Influence Maximization at Community Level:A New Challenge with Non-submodularity

---

背景

在线社交网络信息传播是最近比较流行的话题，其中影响力最大化（IM）问题被广泛的研究，之前的IM问题是针对影响社交网络中的每个用户来说的，而实际上社交网络中会存在很多组用户，每组用户被称为社区。因此有可能只需影响社区某几个用户就可以影响到这个社区的任何用户，为了适应这种场景，在社区层面上的影响力最大化问题就此提出。

创新点

1. 问题上的创新：提出了新的基于社区层面的影响力最大化问题，但它具有非子模性。
2. 方法上的创新：提出了三种的近似算法来解决上述问题，其中包含一种新的基于社区层面的采样方法。

相关工作

有关非子模性的影响力最大化问题的研究：

- k-boosting problem
- Comparative Independent Cascade Model——Sandwich Approximation strategy（本文涉及的一种近似算法基于这个思想）
- a seed selection strategy using network graphical properties——influence barricade model

解决方法

对于图$G=(V,E,w)$，它的生成图$\mathcal{G}=(V,E_{\mathcal{G}})$通过概率$w(u,v)$独立的选择每条边$(u,v)\in E$生成。传播模型采用独立级联模型，基于社区的影响力最大化（IMC）问题定义如下：

给定一个图$G=(V,E,w)$和一个正整数$0<k<|V|$，以及一个传播模型和社区集合Com，IMC问题要求寻找k个种子节点集合$S\subset V$，最大化影响社区的期望收益$c(S)$。

IMC问题假设社区是不重叠的。并且IMC问题的目标函数不满足子模性，在给定的指数时间内不可近似。

利益评估——RIC采样

- 定义一个社区的概率分布$\rho : Com \to \mathbb{R}$，$\rho(C_i) = b_i/b, b = \sum_{j=1}^{r} b_i$。其中$C_i$代表第$i$个社区，$r$为社区个数，$b$为收益。
- RIC样本g的生成：1）根据概率分布ρ随机选择一个社区C_g；2）从G中生成一个样本图$\mathcal{G}_g$；3）返回在$\mathcal{G}_g$中能够触碰到C_g的节点集合。如果在$\mathcal{G}$中存在一条路径连接u和社区C中的任意节点，称社区C被u触碰。
- 评估$c(S)$：如果S可以到达C_g至少h_g个节点，那么说g被S影响。定义$X_g(S) : 2^V \to \{0, 1\}$ g被S影响的评价函数。

随机RIC图的生成：

1. 随机选择一个源社区C_g，创建一个无边G的生成图$\mathcal{G}_g$，将边状态数组全设置为未知。并将C_g中所有节点加入到队列中。
2. 取出队头元素，访问它。遍历它的邻居节点和临边，依概率w创建边，并将选择的节点加入到队列中。直到没有节点再被选择为止。
3. 通过DFS搜索C_g每个节点的路径来寻找能够到达C_g中节点的节点集合$R_g(u)$。

$c(S)$的计算：

1. $c(S) = b \cdot \mathbb{E}(X_g(S))$
2. 让$\mathcal{R}$作为随机RIC样本的集合，$\hat{c}_{\mathcal{R}}(S) = \frac{b}{|\mathcal{R}|} \cdot \sum_{g \in \mathcal{R}} X_g(S) \quad \forall S \subseteq V$
3. 因此只要$|\mathcal{R}|$足够大，$\hat{c}_{\mathcal{R}}(S)$就可以近似计算$c(S)$

MAXR：

给定RIC样本集合$\mathcal{R}$，寻找一个种子集合S，$|S| = k$，最大化影响$\mathcal{R}$中RIC样本g的数量。如果S至少可以到达C_g中h_g个节点，那么g被影响。此问题是满足非子模性。

因此解决IMC问题需要：

- 求解MAXR问题，得到有最大影响RIC样本数量的S
- 生成足够多的RIC样本保证MAXR问题的解的界限误差。

三种近似算法

1. Upper Bound Greedy
    - 思想

      UBG基于三明治近似将$\hat{c}_{\mathcal{R}}(S)$夹在两个具有子模性的函数之间。
    - 算法

      Algorithm 2 UBG algorithm

      Input $\mathcal{R}, k$

      Output S

      1: $S_\nu \leftarrow$ greedy selection with objective function $\nu_{\mathcal{R}}(\cdot)$
      2: $S_c \leftarrow$ greedy selection with objective function $\hat{c}_{\mathcal{R}}(\cdot)$
      3: $S \leftarrow \arg\max_{S_\nu, S_c} \hat{c}_{\mathcal{R}}(S)$

      Return S

      通过贪心算法得到它上界函数和它本身的最优解，取最大的最为返回值。
2. Most Appearance First

- 思想

  MAF思想是考虑$\mathcal{R}$中节点或者社区出现的频率，尝试激活最有影响力的那些节点。

- 算法

Algorithm 3 `MAF` algorithm

Input $\mathcal{R}, k$
Output S

Initiate $S_1, S_2 \leftarrow \emptyset$
`SC` $\leftarrow$ sorted list of `Com` in order of their appearance in $\mathcal{R}$
while `SC` *is not empty* **do**
    `C` $\leftarrow$ take out 1st community of `SC`;
    `X` $\leftarrow$ pick `h` nodes in `C`
    if $|S_1 \cup \mathtt{X}| \leq k$ **then** $S_1 = S_1 \cup \mathtt{X}$
$S_2 \leftarrow k$ nodes that appear the most in $\mathcal{R}$
$S = \arg\max_{S' \in \{S_1, S_2\}} \hat{c}_{\mathcal{R}}(S')$
Return S

  i. MAF首先计算社区在$\mathcal{R}$中出现的频率，从大到小排序，依次从这些社区中选择阈值h个节点加入种子集合S_1，直到选够k个。
  ii. 选择k个在$\mathcal{R}$中出现频率最高的k个节点最为种子集合S_2。
  iii. 取让目标函数值最大的那个种子集合返回。

3. Algorithm for Bounded Activation Threshold

对于阈值$h \leq 2$的情况：

对于每个节点$u \in V$，寻找种子集合$K(u)$，最大化影响$\mathcal{G}_{\mathcal{R}}(u)$中RIC样本的数量。$\mathcal{G}_{\mathcal{R}}(u)$表示包含$u$能够触碰$C_g$的RIC样本集合。首先将$u$加入到$K(u)$，因为节点$u$一定触碰$\mathcal{G}_{\mathcal{R}}(u)$中的$g$，又$h_g \leq 2$，所有只需连接至多一个$g$中的节点即可。可以通过贪心算法得到近似解$K(u)$。

可以扩展到d-界限阈值。

实验

- 实验设置

  数据集采用：

Data	Type	Nodes	Edges
Facebook	Undirected	747	60.05 K
Wiki-vote	Directed	7.1 K	103.6 K
Espinions	Directed	76 K	508.8 K
DBLP	Undirected	317 K	1.05 M
Pokec	Directed	1.6 M	30.6 M

  用Louvain算法分割不重叠社区，并以随机算法分割作为对比。设置一个比较系数$s(init = 8)$，避免分割得到的社区过大。

  社区的收益等于社区节点个数，边权值设置为$1/d(v)$。预先设置其阈值为社区节点数的一半。

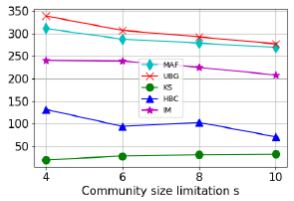
  由于没有存在的算法适应于IMC问题，所有和设计的启发式算法比较：

  - HBC(High Beneficial Connection)

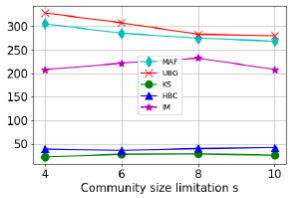
- KS(Knapsack-like Algorithm)，将阈值变成成本放入目标函数优化。
- IM

- 性能比较

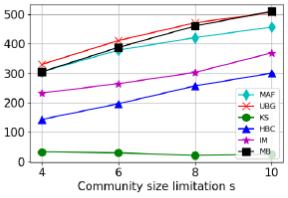
a. 解的质量



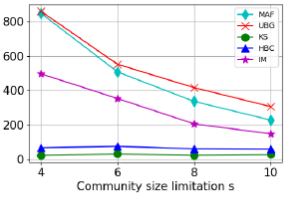
(a) Facebook, Random



(b) Facebook, Louvain

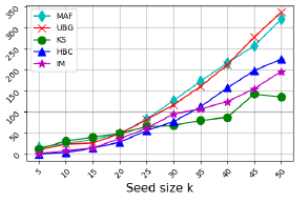


(c) Facebook, Louvain, bounded threshold

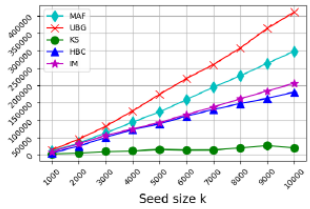


(d) DBLP, Louvain

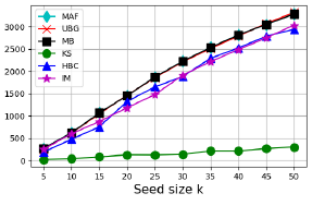
在不同数据集上，随着社区大小的增加，MAF和UBG得到的收益呈现下降趋势，但都要比其它算法性能好，随机划分和Louvain划分相差不大，表明算法的稳定性。在阈值有界限的情况下，Facebook数据集上，表现为随着社区大小增加。收益上升。



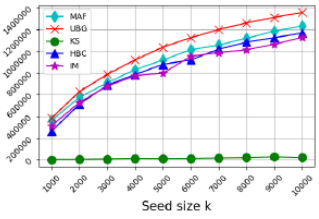
(a) Wiki-vote



(b) Pokec



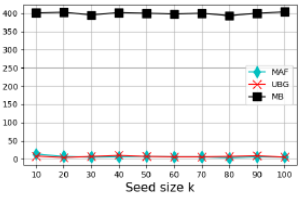
(a) Wiki-vote



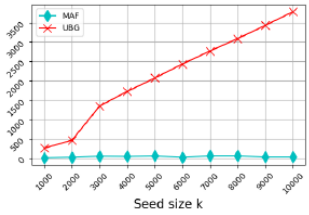
(b) Pokec

分别在Wiki-vote和Pokec数据集上，没有阈值界限和有阈值界限情况下，随种子大小，收益变化情况：MAF和UBG效果更好，而且收益不断增加。

b. 运行时间



(a) Espinions

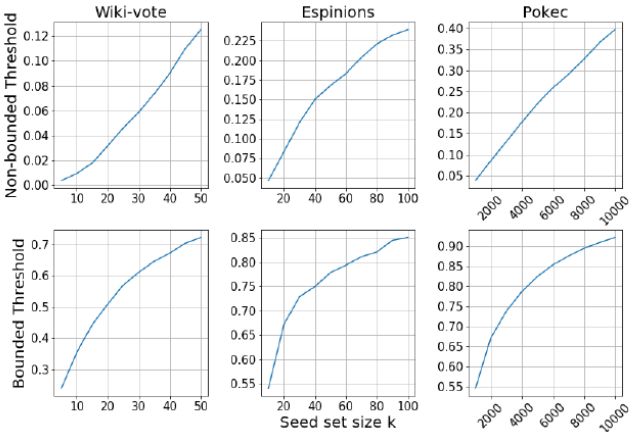


(b) Pokec

数据集较小时，MAF和UBG运行时间很短，随着种子集大小增加几乎不发生变化，而MB方法耗时较长。

数据集较大时，MAF算法运行时间更稳定更快，UBG随种子集大小增加运行时间增加。

c. UBG的评价



该实验测试了UBG在不同数据集下随着种子集大小k增加，在界限阈值和非界限阈值两种情况下的原解值与上界解值的比率。可以看出比率和k有很强的关系。

思考

1. 作者假设社交网络中的用户是以分组的形式即社区的形式存在的，一个网络包含多个社区，且社区是不重叠的。

   由此假设考虑以下问题：

   如果说社区是重叠的呢？

   暂时提出以下思路：

   可以通过图网络进行重叠社区的发现，而重叠的部分更有可能被选为种子。

2. 再者，社区划分在试验中得到的结果差别不大，原因为何？

   作者将社区分割大小限制，社区之间大小差别不大，而又将影响社区收益设置为社区节点个数...这样的设置可能导致了这个问题。这似乎不满足一般性。