在接近最优的时间内求解影响力最大化问题

张伟

摘要

基于图的传播过程广泛存在于现实生活中，比如，传染病的传播，想法或者信息的传播，以及产品的推广过程等。为了更好地研究传播过程，研究者们提出了许多模型，其中，比较广泛的是，独立级联模型（Independent Cascade Model）和线性阈值模型（Linear Threshold Model）。基于传播模型的研究中有一类是**影响力最大化问题**，属于优化问题，目的在于寻找数目约束的最优种子集合，其正式表述是：在种子结点数不超过 k 的前提下，选择哪些种子结点，能使最终受到影响的种子数目最多，即影响力最大。这类问题是 NP-hard 类问题，只能找到近似解，研究发现，影响力扩展度函数具有单调次模性，常用的方法是基于蒙特卡洛的贪心算法，但是，该算法的时间复杂度高，在大型网络的计算中往往捉襟见肘，难以满足运行时间的要求。鉴于此，作者提出了基于反向可达集的贪心算法，达到了几乎最优的近似比 $1-\frac{1}{e}-\varepsilon, \forall \varepsilon>0$，并且运行时间是 $\mathcal{O}((m+n)\varepsilon^{-3}\log n)$，而之前最好的算法的运行时间为 $\Omega(mnk\cdot \mathrm{POLY}(\varepsilon^{-1}))$。

关键词：独立级联模型；影响力最大化；传播过程

1 引言

网络无处不在，任何社会性动物在个体与个体、群体与群体之间都存在着相互影响的关系，而人类作为具有复杂交流手段的高级社会性动物，人际和社会影响力在人们的社会生活中更是无处不在。我们在现实生活中做出的许多选择和决定，也常常受到周围的人和环境的影响。深入认识影响力的产生和传播模式有助于理解人类群体和个体的行为，从而使得我们能够预测人的行为，进而能为政府、机构、企业等部门的决策提供可靠的依据和建议。比如，企业在做新产品的推广时，可以利用对用户影响力及其传播的了解，选择有影响力的用户和传播渠道，从而帮助产品推广；公益机构也可以通过影响力传播推动公益事业的发展，比如增强民众的健康意识、环保意识等；政府可以选择合适的影响力群体和渠道来扩大其政策的影响，或者通过研究传播过程的规律，能够尽早发现谣言并阻断谣言的传播过程。

传播过程是一个基本的图过程，现实生活中的很多现象都是基于传播过程，比如，传染病的传播，想法或者信息的传播。在关于影响力传播的研究中，常采用的模型有独立级联模型（Independent Cascade Model）和线性阈值模型（Linear Threshold Model），两种模型的主要区别在于其随机性体现的地方不同，独立级联模型的随机性多体现边的激活概率上，而线性阈值模型的随机性则体现在结点的受影响阈值上，除此之外，独立级联模型刻画的是结点独立影响相邻结点的过程，而线性阈值模型则刻画的是几个结点联合影响其共同相邻结点的作用。本文中，作者使用的是独立级联模型。基于对传播过程的不同特点的刻画，，其中最广泛，想象的传播过程不同，着重于不同的传播特点，独立级联模型更多地是把传播过程看作结点之间是否有边存在，即是否发生了影响作用，而线性阈值模型则关注的是多人对单人的影响作用是否超过其阈值。

影响力传播中有一类问题就是影响力最大化问题（influence maximization problem），这一问题得到了广泛的关注，而影响力最大化问题就是寻找一个包含 k 个结点的种子集合 S_0，使得在传播结束

时，被激活的结点数目最多，也就是说，用受影响的种子个数来表示某个种子集合 S_0 对剩余结点所产生的影响力，并且，由于传播过程是一个随机过程，受影响的种子数目并非一个确定值，而是一个随机变量，因此，我们需要对同一种子集合的传播过程进行多次独立重复试验，然后对每次产生的结果求期望，最终得到的这个期望值就被定义为种子集合 S_0 的影响力扩展度（influence spread）。也就是说，影响力最大化问题就是要寻找一个具有 k 个种子的集合 S_0，使得以该种子集合为起点，其最终的传播影响力值最大。

关于影响力最大化问题的算法有很多，常见的算法主要是基于影响力扩展度函数单调次模性的贪心算法，这一算法的时间复杂度高，在小规模网络计算中能够实现比较好的性能，但近年来，随着网络规模的不断增大，以及动态网络中的实时计算的需求，对运行时间提出了更高的要求，基于此，本文作者提出了一个快速算法来应对这一挑战，该算法获得了近似最优的系数 $1-\frac{1}{e}-\varepsilon$，对任意的 $\varepsilon>0$ 成立，其中，ε 表示精度要求，并且算法的运行时间为 $\mathcal{O}((m+n)\varepsilon^{-3}\log n)$，这一公式也说明了，$\varepsilon$ 越大，则 $1-\frac{1}{e}-\varepsilon$ 也就越小，结果与最优解的距离也就越远，同时，所需要的运行时间也就相对越小，这也比较符合直觉，当要求的精度比较低时，可以适当缩短运算时间，提前结束，或者说，算法运行时间随着精度要求的降低而降低。作者在文中用大量的推到证明了算法的有效性，并且，该算法相较于之前的算法都有很大程度的提升，提升到了对数级别（log），之前最好的算法的时间复杂度也只是 $\Omega(mnk\cdot \mathrm{POLY}(\varepsilon^{-1}))$。

2 相关工作

传播是复杂网络研究的基础过程，疾病的传播、想法或产品的推广等等都是传播过程，可以通过建立传播模型分析并研究这些过程，这些过程的共同特点是局部的个体间交互行为能够引发传染性结果，这也是口碑广告背后的理念，关于产品的信息通过个体间的链接传递，在这方面有很多研究[1][2][3][4][5][6]。一个越来越显著的应用领域是病毒式营销，在这一推广过程中，目标是利用局部的干预来引发影响力的级联效应，以此推广产品或者理念[7][8][9][5]。这一应用带来了新的算法问题，也就是：给定一个网络，应该选择哪些个体使得最终的传播效果最好，也就是接受所传播对象的人数最多。这一问题就是影响力最大化问题，关于该问题的研究有很多[7][8][10]。该问题的目标是在节点数 k 有限的前提下，寻找合适的结点，以达到最终传播的效果最佳，传播效果使用最终受到影响的结点个数为衡量标准。

2.1 网络传播模型的概述和分类

通常，我们把一个社交网络抽象为一个有向图 $G=(V,E)$，其中 V 是结点的集合，而 $E\subseteq V\times V$ 影响力最大化问题就是找出社交网络中的少数结点作为种子结点，使得这些种子结点的影响力扩展度最大。在影响力最大化问题中，给定一个有向图 $G=(V,E)$ 及其上的一个二值离散时间递进性传播模型，在预算 k 有限的前提下，找到一个最多 k 个结点的种子集合 S_0^*，使得该种子集合的影响力扩展都最大，即

$$S_0^*\in \mathrm{argmax}_{S_0\subseteq V,|S_0|\leqslant k}\,\sigma(S_0)$$

影响力最大化问题的一个直接的应用场景就是病毒式营销（viral marketing），想象这样的一种场景，一家企业想要推广自己的产品，需要在网络中挑选一些人作为初始用户，并且希望选中的用户在试用

产品之后能够主动向别人推广，关于如何选择用户是一个重要的问题，一个可能的策略是选择网红（influencer），但是不是只要选择粉丝（fans）最多的网红就能最大化自己的利益呢？其实，每个网红所能影响到的群体不同，即便是最有名的网红，也有很多用户没有关注，并且，厂商的预算往往有限，如何合理利用预算做出最好的选择，这是一个需要研究的问题。而这个问题的研究首先需要对传播过程进行建模，通过数学方法对传播模型进行分析求解，找到最佳用户群（种子结点集合）使得最终接受产品的人最多（影响力扩展度最大），这便是影响力最大化问题的优化目标。由于影响力最大化问题在独立级联模型和线性阈值模型上都是 NP 难（NP-hard）问题，所以要解决这一问题就需要另辟蹊径，其中的一个重要方法就是利用有效的近似算法，这也意味着，在影响力最大化这个具体问题中，即使找不到使影响力扩展都达到最大的种子集合，但依然可能找到一个较好的集合，使得该集合的影响力扩展度与最优值的差距较小，这个近似解和最优值之间的比例便是近似算法的近似比。关于近似算法，最早的便是基于影响力扩展度函数的次模性的贪心算法。集合函数是指这样的一类函数，其自变量为集合，而因变量是一个数值，也就是刻画了从集合到实数的映射关系，影响力最大化问题所研究的函数指的是 $f:2^V \to \mathbb{R}$，其中 2^V 表示集合 V 的所有子集，2 的含义是集合 V 中的每个元素都是二元状态，即选择或不选择，可以用 $0,1$ 来表示，所以，2^V 也表示一个集合，该集合中的元素是 V 的所有子集。例如，一个集合 $V=1,2,3$，则相应地 $2^V=\{\emptyset,\{1\},\{2\},\{3\},\{1,2\},\{2,3\},\{1,3\},\{1,2,3\}\}$，而相应的映射关系也就是集合中的每个元素对应的实数轴上的点。集合函数中有一类函数具有单调次模性，其中，我们说一个函数是单调的（monotone），指的是对于所有满足 $S \subseteq T$ 的集合来说，其 $f(S) \leqslant f(T)$；而次模性指的是，若 $S \subseteq T$，对结点 $u \notin T$，那也一定 $u \notin S$（因为 $S \subseteq T$）, 如果满足 $f(S \cup \{u\}) - f(S) \geqslant f(T \cup \{u\}) - f(T)$，则函数具有次模性（submodular）。

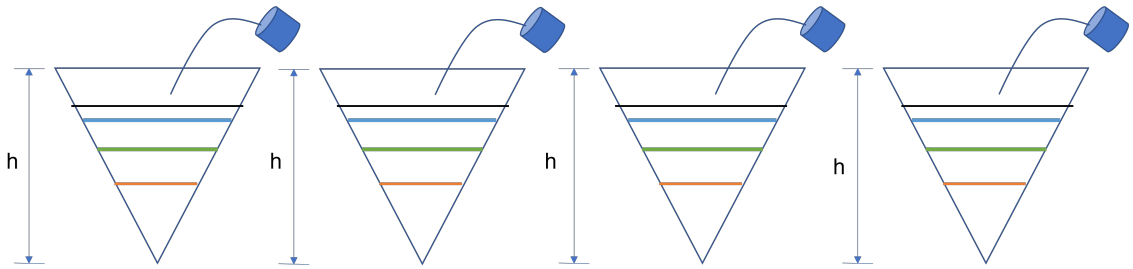

图 1: 单调次模性

进一步，单调性和次模性可以由图 1中的现象来解释，假定我们往一个锥体容器中加水，每次都往里倒入 1 杯水，即每次倒入水的体积相同，如果以水的高度 h 作为研究对象，只要一直加水，高度就会增加，这体现了单调性，但是，随着容器内的水面升高，再加水，其液面的高度增量会递减，这也就是次模性。

单调次模函数的一个重要的性质就是能用贪心算法找到一个近似最优解，其算法过程分为 k 轮，初始时种子集合 S 为空集

2.2 基于单调次模性的贪心算法

假定有一个传播网络可以抽象为如图 2所示的独立级联模型，在传播过程中，每条边被选中的概率为 p，未被选中的概率就为 $1-p$，对于不同的边来说，选择与否相互独立，比如，若两条边都未选中，那么其概率为 $\Pr = 1-(1-p_1)(1-p_2)$，这便是独立级联模型的核心特点。

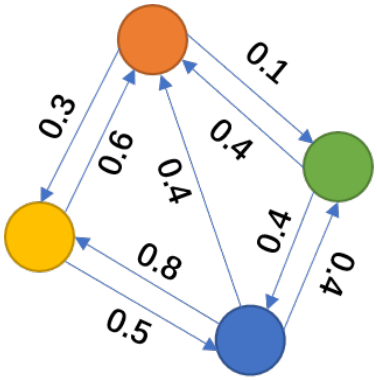

图 2: 独立级联模型

基于蒙特卡洛的贪心算法利用了扩展度函数的单调次模性，其伪代码如下所示，其中输入为图 G 和种子大小 k，一般我们取 $k = 50$，初始化种子集合 $S = \emptyset$，以及轮数 $R = 20000$，总共需要进行 k 次选择，并且每次选择需要在从 $O(n)$ 级别的种子个数中选出一个作为种子结点，每选择一个种子 v_i，将影响力扩展度初始化为 $s_v = 0$，然后进行 R 次模拟，这相当于独立重复实验 R 次，然后求这 R 次独立重复实验的期望，其中 RanCas($S \cup \{v\}$) 表示的是以 $S \cup \{v\}$ 为种子结点得到的最终结点集合，而集合的模 |RanCas($S \cup \{v\}$)| 则表示该集合中元素的个数，每次传播过程的模拟需要选择边，因此时间复杂度为 $O(m)$，最后通过 $s_v = s_v/R$ 求得最终的期望值，作为种子 v_i 的影响力扩展度，在求完每个种子的影响力扩展度以后，选出影响力扩展度最大的结点作为候选种子结点，放入种子集合 S 中，然后再进行下一轮选择。所以其时间复杂度为 $O(knRm)$。

Algorithm 1 GeneralGreedy(G, k)

Input: Graph G and seed number k
Output: Seed Set S with k nodes

initialize $S = \emptyset$ and $R = 20000$
for $i = 1$ to k **do**
  for each vertex $v \in V \setminus S$ **do**
    $s_v = 0$
    for $i = 1$ to R **do**
      $s_v += |\text{RanCas}(S \cup \{v\})|$
    end for
    $s_v = s_v/R$
  end for
  $S = S \cup \{\text{argmax}_{v \in V \setminus S}\{s_v\}\}$
end for
return S

这一基本过程如图所示，其中为了方便，省去了结点之间的边，最左侧未初始状态，接下来，每轮选择一个结点，并在图上进行蒙特卡洛模拟，假设第一轮选择了蓝色作为种子结点，那么第二轮开始有三个结点可供选择，依次选择这三个结点，并比较影响力扩展度大小，最终确定哪个结点与蓝色结点一同作为种子集合中的结点，这就是该算法的基本过程，假设所有结点用 V 表示，进而节点数目为 $|V|$，对于该贪心算法来说，总共需要进行的次数可以表示为 $C^1_{|V|} \cdot C^1_{|V|-1} \cdots C^1_{|V|-k+1} = \dfrac{|V|!}{(|V|-k)!}$，最后，还要乘以 R，即每次需要进行的蒙特卡洛模拟次数，所以，总的次数是 $\dfrac{|V|!}{(|V|-k)!}R$，不难看出，当图的规模很大时，即 $|V|$ 的值很大时，计算的复杂度很高，相应地，其运行时间也会大幅提升。对于一些动态更新的网络来说，可能计算出一个结果时，已经经历了很长时间，导致结果失去可用性，因此，需要提出更为高效的算法。

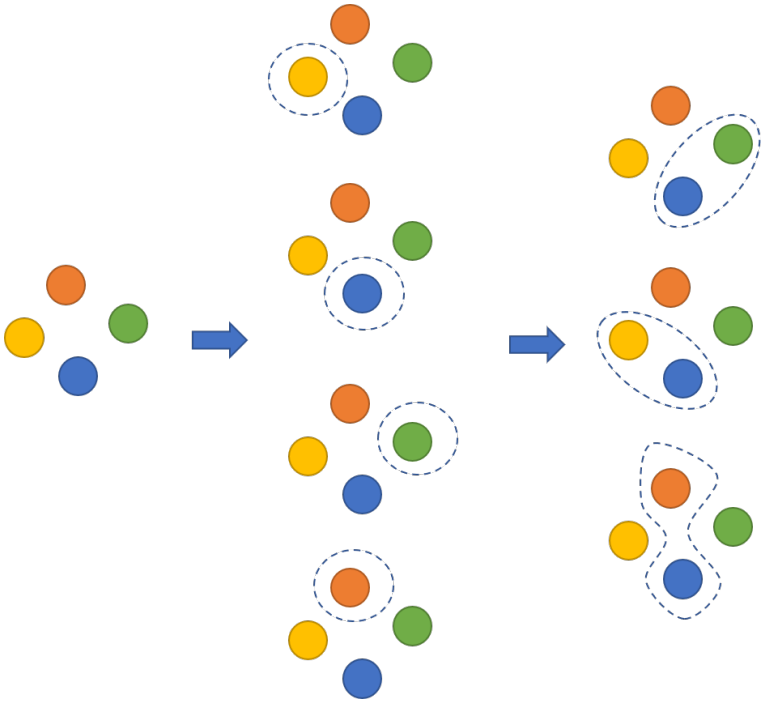

图 3: 基于单调次模性的贪心算法

3 本文方法

3.1 本文方法概述

本文使用了民意调查（polling）的思想，其基本思想如下图所示，常规的网络传播过程是从左到右，也就是从绿色和蓝色结点开始，经过许多中间结点，传递到右侧的结点中，而反向传播网络是将原网络中的有向边颠倒，取反方向，从右侧结点开始向左侧传播的过程，根据反向传播网络可得，灰色结点可以到达的节点有蓝色和绿色结点，橙色结点可以到达的集合亦是蓝色和绿色结点，而黄色结点能够到达的结点集合只含有绿色结点，经过简单的统计，绿色结点所处的集合有 3 个，而蓝色结点可以到达的集合有 2 个，所以，绿色结点的影响力更大，而蓝色结点的影响力相对更小，如果选择两个结点，就选择绿色和蓝色结点，这就是基于反向可达集的算法思想，作者花费了大量的篇幅证明了该算法的有效性。

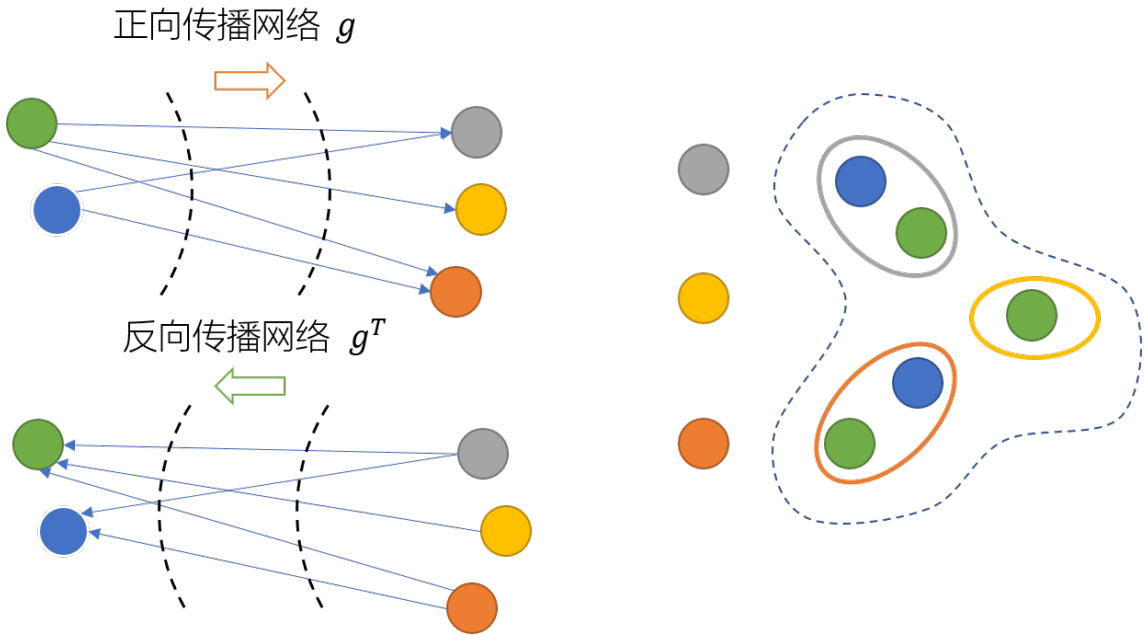

图 4: 反向可达集算法

3.2 基于反向可达集的算法

该算法主要由两个部分组成：基于概率图生成反向可达集合，根据得到的反向可达集进行度结点排序，然后从拥有最大度的结点开始选择，依次选择 k 轮，得到最终的种子结点集合 S_0。

Algorithm 2 Maximum Influence

Input: Precision parameter $\varepsilon \in (0,1)$, directed edge-weighted graph $\mathcal{G}$.

$R \leftarrow 144(m+n)\varepsilon^{-3}\log(n)$
$\mathcal{H} \leftarrow$ BuildHypergraph(R)
return BuildSeedSet($\mathcal{H}$,k)

BuildHypergraph (R):

Initialize $\mathcal{H} = (V, \emptyset)$
repeat
    Choose node u from $\mathcal{G}$ uniformaly at random
    Simulate influence spread, starting from u, in $\mathcal{G}^T$
    Let Z be the set of nodes discovered
    Add Z to the edge set of $\mathcal{H}$
until R steps have been taken in total by the simulation process
return $\mathcal{H}$

BuildSeedSet ($\mathcal{H}$, k):

for $i = 1, \ldots, k$ **do**
    $v_i \leftarrow \text{argmax}_v\{deg_{\mathcal{H}}(v)\}$
    Remove v_i and all incident edges from $\mathcal{H}$
end for
return $\{v_1, \ldots, v_k\}$

4 复现细节

4.1 基于单调次模的贪心算法

关于函数的单调次模性，其数学表达如下所示，其中，单调性指的是，对于两个集合来说，若满足 $S \subseteq T$，则满足 $f(S) \leqslant f(T)$。而次模性指的是对于结点 $u \notin T$ 来说，满足 $f(S \cup \{u\}) - f(S) \geqslant f(T \cup \{u\}) - f(T)$，也就说函数具有次模性。

具有单调次模性的函数能够使用贪心算法求得近似解。

4.2 基于

5 实验结果分析

本部分对实验所得结果进行分析，详细对实验内容进行说明，实验结果进行描述并分析。时间

表 1: 运行时间 (单位：s)

Greedy	704.897	718.554	787.481	772.025	819.826	681.62	812.324	846.325	895.381	807.516
Reverse Set	39.3442	39.3931	39.4749	39.387	39.523	40.2228	40.4572	40.2463	40.2644	40.2714

的均值，对于贪心算法来说，其运行时间均值为 $784.595s$，而基于反向可达集的算法时间复杂度为 $39.85843s$。加速比达到了 $\frac{784.595}{39.85843} \approx 19.68$ 倍，大幅降低了时间复杂度，提升了运行效率。关于准确性的比较：前 50 名种子结点的总影响力值分别为，10 次取平均，其中对于贪心算法来说，326.13327，对于基于反向可达集的算法得到的影响力值为，328.332842。并且我们通过对 10 次试验中得到的各种子影响力值取平均获得前 50 名种子的影响力值。

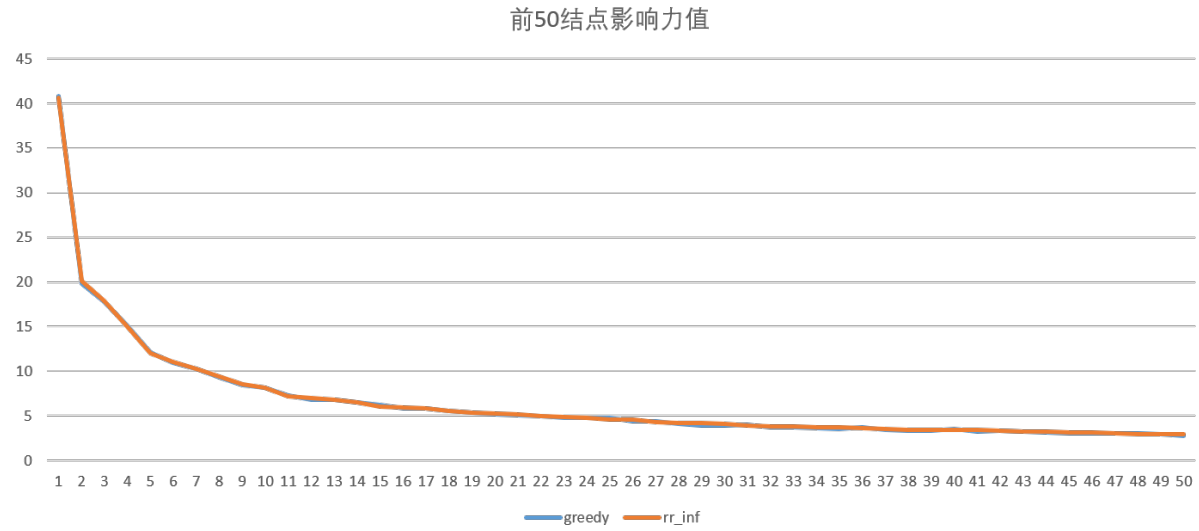

图 5: 前 50 名种子影响力

两条曲线基本重合，精确度不太明显，计算一下各位置的差值，由图中可以看出，影响力差值的范围在 $(-0.25, 0.15)$ 之间波动，波动范围小，进一步说明精确度。

图 6: 前 50 名种子影响力差值

6 总结与展望

作者提出的算法在达到与传统算法（基于单调次模性的贪心算法）相匹配的精确度的前提下，在运行时间方面展现出巨大的优势，但是，作者在文中仅采用了独立级联模型作为研究对象，关于该算法是否适应于其他模型，比如线性阈值模型，以及相应的运算效率，未给出明确的答复。

参考文献

[1] ROGERS E M, SINGHAL A, QUINLAN M M. Diffusion of innovations[G]//An integrated approach to communication theory and research. Routledge, 2014: 432-448.

[2] BAKSHY E, KARRER B, ADAMIC L A. Social influence and the diffusion of user-created content[C]//Proceedings of the 10th ACM conference on Electronic commerce. 2009: 325-334.

[3] BROWN J J, REINGEN P H. Social ties and word-of-mouth referral behavior[J]. Journal of Consumer research, 1987, 14(3): 350-362.

[4] CENTOLA D, MACY M. Complex contagions and the weakness of long ties[J]. American journal of Sociology, 2007, 113(3): 702-734.

[5] 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.

[6] GOMEZ-RODRIGUEZ M, LESKOVEC J, KRAUSE A. Inferring networks of diffusion and influence [J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2012, 5(4): 1-37.

[7] DOMINGOS P, RICHARDSON M. Mining the network value of customers[C]//Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. 2001: 57-66.

[8] KEMPE D, KLEINBERG J, TARDOS É. Maximizing the spread of influence through a social network [C]//Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. 2003: 137-146.

[9] LESKOVEC J, ADAMIC L A, HUBERMAN B A. The dynamics of viral marketing[J]. ACM Transactions on the Web (TWEB), 2007, 1(1): 5-es.

[10] RICHARDSON M, DOMINGOS P. Mining knowledge-sharing sites for viral marketing[C]// Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. 2002: 61-70.