计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
49(10): 2181-2188, 2012

基于阈值的社交网络影响力最大化算法

陈 浩 王轶彤
(复旦大学计算机科学技术学院 上海 201203)
(09210240056@fudan.edu.cn)

Threshold-Based Heuristic Algorithm for Influence Maximization

Chen Hao and Wang Yitong
(*School of Computer Science, Fudan University, Shanghai* 201203)

Abstract Influence maximization is a problem of finding a subset of nodes in a social network that can maximize the spread of influence. This optimization problem of influence maximization is proved NP-hard. Kemple and Kleinberg proposed a natural climbing-hill greedy algorithm that chooses the nodes which could provide the best marginal influence (KK algorithm). KK algorithm has large spread of influence, but is too costly for large social network. We propose a threshold-based heuristic algorithm (TBH) for social network influence maximization based on activation threshold of nodes. The algorithm computes potential influenced nodes (PIN) according to dynamically changing activation threshold of nodes in activating process. In heuristic phase, we select the nodes with maximal PIN as seed nodes. Then, in greedy phase, we greedily select the nodes having maximal margin gain of influence as seed nodes. Our experiments demonstrate that, even without the greedy phase, the performance of our algorithm is close to that of KK algorithm, and our algorithm has relatively very short running time. The experimental results also show that our algorithm outperforms HPG with the same heuristic factor c.

Key words social network; influence maximization; heuristic algorithm; greedy algorithm; TBH

摘 要 对于社交网络影响力最大化问题,Kemple 和 Kleinberg 提出了有较好影响范围的贪心算法,但是 KK 算法的复杂度非常高,并不实用.利用线性阈值模型提出了一种基于节点激活阈值的启发式算法.它综合考虑了节点之间的影响力和节点的激活阈值,根据每个节点在激活过程中动态变化的阈值来计算 PIN 值,启发过程中,每一次都选取 PIN 最大的节点作为种子节点进行激活,贪心阶段中再贪心地挑选那些具有最大影响范围增量的节点作为种子节点.通过实验表明,即使在完全不采用贪心阶段,该算法的激活范围与 KK 算法都非常接近,而算法的复杂度则相对非常小.实验还表明该算法相对于 HPG 算法在相同启发因子 c 的情况下具有更大的激活范围.

关键词 社交网络;影响力最大化;启发式算法;贪心算法;TBH

中图法分类号 TP311.13

收稿日期:2012-06-05;**修回日期**:2012-07-25
基金项目:国家自然科学基金重点项目(61033010)
通信作者:王轶彤(yitongw@fudan.edu.cn)

社会网络是指社会个体成员之间因为互动而形成的相对稳定的关系体系,社会网络关注的是人们之间的互动和联系,社会互动会影响人们的社会行为.传统社会网络由于研究手段的限制,只能研究小规模群体之间的关系.

近年来,由于大型社交网络(SNS)如 Facebook, Flickr, Twitter 等的兴起,大型社交网络下的社会网络研究越来越成为研究的热点.

社交网络也就是通过网络这一载体把人们连接起来,从而形成具有某一特点的团体.社交网络中的个体可以与和他关联的个体进行互动,交流以及分享和推荐信息等.社交网络中的信息传播和扩散是通过个体与个体之间的交互来实现的.而社交网络影响力最大化问题是社交网络研究的一个热点问题.

社交网络影响力最大化问题是如何选取 k 个种子节点进行传播,从而使最终传播的影响范围最大. Kempe 和 Kleinberg 等人证明了这个问题是一个 NP 完全问题[1],并提出了一个贪心算法.它能保证在 $1-1/e$ 的范围内接近最优解.但是对于大型社交网络它的复杂度太高,因此并不实用.我们想要在提高影响范围的同时降低算法的复杂度.

田家堂等人基于线性阈值模型(LT)提出了一种利用节点的度数和潜在影响力的混合式启发式算法 HPG[2],但是 HPG 算法没有考虑到不同节点不同激活阈值的问题,而且无法在启发因子 c 较小的情况下得到较好的激活范围,即无法在较小的复杂度下得到好的激活范围.

目前影响力最大化问题的研究工作主要集中在 IC 模型下利用网络的次模特性(sub-modularity)来减少贪心算法的复杂度[3].利用次模特性可以在 KK 算法挑选种子节点时少计算大量节点的范围增量,但是它的算法复杂度对于大规模社交网络来说依然难以接受.

在本文中,我们基于线性阈值模型提出了一种基于节点的阈值的启发式算法.我们可以看出,在相同的 c 下我们的算法可以比 HPG 的影响范围更大.而且在 $c=0$ 的情况下(完全不利用 KK 算法),我们也能得到和 KK 算法很接近的激活范围,同时我们的复杂度相对 KK 算法非常得小.

本文的主要贡献如下:

1) 基于线性阈值模型提出了基于节点激活阈值的启发式算法;

2) 算法具有很好的激活范围以及非常低的复杂度;

3) 通过实验在不同的数据集上验证了算法的性能;

4) 我们实验了不同的阈值下算法和 HPG 和 KK 算法的性能比较.

1 背景知识

我们研究了一般将社交网络抽象成一个图 $G(V,E)$,其中 V 为网络中个体的集合,E 为网络中个体之间交互和关系的集合;社交网络影响力最大化问题是如何选取 k 个种子节点进行信息的传播和扩散,使得最终被激活的节点个数最多.社交网络中传播过程是,当一个节点被激活时,它会尝试激活与它连接的每一个未被激活的邻居节点,当邻居节点被激活时,它又会尝试激活它自己的邻居节点,这种过程会一直持续,直到没有新的节点被激活为止.

目前社交网络影响力最大化问题有两种激活模型,即线性阈值模型(LT)以及独立级联模型(IC).下面我们将介绍这两种模型.

1.1 独立级联模型

独立级联模型(independent cascade model)[1,4-5]是一种概率模型,当一个节点 v 被激活时,它会以概率 p_{vw} 对它未激活的出边邻居节点 w 尝试激活,这种尝试仅仅进行一次,而且这些尝试之间是相互独立的,即 v 对 w 的激活不会受到其他节点的影响.

目前关于独立级联模型的影响力最大化研究很多,IC 模型的特点是它仅仅考虑 u 与出边邻居 w 之间的激活关系,完全不考虑 w 的其他入边邻居对 w 的影响.但是由于是概率模型,它的激活过程是不确定的,对同一个网络,同样的种子节点进行激活得到的最后结果可能会差异很大.

1.2 线性阈值模型

线性阈值模型(linear threshold model)[1,6]是一种价值积累模型,它对每个节点 v 都有个激活阈值 $\theta_v \in [0,1]$,v 被它的入边邻居 w 以 b_{wv} 影响,b_{wv} 满足:

$$\sum_{w \in in(v)} b_{wv} \leqslant 1,$$

这里 $in(v)$ 是 v 的入边邻居节点集合.节点 v 被激活的条件是

$$\sum_{w \in in(v), active(w) \neq 0} b_{wv} \geqslant \theta_v,$$

即 v 已激活的入边邻居对 v 的累积影响大于 v 的激活阈值.在后面的实验过程中,我们取 $\sum_{w\in in(v)} b_{wv}=1$.

LT 模型的特点是它的激活过程是确定的.当我们对同一个图以同样的种子节点来激活时,最后的传播范围是完全一样的.而且当一个节点 v 被激活时,它会尝试激活它的每一个未激活出边邻居 w,即使这次的激活尝试并没有使得 w 被激活,但是 b_{vw} 会被累积下来,对此后其他节点对 w 的激活产生帮助.这表明 LT 模型的激活过程是一种合作激活的过程,每次的激活尝试都被累积下来.

1.3 相关工作

1.3.1 KK 算法[1]

Kemple 和 Kleinberg 提出了一种贪心算法,我们最终要寻找初始种子节点 S_k,我们令:

1) $S_0=\varnothing$;

2) $I(S_i)$:种子节点集合 S_i 最终激活节点;

3) $m(u|S_i)=|I(S_i\cup u)|-|I(S_i)|$:$u$ 作为种子节点添加到 S_i 中带来的激活范围的增量,则 KK 算法的思想就是每次寻找带来最大激活范围增量的节点 s 作为种子节点,即从 S_0 开始,在第 i 步,选取 $s=\underset{v}{\operatorname{argmax}}\, m(v|S_{i-1})$ 作为种子节点,令 $S_i=S_{i-1}\cup\{s\}$.

从 KK 算法的过程可以看出,由于 KK 算法每一步都需要就算每一个未激活节点作为种子节点而带来的激活范围增量 $m(u|S_i)$,它非常地耗时,对于大型网络它并不实用.

而且,对于 KK 算法来说,它仅仅考虑点 u 是否激活了邻居 v,而不考虑即使 u 未能激活邻居 v,但是 u 对激活 v 所做的贡献的大小.

1.3.2 对网络进行稀疏化[7-8]

Michael 提出了一种对网络结构进行稀疏化的方法来大量减少网络中边的数量[7-8],从而达到在较少的损失影响范围的情况下大量减少算法运行时间.

他考察了现实社交网络中一些行为 α 的传播路径,基于这些行为对网络进行稀疏化,使得稀疏化以后的网络让 α 中每一个可达的点仍然可达.而且他计算了所有这些行为的总的概率:

$$\log L_\alpha(G)=\sum_{v\in V}(\log P_\alpha^+(v)+\log P_\alpha^-(v)),$$

这里 $P_\alpha^+(v)$ 表示 v 在 α 行为中可达的概率,相对的 $P_\alpha^-(v)$ 表示 v 在 α 行为中不可达的概率.

最后他使得稀疏化后的图中 $\log L_\alpha(G)$ 能达到最大,而且 $\log L_\alpha(G)>-\infty$.

最终图中的边能得到大量的削减.但是,他的方法是基于 IC 模型的,利用了概率最大化来稀疏化最终图,所以对 LT 模型并不适用.

1.3.3 混合式影响最大化算法 HPG[2]

田家堂基于 LT 模型提出了一种利用节点的度数 $outDegree(u)$ 和节点 u 的潜在影响力的启发式算法,他混合了启发式算法和 KK 算法.

首先利用启发式算法来选择 $k-\lceil ck\rceil$ 个种子节点来激活网络:

$$inf(u)=\sum_{v\in out(u),active(v)=0} b_{uv};$$

$$PI(u)=outDegree(u)+(1-e^{-inf(u)}).$$

启发阶段每一步选取 PI 值最大的节点 u 作为种子节点激活网络.

贪心阶段利用 KK 算法选取剩余 $\lceil ck\rceil$ 个种子节点.但是 HPG 算法的启发阶段并没有考虑每个节点阈值不同的情况,仅仅将 b_{uv} 进行相加.而且若 $b_{uv}>\theta_v$,则 b_{uv} 表明 v 节点提供了超过一个节点的价值,这显然不太合理.

考虑图 1 的情况:

$b_{u_1v_1}=0.8,\theta_{v_1}=0.9;b_{u_2v_2}=0.5,\theta_{v_2}=0.3.$

显然 u_1 不能够激活 v_1 节点,而 u_2 能激活 v_2 节点,但是 $b_{u_1v_1}>b_{u_2v_2}$,因此,按照 HPG 算法,则认为 u_1 节点的潜在价值大于 u_2 节点的潜在价值;然而事实上 u_2 节点能够激活一个节点,而 u_1 节点并不能激活一个节点,所以 u_2 更具有价值.

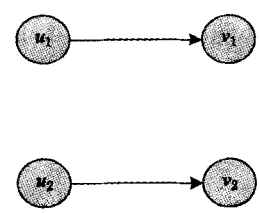

Fig. 1 u_1 and u_2 both have one out-neighbor v_1 and v_2.

图 1 u_1 和 u_2 都只有 1 个出边邻居 v_1 和 v_2

2 基于阈值的影响力最大化算法

2.1 算法框架

我们提出的算法分为两个阶段:第 1 个阶段为启发阶段,即并不像 KK 算法那样寻求局部最优,我们通过 LT 模型的影响累积特性来启发式寻找那些能提供最大潜在影响的节点作为种子节点;第 2 个阶段则在第 1 个阶段激活的基础上,利用 KK 算法局部最优的特性来尽可能扩展影响范围.

由于第 1 阶段仅仅是启发地选取种子节点，它的算法复杂度和 KK 算法相比几乎可以忽略，因此，我们算法的复杂度仅仅与第 2 阶段 KK 算法所占的比例相关，即 KK 算法所占比例越小我们算法的复杂度越小.

而通过实验我们发现，对大部分实验数据集来说，即使我们完全不利用 KK 算法，仅仅利用第 1 阶段基于阈值的启发式算法来选取种子节点激活网络，最后影响范围和 KK 算法的影响范围也非常地接近，而这时算法复杂度相比 KK 算法则极小.

而增加 KK 算法所占的比例也是可以提升整个算法的影响范围. 因此，我们可以根据具体对影响范围和运行时间的要求来合理选取 KK 算法的比例.

我们提出基于阈值的影响力最大化算法是根据这样一个发现：

考虑一条边(u,v)的价值要看这条边对激活 v 节点的贡献大小，这和 v 的激活阈值相关，记：

1) θ_v 为节点 v 的初始激活阈值；

2) c_v 为节点 v 已激活的入边邻居对 v 的累积影响值；

3) b_{uv} 为节点 u 对节点 v 的影响力.

设 w 为一个未激活的节点，边(v,w)的影响力为 b_{vw}，则 θ_w-c_w 为 w 的即时激活阈值，我们定义：

$$p(v,w)=\begin{cases}\dfrac{b_{vw}}{\theta_w-c_w}, & \dfrac{b_{vw}}{\theta_w-c_w}<1\\ 1, & \dfrac{b_{vw}}{\theta_w-c_w}\geqslant 1\end{cases}, \tag{1}$$

$p(v,w)$为边(v,w)的潜在影响力，它表示边(v,w)对激活节点 w 能产生的贡献. 当 $b_{vw}\geqslant\theta_w-c_w$ 时，表明(v,w)足够激活 w 节点. 这时，b_{vw} 带来的效果都是激活了一个节点，我们认为一条边不能带来超过激活一个点带来的影响力. 因此当$\dfrac{b_{vw}}{\theta_w-c_w}\geqslant 1$ 时，我们使得 $p(v,w)=1$. 我们定义节点 v 的潜在影响节点数 $PIN(v)$：

$$PIN(v)=\sum_{w\in out(v),\,active(w)=0} p(v,w), \tag{2}$$

$PIN(v)$为 v 对其指向的未激活节点的影响总和.

我们利用 $PIN(v)$作为启发算法来快速的选取种子节点. 从实验中可以看出本文的算法和 KK 算法的激活范围非常接近，而且本文算法的复杂度相对则非常得小.

选取种子节点时，我们每次挑选 PIN 值最大的未激活节点作为种子节点.

从 PIN 的定义可以看出，当一个节点的累积影响变化时，它的所有未激活入边邻居的 PIN 值都会发生改变，因此我们需要不断地更新 PIN 值.

然而我们发现，我们仅仅在启发选取下一个种子节点时需要利用节点的 PIN 值，因此，尽管在激活的过程中，节点的激活阈值可能发生多次改变，并不需要即时更新入边邻居的 PIN 值，只要在当前种子节点的激活过程结束以后，对激活过程中覆盖到的节点的入边邻居的 PIN 值更新即可.

我们利用 *cover* 数组来存储种子节点 *seed* 在激活过程中覆盖到的节点. 每个节点都有一个覆盖标记 $C(v)$，代表节点 v 被哪个种子节点影响到.

seed 作为种子节点开始激活过程时，我们首先将种子节点 *seed* 加入 *cover* 数组中. 在激活过程中，当一个节点 v 被激活时，它会对它的所有未激活的出边邻居 w 施加影响值，这时，我们查看 C_w，若 $C_w\neq seed$，则说明 w 节点第 1 次在 *seed* 激活过程中被覆盖到，我们将 w 节点加入 *cover* 数组中，并令 $C(w)=seed$.

当 *seed* 节点的激活过程结束后，我们就需要对 *cover* 数组中的每一个节点的未激活入边邻居的 PIN 值更新.

我们来考察 *cover* 数组中一个节点 v 的未激活入边邻居 u 的 PIN 值变化. 设 r_v 中记录了 v 在 *seed* 节点激活过程中第 1 次被影响到之前的累积影响值，这在 *seed* 激活中第 1 次影响到 v 节点时会进行记录. 则首先 $PIN(u)$要消去之前(u,v)边带来的影响值：

$$PIN(u)=\begin{cases}PIN(u)-\dfrac{b_{uv}}{\theta_v-r_v}, & \dfrac{b_{uv}}{\theta_v-r_v}<1\\ PIN(u)-1, & \dfrac{b_{uv}}{\theta_v-r_v}\geqslant 1\end{cases}. \tag{3}$$

若 v 在 *seed* 过程中被激活，则 $PIN(u)$值不需要再改变；否则，$PIN(u)$需要增加(u,v)边这时带来的影响值：

$$PIN(u)=PIN(u)+p(u,v), \tag{4}$$

$$p(u,v)=\begin{cases}\dfrac{b_{uv}}{\theta_v-c_v}, & \dfrac{b_{uv}}{\theta_v-c_v}<1\\ 1, & \dfrac{b_{uv}}{\theta_v-c_v}\geqslant 1\end{cases}, \tag{5}$$

这里，$\theta_v>c_v$.

我们发现，若 v 在 *seed* 激活过程中被影响到但是没有被激活，则 v 的未激活邻居 u 的 PIN 值会变大，这提升了在下一个种子节点选取 u 的概率，说明我们的启发式算法考虑到了网络的结构，它更倾向于配合之前已激活的节点.

在本文的算法中,我们设 k 为选取的种子节点数,$c(0\leqslant c\leqslant 1)$为第 2 阶段 KK 算法所占的比例,则 $k_1=k-\lceil ck\rceil$ 为第 1 阶段选取的种子节点数,$k_2=\lceil ck\rceil$ 为第 2 阶段 KK 算法选取的种子节点数,S 为种子节点的集合.

2.2　算法复杂度分析

设网络中的节点数为 N,每个节点的平均入度以及出度为 D,则 $N\times D=|E|$,启发阶段每个种子节点的平均覆盖范围为 H,KK 算法每个节点的平均覆盖范围为 K. 这里我们假设激活的节点占总节点数较小一部分.

则图 2 的 TBH 算法中,启发阶段的复杂度为 $O((H\times D)\times(k-\lceil ck\rceil))$;贪心阶段的复杂度为 $O((N\times K)\times\lceil ck\rceil)$.

TBH 总的复杂度为 $O((H\times D)\times(k-\lceil ck\rceil)+(N\times K)\times\lceil ck\rceil)$;KK 算法的复杂度为 $O((N\times K)\times k)$.

从我们的实验可以得出 K 和 H 是比较接近的,而 $\lceil ck\rceil$ 和 $(k-\lceil ck\rceil)$ 代表了每个阶段取的种子节点数. $D=\frac{|E|}{N}$,相对于 N 是非常小的,因为现实的社交网络中每个节点的度数相对于网络中的节点数是非常小的,因此 $D\ll N$. 所以 $O(H\times D)\ll O(N\times K)$,因此,TBH 算法取决于 c 的大小,即贪心阶段所占的比例.

Input: Graph $G(V,E)$, b_{uv}, θ_v, seed nodes number k, c
Output: seed nodes set S_k, number of activated nodes
① Initialize $S_0=\varnothing$, $k_1=k-\lceil ck\rceil$, $k_2=\lceil ck\rceil$
② Compute PIN of each node
③ For $i=1$ TO k_1
④　Select inactivated node s with largest PIN
⑤　$S_i=S_{i-1}\cup\{s\}$
⑥　Active network with seed node s, and add nodes that is covered to cover array
⑦　Update PIN of inactivated in-neighbors of nodes in cover array
⑧ Endfor
⑨ For $i=1$ To k_2
⑩　Calculate the increment of influence range of each inactivated node u being seed node
⑪　Choose node v with the largest increment of influence range as seed node
⑫　$S_{i+k_1}=S_{i-1+k_1}\cup\{v\}$
⑬　Activate network with seed node v
⑭ Endfor

Fig. 2　TBH Algorithm.
图 2　TBH 算法框架

3　实　　验

3.1　b_{uv} 的计算

3.1.1　带权图的 b_{uv} 计算

对于带权图 $G(V,E)$,每条边 (u,v) 具有权重 W_{uv},对 v 的所有入边邻居 $u(u\in in(v))$:

$$b_{uv}=\frac{W_{uv}}{\sum\limits_{\omega\in in(v)}W_{\omega v}},\tag{6}$$

这样使得 $\sum\limits_{u\in in(v)}b_{uv}=1$.

3.1.2　无权图的 b_{uv} 计算

对于无权图 $G(V,E)$,我们可以认为每条边的权重都相同,令 $deg(v)=|in(v)|$ 为 v 的入度:

$$b_{uv}=\frac{1}{deg(v)},\ u\in in(v).\tag{7}$$

3.2　实验数据集

如表 1 所示:第 1 个数据集是计算几何合作网络,是一个无向带权图,边上的权重代表了 2 个节点之间合作的次数;

第 2 个数据集是 Epinions 网站上数据,有向无权图,节点 u 到 v 的有向边表示 u 对 v 信任;

第 3 个数据集是 Slashdot 网站 2009 年 2 月的数据,有向无权图,节点 u 到 v 的有向边表示 u 信任 v.

实验的数据集来自 Stanford 大学的大型网络数据搜集网站(http://snap.stanford.edu/data/index.html).

Table 1　Information of Data Sets
表 1　数据集信息

No	Data Set	Nodes	Edges	Average Degree	Graph Type
1	co-operate network	7343	11898	3.2	Undirected weighted
2	Epinions	75879	508837	6.7	Directed unweighted
3	Slashdot	82168	948464	11.5	Directed unweighted

3.3　实验结果

3.3.1　带权网络上的实验结果

1) 所有节点初始激活阈值 $\theta=0.5$ 的实验结果

这里实验采用的是数据集 1 作者合作网络,首先根据数据集中边的权值利用 b_{uv} 计算公式将权重转化为 b_{uv}.

图 3 为数据集 1 上与 KK 算法以及 HPG 算法激活范围的比较,从结果中可以看出,在 $c=0$ 时,

TBH 比 HPG 算法有更好的激活范围，而且随着 c 的增加，激活范围是显著增加的. 而且在种子节点数较小时，我们的算法比 KK 算法有更好的激活范围.

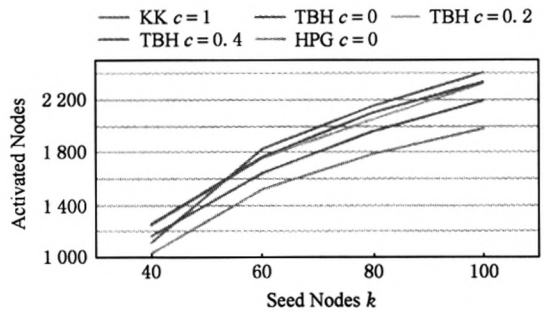

Fig. 3 Activated Nodes of KK, HPG and TBH on data set 1.

图 3 数据集 1 上与 KK 算法以及 HPG 算法激活范围比较

下面我们将比较在 k 相同的情况下($k=100$)，我们的算法和 HPG 的激活范围.

图 4 是当 k 相同($k=100$)时，不同 c 与 HPG 算法影响范围的比较，从图 4 可以看出，TBH 在不同 c 下都比 HPG 有更高的激活范围，而且 c 越小差距越大. 这很容易理解，因为 c 越大 KK 算法所占的比例越大，所以两种算法激活范围也会越接近.

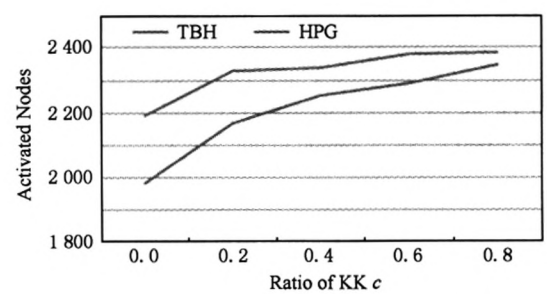

Fig. 4 Activated Nodes of HPG and TBH with different c when $k=100$.

图 4 $k=100$ 时不同 c 下与 HPG 影响范围的比较

2) 所有节点初始激活阈值 $\theta\in[0,1]$取随机的实验结果

下面我们将对每个节点的初始阈值 θ 取[0,1]范围内的随机值，为了使得每次实验的图都是相同的，我们以相同的种子初始化随机函数.

随机阈值以后，所有节点的阈值平均值为 0.5075，从图 5 可以看出，在随机阈值的情况下，TBH 算法依然能与 KK 算法保持很接近的激活范围，而且，在取 $c=0.2$ 以及 $c=0.4$ 的情况下，我们的算法能够在激活范围上超过 KK 算法.

其实激活范围超过 KK 算法是可以理解的，KK

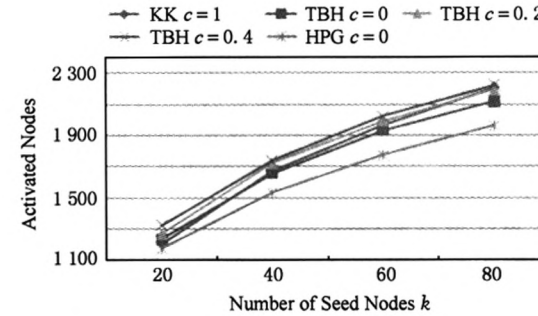

Fig. 5 Activated Nodes with random activation threshold.

图 5 对节点阈值取随机时算法的激活范围比较

算法考虑的是局部最优，对一条边(u,v)，若它不能使得 v 被激活，KK 算法就会在挑选种子节点时忽略这条边带来的价值. 而我们的算法在启发阶段挑选那些具有最大潜在影响节点数的点作为种子节点激活网络，会使得网络中大量点的即时激活阈值变小，从而使得 KK 步骤中这些点更容易得到激活，从而达到更大的激活范围.

图 6 为在随机阈值的情况下，选取 80 个种子节点时不同 c 下两种算法的激活范围. 我们的算法 TBH 在很小的 c 就能达到非常接近 KK 算法的范围，而 TBH 与 HPG 算法复杂度取决于 KK 算法在其中所占的比例，因此，我们可以选取很小的 c 来达到很好的激活范围，从而在激活范围和运行时间上都有很不错的效果.

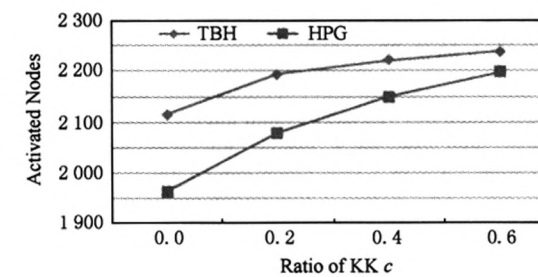

Fig. 6 Activated Nodes of HPG and TBH with randomized activation threshold when $k=80$.

图 6 在随机阈值 $k=80$ 下 TBH 与 HPG 不同 c 下的比较

3.3.2 在无权网络上的实验结果

下面我们将在有向无权网络上验证我们的算法性能. 这两个网络都具有较大的节点数以及较高的平均度数. 我们两个实验都采用激活阈值 $\theta=0.5$，因为即使每个节点的初始阈值相同，在激活过程中每个节点的即时初始阈值也会变化.

1) Epinions 数据集上的实验结果($\theta=0.5$)

以下是在 Epinions 数据集上的实验结果，图 7

为不同种子节点数下各算法的激活范围,从图 7 中可以看出,即使在 $c=0$ 时 TBH 与 KK 算法的激活范围也几乎重合. 图 8 为各算法的运行时间,与 KK 算法比较,TBH 在 $c=0$ 时运行时间几乎可以忽略,在 $c=0.2$ 时,运行时间也约等于 $0.2T(\mathrm{KK})$. 图 9 为 TBH 和 HPG 算法在 $k=100$ 时不同 c 下的影响范围比较. 可以看出,TBH 比 HPG 算法在相同 c 下具有更大的激活范围.

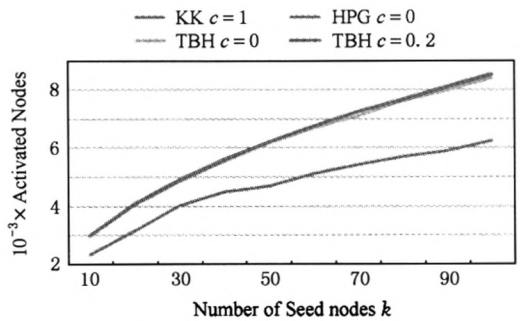

Fig. 7　Activated Nodes on Data set Epinions.

图 7　Epinions 数据集上的激活范围比较

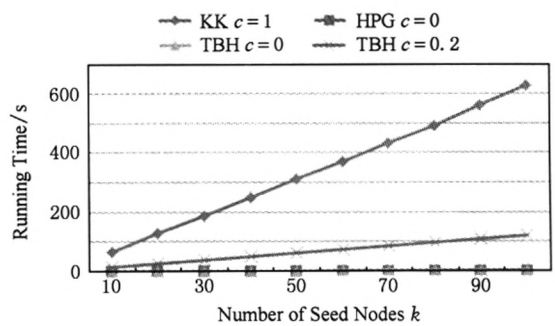

Fig. 8　Running time of the 3 algorithms.

图 8　各种算法的运行时间比较

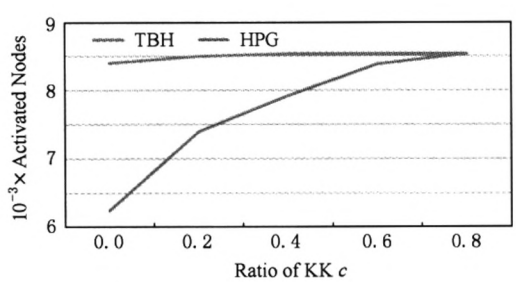

Fig. 9　Activated Nodes of HPG and TBH with different c when $k=100$.

图 9　$k=100$ 时 TBH 和 HPG 算法在不同 c 下的激活范围比较

2) Slashdot 数据集上的实验结果($\theta=0.5$)

图 10 和图 11 是在 Slashdot 数据集上的实验结果,即使在 $c=0$ 时,我们发现算法 TBH 与 KK 算法都有很接近的激活范围,$c=0.2$ 时,激活范围得到进一步提高. 而相对于 KK 算法,$c=0$ 时,TBH 与 HPG 的运行时间可以忽略. $c=0.2$ 时运行时间也相对非常的低.

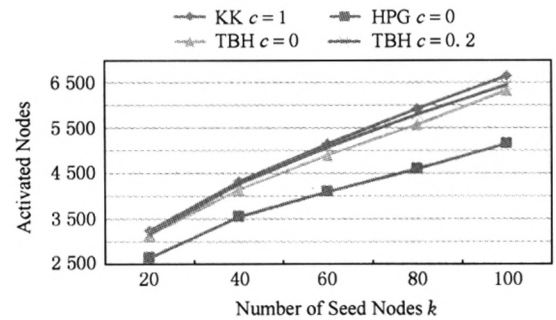

Fig. 10　Activation range of different algorithm on Slashdot data set.

图 10　Slashdot 数据集上不同算法的激活范围

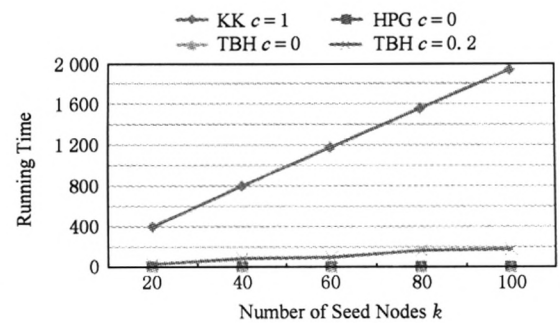

Fig. 11　Running time of the algorithm.

图 11　算法的运行时间

从以上实验可以看出我们的算法 TBH 有着和 KK 算法很接近的激活范围,而算法的复杂度则很小,而且 TBH 算法在相同 c 下比 HPG 算法的激活范围要好,因此我们只需要较小的 c 值就能取得较好的激活范围. 在实际的社交网络应用中,我们可以根据对激活范围和运行时间的具体要求来取不同的 c 值. 特别地,在要求运行时间非常低的情况下,我们可以取 $c=0$,即完全不利用 KK 算法,这样也能得到比较接近 KK 算法的激活范围,而只需要相对可以忽略的运行时间.

4　总　　结

本文利用线性阈值模型,结合节点的激活阈值提出了基于节点激活阈值的启发式算法 TBH. 通过实验我们发现,算法在启发因子 c 很小的情况下就能很接近 KK 算法的激活范围,即使在 $c=0$ 的情况下,算法在激活范围上也比较接近 KK 算法,而且,算法相比较 HPG 算法在 c 相同时具有更好的激活.

范围以及类似的运行时间.

在实际应用时,TBH 算法相对于 HPG 算法在达到相同影响范围只需要较小的 c,因此在运行时间上也更优.

我们可以根据应用时具体的范围和运行时间需求来调整 c 的取值,通常我们只需要取较小的 c 值.

本文的算法依然有需要改进和值得进一步研究的地方,比如本文的算法在带符号网络上的表现;网络中每个节点的价值不同的情况下算法的变化;在现实中信息的传播是需要时间的,而不同次序的激活带来的最终激活会不一样,如何解决这种问题等.

参 考 文 献

[1] Kempe D, Kleinberg J, Tardos E. Maximizing the spread of influence in a social network [C] //Proc of the 9th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2003:137-146

[2] Tian Jiatang, Wang Yitong, Feng Xiaojun. A new hybrid algorithm for influence maximization in social networks [J]. Chinese Journal of Computers, 2011, 34(10): 1956-1965
(田家堂, 王轶彤, 冯小军. 一种新型的社会网络影响最大化算法[J]. 计算机学报, 2011, 34(10): 1956-1965)

[3] Chen Wei, Wang Yajun, Yang Siyu. Efficient influence maximization in social networks [C] //Proc of the 15th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2009: 199-208

[4] Young H P, Blume L, Durlauf S. The diffusion of innovations in social networks [M]. The Economy as a Complex System III. New York: Oxford University Press, 2003: 1-19

[5] Watts D J. A simple model of global cascades on random networks [J]. National Academy of Sciences, 2002: 99(9): 5766-5571

[6] Goldenberg J, Libai B, Muller E. Talk of the network: A complex systems look at the underlying process of word-of-mouth [J]. Marketing Letters, 2001, 12(3): 211-223

[7] Michael M, Francesco B, Carlos C. Sparsification of Influence Networks [C] //Proc of the 17th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2011: 529-537

[8] Manuel G, Jure L, Andreas K. Inferring networks of diffusion and influence [C] //Proc of the 16th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2010: 1019-1028

Chen Hao, born in 1987. MSc candidate. His research interests include social networks, Web mining.

Wang Yitong, born in 1973. PhD and associate professor. Member of China Computer Federation. Her research interests include database, Web mining, data mining and Web infromation retrieval.