnode* = *H. top*();
7) while(*tnode. status* ! $= i$) do
8) compute $gain(tnode.id \mid S)$;
9) *tnode. spread* = $gain(tnode.id \mid S)$;
10) *tnode. status* = i;
11) *H. sort*();
12) *tnode* = *H. top*();
13) *tnode* = *H. pop*();
14) $S = S \cup \{tnode.id\}$;
15) update $pp(S,v)$, for $\forall v \in V$;
16) *H. sort*();
17) return S;

算法 5 通过构造树模型近似计算传播概率(第 1) 行)，时间复杂度是 $O(nt_\theta)$，其中 t_θ 是所有 $ITree(v)$ 和 $OTree(v)$ 的最大顶点数。然后依据独立影响假设简化了影响增益计算，构造大根堆避免了很多影响力小的节点的计算(第 6) ~16) 行)。每次计算 $gain(tnode.id \mid S)$ 的时间复杂度是 $O(t_\theta)$，调整堆 H 的时间复杂度是 $O(\log n)$，每次迭代更新常量次节点 *status* 即可选出最优 *tnode*，所以第 6) ~16) 行的时间复杂度是 $O(t_\theta + \log n)$。因此 GIMS 的时间复杂度为 $O(nt_\theta + t(t_\theta + \log n))$。在此基础上，本文应用 k-核优化 GIMS 算法，称为 KCoreGIMS 算法。KCoreGIMS 算法先通过计算 k-核剪枝部分影响力小的节点，时间复杂度为 $O(m)$，在计算算法 5 的第 1) 行和第 2) 行时无需计算全部网络节点，而只需在 k-核子图上执行 GIMS 算法从而减少计算量，时间复杂度为 $O(n_k t_\theta + t(t_\theta + \log n_k))$。因此 KCoreGIMS 的算法时间复杂度为 $\max\{O(m), O(n_k t_\theta + t(t_\theta + \log n_k))\}$。

2.5 算法复杂度比较分析

经过 2.4 节对各算法分析，总结各算法和其 k-核过滤算法的时间复杂度如表 1。可以看到，对于影响范围效果好的算法 GIMS、PMIA、CCA、OutDegree，计算 k-核的时间复杂度是 $O(m)$，但执行现有影响最大化算法时由于只需要考虑 k-核子

图的节点和边,使得原算法的复杂度有所降低。对于效果较差的 Random 算法,由于是随机选择种子节点,使用 k-核过滤时计算 k-核使时间复杂度有所增加。因此,对于一般影响最大化算法而言,k-核过滤算法使得原算法的时间复杂度有所降低。

表 1 不同影响最大化算法及其 k-核过滤算法的时间复杂度比较

Tab. 1 Time complexity comparison of different IM algorithms and their k-core filtered versions

算法	时间复杂度
GIMS	$O(nt_\theta + t(t_\theta + \log n))$
KCoreGIMS	$\max\{O(m), O(n_k t_\theta + t(t_\theta + \log n_k))\}$
PMIA	$O(nt_{i\theta} + tn_{o\theta}n_{i\theta}\log n)$
KCorePMIA	$\max\{O(m), O(n_k t_{i\theta k} + tn_{o\theta k}n_{i\theta k}\log n_k)\}$
CCA	$O(tm)$
KCoreCCA	$\max\{O(m), O(tm_k)\}$
OutDegree	$O(n + m)$
KCoreOutDegree	$\max\{O(m), O(n_k + m_k)\}$
Random	$O(t)$
KCoreRandom	$O(m)$

3 实验与分析

3.1 数据集

本文实验在三个真实数据集上进行,分别是 ArXiv (https://arxiv.org/) 物理领域作者合作关系网络 NetHEPT (collaboration Network of High Energy Physics Theory from arXiv.org)、社交网络 Slashdot 和商品团购网络 Amazon (http://snap.stanford.edu/data/index.html)。这三个数据集的统计特性如表 2 所示。

表 2 实验中的数据集

Tab. 2 Experimental datasets

数据集	节点数	边数	平均度
NetHEPT	15 233	32 235	4.23
Slashdot	82 168	948 464	23.09
Amazon	403 394	3 387 388	16.79

3.2 实验设计

本文实验比较了 GIMS 算法、PMIA 算法、IRIE 算法、CCA、OutDegree 算法、PageRank 算法和 Random 算法,以及它们的 k-核过滤算法。其中 IRIE 算法和 PageRank 由于依赖全局节点排名不适合 k-核过滤; CCA 的覆盖距离 d 设置为 2; GIMS 算法、PMIA 算法传播概率阈值 θ 设置为 1/320; 社交网络中节点 u 到节点 v 的传播概率初始化为 $1/InDeg(v)$,其中 $InDeg(v)$ 表示节点 v 的入度。由于模型的随机性,在计算选择的 t 个节点的影响范围时采用蒙特卡罗重复模拟10 000次,取平均值。

所有程序代码都是基于 C++ 语言编程,计算机配置为: centos 6.6, Intel Xeon CPU E5-2640 v3 2.60 GHz,8 GB 内存。

为了评估 k-核过滤算法对现有算法的优化效果,本文定义两个评估指标,影响范围优化百分比 *influ_opt* 和执行时间优化百分比 *time_opt*。假设 A 算法的 k-核过滤算法是 KCoreA,当选择 t 个种子节点时,算法 KCoreA 的影响范围和执行时间分别是 *kcore_influ* 和 *kcore_time*, 算法 A 的影响范围和执行时间分别是 *influ* 和 *time*, 那么:

$$influ\_opt = (kcore\_influ - influ)/influ$$

$$time\_opt = (time - kcore\_time)/time$$

当种子节点数 $1 \leqslant t \leqslant 50$ 时,定义平均影响范围优化百分比 *avg_influ_opt* 和平均执行时间优化百分比 *avg_time_opt* 分别为所有 t 值下 *influ_opt* 和 *time_opt* 的平均值。

3.3 实验结果与分析

图 1(a)、(b) 分别是 NetHEPT 上各个算法及其 k-核过滤算法的传播范围和执行时间。通过算法 1 的预训练 k 发现选取不同种子个数时取得最佳优化效果的 k 是固定值: KCoreGIMS 最佳优化效果 k 为 3 或 4,KCorePMIA 的最优 k 为 1 或 2,KCoreCCA 的最优 k 为 8,KCoreOutDegree 的最优 k 为 3,KCoreRandom 的最优 k 为 4 或 5。从图 1 可以看到,本文提出的 GIMS 算法及其优化算法 KCoreGIMS 比现有算法中效果较好的 PMIA 算法和 IRIE 算法选出的节点传播效果更好,执行时间保持在毫秒级别,比 IRIE 算法更有效率。k-核过滤对于不同算法的优化效果不同。对算法本身效果较优的 GIMS 算法、PMIA 算法来说,k-核优化效果比原算法的影响范围扩大了 1% 左右; 对 OutDegree 算法和 Random 算法影响范围扩大了 10% 以上; 对 CCA 传播范围没有影响,但是执行时间缩短了 27%。也即,对所有算法影响范围都扩大了,执行时间都缩短了。

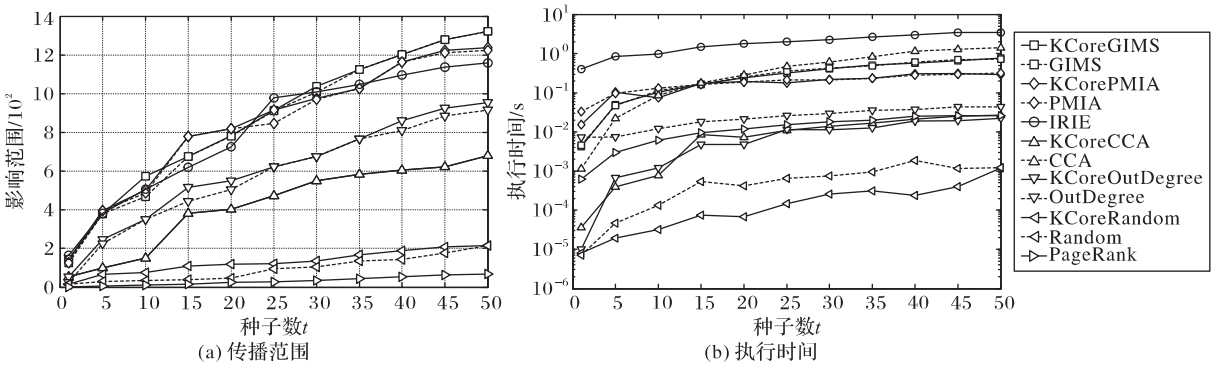


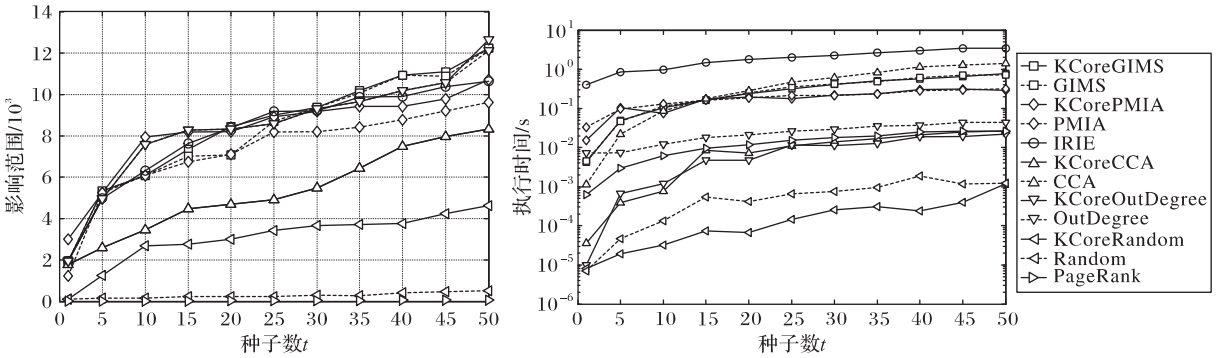
图 1 NetHEPT 数据集上的传播范围和执行时间

Fig. 1 Influence range and execution time on NetHEPT dataset

图 2(a)、(b) 分别是 Slashdot 数据集上各个算法及其 k-核过滤算法的传播范围和执行时间。通过算法 1 的预训练 k 发现选取不同种子个数时取得最佳优化效果的 k 是固定值: KCoreGIMS 的最优 k 为 22 或 44,KCorePMIA 的最优 k 为 5 或 6,KCoreCCA 的最优 k 为 50,KCoreOutDegree 的最优 k 为 40 或 50,KCoreRandom 的最优 k 为 42 或 44。从图 2 可以看到,

本文提出的 GIMS 算法及其优化算法 KCoreGIMS 仍然是传播范围最好的算法,比 PMIA 算法和 IRIE 算法高出 2 000 多节点,执行时间远少于 IRIE 算法和 PMIA 算法。*k*-核过滤算法对 GIMS 算法和 PMIA 算法影响范围扩大了 2% 以上,执行时间缩短了 2% 以上,对 CCA 和 OutDegree 算法执行时间缩短了 27% 左右。

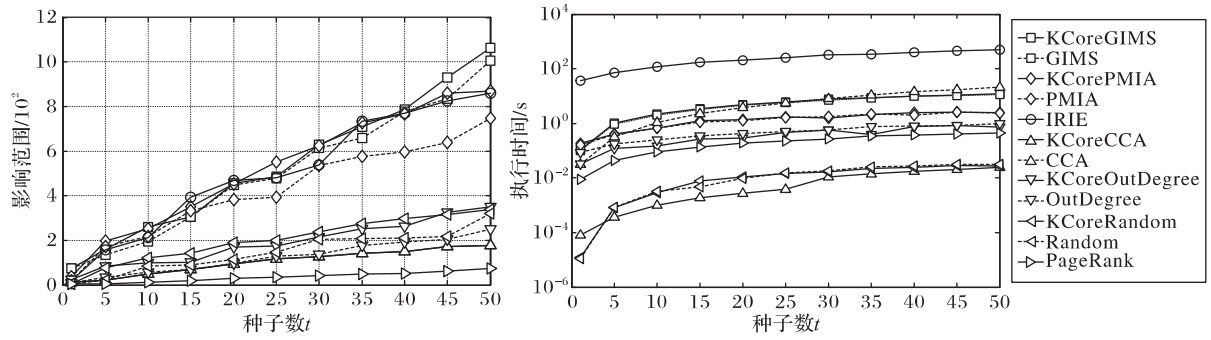
图 3(a) 、(b) 是 Amazon 数据集上各算法及其*k*-核过滤算法的影响范围和执行时间。通过算法1的预训练*k*发现选取不同种子个数时取得最佳优化效果的 *k* 是固定值: GIMS 的最优 *k* 为 11,KCorePMIA 的最优 *k* 为 10,KCoreCCA 的最优 *k* 为 19,KCoreOutDegree 的最优 *k* 为 5 或 7,KCoreRandom 的最优 *k* 为 3 或 6。从图 3 可以看到,本文提出的 GIMS 及其优化算法 KCoreGIMS 仍是影响范围最大的算法,执行时间也远低于 IRIE 算法。*K*-核过滤算法对 GIMS 算法和 PMIA 算法影响范围扩大了 1% 以上,对 OutDegree 算法影响范围扩大了 21% 以上,对 CCA 的执行时间缩短了 28% 左右。



(a) 传播范围　　(b) 执行时间

图 2　Slashdot 数据集上的传播范围和执行时间

Fig. 2　Influence range and execution time on Slashdot dataset



(a) 传播范围　　(b) 执行时间

图 3　Amazon 数据集上的传播范围和执行时间

Fig. 3　Influence range and execution time on Amazon dataset

从图性质来看,Amazon 是稀疏的大网络,Slashdot 和 NetHEPT 是稠密的小网络,但 GIMS 和 KCoreGIMS 保持了更大的影响范围和有竞争力的执行时间,*k*-核过滤算法对各算法的影响范围和执行效率都有不错的优化效果。这说明了 GIMS 算法和 *k*-核过滤算法的鲁棒性。为了更好地展示 *k*-核过滤算法对各算法的影响范围和执行效率的优化效果,表 3 给出了各算法的平均影响范围和执行时间优化百分比,也就是定义在 3.2 节的 *avg_influ_opt* 和 *avg_time_opt*。

从表 3 可以看到,*k*-核过滤算法对 GIMS 算法和 PMIA 算法的优化效果较弱,但是也有提高; 对 CCA 的影响范围没有扩大,但是大大缩短了 CCA 的执行时间; 对 OutDegree 算法和 Random 算法扩大了影响范围,缩短了执行时间。

通过对图 1 ~ 3 和表 3 的综合分析可以得出以下结论: 1) 本文提出的 GIMS 算法及其 *k*-核过滤算法影响范围比现有的算法更好,执行时间也更有竞争力; 2) *k*-核过滤算法对现有大多数算法都可以扩大影响范围,减少执行时间; 3) *k*-核过滤算法在保证不降低影响范围的情况下,能大大减少 CCA 的执行时间。

表 3　*k*-核过滤影响范围和执行时间百分比对比　%

Tab. 3　Comparison of *avg_influ_opt* (*AIO*) and *avg_time_opt* (*ATO*)　%

算法	NetHEPT		Slashdot		Amazon	
	AIO	*ATO*	*AIO*	*ATO*	*AIO*	*ATO*
KCoreGIMS	1.30	-0.30	2.79	2.71	1.09	-0.50
KCorePMIA	1.41	5.66	13.89	8.34	8.74	-1.60
KCoreCCA	0.00	27.93	0.00	27.29	0.00	28.50
KCoreOutDegree	10.18	18.16	0.6	26.96	21.81	7.97
KCoreRandom	21.43	12.00	71.99	24.21	10.12	2.59

4　结语

针对近几年研究热点社交网络影响最大化问题,提出了一种影响范围和执行时间更优的 GIMS 算法,并通过 *k*-核计算剪枝影响范围小的节点,提出了一种可以扩大现有算法影响范围和减少它们执行时间的 *k*-核过滤算法。实验表明: GIMS 算法影响范围超过现有算法,执行时间也很有竞争力; *k*-核过滤算法能扩大现有算法的影响范围并减少它们的执行时间,对 CCA 尤其有效; 并且,首次发现同一算法在同一数据

集上预训练 k 选取不同种子个数时取得最佳优化效果的 k 是固定值。下一步的研究方向可以考虑使用其他方式来剪枝影响力小的节点,还可以考虑优化带有成本或地理位置限制的影响最大化问题[14]。

参考文献:

[1] WASSERMAN S, FAUST K. Social Network Analysis: Methods and Applications [M]. New York: Cambridge University Press, 1994: 148 - 161.

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

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

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

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

[6] 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.)

[7] WANG C, CHEN W, WANG Y. Scalable influence maximization for independent cascade model in large-scale social networks [J]. Data Mining & Knowledge Discovery, 2012, 25(3): 545 - 576.

[8] JUNG K, HEO W, CHEN W. IRIE: scalable and robust influence maximization in social networks [C]// ICDM 2012: Proceedings of the 2012 IEEE 12th International Conference on Data Mining. Washington, DC: IEEE Computer Society, 2012: 918 - 923.

[9] 夏涛,陈云芳,张伟,等. 社会网络中的影响力综述[J]. 计算机应用,2014,34(4):980 - 985. (XIA T, CHEN Y F, ZHANG W, et al. Survey of influence in social networks [J]. Journal of Computer Applications, 2014, 34(4): 980 - 985.)

[10] 李国良,楚娅萍,冯建华,等. 多社交网络的影响力最大化分析[J]. 计算机学报,2016,39(4):643 - 656. (LI G L, CHU Y P, FENG J H, et al. Influence maximization on multiple social networks [J]. Chinese Journal of Computers, 2016, 39(4): 643 - 656.)

[11] SEIDMAN S B. Network structure and minimum degree [J]. Social Networks, 1983, 5(3): 269 - 287.

[12] BATAGELJ V, ZAVERŠNIK M. An $O(m)$ algorithm for cores decomposition of networks [J]. Computer Science, 2003, 1(6): 34 - 37.

[13] BRIN S, PAGE L. The anatomy of a large-scale hypertextual Web search engine [J]. Computer Networks and ISDN Systems, 1998, 30(1/2/3/4/5/6/7): 107 - 117.

[14] 刘院英,郭景峰,魏立东,等. 成本控制下的快速影响最大化算法[J]. 计算机应用,2017,37(2):367 - 372. (LIU Y Y, GUO J F, WEI L D, et al. Fast influence maximization algorithm in social network under budget control [J]. Journal of Computer Applications, 2017, 37(2): 367 - 372.)

This work is partially supported by the National Natural Science Foundation of China (61502349), the Hubei Provincial Natural Science Foundation (2015CFB339), the Scientific and Technologic Development Programm of Suzhou (SYG201442).

LI Yuezhi, born in 1993, M. S. candidate. His research interests include social network, data mining.

ZHU Yuanyuan, born in 1984, Ph. D., associate professor. Her research interests include graph database, large-scale data analysis, data mining.

ZHONG Ming, born in 1982, Ph. D., associate professor. His research interests include large-scale data analysis, graph query and processing, data mining.

(上接第 447 页)

[8] RAO R V, PATEL V. Comparative performance of an elitist teaching-learning-based optimization algorithm for solving unconstrained optimization problems [J]. International Journal of Industrial Engineering Computations, 2013, 4(1): 29 - 50.

[9] OUYANG H, WANG Q, KONG X. Modified teaching-learning based optimization for 0-1 knapsack optimization problems [C]// Proceedings of the 29th Chinese Control and Decision Conference. Piscataway, NJ: IEEE, 2017: 973 - 977.

[10] 王培崇. 改进的动态自适应学习教与学优化算法[J]. 计算机应用,2016,36(3):708 - 712. (WANG P C. Improved dynamic self-adaptive teaching-learning-based optimization algorithm [J]. Journal of Computer Applications, 2016, 36(3): 708 - 712.)

[11] ZHANG H, LI B, ZHANG J, et al. Parameter estimation of nonlinear chaotic system by improved TLBO strategy [J]. Soft Computing, 2016, 20(12): 4965 - 4980.

[12] DASHTI S E, RAHMANI A M. Dynamic VMs placement for energy efficiency by PSO in cloud computing [J]. Journal of Experimental & Theoretical Artificial Intelligence, 2016, 28(1/2): 97 - 112.

[13] GARRO B A, RODRÍGUEZ K, VÁZQUEZ R A. Classification of DNA microarrays using artificial neural networks and ABC algorithm [J]. Applied Soft Computing, 2016, 38: 548 - 560.

This work is partially supported by the National Natural Science Foundation of China (61675108), the Scientific Research Project of Zhejiang Provincial Education Department (Y201326770).

TONG Nan, born in 1981, M. S., lecturer. Her research interests include intelligent control, optimization algorithm, data mining.

FU Qiang, born in 1975, Ph. D. candidate, associate professor. His research interests include intelligent optimization algorithm, automatic design of integrated circuit.

ZHONG Caiming, born in 1970, Ph. D., professor. His research interests include machine learning, pattern recognition.