## 前言

下面是几个摆在前面的问题:怎样处理一篇完全英文的论文?如果时间紧迫,怎样在限定时间内读懂、实现这篇论文所提及的东西?

所以,第一部分叫做理解,第二部分叫做实现.

## 理解

论文题目:In Search of Influential Event Organizers in Online Social Networks

搜索在线社交网络中有影响力的事件组织者

**abstract**

Influential Event Organizers:①skilled ②influential

在影响力集合G中寻找集合k个元素,这k个元素能够邀请来参加这个事件的人数最大.这是一个NP-hard问题,所以只找到了三个近似的算法来解决此问题,前两个算法是贪心算法,对解没有保证,第三个算法是一个2-近似的算法,如果有可行解的话也会找到一个可行解.

另外实际的测试表明了这个算法具有优越性.

**introduction**

Obviously, selection of influential event organizers is critical to the success of an event.

显然,对关键组织人物的选取是事件成功的关键

数学模型:Social Network--🡪G,v是其中的节点,A(v)是v属性的集合;问题转换为①寻找的k个user能够使(union A1 A2 … Ak) 能够覆盖给定的集合Q;即所有对于此次活动需要的属性都具备 ②k个人的集合能够影响G中最多的人

第一个算法叫做ScoreGreedy;第二个算法叫做PigeonGreedy(鸽巢贪心算法),它使用了鸽巢原理,对于k个user,为了满足|Q|个属性,至少有一人属性在|Q|/k之上,于是下一次选取的就是满足这一条件的user.尽管两个贪心算法运行速度快,但是不能保证即使真的存在满足Q的集合而找得到它,由此设计了”基于分区的ICS算法”.

下面是论文的结构:3给出问题的定义 4 给出贪心算法5给出P和P+算法 6给出研究

**definition of the ques.**

G中的每条边(v,u), w(v,u)在(0,1]之间,代表v对u的影响概率.我们寻找的解是一个集合S,|S|=k,且S所影响的点集合δS较大和S所覆盖的属性ρS接近于Q.

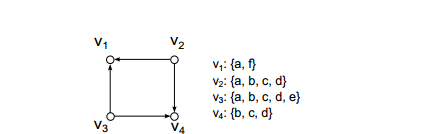
定义F1(S)为S所能影响所有顶点的数量,F1(S1,S2)=F1(S1+S2)-F(S2),为受S1但不受S2影响的数量

分数函数**Score(v,S’,Q’)**定义为综合了两部分的和:v能提供的属性价值+v能提供的数量影响价值;每一次都以分数来确定下一个应当选择的顶点.算法在k次循环后退出,如果k次循环没有找到Q,则返回空集.该算法在k较小的时候性能较好

注:关于影响概率对本问题并不构成影响,也超出了本问题的范围.通常假定w已经求得

**实例**

假设G为下图,Q={a,b,c,d,e,f},k=2;



**ScoreGreedy**

Input:Q,k,G

Output:Seed Set S

while |S|<k do

scores.update();

if(scores=NULL)return NULL;

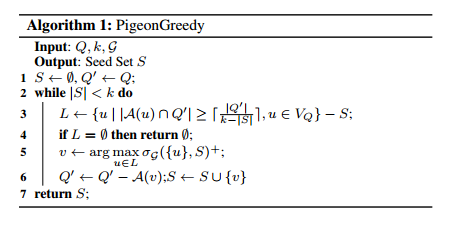
v=maxScoreNode(scores);

Q’<-Q’-A(v);S<-S+{v};

return S;

**PigeonGreedy**

若k>=|Q|,且存在解,那么该算法总能找到解,但是对于k<|Q|,没有这个保证



对于下一个顶点v,并不要求其必须是S的邻接点.也就是最后的S不必是一个互联的网络;如果两个顶点的影响数量相同,就比较他们的影响力之和,谁大谁优先;或者,也可以忽略这一点,从而任意选取一点.

图的节点具有的属性记为1-|A|的整数,用一个数组properties[][]来记录,true或者false来进行记录

## 两种贪心算法的java实现

### 精确解算法PCSI