

基于动态网络影响扩散问题研究

王学光

(华东政法大学信息科学与技术系 上海 201620)

摘要 影响扩散是复杂网络上动态过程研究的关键问题之一,而且基于动态网络的影响扩散问题的相关成果很少。讨论了动态独立级联模型和动态线性阈值模型以及基于这两个模型的动态影响最大化问题,提出了一种改进的贪婪算法,该算法消除了随机模型的不确定性并采用连通图方法来提高算法性能,并在不同规模的4个数据集上进行了验证。实验结果表明,与HT算法相比,提出的算法在影响扩散范围方面具有明显的优势,且在时间效率方面要好于HT算法。

关键词 动态网络,影响最大化,信息融合模型,启发式算法

中图分类号 TP393

文献标识码 A

Research on Influence Maximization Problem Based on Dynamic Networks

WANG Xue-guang

(Department of Computer Science, East China University of Political Science and Law, Shanghai 201620, China)

Abstract Influence spread is one of key problems about dynamic process problems on complex networks, and the research results of influence maximization problem based on dynamic networks are less. We discussed dynamic independent cascade model and dynamic linear threshold model and proposed a dynamic influence maximization problem based on above two models. Then, we presented an improved greedy algorithm, which eliminates the uncertainty of stochastic models and improves its performance by using connected graph approach. The algorithm was validated on four datasets with different sizes including AS, EMAIL, DELICIOUS and DBLP. The results show that, the size of influence spread of our algorithm has an obvious advantage and time efficiency is better compared with HT algorithm.

Keywords Dynamic network, Influence maximization, Information diffusion model, Heuristic algorithm

1 引言

创新、信息、疾病等的扩散、融合以及级联行为是发生在网络中的基本过程。比如,传染病在人类或动物网络中的传播是一种融合过程,像禽流感、口蹄疫等;类似地,计算机病毒也会在Internet上大规模地蔓延而导致严重的损失;一个人使用了一种新的产品后可能会告诉他的朋友们也去使用这种产品;而在Facebook和Twitter上,信息在个体间的扩散又是不同类型的过程;一些服务信息也可以通过E-mail网络大量传播,等等。因此,当我们计划推出新产品、传播新思想时,就可以利用网络的这种传播特征或称之为“word-of-mouth”、“viral marketing”效应。也就是说,我们可以先找到一些具有影响力的个体,让他们推荐产品给他们的朋友们,以使得这样的级联在人群中最大范围地传播。如何选择这些具有影响力的个体被称为“影响最大化问题(influence maximization problem)”。这里“影响”指的是信息、思想、产品、病毒等的传播。Domingos和Richardson最先将上述问题作为算法问题来进行研究^[1,2],Kempe等人将该问题形式化为离散优化问题,证明了该问题是NP-Hard问题并提出了贪婪近似算法^[3]。然而,他们方法的效率严重制约了它的可扩展性。Leskovec等

人提出了一个优化方法 CELF(Cost-Effective Lazy Forward)来加速这些个体的选择^[4]。然而,最值得关注的是Chen等人在解决影响最大化问题方面所做的工作,他们着眼于在网络结构局部范围内计算节点之间的相互影响从而得到具有影响力的节点,而不是去改进贪婪算法本身。首先,他们针对基于独立级联模型的融合过程进行了优化,提出了Degree Discount启发式算^[5]。该方法在影响扩散的精确度方面接近于贪婪算法而其运行时间较贪婪算法提高了6个数量级。随后,他们针对通用独立级联模型构建节点的局部树结构来限制影响传播的范围,并提出了MIA(Maximum Influence Arborescence)模型及其算法^[6]。该方法能够处理大规模网络的影响最大化的问题并具有较好的效率。同时,他们也提出了一个LDAG(Local Directed Acyclic Graph)方法来处理基于通用线性阈值模型的最大影响化问题并取得了较好的效果^[7]。上述工作都是在静态网络上展开相关研究的。

然而,网络本身是动态的,其中的个体和它们之间的交互随着时间的推移而变化。因此,在动态网络上的影响扩散过程与静态网络不同。目前,基于动态网络的影响最大化问题研究较少。Habiba等人首先利用一种贪婪算法研究了该问题^[8],通过实验证实了忽略网络的动态特性会导致不精确的

结论。但是,他们同样面临着该方法具有较低效率的问题,而且其实验规模较小。本文提出了一种解决动态影响最大化问题的改进贪婪算法 IGA,并在不同规模的 4 个数据集上进行了验证。与 Habiba 等人的 HT 算法相比,IGA 算法在影响扩散范围方面具有明显的优势,在时间效率方面 IGA 算法好于 HT 算法。

2 背景知识

2.1 融合模型

信息在网络上传播的模型已经被广泛研究^[3,9,10],两类基本的模型包括独立级联模型^[11](Independent Cascade Model, ICM)和线性阈值模型(Linear Threshold Model, LTM)^[12]。在 ICM 中,对于某一节点的激活尝试都是独立于其他已激活邻居节点的;而在 LTM 中,某一节点是否被激活受其所有已激活邻居节点的权重和的影响。

2.2 影响最大化问题

给定网络 $G=(V, E)$ 、影响扩散模型 M 以及正整数 k , 影响最大化问题(Influence Maximization Problem, IMP)是找到 k 个节点使得在当前模型 M 下由这些节点所影响的节点数目最大,即找到

$$A^* = \operatorname{argmax}_{A \in \{S \subseteq V, |S|=k\}} \sigma(A) \quad (1)$$

式中, $\sigma(A)$ 表示 k 节点集合 A 在影响扩散过程完成后激活节点的期望数; $\sigma(A_k^*) \geq \sigma(S)$, S 表示 G 中任意 k 节点的集合。

2.3 子模块函数

如果函数 f 对于所有的 $S \subseteq T \subseteq V$ 及任意 $v \in V$ 都有 $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$, 则称 f 具有子模块性(Submodularity)。子模块性说明随着集合规模的增加其中每一个体对它的影响在减少。

影响最大化问题是一个 NP-hard 问题,但是 Nemhauser 等人已经证明存在一个贪婪爬山算法使得在 $(1-1/e)$ 范围内近似最优解^[13]。该算法在定理 1 中给出。

定理 1 f 为非负单调子模块函数; S 是具有 k 元素的集合且每次选择一个元素进入 S 时使得 f 具有最大增量; S^* 为所有 k 元素集合中使得 f 具有最大值的一个集合。那么,有 $f(S) \geq (1-1/e)f(S^*)$ 。也就是说, S 保证了 $(1-1/e)$ -approximation。

对于影响最大化问题,具体的贪婪算法见算法 1。在该算法中,如果 $\sigma(\cdot)$ 函数为非负单调子模块函数,则根据定理 1 有如下结论:

如果在每一循环中都选择最优节点 v_k , 那么该算法保证 $(1-1/e)$ -approximation, 即, 集合 A 满足 $\sigma(A) \geq (1-1/e)\sigma(A^*)$, A^* 为所有 k 节点集合中使得 $\sigma(\cdot)$ 具有最大值的集合。如果在每一循环中节点 v_k 以 $(1-\epsilon)$ 近似最优节点, 那么该算法保证 $(1-1/e-\epsilon')$ -approximation, ϵ' 线性依赖于 ϵ 。也就是说, 该算法在性能方面可以获得确定性的保障。然而, 并不是所有的影响扩散模型都能够满足子模块性质。幸运的是, 对于 ICM 和 LTM 模型, Kempe 等人已经证明了它们具有这种性质^[3]。

定理 2 对于任意 ICM 实例, 函数 $\sigma(\cdot)$ 具有子模块性; 基于 ICM 的 IMP 是 NP-hard 问题。

定理 3 对于任意 LTM 实例, 函数 $\sigma(\cdot)$ 具有子模块性; 基于 LTM 的 IMP 是 NP-hard 问题。

算法 1 贪婪算法 Greedy

1. $A = \Phi$
2. for $i=1$ to k do
3. $v_i = \operatorname{argmax}_{u \in V \setminus A} (\sigma(A \cup \{u\}) - \sigma(A))$
4. $A = A \cup \{v_i\}$
5. end for

在算法 1 中一个关键问题是需要计算函数 $\sigma(A)$, 然而目前还没有有效的方法得到其精确解。但是, 可以采用 Monte-Carlo 方法多次模拟影响扩散过程, 以较高的概率来获得近似的结果。更具体地讲, 给定充分大的正整数 R 和已激活节点集合 A , 对于任意非集合 A 节点 $v \in V$, 从初始集合 $A \cup \{v\}$ 开始运行融合模型并统计激活节点数 $c_v^i, i=1, \dots, R$ 。 $\sigma(A \cup \{v\})$ 的估计值取 R 次运行的均值 $(1/R) \sum_{i=1}^R c_v^i$, 如算法 2 所示。由算法 1 和算法 2 可以得到贪婪近似算法的时间复杂度为 $O(knmR)$, 其中 n, m 分别为网络的节点数和边数, $\text{DiffMod}(\cdot)$ 花费的时间为 $O(m)$ 。

算法 2 评估 $\sigma(A \cup \{v\})$

- 1 while $v \in V \setminus A$ do
2. for $i=1$ to R do
3. $c_v^i = \text{DiffMod}(A \cup \{v\}) // \text{DiffMod}(\cdot)$ 执行融合过程
4. end for
5. $\sigma(A \cup \{v\}) = (1/R) \sum_{i=1}^R c_v^i$
6. end while

3 动态网络上的影响最大化问题

3.1 动态网络

许多之前的工作都是基于一种合成网络, 也就是说在一定的时间内所观察到的所有个体以及个体之间的交互同时展现在这种网络中。然而它忽略了节点和边出现的时序信息, 从而导致基于合成网络的相关结论并不适用于动态网络。动态网络是多个静态网络的时间序列, 静态网络是在某一时间点所观察到的个体以及个体间的交互构成的网络, 在这里假定时间是离散的。在某个时间的静态网络中出现的节点或边可能不会出现在下个静态网络中, 也就是说动态网络中的节点和边是随时间变化的。下面给出静态网络和动态网络的定义。

定义 1 令 $G_t = (V_t, E_t)$ 表示时刻 t 时的网络图, 其中 $V_t \subseteq V$ 是在时刻 t 观察到的所有个体集合, $(u_t, v_t) \in E_t$ 表示时刻 t 观察到的个体 u 与 v 之间的交互, $t \in \{1, \dots, T\}$ 。称 G_t 为静态网络。

定义 2 动态网络 $G = \langle G_1, \dots, G_T \rangle$ 是图 $G = (V, E)$ 的离散时间序列, 满足 $V = \bigcup_t V_t, E = \bigcup_t E_t \cup \bigcup_{t-1} (v_t, v_{t+1})$ 。注意 $(v_t, v_{t+1}) \in E$ 是跨时间的有向自边。

图 1(a) 给出了具有 5 个节点并由 5 个时间点序列构成的动态网络示意图。其中, 虚线圆圈所表示的节点意味着它在当前时刻未能被观察到, 而黑色圆圈则与之相反。实线表示当前时刻它所连接的两个节点之间存在相互作用, 而不同时间点之间的虚线箭头表示节点本身的自边, 也就是说上一时间点处于激活态的节点在下一时刻仍被激活。图 1(b) 是图 1(a) 的合成网络, 它无法体现节点及节点间交互的时序信息。

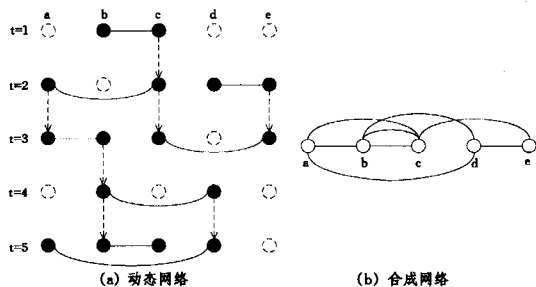


图1 一个动态网络示例

3.2 动态融合模型

2.1节所述ICM和LTM模型并不适用于动态网络。下面给出动态网络中的ICM和LTM模型。假定网络中初始激活节点集为 A_0 ,节点只能由非激活态转换为激活态, $t \in \{1, \dots, T\}$ 。

• 动态独立级联模型 (Dynamic Independent Cascade Model, DICM)

在DICM中,一个激活节点在时刻 t 尝试以概率 p_{v_t, u_t} 激活其邻居节点中未被激活的每一节点 u_t ,且其只有一次机会。如果 v_t 在时刻 t 成功激活 u_t ,那么在时刻 $t+1$ 无论 (u_{t+1}, v_{t+1}) 是否属于 E_{t+1} ,都有 u_{t+1} 保持激活。如果 v_t 在时刻 t 未能激活 u_t 且在接下来的某一时刻 v_{t+i} 重新连接到 u_{t+i} ,那么它将会再次尝试去激活 u_{t+i} 。这一过程直到时刻 t 达到 T 结束。DICM与ICM的区别关键在于已激活节点可以在不同时刻尝试去激活其同一邻居节点。

• 动态线性阈值模型 (Dynamic Linear Threshold Model, DLTM)

在DLTM中,每一非激活节点 v_t 受其已激活邻居节点的影响。如果 $\sum_{u_t} w_{u_t, v_t} \geq \theta_{v_t}$,则在时刻 $t+1$ 节点 v_{t+1} 被激活;否则, v_{t+1} 处于非激活态。其中, $0 \leq \theta_{v_t} \leq 1$ 为 v_t 的阈值,权重 w_{u_t, v_t} 为节点 u_t 对 v_t 的影响。在每一时刻 t 节点 v_t 都会被尝试去激活且不同时刻的尝试激活是独立的。这一过程执行 T 步结束。

3.3 动态影响最大化问题

给定动态网络 $G = \langle G_1, \dots, G_T \rangle$ 、融合模型 M 和正整数 k ,动态影响最大化问题 (Dynamic Influence Maximization Problem, DIMP) 的目的在于找到 k 个已激活节点的集合 A^* ,使得对于 $\forall A \in V$ 且 $|A| = k$ 都有 $\sigma_D(A^*) \geq \sigma_D(A)$ 。 $\sigma_D(A)$ 表示动态影响扩散过程完成后激活节点的期望数。如前文所述,如果 $\sigma_D(\cdot)$ 是单调子模块函数,那么通过逐次选择影响扩散增量最大的节点作为初始集合元素而得到的解 A_k 可以确保 $(1-1/e)$ -approximation,即 $\sigma_D(A_k) \geq (1-1/e)\sigma_D(A^*)$ 。Habiba等人^[8]证明了对于DICM和DLTM的任意实例,函数 $\sigma_D(\cdot)$ 都具有子模块性;基于DICM和DLTM的DIMP是NP-hard问题。同样,DICM和DLTM的函数 $\sigma_D(\cdot)$ 无法精确知道或计算,但可以通过大量模拟影响扩散过程来获得近似值。他们给出了一个贪婪近似算法,叙述见算法3。

算法3 HT算法

1. 对于网络中的每一节点计算其作为初始激活集合唯一元素时的影响扩散范围;为网络中的每一节点重复上述实验 R 次;
2. 计算每一节点作为初始单点集时 R 次实验结束后激活节点的平均数目;

3. 选择具有最大激活节点数目平均值的节点;
4. 移除该节点以及那些在 R 次实验中有 $R/2$ 次以上被激活的节点;
5. 重复步骤1-4直到选出 k 个节点。

在上述算法中需要注意几点:(1)无论是DICM还是DLTM,每次实验都要执行 T 步;(2)由于文中所用融合模型本质上是一种随机过程,因此实验次数 R 对于结果集合有着重要的影响,其值越大效果会更好;(3)该算法中在选出一个节点后,从网络中移除该节点以及一些相关节点,这将会影响接下来的融合模型运行和解集的精确度。鉴于此,在第4节中将给出一种新的贪婪算法来对其进行优化。

4 一种改进贪婪算法

我们采用算法1中的框架,其关键问题是如何评估DICM和DLTM模型的函数 $\sigma_D(A \cup \{v\})$ 。采用蒙特卡罗方法模拟影响扩散过程获得较为精确的值是一种通常使用的方法(即算法2中的方法),然而这种方法的效率是低下的。对于大规模的网络而言,即使找到较小的初始激活节点集合并使用现代化计算机仍需要几天的时间。下面给出一种有效的评估方法,其基本思想是首先将上述两种融合模型的随机性消除,然后利用图论方法计算每一节点的增量影响程度。

4.1 消除融合模型的随机性

为评估 $\sigma_D(A \cup \{v\})$,需要运行融合模型 R 次,它每次运行后实际上产生了一个确定图 $G_D = (V_D, E_D)$,其包括所有已激活的节点和链接这些节点的边。这也相当于对图中的边进行选择并与采用何种融合模型有关。因此,我们分别对DICM和DLTM模型进行讨论。

对于DICM模型而言,考虑动态网络 G 中节点 v_t 激活时的级联过程,它尝试以概率 p_{v_t, u_t} 去激活它的邻居节点 u_t 。这一随机过程的结果可以通过以偏置 p_{v_t, u_t} “抛硬币”的方式来决定,无论是在节点 v_t 激活时还是在DICM过程开始时。因此,可以在影响扩散过程开始前,对边 (v_t, u_t) 以概率 p_{v_t, u_t} 来判断其在这次融合模型运行中是否被保留。对 G 中的所有边都加以判断后,保留的边及其连接的节点便构成了运行一次融合模型后所得到的确定图 G_D 。这样做的目的是当 v_t 激活时能被它所激活的那些节点在 G_D 中必然存在一条到 v_t 的路径,那么就可以通过跟踪初始节点集 A_0 到其他节点的路径来计算影响扩散的范围。需要注意的是所有 (v_t, v_{t+1}) 形式自边的概率 $p_{v_t, v_{t+1}}$ 设置为1。

在DLTM中,节点 u_t 对其邻居节点 v_t 的影响为 $w_{u_t, v_t} \geq 0$,且满足时刻 t 对 v_t 的全部影响为 $\sum_{(u_t, v_t) \in E_t} w_{u_t, v_t}$ 。假定时刻 t 节点 v_t 以概率 w_{u_t, v_t} 随机选择一条邻居边 (v_t, u_t) ,而不选择其他边的概率为 $1 - \sum_{(u_t, v_t) \in E_t} w_{u_t, v_t}$ 。如果 u_t 没有边与 v_t 相连,则有 $w_{u_t, v_t} = 0$ 。从初始节点集 A_0 通过被选择的边所得可达集合的分布实质上与运行DLTM所得集合的分布是相同的。需要注意的是,在DICM中每一条被选择的边是独立于其他边的,所以就有可能一个节点存在多条边路径;而在DLTM中,每一激活节点仅有一条边路径存在。

由上所述,我们就可以不用运行融合模型而是通过产生 R 个图 G_D 的方式来评估 $\sigma_D(A \cup \{v\})$,那么如何在 G_D 上计算 $A \cup \{v\}$ 所激活的节点数目将在下一节给出。

4.2 计算节点影响

对于已获得激活节点集 A 和任意节点 $v \in V_D \setminus A$,令 F_{G_D}

$(AU\{v\})$ 表示 $AU\{v\}$ 在 G_D 上所能激活的节点集, $\varphi_v = |F_{G_D}(AU\{v\})|$ 表示集合的基, 我们借助连通图来计算 φ_v 。方法如下: 首先找到集合 $C = F_{G_D}(A)$, $\forall v \in C \setminus A$ 都有 $\varphi_v = |C|$ 。然后找到节点 $u \in V_D \setminus C$ 所在连通图的节点集合 $H_u \subseteq V_D$, 则 $\forall v \in H_u$ 都有 $\varphi_v = |C| + |H_u|$ 。令 $C = F_{G_D}(A) \cup H_u$ 后重复上述过程, 直到 $V_D \setminus C = \Phi$ 。这样, 对于给定的任何节点 v , 都可以直接得到 φ_v 。

综上, 具体给出新的贪婪算法 IGA。

算法4 improvedGreedyAlgorithm(G, k)

```

1.  $A = \Phi, R = 10000$ 
2. for  $i = 1$  to  $k$  do
3.   for  $j = 1$  to  $R$  do
4.     Construct Graph  $G_D^j = (V_D^j, E_D^j)$ 
5.     Compute  $\varphi_v^j = |F_{G_D^j}(AU\{v\})|$  for all  $v \in V_D^j \setminus A$ 
6.   end for
7.    $A = A \cup \{\arg\max_{v \in V_D \setminus A} (\frac{1}{R} \sum_{j=1}^R \varphi_v^j)\}$ 
8. end for

```

5 实验

5.1 实验设置

我们将在 4 个真实网络数据上验证所提算法的性能。第一个数据集来自于由 Michael Ley 维护的 DBLP 计算机科学文献数据库(<http://dblp.uni-trier.de/db/index.html>)。我们处理它们提供的 XML 格式数据, 选择从 1980 年到 2009 年间发表的所有文章以年为单位构建共作者动态网络, 该网络中共有 655491 个节点和 2007381 条边(不包括自边, 下同)。第二个数据集来自于 Oregon 大学的 Route Views 项目(<http://www.routeviews.org>)。该数据集中包含了从 1997 年 11 月到 2000 年 1 月间 Internet 上自治系统(Autonomous System, AS)之间的流量交互信息。由此构建的网络中包含 6474 个节点和 26467 条边。其他两个数据来自于文献[14]: 一个是 EMAIL 网络, 收集时间从 2003 年 10 月到 2005 年 5 月, 该网络中有 35756 个节点和 123254 条边; 另一个是在线社会网络 DELICIOUS, 由 203234 个节点和 430707 条边构成, 收集周期 10 个月。表 1 给出了上述 4 个数据集的相关信息。

表 1 数据集信息

数据集名称	节点数	边数	时间周期
AS	6474	26467	27 months
EMAIL	35756	123254	18 months
DELICIOUS	203234	430707	10 months
DBLP	655491	2007381	30 years

我们对比 IGA 算法、HT 算法、通用贪婪算法 Greedy、度启发式方法 Degree 以及随机选择法 Random。这里 Greedy 算法是算法 1 与算法 2 在动态网络上的直接应用, 它可以作为衡量初始激活节点集合影响范围的基准。Degree 算法选择网络中 k 个最大度节点作为初始集合, 我们使用

$$DEG_T(v) = \sum_{1 \leq t \leq T} \frac{|N(v_{t-1}) - N(v_t)|}{|N(v_{t-1}) \cup N(v_t)|} |N(v_t)| \quad (2)$$

来计算动态网络中的节点度^[15], 其中 $N(\cdot)$ 表示邻居节点集合; Random 算法随机选择 k 个节点作为初始集合。

为了获得上述 5 种算法的影响扩散范围, 对每一初始集合分别运行 DICM 和 DLTM 模型 10000 次并取其均值。在

DICM 模型中采用同一激活概率 $p = 0.05$, DLTM 模型中节点 v 的边权重设置为 $1/DEG_T(v)$ 。假定初始集合的大小从 1 到 30 依次增加。这些实验都是在 Intel CPU 3.2GHz 主频、4G 内存、Windows 7 系统上运行。

5.2 实验结果和分析

现讨论上述算法在不同数据集上运行 DICM 和 DLTM 模型的时间效率和影响扩散结果。

• DICM

图 2 显示了不同算法在 4 个数据集上的运行时间。横坐标表示 4 个数据集, 纵坐标采用对数坐标表示不同算法的运行时间, 以秒为单位。由于有些算法在选择较少初始节点时影响扩散要好于其他算法, 而在选择较多初始节点时则较其他算法有明显的劣势, 因此选择初始节点集合规模为 25 时各个算法的平均运行时间进行比较。从图 2 中可以看到, 对于具有几万到十几万条边的较小规模数据集 AS 和 EMAIL 而言, Greedy 算法仍旧是可行的, 在当前实验环境下所花费的平均时间分别为 953 秒和 7161 秒; 然而, 当数据集中的边数达到几十万到几百万时, Greedy 算法是不可行的, 估计在当前实验环境下要花几天的时间, 因此, 图 2 中没有显示 Greedy 算法在 DELICIOUS 和 DBLP 数据集上的运行时间。在不同规模的数据集上, IGA 算法较 HT 算法在运行时间上大约有 9% 到 24% 的提升; 相对于 Greedy 算法, IGA 算法在 AS 和 EMAIL 上运行时间则有 90.4% 和 89.7% 的大幅提升, 这得益于 IGA 算法在消除随机性和连通图方法的运用上。Random 算法是所有算法中运行时间最少的, 处于在微妙级和毫秒级之间。Degree 启发式算法的运行时间则保持在毫秒级。

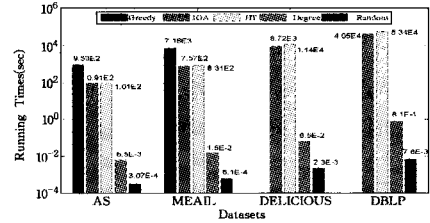


图 2 不同算法在 4 个数据集上运行 DICM 模型的时间

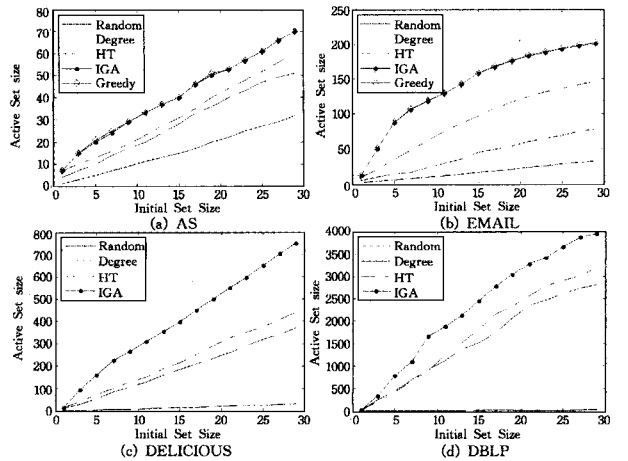


图 3 不同算法在 4 个数据集上运行 DICM 模型的影响扩散结果

图 3 显示了不同算法在 4 个数据集上的影响扩散结果。横坐标表示初始激活节点集合, 而纵坐标表示 DICM 模型影响扩散的范围, 以最终激活节点的平均数目来衡量。由于

Greedy 算法在数据集 DELICIOUS 和 DBLP 上的不可行性, 图 3(c)和图 3(d)中未给出使用 Greedy 算法的运行结果。从图 3 中可以看到, 使用 Random 算法得到的初始节点集具有最小的影响范围。简单的 Degree 启发式算法在各个数据集上影响扩散的结果要好于 Random 算法, 但是仍旧落后于其他算法。在较小规模数据集 AS 和 EMAIL 上 IGA 算法几乎与 Greedy 算法得到的结果一致, 而与 HT 算法相比, 它有着 24.1%和 41.2%的明显优势。在较大规模数据集 DELICIOUS 和 DBLP 上, IGA 算法较其他算法依然保持着较好的性能。在图 3(c)和图 3(d)中, IGA 算法影响扩散的范围要比 HT 算法多 46.8%和 29%。

• DLTM

图 4 和图 5 分别描述了不同算法使用 DLTM 模型在 AS、EMAIL、DELICIOUS 和 DBLP 4 个数据集上的运行时间和影响扩散规模。我们得到了与 DICM 一致的结论。在 DLTM 中, IGA 算法在时间上要优于 Greedy 算法和 HT 算法; 而在影响扩散规模方面, IGA 算法接近于 Greedy 算法且明显优于 HT 算法。Degree 算法和 Random 算法较上述 3 种算法虽然在获得初始节点集的时间上优势突出, 然而其最终激活节点数目与它们相比较差, 因此只能作为衡量其他算法的基准。

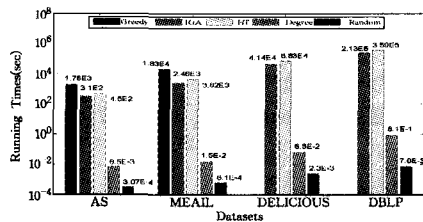


图 4 不同算法在 4 个数据集上运行 DLTM 模型的时间

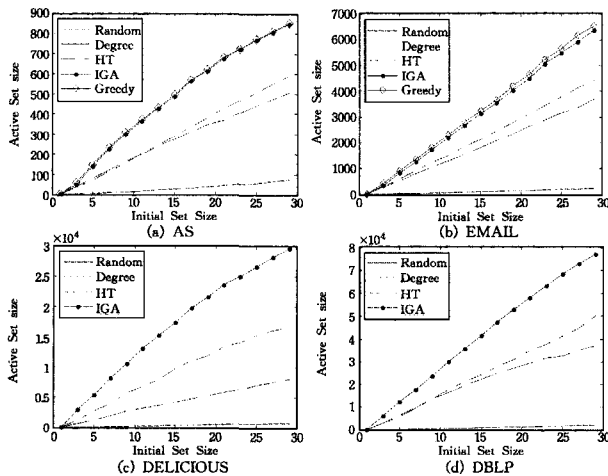


图 5 不同算法在 4 个数据集上运行 DLTM 模型的影响扩散结果

结束语 本文讨论了动态网络上的影响最大化问题, 描述了动态独立级联模型和动态线性阈值模型的基本过程, 给出了基于这两个模型的动态影响最大化问题是 NP-hard 问题的结论。为了解决这一问题, 提出了一种改进的贪婪算法 IGA, 并在不同规模的 4 个数据集 AS、EMAIL、DELICIOUS 和 DBLP 上进行了验证, 对实验结果进行了详细的讨论。与 HT 算法相比, 本文所提出的 IGA 算法在影响扩散范围方面

具有明显的优势, 且在时间效率方面 IGA 算法要好于 HT 算法。

参考文献

- [1] Domingos P, Richardson M. Mining the network value of customers[C]//7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). San Francisco, CA, 2001:57-66
- [2] Domingos P, Richardson M. Mining knowledge-sharing sites for viral marketing[C]//The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). Edmonton, Canada, 2002:61-70
- [3] Kempe D, Kleinberg J M, Tardos é. Maximizing the spread of influence through a social network[C]//The 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). Washington, DC, 2003:137-146
- [4] Leskovec J, Krause A, Guestrin C, et al. Cost-effective outbreak detection in networks[C]//The 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). San Jose, California, 2007:420-429
- [5] Chen W, Wang Y, Yang S. Efficient influence maximization in social networks[C]//The 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). Paris, France, 2009:199-207
- [6] Chen W, Wang C, Wang Y. Scalable influence maximization for prevalent viral marketing in large scale social networks[C]//The 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). Washington, DC, 2010:1029-1038
- [7] Chen W, Yuan Y, Zhang L. Scalable influence maximization in social networks under the linear threshold model[C]//The 2010 IEEE International Conference on Data Mining (ICDM). Sydney, Australia, 2010:88-97
- [8] Habiba, Berger- TWolf. Maximizing the extent of spread in a dynamic network[R]. Technical Report 20. DIMACS, 2007
- [9] Even-Dar E, Shapira A. A note on maximizing the spread of influence in social networks[C]//WINE 2007. LNCS 4858, 2007:281-286
- [10] Lahiri M, Cebrian M. The genetic algorithm as a general diffusion model for social networks[C]//The Twenty-Fourth AAAI Conference on Artificial Intelligence. Atlanta, Georgia, 2010:494-499
- [11] 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
- [12] Granovetter M. Threshold models of collective behavior[J]. American Journal of Sociology, 1978, 83(6):1420-1443
- [13] Nemhauser G, Wolsey L, Fisher M. An analysis of the approximations for maximizing submodular set functions [J]. Mathematical Programming, 1978, 14(1):265-294
- [14] Leskovec J. Dynamics of Large Networks[D]. Carnegie Mellon University, Pittsburgh, PA, USA, 2008
- [15] Habiba, Yu Y, Berger-Wolf T Y, et al. Finding spread blockers in dynamic networks[C]//The 2nd SNA-KDD Workshop. Las-Vegas, Nevada, USA, 2008