**社交网络和影响力的传播**

社交网络是人们传播影响力的媒介，人们在网络上传播着观点、想法、信息以及创新。消费市场中人们也发现依靠口碑效应能显著的提高利润。

这个问题的概述：

**前提**：一个有限的预算B（给予一些有影响力的人S一些免费的商品样品）；

**目标**：通过这些人在社交网络上的影响力，期望更多的人受到影响并且购买该商品；

**问题**：如何确定这些有影响力的人？

**研究方向**：社交网络的影响力传播模型研究；

**在这篇论文中所做的事情：**

**1.影响力模型**

1.1 线性阈值模型

1.2 独立级联模型

**2.影响力最大化问题**

2.1 算法

2.2 性能约束的证明

2.3 计算目标函数

**3.实验**

3.1 数据集和设置

3.2 结论

**4.影响最大化的一般化框架**

**5.非渐进式过程：活跃节点可以变成不活跃**

**6.更现实的营销**

**1.影响力模型**

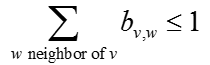
一个社交网络由一个有向图表示，节点表示用户，边表示用户间的影响力。每个节点初始状态要么是活跃的要么是非活跃的，一个活跃的节点可能会激活邻近节点。且假设:活跃节点永远不会变成不活跃的节点；

**1.1 线性阈值模型：**

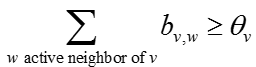
为每个节点随机的初始化一个阈值：

http://img.blog.csdn.net/20151121151549458

每个节点v受到其邻居节点w的影响力表示为：



一个非活跃的节点被激活的条件是：



**1.2 独立级联模型：**

独立级联模型中每个活跃节点只有一次机会可以激活其邻居节点，当v在t 时刻被激活了，那么在 t +1时刻它可以尝试激活其邻居，之后就不再尝试激活任何节点。且其尝试激活邻居的概率表示为Pvw。

**2.影响力最大化问题**

假设初始化活跃的节点为S，用f(S)表示最终活跃的节点数目。影响力最大化问题就转化为最大化f(S)；

**2.1 算法**

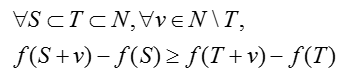
问题描述：

找到一个由k个节点组成的集合S，使得f(S)最大。但是，对这样一个函数f(S)求最大化是一个NP难问题；

解决方案：贪心算法，初始化一个空的S集合，然后每次添加一个节点，添加原则是使得f(S+v)-f(S)的值最大那个点将被加进S；

贪心算法所能达到的精度是1-1/e，也就是说用这种算法求得的S能激活至少63%（分母：k个任意节点的S集合所能激活的节点总数）的节点；

当一个函数满足下列条件，即称这个函数是子模函数：



**2.2 f(S)子模性证明**

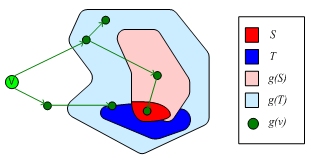
首先证明影响传播函数f(S)是子模的：

在独立级联模型中，我们是一步步激活非活跃的节点，概率是Pvw，此时每个节点被其所有邻居节点激活的顺序不确定，此时分析很困难，那么换一个思路，建立一个和该模型等价的模型。假设用一个概率不均等的硬币来决定是否激活，每次激活时进行抛硬币行为来确定是否激活邻居节点。那么我们可以在一开始就将所有的边的激活概率进行抛硬币确定是否激活，这样所有的边都变成了活边或者死边（要么激活，要么不激活，已经全部确定），很显然，这和一步步抛硬币是等价的，那么这个模型转换成了一个由确定活边和死边的有向图，求f(S)的最大化；如果给定初始S，那么最后f(S)是确定的，因为任意一个S中的点v能激活S集合外的一个点u当且仅当v到u有一条活边路径。

先定义：

g(S)：S集合中的点可达的且不在S中的节点的集合

T：S的一个超集



可得：g(T+v)-g(T)包含于g(S+v)-g(S)，即g(S)是子模的。

其中的g(S)是一次抛硬币的样本点，那么在整个样本空间中，f(S)可由以下函数得到：

http://img.blog.csdn.net/20151123100120255

子模的性质：子模函数的非负线性组合仍然是子模函数。

因此：

f(S)是子模函数。

**2.3 计算目标函数**

Nemhauser证明了一般化的贪心算法可以达到1-1/e-x的精度，其中x是任意一个大于0的数。

**3. 实验**

**3.1 数据集和设置**

arXiv高能物理理论部分的共同创作论文（合著论文）的协作图，共同创作网络广泛的捕获了社交网络的许多关键特性，图中包括10748个节点，53000条边；

**线性阈值模型实验设置：**

当两个作者v,w合著了C篇文章时，那么v对w的影响可以表示成：C/dw；那么w对v的影响可以表示成：C/dv；

**独立级联模型的实验设置（两种类型）：**

（1）对每条边初始化一个概率p，那么v激活w的概率为1-（1-p）^C——注意：因为v和w合著C篇，因此是C条边的激活概率之和。

（2）带权值的级联模型，v激活w的概率为1/dw。

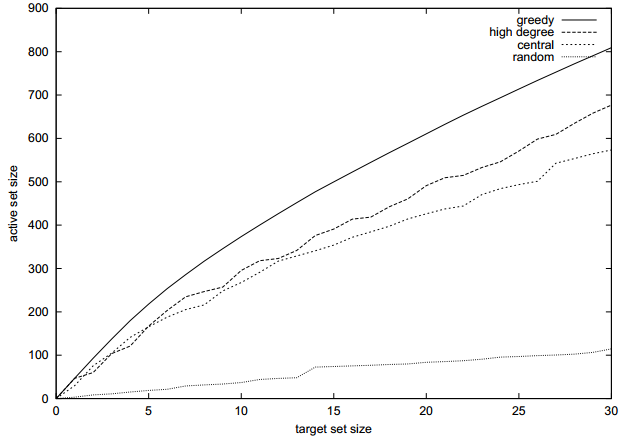
 对每个传播过程进行10000次的模拟，每次都重新设置阈值和p

进行对比的三个算法：

1. **基于节点度的启发式算法（选择度最高的k个节点作为初始的活跃节点）**
2. **基于距离的启发式算法（选择平均路径（到其他所有节点的距离）最小的k节点作为初始活跃节点）**
3. **随机算法（随机初始化k个活跃节点）**

**3.2 结果**

一：线性阈值模型

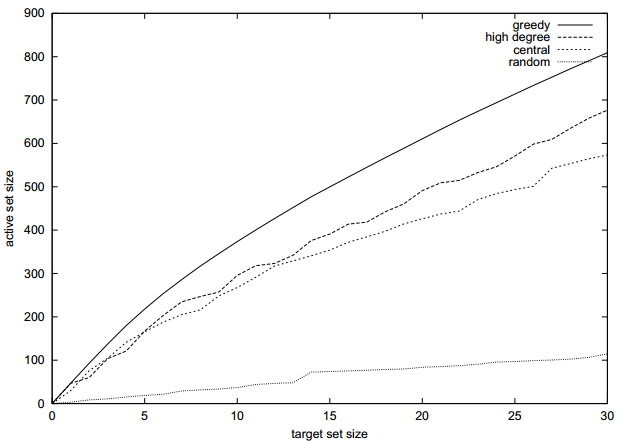


从图中可以看出：

贪心算法的性能比基于度的算法的性能要高18%，比基于距离的算法性能要高40%，随机的初始化S的结果性能很差；这表明明确的考虑一个网络中的动态信息比单纯的依赖图的结构信息更能获得好的营销结果。

后两种算法的缺陷分析：因为节点度很高的节点和中心性节点一般是聚成簇的，所以选取这些节点作为初始活跃节点并不能将影响更好的扩散下去。并且这两种算法的曲线不是平滑的，表面许多节点的度或者中心性不能准确的反映的节点的影响力。

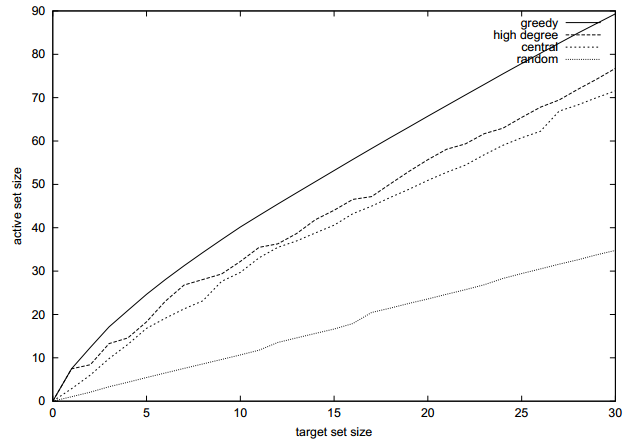
二：带权值的级联模型



从图中可以发现：

带权值的级联模型和线性阈值模型的结果惊人的相似，只是比例略有不同（所有的值都降低了25%）。

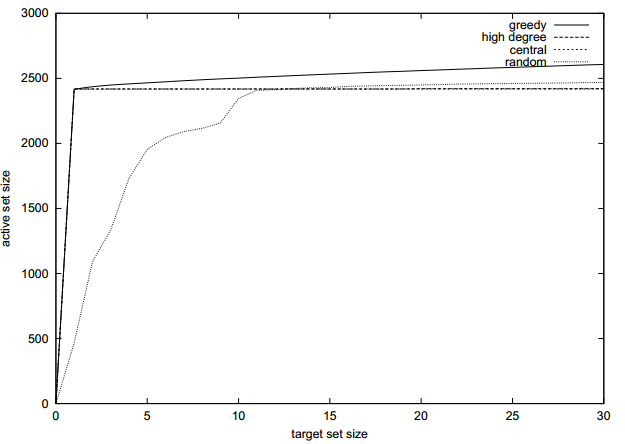
三：独立级联模型（p=1%）



从图中可以发现：

和上面的结果很相似，只是比例有很大的区别，平均每个节点只能激活其他的三个节点。这表明独立级联模型中当p很小时，性能比其他的模型都要差。有几个节点度超过100，那么在带权级联模中该点每个邻居激活它的概率甚至会小于1%（1/dv）。从线性阈值模型和独立级联模型中可以观察到网络的影响很大程度上依赖于低度的节点。

四：独立级联模型（p=10%）



从这个图中可以发现：

当p=10%时，随机模型的性能明显增加了好多，当初始节点数定为12的时候，随机模型的性能要超过那两个基于启发式的算法。主要原因是：当一开始就选取那些中心性节点，那么在下一个时间点会激活几乎25%的节点，但是接下来由于这些节点大部分都是聚成簇，因此不会再有更多的外部节点被激活，这解释了图中的那个陡峭的转折点。贪心算法在转折点之后还是有变大，只是幅度很小。而随机模型它完全没有关注中心节点，所以可以避开那个转折点，渐渐的增加，直到超过了那两个启发式算法。

**4. 影响最大化的一般化框架**

目的：研究更一般的影响模型，找到普遍性和可行性的平衡点。

**一般化线性阈值模型：**

节点v的邻居对其的影响不再是求和，而是一个一般化的函数f()。

**一般化级联模型：**

节点v激活其邻居的概率不再是Pvu，而是Pv(u,S),其中S是激活u失败的其他w的邻居节点。

**这两个模型的统一：**

对于任意一个一般化线性阈值模型（阈值函数是f()），都可以找到一个等价的一般化级联模型（激活概率是Pv(u,S)）。

对于该线性阈值模型实例，pv(u,S)表示集合S激活v失败，此时u激活该节点的概率，那么对应到线性阈值模型中既是：S激活节点v失败，表示fv(S)小于节点v的阈值（v的阈值范围此时是fv(S)到1）。那么此时u激活v的概率为(fv(SUu)-fv(S)\_/1-fv(S)。得到pv(u,S)和fv(u)之间的对应关系如下:

http://img.blog.csdn.net/20151130161232863

同理，给定一个一般化的级联模型实例，也可以得到它对应的线性阈值模型，fv(S)表示的是S的邻居激活v的概率，可以用1减去它的所有邻居都没有激活v的概率得到：

http://img.blog.csdn.net/20151130161919789

由于一般化的模型中有很多f()实例不是子模性的，因此近似算法性能比无法达到一个常数级n,是一个NP难问题。