位置敏感的社交网中最小种集选取算法研究

李智慧[1)] 张兆功[1)] 李建中[2)]
[1)](黑龙江大学计算机科学技术学院 哈尔滨 150080)
[2)](哈尔滨工业大学海量数据计算研究中心 哈尔滨 150001)

摘 要 最小种子集合选取问题(*J*-MIN-Seed 问题)的目标是选择一个种子集合 *S*,在影响传播结束后,它不仅需要影响一定数量的用户(如 *J* 个用户),同时 *S* 是最小的集合.虽然该问题得到了广泛的研究,但是现有工作却忽略了一个重要事实,即地理位置信息对于 *J*-MIN-Seed 问题是非常重要的.在许多真实的应用中,例如位置敏感的口碑营销,都有地理位置的需求.因此,该文将地理位置因素融入到 *J*-MIN-Seed 问题中,提出了位置敏感的 *J*-MIN-Seed 问题,并证明了该问题是 NP-hard 问题.该问题的一个挑战是如何高效且有效地计算给定区域的影响范围.为了解决这个挑战,该文对现有的树模型进行扩展,设计出一种高效且有效的近似模型.基于此模型,该文首先提出了朴素的贪心算法 MS-Greedy.MS-Greedy 虽然有近似保证,但其计算量太大.为满足在线查询的需求,该文又提出了另外两种高效的算法 Bound-based 和 Partition-Assembly-based.大量真实数据的实验结果表明:文中算法能有效地解决位置敏感的 *J*-MIN-Seed 问题.

关键词 *J*-MIN-Seed 问题;地理位置;基于树的近似模型;影响最大化;社会网络
中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2017.02305

Research on Algorithms for Selecting Minimum Seed Set on Location-Aware Social Networks

LI Zhi-Hui[1)] ZHANG Zhao-Gong[1)] LI Jian-Zhong[2)]
[1)](*School of Computer Science and Technology, Heilongjiang University, Harbin* 150080)
[2)](*Research Institute of Massive Data Computing, Harbin Institute of Technology, Harbin* 150001)

Abstract The goal of the smallest seed set selection problem(*J*-MIN-Seed problem) is to select a seed set *S*, at the end of the influence spread, it calls for influencing a certain amount of users (for example *J*) while the size of *S* is the smallest. Although the problem has been extensively studied, existing works neglected the fact that the location information can play an important role in *J*-MIN-Seed problem. In many real-world applications, such as location-aware word-of-mouth marketing, have location-aware requirement. Therefore, we integrate the geographical position factor into *J*-MIN-Seed problem, and we propose the location-aware *J*-MIN-Seed problem, and prove that it is NP-hard. One challenge in the problem is how to compute the influence spread of the given region effectively and efficiently. To address this challenge, we extend the existing tree model and design an effective and efficient approximate model. Based on the approximate model, we firstly propose a naive greedy algorithm MS-Greedy. Although MS-Greedy has approximate guarantee, its computation is rather large. In order to meet the needs of online query, we then propose another two effective algorithms Bound-based and Partition-Assembly-based. Experimental results on real data show that our algorithms can solve the location-aware *J*-MIN-Seed problem effectively.

收稿日期:2016-03-05.在线出版日期:2016-11-20.本课题得到国家"九七三"重点基础研究发展规划项目基金(2012CB316200)、国家自然科学基金(61302139)和和黑龙江省自然科学基金(F2017024,F2017025)资助.**李智慧**,女,1990 年生,硕士研究生,主要研究方向为社会网和海量数据管理与计算等.E-mail:15114669830@163.com.**张兆功**(通信作者),男,1963 年生,博士,教授,硕士生导师,主要研究领域为数据挖掘、生物信息学等.E-mail:zhaogong.zhang@qq.com.**李建中**,男,1950 年生,教授,博士生导师,中国计算机学会(CCF)理事,主要研究领域为海量数据管理与计算、无线传感网络等.

Keywords *J*-MIN-Seed problem; geographical position; tree-based approximate model; influence maximization; social networks

1 引 言

近年来,随着互联网和 Web 2.0 技术的迅猛发展,社会网络得到了工业界和学术界越来越多的关注和研究,例如 Twitter、Facebook、微信和微博.社会网络作为沟通现实人类世界的桥梁,俨然已经成为信息传播、交互沟通和知识共享的重要媒介和平台.社会网络分析领域的一个关键性问题是影响传播问题,其传播手段是病毒式营销.病毒式营销是一种广告策略,它通过个体之间的社会关系,利用"口碑效应"来推销产品或思想.现有的广告策略是直接选取大量的用户并对其进行产品或思想的宣传[1],而病毒式营销的思想是选取一定量的初始用户(给这些用户提供奖励或免费的产品),利用这些用户的社会关系,如朋友、家人和合作者,在个体之间进行产品或思想的传播.传播过程中,用户会从对产品的不知道状态转换成对产品的知道状态,这种状态发生转换的用户称之为被影响的用户.影响传播结束后,所有被影响用户的总数称为初始用户的影响范围.最近调查研究表明,人们更加倾向于从朋友、家人或亲属获得的信息,而不是像电视一类的一般广告媒介[2].因此,病毒式营销是最为有效的营销手段之一.

社会网中病毒式营销的过程可以描述如下:首先,广告商会选取一个初始用户集合,为了激励初始用户在社会网中传播其产品或思想,广告商会给予每个初始用户免费的产品或奖励.我们将这些初始用户称之为种子节点.传播被初始化后,产品或思想就会通过社会网中用户的关系进行传播.很多模型用来描述上述的传播过程,现在被广泛使用的两个模型是独立级联模型 IC(Independent Cascade Model)和线性阈值模型 LT(Linear Threshold Model).

考虑病毒式营销中这样一种场景:某公司希望在社交网中通过病毒式营销手段来推销其新产品,由于费用有限,它希望用于病毒式营销的初始种子用户最少(这样提供给种子用户的奖励最少),同时传播结束后最终能影响一定数量的用户(如 J).该问题就是最小种集选取问题(J-MIN-Seed).J-MIN-Seed 问题的目标是至少影响社会网中一定量的用户(如 J),同时用于病毒式营销的种子节点个数最少.

我们使用图 1 来具体描述 J-MIN-Seed 问题.图 1 将 4 个朋友之间的关系抽象成一个有向图.其中节点分别代表 4 个朋友成员,即 $Lily$、$Lucy$、Tom 和 $John$,边代表成员之间的关系,如($Lily$,$John$)表示 $Lily$ 是 $John$ 的朋友.边上的概率表示在病毒式营销中一个节点对另一个节点的影响力,如 $w(Lily,John)=0.8$,表示 $Lily$ 对 $John$ 的影响力是 0.8.假设在图 1 中,我们的目标是找到一个最小种子集合,它满足至少影响该社交网中 3 个朋友成员的条件.由于节点 $Lily$ 对节点 $John$ 的影响路径有两条,分别为〈$Lily$,$John$〉和〈$Lily$,$Lucy$,$John$〉,所以在 IC 模型下,集合{$Lily$}对 $John$ 的影响为 $1-(1-0.8)\cdot(1-0.6\cdot0.7)=0.884$.以此类推,可以计算出{$Lily$}的最终影响范围为 3.57.通过计算发现,没有比集合{$Lily$}更小的集合满足至少影响 3 个朋友成员的条件,所以目标种集为{$Lily$}.

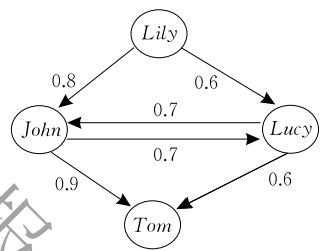


图 1 J-MIN-Seed 问题的社会网

J-MIN-Seed 问题的实际意义是利用最小的花费同时达到明确的病毒式营销目标.然而,现在的研究忽略了一个重要的事实,即对于 J-MIN-Seed 问题,位置信息是十分重要的.许多真实的实际应用,如位置敏感的口碑营销,对 J-MIN-Seed 问题有位置限制的要求.例如,社会网系统(如 Twitter)通过定位空间区域(如哈尔滨市南岗区)中的一定量的潜在客户为新企业(如餐厅)提供市场营销的服务.具体描述为,企业希望特定空间区域中至少一定量的用户(如 J)被影响,同时用于病毒式营销的初始费用(种子节点个数)最少.当公司的费用有限时,选取的种子节点是至关重要的,它们自身可以不在公司指定的区域中,但它们在影响传播中能够影响到指定区域中的朋友、朋友的朋友等等.通过这种病毒式营销,公司可以花费最少的资源并且保证公司附近至少一定量的用户知道该公司.

上面的问题我们将其称为位置敏感的最小种子集合问题(LA-J-MIN-Seed 问题)，它可以描述如下：给定位置敏感的社会网络，其中每个用户都包含经度和纬度的地理位置，包含特定位置区域 R 和整数 J 的查询 Q，LA-J-MIN-Seed 问题的目标是选择初始用户集合，在影响传播结束后，该集合至少能影响特定位置区域中 J 个用户，且该集合是满足上述条件的最小集合.

尽管 LA-J-MIN-Seed 问题在病毒式营销领域中非常重要，但是现在的研究尚未考虑该问题. 首先，大部分涉及病毒式营销的研究都致力于影响最大化问题，该问题旨在选择 k 个种子节点，在影响传播结束后，这 k 个种子节点带来的影响范围最大. 很明显，本文问题与影响最大化问题的给定条件和追求目标是不同的. 其次，虽然有一些研究的研究问题与本文问题相似，但是它们并没有考虑到位置因素，不能直接用来解决本文的问题.

LA-J-MIN-Seed 问题存在三方面的挑战：第 1 点是获得用户的位置信息. 非常幸运的是，现实世界中存在很多位置敏感的社会网络，如 Gowalla(http://snap.stanford.edu/data/loc-gowalla.html)，它们自身内部含有每个用户的位置信息. 另外，也有一些研究[3-4]致力于从社会网中获得用户的位置信息. 通过对这些研究中涉及的技术进行学习和实现，可以获得社交网中用户的位置信息，进而为 LA-J-MIN-Seed 问题的研究提供了支持. 第 2 点是如何满足高性能需求. 例如在上面的例子中，大量的企业希望在进行营销服务的同时，系统能够尽快地给出解决方案，因此这就需要设计高效的算法满足有效且高效地支持在线 LA-J-MIN-Seed 问题查询. 在前人工作的基础上可以提出 LA-J-MIN-Seed 问题的一种朴素的解决方法，即利用已有影响最大化问题算法和已有 J-MIN-Seed 问题算法进行求解. 该方法虽然能有效地解决 LA-J-MIN-Seed 问题，但不能满足其高性能的需求，因为影响最大化问题和 J-MIN-Seed 问题本身就是 NP-hard 问题，它们不得不枚举大量的节点来选取种子节点，并且不能剪枝掉影响力较小的节点. 我们借鉴文献[5]中的思想设计出高效的算法，在不损失太多影响精度的基础上，提高算法的高效性. 第 3 点是有效且高效地计算特定区域上的影响范围. 由于计算特定区域上的影响范围是 ♯P-hard 问题，即没有多项式时间的算法能够准确求解出特定区域上的影响范围，所以本文在基于树的近似模型的基础上进行了相应的扩展，使得扩展后的模型能够很好地近似求解特定区域上的影响范围.

本文的主要贡献如下：

(1) 据我们所知，本文首次将地理位置因素引入到 J-MIN-Seed 问题，将其定义为 LA-J-MIN-Seed 问题，并形式化了该问题.

(2) 本文在基于树的近似模型的基础上进行了相应的扩展，使得扩展后的模型能很好地近似求解查询区域上的影响范围.

(3) 本文证明了 LA-J-MIN-Seed 问题是 NP-hard 问题，并且提出朴素的贪心算法框架 MS-Greedy，用于解决 LA-J-MIN-Seed 问题，证明了它的绝对误差值和比例误差值.

(4) 本文提出两种高效的贪心算法 Bound-based 算法和 Partition-Assembly-based 算法，它们可以满足高效性，同时又不会损失太多精度.

(5) 大量真实数据集上的实验结果证明了本文算法的有效性和高效性.

本文第 2 节介绍相关工作；第 3 节进行问题定义；第 4 节描述贪心算法框架 MS-Greedy，并证明其绝对误差值和比例误差值；第 5 节描述 Bound-based 算法；第 6 节描述 Partition-Assembly-based 算法；第 7 节阐述实验结果；第 8 节进行总结.

2　相关工作

现在大部分关于病毒式营销的研究都集中于影响最大化问题，其目标是寻找社会网中最终影响范围最大的 k 个节点. 当前，针对影响最大化问题的研究主要分为贪心算法和启发式算法两大类. 贪心算法的研究核心思想是爬山思想，即每次迭代选择的节点都是能带来最大影响值的节点，通过局部的最优解来近似全局的最优解. 贪心算法的优点是所求种集的精度相对较高，与理论最优解有 $1-1/e-\varepsilon$ 的近似比. 但是其存在严重的效率问题，算法的复杂度较高，执行时间较长，很难扩展到大规模的社会网络中. 针对此缺点，已经存在大量的研究致力于相应的改善. 启发式算法并不是基于局部最优解来近似全局最优解的思想，它的思想是利用设定的启发策略来选择最有影响力的节点，这样就避免了复杂地计算节点影响的准确值，从而大大地缩短了运行时间，提高了效率，但是不可避免地导致算法的精度较差，难以和贪心算法相比较.

虽然影响最大化问题的初始条件和追求的目标

和本文的 LA-J-MIN-Seed 问题完全不同，但是我们可以对影响最大化问题的算法进行一些改变，用其形成一种朴素的方法来近似地求解 LA-J-MIN-Seed 问题. 将影响最大化问题中每个节点的影响范围限制在给定的特定区域内，然后多次调用影响最大化问题的算法，直到满足 LA-J-MIN-Seed 问题的目标，算法停止. 令 k 表示种子节点个数. 具体描述为，最初设置 $k=1$，每次迭代，k 增加 1. 每次迭代都调用一次改变后的影响最大化问题的算法，计算 k 个种子节点给查询区域带来的影响 I. 如果 $I\geq J$，则迭代停止，返回 k；否则 k 增加 1，进行下一轮迭代. 虽然该方法可以解决本文问题，但是相当耗时，存在严重的效率问题. 由于影响最大化问题本身是 NP-hard，上述方法又多次调用影响最大化问题的算法，导致该方法的计算代价相当大. 因此，我们应该设计更高效的算法来解决 LA-J-MIN-Seed 问题.

关于病毒式营销的另一些研究所致力于的问题与最小种集问题十分相似，它们将其称之为目标集合选择 TSS(Target Set Selection)问题. TSS 问题最初是由文献[6]提出的，其目的也是选择种集，但是该种集主要影响整个网络中所有的节点，同时是满足上述条件的最小集合. 在 TSS 问题中，底层的传播模型是确定线性阈值模型 DLT(Deterministic Linear Threshold)，它与线性阈值模型的唯一不同之处在于，社会网中所有节点的阈值是固定的.

随后，一些学者也对 TSS 问题进行了研究[6-8]. 在文献[6]中，Chen 推导出 TSS 问题的非近似结果. 具体为，对任意固定的 $\varepsilon>0$，TSS 问题的近似比不小于 $O(2^{\log(1-\varepsilon)n})$，除非 $NP\subseteq DTIME(n^{ploylog(n)})$. 在文献[7-8]中，Ben-Zwi 等人通过考虑图的树宽参数探索了 TSS 问题的难易程度. 他们设计了一个运行时间为 $n^{O(w)}$ 的准确算法用于解决 TSS 问题，其中 w 是底层社会网络的树宽. 随后，文献[9-10]提出一些剪枝策略用于解决 TSS 问题. 最近，Long 和 Wong[11] 提出了 J-MIN-Seed 问题，其底层传播模型是 IC 模型和 LT 模型.

与现有的研究工作不同，本文研究的问题是 LA-J-MIN-Seed 问题，并且旨在实现实时的在线查询.

3 问题定义

3.1 预备知识

社会网是一个由社会个体成员以及个体成员之间的社会关系所组成的一种复杂的网络结构. 在有关社会网的研究中，通常将社会网建模为图 $G=(V,E,W)$，其中图 G 中节点集合 $V=\{v_1,v_2,\cdots,v_n\}$ 中的节点对应社会网中个体成员；图 G 中边的集合 $E\subseteq V\times V$ 对应社会网中个体成员之间的社会关系；集合 $W=\{w_1,w_2,\cdots,w_n\}$ 对应社会网中个体的权重. 我们将位置敏感的社会网络建模为有向图 $G=(V,E)$，其中 V 中的节点表示网络中的用户，E 代表用户之间的关系. V 中的每个节点都有一个地理位置 (x,y)，其中 x 表示经度，y 表示纬度.

影响传播模型定义了社会网络中影响力传播的方式和机制，它是研究本文 LA-J-MIN-Seed 问题的基础. 目前学术界对影响传播模型已经进行了深入的研究，其中被广为研究的两个影响传播模型分别是 IC 模型和 LT 模型. 这两个模型分别从不同的角度对影响传播过程进行了阐述，IC 模型是从概率角度，LT 模型是从阈值角度.

给定初始活跃节点集合 S 后，IC 模型的影响传播过程如下：在时间 0，仅仅活跃集合 S 中的节点为活跃状态，其余节点皆为非活跃状态；如果节点 v 在时间 t 从非活跃状态变为活跃状态，则节点 v 试图以一定概率去激活其处于非活跃状态的后继节点. 节点 v 成功激活其后继节点 w 的概率为 $p(v,w)$. 如果节点 v 成功激活节点 w，则节点 w 在时间 $t+1$ 变为活跃状态. 如果节点 v 没有成功激活节点 w，则节点 w 在时间 $t+1$ 仍为非活跃状态. 该过程一直持续到没有节点被激活为止. 该过程结束后，将集合 S 激活的节点集合称为集合 S 的影响范围. 需要注意的是，上述过程中节点 v 对于其所有的邻居节点均有且仅有一次机会去尝试激活，而且节点 w 一旦被激活则不能从活跃状态恢复到非活跃状态.

LT 模型是基于阈值的，并且多个处于活跃状态的节点尝试影响同一个后继节点的行为不是独立的，影响成功与否取决于所有活跃节点对后继节点影响权重的和是否超过后继节点的阈值. 在 LT 模型中，定义 $N(v)$ 表示节点 v 的前驱节点集合，节点 v 受其活跃的前驱结点 w 影响的程度由权重 $p(w,v)$ 衡量. 每个节点都有一个阈值 θ_v，$\theta_v\in[0,1]$. 该阈值表明了节点从非活跃状态转变到活跃状态的难易程度，阈值越大表明其越不容易被影响，相反则越容易被影响. 同 IC 模型类似，在时间 0，只有活跃集合 S 中的节点处于活跃状态，其余节点皆处于非活跃状态；如果节点 v 在时间 t 变为活跃状态，则节点 v 将其影响权重 $p(v,u)$ 加到其处于非活跃状态的后继

节点 u 上. 如果 u 的所有处于活跃状态的前驱结点对 u 的影响权重累加和超过 u 的阈值 θ_u，则节点 u 在时间 $t+1$ 变成活跃节点. 该过程一直重复进行下去，直到没有节点被激活为止. LT 模型体现了影响累计的特性.

独立级联模型 IC 是最基本的影响传播模型，它能很好地反映许多真实情况下的影响传播方式. 本文研究的 LA-J-MIN- Seed 问题的底层传播模型采用 IC 模型.

3.2　问题定义

在社会网影响传播模型及其相关算法的研究领域中，通常采用符号 $\sigma(S)$ 表示集合 S 的影响值. 具体为将集合 S 作为初始活跃节点集合，在给定的影响传播模型下，经过上述的影响传播过程，最终网络中被激活的结点个数.

定义 1(边际收益).　在影响传播过程中，影响值函数 $\sigma(\cdot)$ 的边际收益是指在当前活跃节点集合 S 的基础上，额外增加一个节点 v 作为初始活跃节点，所带来的最终影响值的增加量，即

$$\sigma_v(S)=\sigma(S\bigcup\{v\})-\sigma(S) \tag{1}$$

仍以上面的图 1 为例，假设当前活跃种集 S 为 $\{Lily\}$，将节点 $John$ 加入到 S 后，节点 $John$ 带给函数 $\sigma(\cdot)$ 的边际收益为 $\sigma(S\bigcup\{John\})-\sigma(S)=3.83-3.57=0.26$.

定义 2(单调性质).　$f(\cdot)$ 是将有限集合 S 映射为非负实数的一个函数. 当 $f(\cdot)$ 的自变量在其定义区间内增大时，函数值 $f(\cdot)$ 也随着增大，则称该函数为在该区间上具有单调性质. 形式化描述为 $S\subseteq T, f(S)\leq f(T)$.

定义 3(子模性质).　$f(\cdot)$ 是将有限集合 S 映射为非负实数的一个函数. 如果函数 $f(\cdot)$ 满足收益递减性质，即增加任意一个元素 v 到集合 S 所带来的边际收益不低于增加元素 v 到 S 的超集 $T(S\subseteq T)$ 所带来的边际收益，则称函数 $f(\cdot)$ 具有子模性质. 具体为 $S\subseteq T, v\notin T$，函数 $f(\cdot)$ 具有子模性质，则有以下不等式成立：

$$f(S\bigcup\{v\})-f(S)\geq f(T\bigcup\{v\})-f(T) \tag{2}$$

文献[12]已经证明函数 $\sigma(\cdot)$ 在 IC 模型和 LT 模型下都具有子模性质和单调性质.

例 1.　为了更好地理解单调性质和子模性质，我们仍以图 1 为例. 假设种集 T 为 $\{Lily\}$，设定它的子集 S 为 $\varnothing$，将节点 $John$ 分别加入到种集 T 和种集 S 中. 在 IC 模型下，很容易计算出 $\sigma(\varnothing)=0$，$\sigma(\{Lily\})=3.57$，$\sigma(\{John\})=2.64$，$\sigma(\{Lily,John\})=3.83$. 符合函数 $\sigma(\cdot)$ 具有的子模性质，即 T 的边际收益 $\sigma_{John}(T)$ 小于 S 的边际收益 $\sigma_{John}(S)$，并且满足 $\sigma(\{Lily\})\leq\sigma(\{Lily,John\})$.

给定位置区域 R，将 V_R 定义为位置区域 R 中的节点集合. 图 2 中虚线框描述了一个位置区域 R，则 $V_R=\{2,4,5,8,9,10,11,12,13,14\}$. 给定种集 S，在影响传播过程结束后，V_R 中被集合 S 激活的节点总数称为 S 在区域 R 的影响范围(后面为了方便，简称为影响范围)，将其定义为 $\sigma(S,V_R)$.

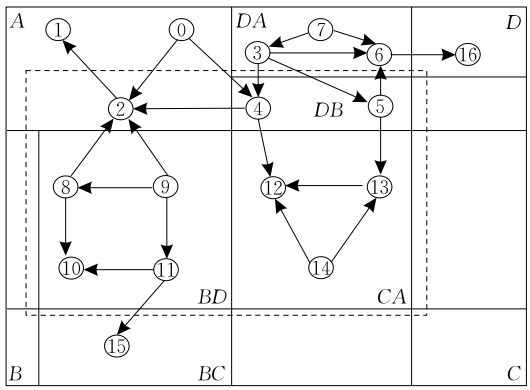


图 2　运行例图

定理 1.　函数 $\sigma(S,V_R)$ 在 IC 模型和 LT 模型下都具有子模性质.

证明.　证明方法与文献[12]证明的思想类似. 在给定初始种集 S 和位置区域 R 前提条件下，当将一个新的节点 v 加入到种集 S 中时，节点 v 给区域 R 所带来的影响可能与集合 S 给区域 R 所带来的影响有重叠. 种集 S 越大，重叠的影响可能越多，因此，种集 S 的边际收益会小于 S 的任意子集的边际收益，所以函数 $\sigma(S,V_R)$ 在 IC 模型和 LT 模型下都具有子模性质.　证毕.

定理 2.　函数 $\sigma(S,V_R)$ 在 IC 模型和 LT 模型下都具有单调性质.

证明.　假设当前种集为 S，在 IC 模型和 LT 模型下，S 在给定区域 R 的影响范围为 $\sigma(S,V_R)$. 将任意节点 u 加入到当前种集 S 中，由于 u 在给定区域 R 的影响范围 $\sigma(\{u\},V_R)\geq 0$，所以 $\sigma(S\bigcup\{u\},V_R)\geq\sigma(S,V_R)$. 由单调性的定义可知，函数 $\sigma(S,V_R)$ 在 IC 模型和 LT 模型下都具有单调性质.　证毕.

定理 3.　计算准确的 $\sigma(S,V_R)$ 是 #P-hard 问题.

证明.　设 $V_R=V$，则计算准确的 $\sigma(S,V_R)$ 变成计算准确的 $\sigma(S)$. 由定理 1 和定理 2 可知，$\sigma(S,V_R)$ 在 IC 模型和 LT 模型下都具有子模性质和单调性质，并且计算准确的 $\sigma(S)$ 是 #P-hard[13]，所以计算准确的 $\sigma(S,V_R)$ 也是 #P-hard.　证毕.

文献[13]证明了计算给定种集 S 的 $\sigma(S)$ 是 #P-hard 问题,因此准确地计算出 $\sigma(S)$ 的代价非常昂贵.在病毒式营销领域中,通常采用蒙特卡罗模拟(Monte-Carlo simulation)方法估计 $\sigma(S)$.蒙特卡罗模拟的主要思想如下:假设初始种集为集合 S,以 S 为活跃节点集合进行多次模拟影响传播过程.在每次模拟过程中,都需要计算,影响传播过程结束后,最终被影响的结点个数.然后,取这些模拟过程结果的平均值作为 $\sigma(S)$ 的估计值.在一个包含 10k 个节点和 50k 条边的社会网上进行蒙特卡罗模拟,进行 10 000 次模拟影响传播过程获得的准确性已经足够好,不需要进行 20 000 次的模拟[12].虽然蒙特卡罗模拟方法估计 $\sigma(S)$ 的准确度较高,但是每次模拟过程需要遍历一次图结构,导致该方法代价昂贵,并且难以扩展到大规模的社会网络中.

定理 3 表明任何算法都不能在多项式时间内计算出 $\sigma(S,V_R)$ 的准确值.为了避免蒙特卡罗模拟方法的昂贵代价,本文将基于树的近似模型[13]进行了扩展,扩展后的模型能够高效地计算影响范围 $\sigma(S,V_R)$,同时也不会损失太多的影响精度.本文的扩展方法也可以应用到其他的近似模型中.

IC 模型中,当活跃节点 u 激活其非活跃的后继节点 v 后,节点 v 也会去尝试激活其非活跃的后继节点 w.上述影响传播过程中,虽然节点 u 没有到达节点 w 的直接边,但是节点 u 也会以一定的概率去尝试激活节点 w.受上述启发,我们引入影响路径的概念.

定义 4(影响路径). $G=(V,E)$ 是一个边带权值的有向图,边 (u,v) 上的概率为 $p(u,v)$.定义 $p=\langle u=w_1,w_2,\cdots,w_n=v\rangle$ 表示一条从 u 到 v 的路径,其中 w_i 表示 u 到 v 的路径所经过的节点.

显然,在 IC 模型下,u 沿着路径 p 影响 v 的概率(或者 v 被 u 影响的概率)为路径上每条边概率的乘积,将其定义为 $P(p)=\prod_{i=1}^{n-1}p(w_i,w_{i+1})$.$u$ 到 v 有很多不同的路径,沿着不同的路径 u 对 v 的激活概率不同.IC 模型中,由于每个节点有且仅有一次机会去尝试激活其处于非活跃状态的后继节点,所以 u 试图激活 v 的最好选择是沿着 u 到 v 的概率最大的路径,因此将 u 对 v 的影响定义为 $P(u\to v)=\max\{P(p)\mid p$ 是 u 到 v 的一条路径$\}$.对于图 G 中的边 (u,v),如果将边的传播概率 $p(u,v)$ 转换成边 (u,v) 的距离权重 $-\log p(u,v)$,则图 G 中 u 到 v 的最大影响概率路径对应于距离权重图中的 u 到 v 的最短路径,后面介绍的影响树直接对应于距离权重图中最短路径组成的树,并且它们可以利用有效的算法来进行求解,如迪杰斯特拉算法.

给定种集 S,将集合 S 对 v 的影响定义为 $P(S,v)$.注意 $P(S,v)\neq\sum_{u\in S}P(u\to v)$,因为集合 S 中的不同节点在影响 v 时可能有相关性.解决该问题的一个较好方法是采用基于树的模型来计算 $P(S,v)$.具体描述为,为集合 S 中的每个节点 u,在图 G 中找到一条 u 到 v 影响概率最大的路径,然后对这些最大影响概率路径进行组装,形成一个影响树 $Tree(S,v)$,其以 v 为树根,u 为叶子.如果 $v\in S$,则 $P(S,v)=1$;否则,v 一定是被影响树 $Tree(S,v)$ 中 v 的孩子所影响.由于 v 的孩子 c 也被集合 S 影响,所以 v 被 c 影响的概率为 $P(S,c)\cdot p(c,v)$.我们可以通过组合所有孩子 c 对 v 的影响,计算出 $P(S,v)=1-\prod_{c\in child(v)}(1-P(S,c)\cdot p(c,v))$,其中 $child(v)$ 为 v 的影响树 $Tree(S,v)$ 中 v 的孩子集合.

综上所述,$P(S,v)$ 的计算公式如下:

$$P(S,v)=\begin{cases}1, & v\in S\\ 1-\prod_{c\in child(v)}(1-P(S,c)\cdot p(c,v)), & v\notin S\end{cases}\tag{3}$$

例 2. 仍以图 2 为例,假设我们已经选择节点 4 和节点 8 作为种子节点,下面我们详细描述怎样计算它们对节点 1 的共同影响 $P(\{4,8\},1)$.本文例子中图的边概率采用权重级联模型,即边 (u,v) 的概率 $p(u,v)=1/|N_{in}(v)|$,其中 $N_{in}(v)$ 是节点 v 的所有前驱节点集合.如节点 2 的前驱节点集合为 $\{0,4,8,9\}$,则 $p(9,2)=1/4=0.25$.由于节点 4 和 8 到节点 1 的最大影响概率路径都经过节点 2,所以 $P(\{4,8\},1)=1-(1-P(\{4,8\},2)\cdot p(2,1))$.由于 $P(\{4,8\},2)=1-(1-p(4,2))\cdot(1-p(8,2))=0.4375$,$p(2,1)=1$,所以 $P(\{4,8\},1)=0.4375$.

定义 5. 在基于树的模型中,可以用如下公式来近似计算 $\sigma(S,V_R)$,

$$\sigma(S,V_R)=\sum_{v\in V_R}P(S,v)\tag{4}$$

例 3. 仍以图 2 为例,假设我们已经选择节点 7 和节点 9 作为种子节点,图中虚线框表示 V_R,则 $V_R=\{2,4,5,8,9,10,11,12,13,14\}$.利用式(3)可以计算出 $P(\{7,9\},v)$,$v\in V_R$,然后利用式(4)可以计算出 $P(\{7,9\},V_R)=6.008$.

定义 6(LA-J-MIN-Seed 问题). 给定位置敏感的社会网络 G 和查询 $Q=(R,\mathrm{J})$,其中 R 是位置区域,J 是正整数. LA-J-MIN-Seed 问题旨在选择最小的节点集合 S:$\sigma(S,V_R)\geq J$. 其中 S 称为种子集合,简称种集,S 中的每个节点称为种子节点. 形式化为

$$S=\{V|V\text{ 为种子集合},|V|=n,\sigma(V,V_R)\geq J\},$$

其中

$$n=\min_V\{|V||V\text{ 为种子集合且}\sigma(V,V_R)\geq J\}\quad(5)$$

例 4. 图 2 描述了包含 17 个节点的位置敏感的社会网络. 给定查询 $Q=(R,J)$,其中图中的虚线框给出了查询区域 R,$J=5$,则 LA-J-MIN-Seed 问题的目标种集 S 为 $\{7,9\}$,其中,$\sigma(\{7,9\},V_R)=6.008$,$\sigma(\{9\},V_R)=3.75$,$\sigma(\{7\},V_R)=2.29$. 注意节点 7 并不在 R 中,所以我们不能仅使用 R 中的节点来选择种子节点.

定理 4. LA-J-MIN-Seed 问题在IC模型和LT模型下是 NP-hard.

证明. 设 $V_R=V$,则 LA-J-MIN-Seed 问题变成 J-MIN-Seed 问题,由于在 IC 模型和 LT 模型下,J-MIN-Seed 问题是 NP-hard[11],所以 LA-J-MIN-Seed问题在IC模型和LT模型下也是 NP-hard. 证毕.

由于 LA-J-MIN-Seed 问题是 NP-hard 的,所以没有多项式时间的算法能够准确地解决该问题. 我们需要设计近似算法来解决该问题.

4　MS-Greedy 算法

第 3 节的定理 4 已经证明了 LA-J-MIN-Seed 问题是 NP-hard 问题,所以没有高效而准确的算法能够求解该问题. 在第 2 节中,我们描述了一种朴素的方法求解 LA-J-MIN-Seed 问题,即调用现存的用于求解影响最大化问题的算法. 虽然该朴素方法能够求解本文问题,但其相当耗时. 主要原因在于需要迭代地调用已有方法很多次,并且每次迭代过程中,对已有方法的本次调用与上次调用是相互独立的,从而导致大量的计算都是重复的,造成了时间的浪费. 受上述启发,我们提出了一种贪心算法 MS-Greedy 用于解决 LA-J-MIN-Seed 问题. 该算法避免了上述朴素方法的大量重复计算,该算法中每轮迭代是在上轮迭代结果基础上进行计算选择种子节点.

算法 MS-Greedy 的主要思想是:初始种集 S 为空集,选择一个非种子节点 u,u 满足将其插入集合 S 中带来的边际收益最大,并将 u 加入到 S 中. 重复上述过程,直到 S 至少影响了 J 个节点.

算法 1. MS-Greedy.

输入:$G(V,E)$:位置敏感的社会网络;$Q=(R,J)$:查询,其中 R 是查询区域,J 是正整数

输出:S:种子集合

1. $S\leftarrow\varnothing$
2. WHILE ($\sigma(S,V_R)<J$)
3. $\quad u\leftarrow\arg\max_{x\in V-S}(\sigma(S\cup\{x\},V_R)-\sigma(S,V_R))$
4. $\quad S\leftarrow S\cup\{u\}$
5. RETURN S

算法 1 是算法 MS-Greedy 的伪代码. 算法 1 的具体描述为:第 1 行初始种集 S 为空集. 第 2～4 行迭代地选取种子节点. 在选取下一个种子节点时,第 2 行先判断当前种集 S 对给定区域 R 的影响是否不小于 J,若不小于 J 则算法结束,当前种集 S 就是目标种集,第 5 行返回目标种集 S;否则进行下一次迭代. 每次迭代的过程都是选择给当前种集 S 带来最大边际收益的节点(第 3 行),并将其加入到当前种集 S 中(第 4 行).

例 5. 仍然使用例 4 中的查询. 初始 S 为空集,由于 S 对 R 的影响 $\sigma(\varnothing,V_R)=0$ 小于 J,所以进行第 1 轮迭代,选择带来边际收益最大的节点 9,并将节点 9 加入到 S 中,S 变为 $\{9\}$;此时 S 的影响 $\sigma(\{9\},V_R)=3.75<5$,所以进行第 2 轮迭代,选择带来边际收益最大的节点 7,并将节点 7 加入到 S 中,S 变为 $\{7,9\}$;此时 S 的影响 $\sigma(\{7,9\},V_R)=6.008>5$,结束迭代,返回当前种集 S,算法结束. 算法 MS-Greedy 求得的目标种集为 $\{7,9\}$.

MS-Greedy 算法与影响最大化问题算法类似,但是算法的终止条件不同,并且它们有不同的理论结果. 前者算法的终止条件是目标种集 S 对指定区域 R 的影响范围 $\sigma(S,V_R)$ 不小于 J,后者算法的终止条件是目标种集 S 的大小为算法指定的正整数 k. 前者算法求得的目标种集保证了最终被影响的节点个数,后者算法求得的目标种集保证了初始种集的大小.

下面我们证明算法 1 中 MS-Greedy 算法求得的种子集合 S 的绝对误差值(additive error guarantee)和比例误差值(multiplicative error guarantee).

定理 5. 函数 $P(S,v)$具有单调性质.

证明. 假设当前种集为S,在v的影响树$Tree(S,v)$中,令 $P(S,v)=t$. 将任意节点 u 加入到当前种集 S 中,有 $P(S\cup\{u\},v)\geq P(S,v)$成立,因为 u 对 v 的最大影响概率 $p(u\rightarrow v)\geq 0$. 由单调性的定义可知,函数 $P(S,v)$具有单调性质. 证毕.

定理 6. 函数 $P(S,v)$具有子模性质.

证明. 给定初始种集 S 和节点 v,当将一个新的节点 u 加入到种集 S 中时,u 对 v 的影响可能与集合 S 对 v 的影响有重叠. 种集 S 越大,重叠的影响可能越多,因此,种集 S 的边际收益会不大于 S 的任意子集的边际收益,所以函数 $P(S,v)$具有子模性质. 证毕.

定理 7(additive error guarantee). 假设 h 是 MS-Greedy 算法返回种集 S 的大小,t 是 LA-J-MIN-Seed 问题的最优种集的大小. MS-Greedy 算法能够保证 $h-t\leq 1/e\cdot J+1$,其中 e 是自然对数.

证明. 假设 $V_R=V$,则 LA-J-MIN-Seed 问题变成 J-MIN-Seed 问题. 定理 5 和定理 6 表明,函数 $P(S,v)$具有单调性和子模性质,并且 J-MIN-Seed 问题的贪心框架能够保证 $h-t\leq 1/e\cdot J+1$[11],所以定理成立. 证毕.

为了方便证明 MS-Greedy 算法的比例误差值,我们给出以下相关的概念. 假设 MS-Greedy 算法在 h 轮迭代之后结束,S_i 表示第 i 轮迭代之后 MS-Greedy 算法求出的种集,其中 $i=1,2,\cdots,h$. S_0 表示 MS-Greedy 算法初始的种集,即为空集. 对任意的 $i=1,2,\cdots,h-1$,都有 $\sigma(S_i,V_R)<J$ 成立,并且 $\sigma(S_h,V_R)\geq J$.

定理 8(multiplicative error guarantee). 假设 $\sigma'(S,V_R)=\min\{\sigma(S,V_R),J\}$,$h$ 是 MS-Greedy 算法返回的种集 S 的大小,t 是 LA-J-MIN-Seed 问题的最优种集的大小. MS-Greedy 算法的比例误差值为 $h/t\leq 1+\min\{k_1,k_2,k_3\}$,其中

$k_1=\ln J/(J-\sigma'(S_{h-1},V_R))$,

$k_2=\ln\sigma'(S_1,V_R)/(\sigma'(S_h,V_R)-\sigma'(S_{h-1},V_R))$,

$k_3=\ln(\max\{\sigma'(\{x\},V_R))/(\sigma'(S_i\cup\{x\},V_R)-\sigma'(S_i,V_R))\mid x\in V,0\leq i\leq h,\sigma'(S_i\cup\{x\},V_R)-\sigma'(S_i,V_R))>0\}$.

证明. 假设 $\sigma'(S,V_R)=\min\{\sigma(S,V_R),J\}$. 定理证明过程主要分为四部分:第一部分,基于函数 $\sigma'(S,V_R)$定义一个新问题 P',其定义为 $\arg\min\{|S||\sigma'(S,V_R)=\sigma'(V,V_R),S\subseteq V\}$. 第二部分,将算法 1 中的函数 $\sigma(S,V_R)$替换成函数 $\sigma'(S,V_R)$,替换后的算法用于求解 P'问题. 文献[14]研究的是子模集合覆盖问题,其定义为 $Z=\arg\min\{\sum_{j\in S}f_j:z(S)=z(N),S\subseteq N\}$,其中 z 是定义在有限集合 N 的子集上的函数,且其具有单调性质和子模性质,f 也是定义在集合 N 上的函数. 令集合 N 等于集合 V,函数 $z(\cdot)=\sigma'(\cdot,V_R)$,并且 $\forall x\in V$,都定义 $f_x=1$. 定理 1 和定理 2 表明函数 $\sigma(S,V_R)$具有子模性质和单调性质,很容易证明函数 $\sigma'(S,V_R)$也具有子模性质和单调性质. 所以,问题 P'就转化成子模集合覆盖问题. 由于求解问题 P'的算法思想和求解子模集合覆盖问题的算法思想相同,根据文献[14]中比例误差值的定理,可以推导出求解问题 P'的算法能够保证比例误差值为 $1+\min\{k_1,k_2,k_3\}$,其中

$k_1=\ln J/(J-\sigma'(S_{h-1},V_R))$,

$k_2=\ln\sigma'(S_1,V_R)/(\sigma'(S_h,V_R)-\sigma'(S_{h-1},V_R))$,

$k_3=\ln(\max\{\sigma'(\{x\},V_R))/(\sigma'(S_i\cup\{x\},V_R)-\sigma'(S_i,V_R))\mid x\in V,0\leq i\leq h,\sigma'(S_i\cup\{x\},V_R)-\sigma'(S_i,V_R))>0\}$.

第三部分证明问题 P'的解与本文 LA-J-MIN-Seed 问题的解是一一对应的. 第四部分证明算法 1 求解本文问题的比例误差值与第二部分推导出的 P'问题比例误差值相同. 本定理的证明方法与 J-MIN-Seed 问题比例误差值的证明方法相同,具体详细见文献[13]. 证毕.

定理 8 给出了 MS-Greedy 算法比例误差值的上界,通过这个上界的表达式,可以发现,MS-Greedy 算法的比例误差值取决于算法的执行过程. 理论上,贪心算法 MS-Greedy 的比例误差值不会超过 4[14].

虽然 MS-Greedy 算法能够解决本文问题,但是在每次贪心选择下一个种子节点时,需要重新计算所有节点的影响,所以 MS-Greedy 算法相当耗时. 为了满足在线查询的需求,我们要设计更为有效的算法,不但能够满足高效性的需求,同时也不会损失太多的精度.

5 Bound-based 算法

算法 MS-Greedy 虽然能够求解 LA-J-MIN-Seed 问题,但是非常耗时,不具有良好的扩展性,难以应对大规模的社会网络结构. 算法 MS-Greedy 主要有两个缺点:第 1 个缺点是没有区分节点,同等地对待每个节点,并且在每次迭代过程中都考虑所有的节点来选择种子节点. 针对此问题,我们的解决方法是对所有节点进行分类. 每个节点的类别只能是候选种子节点和非种子节点中的一种. 然后将类别是候选种子节点的所有节点组成一个候选种集,只有该集合中的节点有可能成为种子节点,这样就剪枝掉很多节点,从而避免了对这些节点的影响计算.

第 2 个缺点是算法在找下一个给种集 S 带来最大边际收益 $\sigma(S\cup\{x\},V_R)-\sigma(S,V_R)$ 的种子节点 x 时，需要更新 S 能够影响到的候选种子节点的影响，并且枚举所有的剩余候选种子节点进行选择节点 x. 然而对于很多节点，由于其影响 $\sigma(S\cup\{x\},V_R)$ 非常小，我们不必计算. 对于这样不重要的节点，我们应该避免其不必要的计算. 受上述启发，我们提出了一种上界策略，即估计 $\sigma(S\cup\{x\},V_R)$ 的上界，先访问上界大的节点，从而避免了对影响小的节点的访问.

5.1　候选种集

我们想要识别出一个候选种子节点集合，将其定义为 C，该集合包含了所有可能成为种子节点的节点. 换句话说，不在候选种子节点集合 C 中的节点不可能成为种子节点. 因此，我们只需要考虑集合 C 中的节点来选择目标种集 S. 现在对病毒式营销的研究，将整个节点集合 V 作为候选种集 C，我们的目标是尽可能地缩小候选种集 C.

很明显，查询区域 R 中包含的节点一定是候选种子节点. 另外还有很多节点虽然在查询区域外，但其对查询区域也有影响，这些节点也有可能是候选种子节点. 这些节点中有的影响值大，有的影响值小，为了剪枝掉影响值小的节点，我们设置一个影响阈值 θ. 如果 u 对 v 的影响值 $p(u\to v)<\theta$，则认为 u 对于 v 是不重要的节点. 如果 u 对于 V_R 中每个节点都不重要，则表示 u 对区域 R 的影响非常小，可以忽略其影响，可以将其剪掉，即 u 不是区域 R 的候选种子节点.

定义 7（$influencee(u)$ 和 $influencer(u)$）. 给定两个节点 u 和 v，如果 $p(u\to v)\geq\theta$，则 u 称为 v 的 $influencer$，v 称为 u 的 $influencee$. 因此，$influencee(u)=\{v\mid p(u\to v)\geq\theta\}$，$influencer(v)=\{u\mid p(u\to v)\geq\theta\}$.

定义 8（候选种集 C）. 给定一个查询 $Q=(R,J)$，如果节点 u 是集合 V_R 的 $influencer$，则 u 是一个候选种子节点. 候选种集 $C=\bigcup\limits_{v\in V_R} influencer(v)$.

为了有效地支持在线查询，我们预先计算出图 G 中所有节点的 $influencer(v)$ 和 $influencee(v)$，然后计算出候选种集 C. 为了高效地计算出候选种集 C，我们利用图中节点并基于它们的地理位置信息建立一棵四叉树，并且利用该四叉树计算查询区域 R 包含的节点集合 V_R. 然后对于每个节点 $v\in V_R$，枚举出它的前驱节点 u，如果满足 $p(u\to v)\geq\theta$，则将节点 u 加入到候选种集 C 中，并继续遍历节点 u 的前驱节点. 通过上述的迭代方式可以计算出候选种集 C.

例 6.　图 3 是图 2 中节点建立的一棵四叉树，该四叉树包含树节点 A、B、C、D、BC、BD、CA、DA、DB 和 DD. 假设 $\theta=0.05$，根据定义 7，可以计算出节点 2 的 $influencer(2)=\{0,3,4,7,8,9\}$，$influencee(2)=\{1\}$. 对于查询 Q，查询区域中的节点是候选种子节点. 由于节点 0，3，7 是节点 2 的 $influencer$，即它们对查询区域 R 的影响大于 0.05，所以它们也是候选种子节点. 节点 1，6，15 对查询区域 R 没有影响，所以它们不是候选种子节点. 综上所述，候选种集 C 为 $\{0,2,3,4,5,7,8,9,10,11,12,13,14\}$.

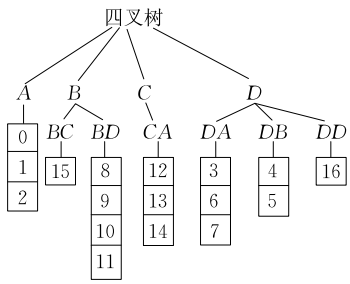


图 3　四叉树

5.2　影响上界

算法 MS-Greedy 需要更新所有与种集 S 有共同影响的节点的影响值. 由于计算 $\sigma(S\cup\{u\},V_R)$ 代价非常大，我们提出一种上界策略，即估计影响值的上界，然后利用上界值去选择种子节点.

选择第 1 个种子节点时，我们需要计算出每个候选种子节点 $u\in V_R$ 的影响值 $\sigma(\{u\},V_R)$，然后选择影响值最大的节点作为第 1 个种子节点. 最初时，种集 S 为空集，其影响值 $\sigma(S)=0$，则节点 u 在种集 S 下的边际收益就是 u 的影响值 $\sigma(\{u\},V_R)$. 为了方便选择种子节点，我们利用候选种集 C 中的节点，以它们的边际收益值 $\sigma(\{u\},V_R)$ 为关键字建立一个大根堆，则其堆顶的节点就是第 1 个种子节点. 假设堆顶节点为 v，将其从大根堆中删除，加入到 S 中，S 变为 $\{v\}$，然后调整大根堆.

在选择第 2 个种子节点时，首先取大根堆的堆顶元素，假设为节点 u. 如果 u 和 S 没有共同影响，即 $influencee(u)\cap influencee(S)=\varnothing$，则对任意节点 w，$P(\{u,v\},w)-P(\{v\},w)=P(\{u\},w)$，从而节点 u 在种集 S 下的边际收益值保持不变. 如果 u 和 S 有共同影响，即 $influencee(u)\cap influencee(S)\neq\varnothing$，则需要重新计算 $\sigma(S\cup\{u\},V_R)$. 由于计算 $\sigma(S\cup\{u\},V_R)$ 代价非常大，我们估计其上界值.

令 $\sigma(\{u\}\mid S,V_R)=\sigma(S\cup\{u\},V_R)-\sigma(S,V_R)$ 表示在当前种集 S 下节点 u 的边际收益. 如果 S 对节点 w 的影响和 u 对节点 w 的影响相互独立，则

$$P(S\cup\{u\},w)=P(S,w)+P(\{u\},w)-P(S,w)\cdot P(\{u\},w),$$

否则

$$P(S\cup\{u\},w)\leq P(S,w)+P(\{u\},w)-P(S,w)\cdot P(\{u\},w) \quad (6)$$

并且在 w 的影响树 $Tree(w)$ 中，对于 w 的任意孩子 c，有 $P(S,c)\leq 1$，则

$$P(S\cup\{u\},w)\leq 1-\prod_{c\in child(w)}(1-p(c,w)) \quad (7)$$

综上所述，我们可以计算出对节点 w 影响的上界值 $\hat{P}(S\cup\{u\},w)$、对集合 V_R 的影响上界 $\hat{\sigma}(S\cup\{u\},V_R)$ 和边际收益的影响上界 $\hat{\sigma}(\{u\}|S,V_R)$，

$$\hat{P}(S\cup\{u\},w)=\min\{(6),(7)\},$$

$$\hat{\sigma}(S\cup\{u\},V_R)=\sum_{w\in V_R}\hat{P}(S\cup\{u\},w),$$

$$\hat{\sigma}(\{u\}|S,V_R)=\hat{\sigma}(S\cup\{u\},V_R)-\sigma(S,V_R) \quad (8)$$

通常情况下，假设当前种集为 $S_i=\{s_1,s_2,\cdots,s_i\}$，则有

$$P(S_i\cup\{u\},w)\leq P(S_i,w)+P(\{u\},w)-P(S_i,w)\cdot P(\{u\},w) \quad (9)$$

$$P(S_i\cup\{u\},w)\leq 1-\prod_{c\in child(w)}(1-p(c,w)) \quad (10)$$

$\hat{P}(S_i\cup\{u\},w)=\min\{(9),(10)\}$，$\hat{\sigma}(S_i\cup\{u\},V_R)=\sum_{w\in V_R}\hat{P}(S_i\cup\{u\},w)$ 成立，我们可以利用如下公式估计出节点 u 在种集 S_i 下带来的边际收益值 $\hat{\sigma}(\{u\}|S_i,V_R)$，

$$\hat{\sigma}(\{u\}|S_i,V_R)=\hat{\sigma}(S_i\cup\{u\},V_R)-\sigma(S_i,V_R) \quad (11)$$

5.3 Bound-based 算法

5.2 节详细地描述了怎样估计影响值的上界，在此基础上，我们提出了 Bound-based 算法. 该算法用到了一个大根堆 H 和一个映射表 M. 借助大根堆 H 可以快速地找到影响值最大的节点. 映射表 M 用来区分节点 u 的边际收益值是初始的影响值 $\sigma(\{u\},V_R)$，或者是边际收益的估计上界值 $\hat{\sigma}(\{u\}|S_i,V_R)$，或者是边际收益的准确值 $\sigma(\{u\}|S_i,V_R)$.

下面我们描述该算法怎样使用影响的上界值选择下一个种子节点. 假设当前种集为 S_i，大根堆的堆顶元素为节点 u. 如果 u 与 S_i 没有共同的影响，即 $influencee(u)\cap influencee(S_i)=\varnothing$，则 u 的边际收益值不变，u 仍然是在种集 S_i 下边际收益最大的节点，所以 u 为下一个种子节点. 如果 u 与 S_i 有共同的影响：(1) 如果此时 u 的边际收益值仍是最初的影响值 $\sigma(\{u\},V_R)$，我们需要用式(11)估计出其新的边际收益的上界值，并将其加入到大根堆中；(2) 如果此时 u 的边际收益值是估计值 $\hat{\sigma}(\{u\}|S_i,V_R)$，我们需要用式(4)计算出其边际收益的准确值 $\sigma(\{u\}|S_i,V_R)$，并将其加入到大根堆中；(3) 如果此时 u 的边际收益值是准确值 $\sigma(\{u\}|S_i,V_R)$，则 u 为下一个种子节点.

算法 2. Bound-based.

输入：$G(V,E)$：位置敏感的社会网络；$Q=(R,J)$：查询，其中 R 是查询区域，J 是正整数

输出：S：种子集合

```
//OFFLINE-INDEXING
1.  预计算所有节点的 influencee(u)和 influencer(u)
    //ONLINE-SEARCH
2.  计算出查询区域 R 的候选种集 C
3.  用 u∈C 的影响 σ({u},V_R)初始化大根堆 H
4.  初始化种集 S=∅
5.  初始化 influencee(S)=∅
6.  WHILE (σ(S,V_R)<J) DO
7.   初始化映射表 M=∅
8.   WHILE (H≠∅) DO
9.    u=H.POP()
10.   IF influencee(u)∩influencee(S)=∅
       THEN S=S∪{u}influencee(S)
                =influencee(u)∪influencee(S)
        BREAK
11.  ELSE IF u∉M
       THEN 估计 σ̂({u}|S,V_R)
            将〈u,σ̂({u}|S,V_R)〉加到 H
            将〈u,1〉加到 M
12.  IF u∈M 且 u.flag==1
       THEN 计算 σ({u}|S,V_R)
            将〈u,σ({u}|S,V_R)〉加到 H
            更新 M 中 u 的对应项为〈u,2〉
13.  IF u∈M 且 u.flag==2
       THEN S=S∪{u}influencee(S)
                =influencee(u)∪influencee(S)
        BREAK
14.  RETURN S
```

算法 2 给出了 Bound-based 算法的伪代码. 第 1 行表示线下预先计算 V 中所有节点的 $influencee$ 集合和 $influencer$ 集合，然后进行在线查询. 第 2 行计算出查询区域 R 的候选种集 C. 第 3～14 行主要借助大根堆 H、映射表 M 和影响的上界来迭代选取种子节点，直到 $\sigma(S,V_R)\geq J$ 成立(第 6 行)，算法停止，S 就是最后的目标种集.

第 3 行用候选种子节点的影响初始化大根堆 H. 第 4 行初始化当前种集 S 为空集. 第 5 行初始化

S 能影响的节点集合 $influencee(S)$ 为空集，因为 S 为空集. 第 6～13 行进行迭代的选取种子节点. 第 7 行初始化映射表 M 为空集. 每次迭代，取 H 的第 1 个节点 u，检查 u 是否与当前种集 S 的影响有相关性. 如果没有相关性(第 10 行)，则 u 是下一个种子节点，将 u 加入到 S，并更新 $influencee(S)$. 如果 u 和 S 的影响有相关性，需要利用映射表 M 来判断怎样计算 u 的边际收益. 如果 $u \notin M$(第 11 行)，则 u 的边际收益仍是初始影响，需要用式(11)估计其上界，将$\langle u, \hat{\sigma}(\{u\} \mid S, V_R)\rangle$加入到大根堆 H，将$\langle u, 1\rangle$加到 M. 如果 $u \in M$ 且 $u.flag=1$(第 12 行)，则 u 的边际收益是在当前种集 S 下计算的估计值，需要使用式(4)计算准确值，将$\langle u, \sigma(\{u\} \mid S, V_R)\rangle$加入到 H，更新 M 中 u 对应的项为$\langle u, 2\rangle$. 如果 $u \in M$ 且 $u.flag=2$(第 13 行)，则 u 的边际收益是在当前种集 S 下计算的准确值，u 为下一个种子节点，将 u 加入到 S，并更新 $influencee(S)$，结束此次迭代.

例 7.　我们仍然使用例 4 的查询区域 R，但是将 J 设为 9. 首先找到候选种集 C 为{0,2,3,4,5,7,8,9,10,11,12,13,14}，共有 13 个节点. 然后用 C 中的节点及影响初始化 H. 首先进行第 1 次迭代，H 中影响最大的节点为 9，它与当前种集 S(空集)无相关性，所以 9 为第 1 个种子节点，9 加入到 S 中，S 的影响为 $3.75<9$，进行第 2 次迭代. 当前 H 中影响最大的节点为 7，其影响为 2.29，由于节点 7 与 S 有相关性，且其影响仍是初始影响，所以估计其影响的上界为 2.2765，然后更新 H；此时 H 中最大的节点为 3，它也与 S 有相关性，且其影响仍是初始影响，估计其影响的上界为 2.2765，并更新 H；此时 H 中最大的节点为 7，由于 7 的影响值为估计值，计算其准确值为 2.258，并更新 H；此时 H 中最大节点又为 3，计算其准确影响 2.258，并更新 H；此时 H 中最大节点为 7，且其影响为准确值，所以节点 7 为第 2 个种子节点. 依次进行迭代，找到种子节点 14、12 和 10. 该算法仅计算了 13 个节点的影响就找到了第 1 个种子节点，而 MS-Greedy 算法要计算 17 个节点的影响才能找到第 1 个种子节点.

6　Partition-Assembly-based 算法

算法 2 有两个缺陷. 第 1 个缺陷是它要计算出候选种集 C，并且为 C 中的每个节点 u 计算其影响 $\sigma(\{u\}, V_R)$. 如果查询区域 R 非常大，导致候选种集 C 非常大，进而影响的计算量就十分大. 为了避免上述情况，我们提出一种索引策略，它能够更为高效地计算候选种集和初始影响. 第 2 个缺陷是将 C 中所有的候选节点都加入到大根堆 H 中. 实际上，C 中也包含许多不重要的候选节点，它们成为真正种子节点的概率相当小，因此不需要计算它们的初始影响，也不用将它们加入到 H 中. 因此，要尽早地识别出这些不重要的节点，将其剪枝掉，避免大量多余的影响计算. 为了实现这个目标，我们提出了 Partition-Assembly-based 算法.

在图 G 对应的四叉树中，令 V_t 表示树节点 t 所包含的图 G 中节点集合. 例如图 3 中四叉树的树节点 A 对应的集合 V_A 为{0,1,2}.

定义 9(树节点的索引表 I).　假设 t 是一个树节点，将 I_t 定义为 t 的索引，它是树节点 t 的 $influencer$ 及其影响值的索引列表. 每个索引项包含两部分，第一部分由 V_t 中所有节点的 $influencer$(如 u)组成，第二部分是对应的影响 $\sigma(\{u\}, V_t)$，即 $I_t = \{\langle u, \sigma(\{u\}, V_t)\rangle \mid u \in influencer(V_t)\}$，且该表按照影响值的降序排列.

我们可以采用自底向上的方法有效地计算出所有树节点的 I 索引. 具体描述为，首先遍历四叉树中叶子节点 t 中的每个节点，计算出 $influencer(V_t)$ 和对应的 $\sigma(\{u\}, V_t)$. 然后对每个非叶节点 t，它的 $influencer$ 集合就是其孩子节点 $influencer$ 集合的并集，即 $influencer(V_t) = \bigcup_{c \in child(t)} influencer(V_c)$，对于 $u \in influencer(V_t)$，$\sigma(\{u\}, V_t) = \sum_{c \in child(t)} \sigma(\{u\}, V_c)$.

定义 10(节点的索引表 F).　对节点 u，令 F_u 表示 u 能影响到的树节点 t 的索引列表，即 $F_u = \{\langle t, \sigma(\{u\}, V_t)\rangle \mid u \in influencer(V_t)\}$.

例 8.　仍以图 2 和图 3 为例，可以根据定义 9 计算出树节点 DB 的 $I_{DB} = \{\langle 3, 1.5\rangle, \langle 7, 1.5\rangle, \langle 0, 0.5\rangle\}$. 根据定义 10 可以计算出节点 4 的 $F_4 = \{\langle A, 0.5\rangle, \langle DB, 1\rangle, \langle CA, 1\rangle\}$.

给定一个查询 Q，识别出完全由 V_R 中节点组成的树节点，将其定义为 R_i，$i=1,2,\cdots,r$；除去所有 R_i 中节点后，V_R 中剩余的节点集合定义为 R_0. 利用定义 9 可以计算出列表 I_{R_i}，其中 $i=0,1,\cdots,r$. 利用定义 10 可以计算出所有节点 u 的 F_u. 利用这些列表，可以计算出任意节点 u 对区域 R 的影响 $\sigma(\{u\}, V_R)$，即

$$\sigma(\{u\}, V_R) = \sum_{0 \le i \le r} \sigma(\{u\}, V_{R_i}) \qquad (12)$$

我们提出了算法 Partition-Assembly-based 来避免算法 2 的缺陷. 该算法根据需要将候选种子节点加入到大根堆 H 中,并利用新的策略来剪枝不重要的节点,即利用上界 B_l 和下界 B_h 进行剪枝.

算法 3. Partition-Assembly-based.

输入:$G(V,E)$:位置敏感的社会网络;$Q=(R,J)$:查询,其中 R 是查询区域,J 是正整数

输出:S:种子集合

//OFFINELINE-INDEXING

1. 预计算所有节点的 $influencee(u)$ 和 $influencer(u)$
2. 预计算 V_R 对应的每个 R_i 的索引列表 I_{R_i}
3. 预计算每个节点 u 的索引列表 F_u

//ONLINE-SEARCH

4. 初始化种集 $S=\varnothing$
5. 初始化大根堆 H 的下界 $B_h=0$,列表的上界 $B_l=0$,列表的索引下标 $d=0$
6. WHILE $\bigcup_{0\le i\le r} I_{R_i}\neq\varnothing$ DO
7. FOR $i\in[0,r]$
   POP u_{id} FROM I_{R_i}
   $\left(u_{id},\sum_{0\le j\le r}\sigma(\{u_{id}\},V_{R_j})\right)$ 加入到 H
8. $u=$Bound-based(G,Q)
9. 用 u 的边际收益来更新 B_h
10. $B_l=\sum_{0\le i\le r}\sigma(\{u_{id}\},V_{R_i})$
11. IF $B_h\ge B_l$
    THEN $u=H.POP()$
    $S=S\cup\{u\}$
    IF $\sigma(S,V_R)\ge J$ THEN RETURN S
12. ELSE $d++$

算法 Partition-Assembly-based 的主要思想如下:给定查询区域 R,可以求出其对应的索引列表 I_{R_i},其中 $i=0,1,\cdots,r$. 由于索引表 I_{R_i} 都是降序排列的,所以每次将这 $r+1$ 个索引表的影响值最大的索引项加入到大根堆 H 中. 将这些最大索引项的影响值的累加和定义为 B_l,由于索引表降序排列,所以 B_l 表示的是所有未访问的候选种子节点的边际收益的最大值. 由于种集 S 是变化的,导致索引表中节点的边际收益不正确,所以采用算法 Bound-based 选择 H 中边际收益最大的节点 u,并将其边际收益值 $\sigma(\{u\}|S,V_R)$ 定义为 B_h,它表示的是下一个种子节点边际收益的下界. 很明显,如果 $B_h\ge B_l$,则 u 为下一个种子节点,将 u 从 H 中删除,调整 H,并更新 B_h;如果 $B_h<B_l$,则继续将 $r+1$ 个 I 索引表的影响值最大项加入到 H 中,更新 B_h 和 B_l. 重复上述过程,直到满足 $\sigma(S,V_R)$ 不小于 J 为止,此时的 S 为目标种集.

算法 3 给出了 Partition-Assembly-based 算法的伪代码. 算法的具体描述为:第 1~3 行进行线下的预计算. 第 4~12 行进行在线查询,迭代地寻找种子节点. 第 4 行初始种集 S 为空集. 第 5 行初始化下界 B_h 为 0,上界 B_l 为 0,列表的索引下标 d 为 0. 令每个索引表 I_{R_i} 对应的第 d 个影响大的节点为 $u_{id}=I_{R_i}[d]$. 在迭代地选择种子节点过程中,按照需要将候选种子节点加入到大根堆 H 中,即每次将所有索引表 I_{R_i} 的第 d 个影响大的节点 u_{id} 及其影响加入到 H 中(第 7 行). 由于 H 中的影响可能不是当前种集 S 下的边际收益,所以利用算法 2 的思想找到 H 中的边际收益最大的节点(第 8 行),并用其影响更新 B_h(第 9 行),则 B_h 表示下一个种子节点在当前种集 S 下边际收益的下界. 令 $B_l=\sum_{0\le i\le r}\sigma(\{u_{id}\},V_{R_i})$(第 10 行),由影响函数的子模性质可知,$B_l$ 表示所有未访问的候选种子节点在当前种集 S 下的边际收益上界. 很明显,如果 $B_h\ge B_l$(第 11 行),则找到了下一个种子节点,将其从 H 中删除,加入到 S 中,并判断 S 的影响是否不小于 J. 如果满足此条件,则 S 为目标种集,返回 S 结束算法;否则结束本轮迭代,进行下一次迭代. 如果 $B_h<B_l$(第 12 行),将所有索引表的第 $d+1$ 个影响大的节点 $u_{i(d+1)}$ 加到 H 中,更新 B_l,重复上述步骤.

例 9. 我们仍然使用例 4 的查询区域 R,但是将 J 设为 9. 对于查询 Q,我们得到 3 个完全覆盖的树节点 BD、CA 和 DB,并找到其对应的 I 列表. 我们也得到一个节点 2,计算其对应的 I 列表. 在找第 1 个种子节点过程中,我们首先访问 4 个 I 列表中的第 1 个节点 2、9、14 和 7,并将它们和它们对 V_R 的影响加入到 H 中. 我们获得 $B_h=3.75$,并且对应的节点为 9. 由于此时的 $B_l>B_h$,然后我们依次插入每个列表对应的第 2、3 节点到 H 中,我们得到 $B_l=3.75$,并且此时 $B_h=3.75\ge B_l$,对应的节点仍为 9,所以我们得到的第 1 个种子节点为 9. 这里我们仅访问了 10 个节点就找到了第 1 个种子节点,而 Bound-based 算法要访问 13 个节点.

7 实 验

数据集. 我们使用 3 个真实数据集进行实验,分别为 Gowalla、Twitter 和 Weibo. Gowalla 可以从开放的数据集网站 http://snap.stanford.edu/data/

index.html 下载，Twitter 和 Weibo 可以分别从网站 twitter.com 和网站 weibo.com 进行爬取，实验中使用用户经常登录的地点信息作为用户的位置信息. 表 1 详细地描述了 3 个数据集的信息.

表 1　数据集

数据集	顶点数	边数	平均度	最大入度	最大出度
Gowalla	197K	1.9M	9.67	739.000	735.0
Twitter	554K	4.29M	7.75	1.143	639.0
Weibo	1.02M	166.7M	166.20	1000.000	4979.0

边的概率. 我们使用权重级联模型和三价模型来设置边上的概率，前者边(u,v)的概率等于$1/|N_{in}(v)|$，其中$N_{in}(v)$是节点v的所有前驱节点集合，后者边(u,v)的概率随机从$\{0.1,0.01,0.001\}$中选取.

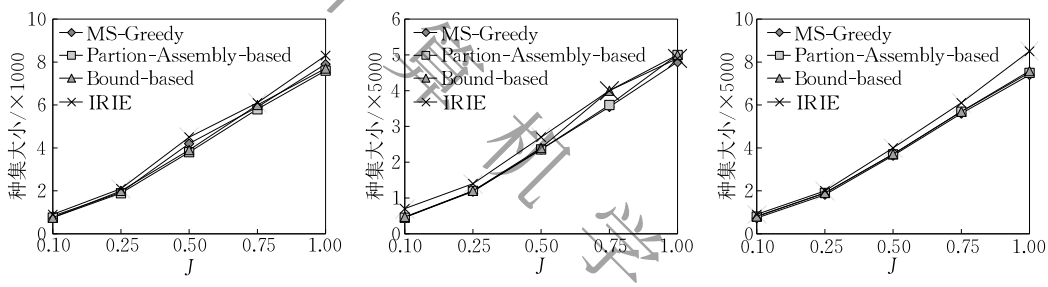
查询. 我们将查询区域R分为两种类别，即包含 10000 个节点的小范围查询和包含 100000 个节点的中等范围查询. 我们将J定义为 0～1 之间的实数，表示给定区域R包含的所有节点的J部分作为最终被影响的节点数. 实验中，设置J为 0.1、0.25、0.5、0.75 和 1.

算法. 我们与 IRIE[15] 进行比较. 为了公平，我们也为比较的方法进行 $influencee(u)$ 和 $influencer(u)$ 的预先计算. 所有的算法都用 C++实现.

实验设置. 所有实验都是在具有两个 3.0GHz 的处理器、48GB 的 RAM 和运行 Linux 系统的机器上运行.

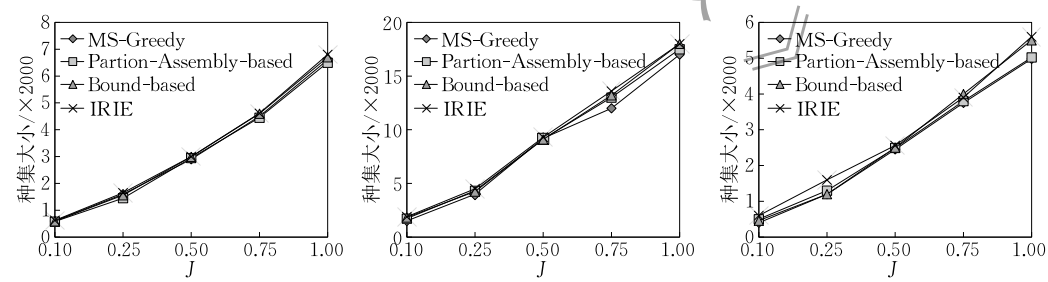
7.1　种子节点个数

我们设置J从 0.1 变化到 1，并且比较不同方法的有效性. 图 4 和图 5 分别表示权重级联模型和三价模型上的结果. 我们在以上 3 个数据集上进行了大量的实验，通过观察发现，对于 IRIE，我们设置$\alpha=0.3$，对于其他方法，针对不同数据集的不同特点设置不同的参数θ. 我们可以观察到这些方法找到的种集大小几乎相同，因为它们计算的影响范围几乎相同，又都是贪心地选择影响范围最大的节点.



(a) Gowalla($|V_R|\approx$10K)　(b) Twitter($|V_R|\approx$100K)　(c) Weibo($|V_R|\approx$100K)

图 4　权重级联模型下种集大小

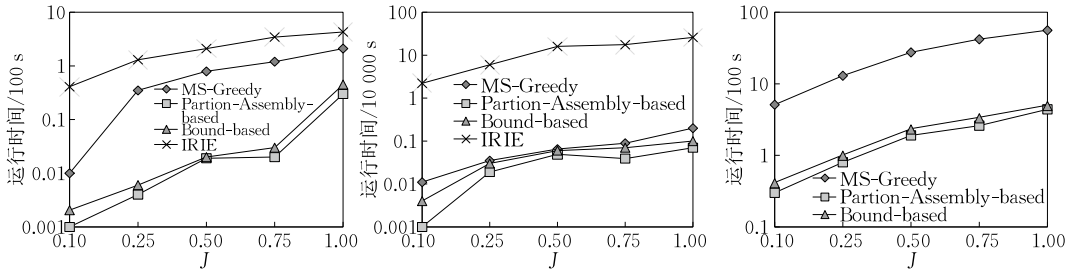


(a) Gowalla($|V_R|\approx$10K)　(b) Twitter($|V_R|\approx$100K)　(c) Weibo($|V_R|\approx$100K)

图 5　三价模型下种集大小

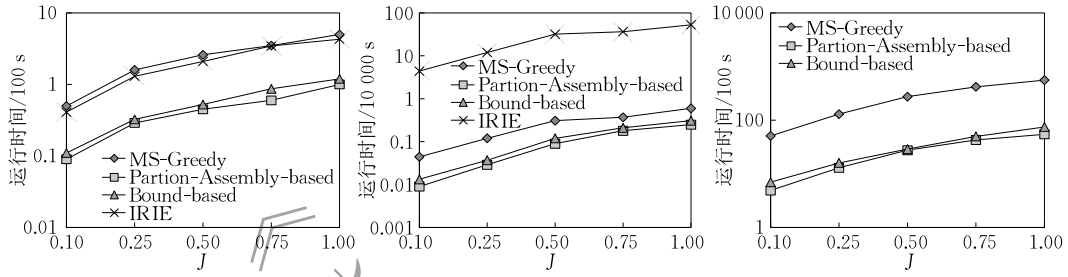
7.2　运行时间

我们评价了不同算法的高效性. 图 6 表示在权重级联模型下两种类型查询在 3 个数据集上的结果，图 7 表示三价模型下两种类型查询在 3 个数据集上的结果. 由于 IRIE 算法在 Weibo 数据上的运行时间超出图的表示范围，所以没有显示出来. 我们可以观察到，IRIE 算法最慢主要是因为在每次迭代选种子节点时，IRIE 要更新所有节点的边际收益，而我们的方法只更新与当前种集有相关性的节点的边际收益. 同时还能观察到 Partition-Assembly-based 算法比 Bound-based 算法高效，Bound-based 算法比 MS-Greedy 算法高效. MS-Greedy 算法在每轮迭代时都要更新很多节点的边际收益，而 Bound-based 算法利用估计的上界影响可以剪枝掉大量的不重要的节点，而算法 Partition-Assembly-based 利用子查询区域上预计算的结果极大地减小了大根堆H，并根据需要将节点加入到H中，计算量极大地减少了.



(a) Gowalla($|V_R|\approx$10K) (b) Twitter($|V_R|\approx$100K) (c) Weibo($|V_R|\approx$100K)

图 6 权重级联模型下运行时间

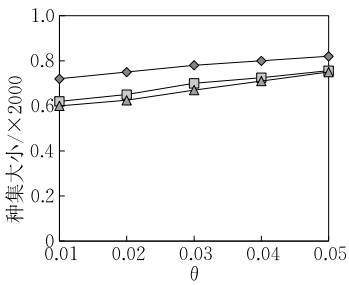


(a) Gowalla($|V_R|\approx$10K) (b) Twitter($|V_R|\approx$100K) (c) Weibo($|V_R|\approx$100K)

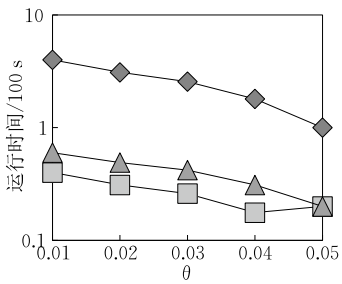
图 7 三价模型下运行时间

7.3 θ 的影响

我们改变 θ 的值从 0.01～0.05 来评估我们的算法. 图 8 表示的是数据集 Gowalla 上的结果,其他数据集上的结果同该结果一致. 从图 8 中我们可以观察到 θ 的值对选取种集的大小和运行时间的影响. 随着 θ 值的增加,导致一个节点的影响减小,从而种集在增大,预计算所需的时间在减少,运行的时间在减少.



(a) 种集大小—Gowalla



(b) 运行时间—Gowalla

图 8 变化 θ ($J=0.1$, $|V_R|\approx 1M$)

8 总 结

本文提出了位置敏感的 J-MIN-Seed 问题,首先证明了该问题在 IC 模型和 LT 模型下是 NP-hard 问题,然后扩展了现有的基于树的模型,并提出有效的朴素贪心算法框架 MS-Greedy. 该算法虽然有理论保证,但其计算量非常大、十分耗时. 为了提高其效率,本文又提出了两种贪心算法 Bound-based 算法和 Partition-Assembly-based 算法. 实验结果表明,本文算法既能有效地解决该问题、提供理论保证,又能高效地实现在线查询.

参 考 文 献

[1] Bryant J, Miron D. Theory and research in mass communication. Journal of Communication, 2004, 54(4): 662-704

[2] Nail J, Charron C, Baxter S. The consumer advertising backlash. Cambridge, USA: Forrester Com, Forrester Research and Intelliseek Market Research Report: 137, 2004

[3] Li G, Hu J, Lee T K, Feng J. Effective location identification from microblogs//Proceedings of the IEEE International Conference on Data Engineering (ICDE). Chicago, USA, 2014: 880-891

[4] Li G, Wang Y, Wang T, Feng J. Location-aware publish/subscribe//Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Chicago, USA, 2013: 802-810

[5] Li G, Chen S. Efficient location-aware influence maximization// Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. New York, USA, 2014: 87-93

[6] Chen N. On the approximability of influence in social networks //Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms. New Orleans, USA, 2008: 1029-1037

[7] Ben-Zwi O, Hermelin D, Lokshtanov D, Newman I. An exact almost optimal algorithm for target set selection in social networks//Proceedings of the 10th ACM Conference on Electronic Commerce. New York, USA, 2009: 355-362

[8] Ben-Zwi O, Hermelin D, Lokshtanov D, Newman I. Tree-width governs the complexity of target set selection. Discrete Optimization, 2011, 8(1): 87-96

[9] Reichman D. New bounds for contagious sets. Discrete Mathematics, 2012, 312(10): 1812-1814

[10] Shakarian P, Paulo D. Large social networks can be targeted for viral marketing with small seed sets//Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis & Mining. Istanbul, Turkey, 2012: 1-8

[11] Long C, Wong R C. Viral marketing for dedicated customers. Information Systems, 2014, 46(5): 1-23

[12] Kempe D, Kleinberg J, Tardos E. Maximizing the spread of influence through a social network//Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, USA, 2003: 137-146

[13] Chen W, Wang C, Wang Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks// Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, USA, 2010: 1029-1038

[14] Wolsey L A. An analysis of the greedy algorithm for the submodular set covering problem. Combinatorica, 1982, 2(4): 385-393

[15] Jung K, Heo W, and Chen W. Iries: scalable and robust influence maximization in social networks//Proceedings of the 12th IEEE International Conference on Data Mining. Brussels, Belgium, 2012: 918-923

LI Zhi-Hui, born in 1990, M. S. candidate. Her main research interests include social networks, massive data management and computing and so on.

ZHANG Zhao-Gong, born in 1963, Ph. D., professor, M. S. supervisor. His research interests include data mining, bioinformatics and so on.

LI Jian-Zhong, born in 1950, Ph. D., professor, Ph. D. supervisor. His research interests include massive data management and computing, wireless sensor networks and so on.

Background

The prevalence of social networks has prompted both industrial and academic communities to pay close attention to social network propagation problems. Its propagation means is viral marketing that takes the advantage of the effect of "word-of-mouth" among the relationships of individuals, such as friends, families, and co-workers, to promote a product. J-MIN-Seed problem has been extensively studied in the field of social network propagation. The goal is to select a seed set S that calls for influencing a certain amount of users at the end of influence propagation process, while the size of S is the smallest. Although the problem has been extensively studied, existing works neglected the fact that the location information can play an important role in J-MIN-Seed problem. Many real-world applications such as location-aware word-of-mouth marketing have location-aware requirement. For example, a social network system (e. g., Twitter) wants to provide new companies (e. g., restaurants) with marketing services by locating their potential customers in a spatial region (Harbin, Heilongjiang) to promote their businesses. Therefore we proposed the location-aware J-MIN-Seed problem, and proved that it is NP-hard. There are three main issues in location-aware J-MIN-Seed problem. The first is to obtain users' location, and the second is to efficiently and effectively calculate the influence spread of a given area, and the last is to design efficient algorithms to meet the high performance needs. To address the first issue, we assume that the user location was the place the user most frequently checked in. To address the second issue, we extend the existing tree model and design an effective and efficient approximate model. Based on the approximate model, we firstly propose a naive greedy algorithm MS-Greedy. Although MS-Greedy has theoretical guarantee, its computation is rather large, and its efficiency is not high. In order to meet the needs of online query, we then propose another two effective algorithms Bound-based and Partition-Assembly-based. Experimental results on real data show that our algorithms can solve the location-aware J-MIN-Seed problem effectively.

This work is supported by the National Basic Research Program (973 Program) of China (No. 2012CB316200), the National Natural Science Foundation of China (No. 61302139) and Provincial Natural Science Foundation of Heilongjiang (Nos. F2017024, F2017025).