基于距离控制和聚类系数的 Top K 影响力节点挖掘

Zhijie Zhang 18307130184

Tao You 18307110206

Ziqin Luo 18307130198

摘要：

本文将主要分析社会影响力分析(SIA)中的一个重点问题——Top-K 节点影响力最大化问题。本文通过分析在影响力传播过程中网络图的节点间距和节点聚类系数对其影响力的影响，实现了一种线性阈值模型上的基于距离控制与聚类系数度的混合策略(CDD)的启发式 Top-K 节点挖掘算法，并在静态网络与动态网络进行了实验以及与传统启发式算法的对比，进而分析算法的适用范围和评价算法的实际效果，希冀为影响力最大化的研究发展起到一些帮助。

关键词：社交网络、影响力最大化、距离控制、聚类系数、静态与动态网络

1 课题介绍

1.1 影响力最大化的 Top K 节点挖掘

Top K 节点影响力最大化问题是社会影响力分析(SIA)中的一个重点问题，同时也具有重要的现实意义。该问题最早的提出即是基于病毒营销中的实际问题：一家公司需要推广自己的新产品或服务，希望采用免费试用品的策略，说服少部分人试用其新产品或服务，当选中的用户对商品满意时，便会通过社交网络向自己的同事朋友推荐该商品，使得更多的人了解并且最终购买该商品，达到该公司的推广目的。而由于预算有限，免费使用品数量一定，因此，对公司而言，如何找出这“少部分”人来试用商品来使得最终购买人数最多就成为了公司需要解决的核心问题。而随着互联网的不断发展，社交网络的不断大型化，影响力最大化的 Top-K 节点挖掘的应用场景也愈发丰富，包括病毒营销、信息扩散、专家发现、链接预测等等，除了在市场营销和信息传播领域，在公共卫生领域中，也有学者希望通过 SIA，为传染性疾病的传播提供参考。

1.2 问题定义

该问题的算法希望通过对网络中的个体(节点)进行分析，从而挑选出网络中一些具有影响力的个体，作为最初的被激活的个体集，称之为种子集(seed set)。而对于算法的效果评估，则基于特定的影响力传播模型，令种子集中的个体首先被激活，并基于传播模型去影响其他的个体，从而使得其他个体也能被激活，最终被激活的个体数量，则被认为是该种子集的影响力大小。

给定一张有向无权图 G(V, E), 以及种子集的大小 k，算法需要给出大小为 k 的种子集S，SϵV，并希望最大化令S为种子集时在 G 上最终所激活的网络中的个体，将最终激活的节点集合记为$\sigma_G(\boldsymbol{S})$。即求解如下的优化问题：

$$\max\ |\sigma_G(\boldsymbol{S})|\ \text{s.t.}\ |\boldsymbol{S}| = k\ ,$$

$$where\ k\ is\ a\ given\ positive\ \boldsymbol{integer}, G\ is\ a\ given\ directed\ graph$$

1.3 研究概况

该问题最早是由 Domimgos 和 Richardson 等人，基于病毒营销的背景而提出，并将问题建模为马尔可夫场。[1] Kemp 等人于 2003 年提出了至今仍被广泛使用的传播过程模型——IC 模型(独立级联模型)和 LT 模型(线性阈值模型)，并同时证明了基于 IC 和 LT 模型，影响力最大化问题均为 NP-难问题。[2]

因此，求解该问题一般采用近似估计求次优解的方法，其算法主要分为基于传播(propagation-based)

和基于拓扑(topology-based)的两类算法。基于传播的算法主要使用贪婪算法，对每个未选中节点使用传播过程进行模拟，每次选择使得总影响力的一个节点，可以达到不低于最优解$(1-1/e)$倍的影响力效果[2]，如最早的 Greedy 算法，经过优化的 Greedy++，CELF，使用局部代替整体传播效果的 Hop Based 等等；基于拓扑的算法则希望直接通过网络的拓扑结构和节点属性，近似地度量影响力，进而选择种子集。

基于传播的算法尽管效果稳定(存在影响力下界)，但传播过程模拟需要大量使用蒙特卡洛模拟方法，导致其在复杂网络(10^4-10^5)上的运行时间可达数天。虽然基于拓扑的算法结果相对不稳定，但大部分情况下影响力效果足够可观，足以满足现实需求，而且其运行时间极短，同样大小的网络计算只需要数秒到数十秒，在网络日益大型化、复杂化的今天优势更为突出。经过小组成员讨论，由于现实应用的需求和实验设备的约束，本文将对基于拓扑结构的算法展开研究。

目前，基于拓扑结构的算法的研究重点旨在解决以下两个问题：(1)如何度量节点影响力。网络研究中各个中心性属性都被应用并研究：度中心性，中间中心性，接近中心性，Page Rank 中心性，K-shell 分解方法等等，但对于哪一个属性能够更好地对影响力进行近似，学界至今尚未达成共识；(2)如何规避影响力重复(rich-club effect)的问题。将两个影响力很大但影响范围高度重合节点同时加入节点集是不明智的，如 Degree Discount 算法采用更接近传播过程的方法，而 LIR 算法引入了 local index rank 属性，保证种子节点不互为邻居。

本文希望在两者间得到均衡，对于(1), 本文提出了聚类系数与度结合的算法，以此度量其影响力大小；对于(2)，本文在 LIR 算法的基础上提出了进一步的 d-距离控制策略，并将两者混合，设计算法。

2 算法设计

2.1 传播模型

网络图表示：

我们将有向无权图表示为$G = G(\boldsymbol{V}, \boldsymbol{E})$，其中$\boldsymbol{V}$表示网络图的节点集，$\boldsymbol{E}$表示网络图的边集，令节点 w 的入度邻居$N(v) = \{w \in \boldsymbol{V} | (w, v) \in \boldsymbol{E}\}$。

静态网络的传播模型：

线性阈值模型：由于个体间，以及个体间关系的随机性，随机的阈值和权值将会赋给网络 G 的节点和边。对于G中的任一节点v，赋予随机阈值θ_v，$\theta_v \sim Uniform(0,1)$，且对其入边$(w, v), \forall w \in N(v)$，赋予随机权重$b_{w,v}$，且满足$\Sigma_{w \in N(v)} b_{w,v} \leq 1$。对于一个处于未活跃状态的节点 v，当它的活跃邻居节点的影响力之和大于等于其阈值θ_v，即$\Sigma_{w \in N(v), w \text{ is active}} b_{w,v} \geq \theta_v$，节点 v 即被激活。[1]

传播过程算法如下：

(1)激活初始的活跃节点集合 S。

(2)在 T 时刻，节点 v 所有的处于活跃态的邻居节点都来尝试激活 v，如果影响力之和超过了 v 的激活阈值，则节点 v 在 T+1 时刻转换为活跃状态。

(3)上述过程不断进行重复，直到某时刻不再有节点被激活，传播过程结束。得到本次传播的最终激活数$\sigma(S)$。

由于每次传播具有随机性$(b_{w,v}, \theta_v)$，实验将传播过程进行多次，记$\sigma_i(S)$第 i 次模拟的影响力结果，$\sigma(S) = \text{AVG}(\sigma_i(S))$

动态网络的模型与传播：

快照模型：动态网络(TSN)为一个快照(snapshot)序列，即$TSN =< G_1, G_2, \dots, G_K >$，且每一快照对应着一个时间窗口：

$$T_K = \{(t_1, t_1'), \dots, (t_l, t_l'), \dots (t_K, t_K')\}, K \in N_+$$

各个时间窗口相互不重叠。对于快照组中的任意一个快照$G_l, l \in \{1,2, \dots, K\}$，都有$G_l = G_l(V_l, E_l)$的形式。对于任意两个节点$w$和$v$，要使$w, v \in V_l, (w, v) \in E_l$成立，当且仅当在时间窗口$(t_l, t_l')$中，存

在由w指向v的边。每一快照G_l都被视为一张静态图，可以应用 LT 模型模拟在每张快照上的影响力传播。

记任一快照 $G_l = G(V_l, E_l)$，(t_l, t_l')，$l \in \{1,2,\dots,K\}$；在时间窗口l:(t_l, t_l') 内，能影响v_i的节点集记为：$N_i(G_l) = \{w|(w, v_i) \in E_l\}$，被激活的节点集记为 $\Phi(l)$。[3]

传播过程算法如下：

(1)假设存在初始状态快照$G_0 = G(V_0, E_0)$，选择种子节点集$\Phi(0) \in V_0$，并将其激活。

(2)对于快照集合G_k，将$\{\cup_{l=0}^{k-1} \Phi(l)\} \cap V_k$作为种子节点集，在网络$G_k(V_l, E_l)$应用 LT 模型进行传播上，影响 V_k $\{\cup_{l=0}^{k-1} \Phi(l)\}$中的节点，即对于节点$v_i \in V_k$ $\{\cup_{l=0}^{k-1} \Phi(l)\}$，当满足$\Sigma_{w \in N_i(G_l)} b_{w,v_i} \geq \theta_{v_i}$时，$\Phi(l) = v_i \cup \Phi(l)$。同时，为了使模型更加合理，种子集在$G_k$中的影响力扩散范围被限制在 d 阶邻居内，即单次传播过程的结束时刻T_{end}小于等于 d。

动态网络中的 Top K 节点挖掘问题可以描述为如下优化问题[3]：

$$\arg \max_{\Phi(0),|\Phi(0)|=m} |\boldsymbol{\Phi(0)} \cup \{\cup_{l=1}^{K} \{v_i : v_i \in \{\{V_l \backslash \cup_{k=0}^{l-1} \Phi(k)\} \cap \{\Sigma_{w \in N_i(G_l)} b_{w,v_i} \geq \theta_{v_i}\}\} \}\}$$

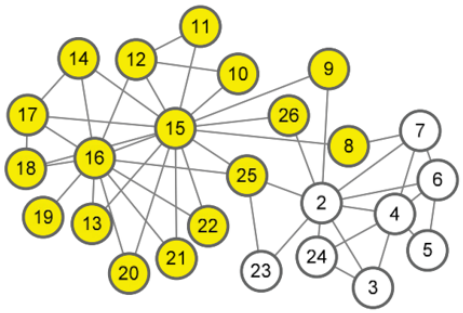
即：

$$\max \cup_{l=0}^{K} \Phi(l),\ \text{s.t.}\,|\Phi(0)| = \text{m}.$$

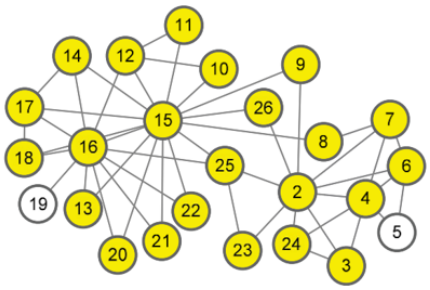
2.2 算法思想

2.2.1 距离控制思想

基于拓扑结构的算法通过图的静态拓扑结构和单个节点的属性以近似单个节点的影响力大小，但往往欠缺了对于多个种子节点共同作用的考量。度中心性(DC)算法作为一种基础的简单算法，经常地被用于解决 Top-K 影响力最大化问题，但在传播过程中，这些具有高度数的节点激活的节点往往大量重叠，简而言之，传播范围的重叠导致了“rich-club”效应，致使本应可以激活更多节点的影响力被浪费了。



(a) 选择degree



(b) 选择degree + distance control

如上例所示，假设种子节点集大小为 2，节点 15、16 拥有最大的度，根据传统的度中心性算法，应当选择节点 15、16 作为种子节点，但 15、16 的邻居高度重合，总影响力范围实则非常受限；而如果采取适当的距离控制策略，选择 4、16 作为种子节点，将能得到更大的影响力效果。而同样的情况，也发生在其他中心性算法上。在现实的社交网络中，“名人”(高中心性节点)与“名人”之间的相互认识，也是符合认知的。

基于如上思想，Dong Liu 等[4]于 2017 年提出了 local index rank 属性，即节点 v 在其邻居中度的排名：

$$\text{LI(v)} = \sum_{w \in N(v)} Q(d_w - d_v)\ ,Q(x) = 1\ when\ Q > 0, otherwise\ Q = 0$$

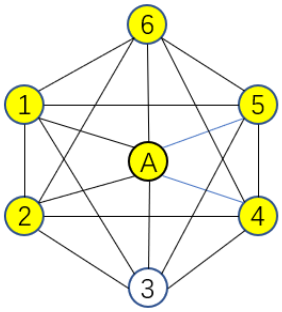
其中d_v, d_w为 v, w 的度。在$\text{LI} = 0$的节点中选择高度中心性点作为种子节点。

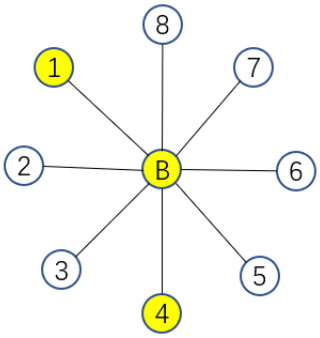
LIR 算法在规避 rich-club 效应，避免影响力过度的同时，但也带来了巨大的影响力不足，不论基

于抽象的传播模型，抑或是现实的社交网络，如上例中所示的，种子节点能够完全影响其邻居的情况都是几乎不可能发生，而严格的 LI=0 条件又导致许多高度数节点被放弃，实际运行时发现，影响力效果并不好(在许多图中，效果甚至不如 DC)。针对这一问题，需要引入一个更好的节点属性以保证其在 1 阶/2 阶子图内的影响力；另一方面，LIR 算法引入了 LIR 属性避免重复影响，本质上为$d = 1$的距离控制策略，其控制的距离大小，依旧存在调整和探索的空间。

2.2.2 聚类系数与度的结合

由对于 LIR 算法缺陷的分析，度中心性对于节点的影响力，尤其是局部的影响力大小难以准确的度量，因此，我们考虑了聚类系数与度的属性相结合。





对比节点 A，B，尽管 B 度数更高，但由于聚类系数小，邻居之间互不联系，影响力范围有限，而节点 A 尽管度数较小，但局部聚类系数高，邻居之间联系紧密，在 A 影响其邻居后，其邻居之间又相互影响，实现了较好的影响力。将聚类系数与度结合的思想，据我们所知，最早来自李梦甜[5]提出了 CLD(Cluster Local Degree Centrality)方法，她将聚类系数与邻居度之和进行结合：

$$\mathrm{CLD(v)} = (1 + \mathrm{Cluster}_v)\sum_{w \in N(v)} d_w$$

在复现实验过程中，我们发现，一方面，如上例所示，结合聚类系数的方法可以对节点在邻居中的影响力可以得到较好的保证，但对于邻居的邻居，效果便开始变得不稳定，因此我们对 CLD 方法改进为聚类系数与度的结合：

$$\mathrm{CLD2(v)} = (1 + \mathrm{Cluster}_v)d_v$$

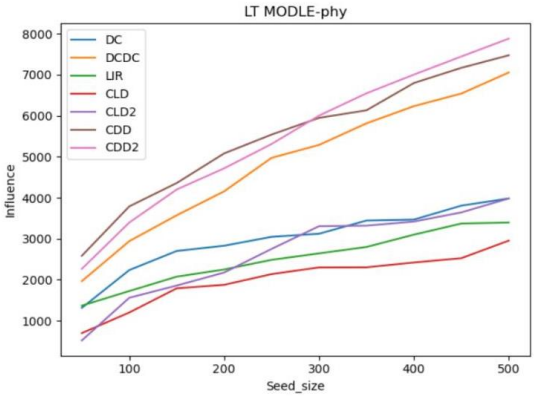
改进后，不仅算法代价降低，影响力效果也得到了提升。另一方面，影响范围重复的问题在 CLD 中心性中依旧存在，有必要引入距离控制策略加以改进。

2.3 算法设计

我们提出了基于混合策略的 CDD(Cluster local Degree centrality with Distance control)算法，在 CLD 和 CLD2 的基础上采用距离控制策略，设置参数最小距离 d，即种子节点集中任意两点间的距离大于等于最小距离：

$$\mathrm{d(v,w)} \geq \mathrm{d}, \mathrm{v}, \mathrm{w} \in \mathrm{S}$$

由于经过预实验，使用节点度的 CLD2 与 CDD2 都显著好于使用邻居度之和的 CLD 与 CDD 算法：



其中 DCDC 为距离控制策略的度中心性，距离控制$d = 3$. 因此，在下文中，CLD 与 CDD 所指，

均为效果更好的、使用度中心性的方法。

算法伪代码如下：

Algorithm – CDD

```
Input: Digraph G, int k, int d
S = G.nodes
for u in G.nodes:
    compute cdd2[u] <- (1+cluster[u])*u.degree
    # compute the cluster degree
for u in G.nodes:
    for v in G.neighbours(u,d): # the d-step neighbours of u
        if cdd2[v] > cdd2[u]:
            S.remove(u)
    # control the distance
S.sort(function = cdd2())
Return S[:k]
```

CDD 算法首先计算出各个点对应的 CDD 中心性大小，再在检查节点 v 的 d 阶邻居中是否存在 CDD 中心性更大的值，若有，则将该点剔除。最后在 S 集中根据 CDD 中心性排序，选择前 K 个节点作为种子节点。

2.4 算法优化

2.4.1. 基于幂律定律的剪枝优化

在社交网络中，节点的度分布往往符合幂律分布，合理地推断，影响力分布也应当符合幂律分布，因此，在挖掘影响力 Top-K 节点时，只需要对尾部结点（少数高度结点）进行分析和排序。

在适用 CDD 算法前，对 V 中结点根据度进行排序，并选择其中前 a*k 节点作为候选。由于种子节点集大小一般远小于网络中节点个数，算法复杂度由$O(nl^2)$变为$O(kl^2)$，速度大大提高。

2.4.2. 距离控制参数 d 的自适应优化

使用上述的距离控制算法，不难预见，当 k 足够大而接近 n 时，选出 k 个距离大于等于 d，不互为邻居的节点是不可能的，当经过剔除后，若$|S| < k$，则令$d = d - 1$，重新运行。而 LIR 算法的论文中并没有考虑到这一点。

2.5 动态网络与遗忘机制

在动态图中，离目前时刻越近的快照图节点得分能够给我们更多的信息；基于这样的考虑，应当给予离目前时刻越近的快照图的节点得分更大的权重。在我们的算法中，我们考虑以下三种较为简单的遗忘机制[3]：

线性遗忘机制：　$LF(v_i) = \Sigma_{l=1}^{K} l m_{v_i}^{l}$　　(1)

双曲线遗忘机制：　$HF(v_i) = \Sigma_{l=1}^{K} \frac{1}{K-l+1} m_{v_i}^{l}$　　(2)

指数遗忘机制：　$EF(v_i) = \Sigma_{l=1}^{K} \frac{1}{\exp(K-l+1)} m_{v_i}^{l}$　　(3)

使用以上三种方法来衡量初始快照$G_0 = G(V_0, E_0)$中节点的重要性，并对其排序。我们由高到低从中选取m个节点作为种子集$\Phi(0)$.

3. 实验

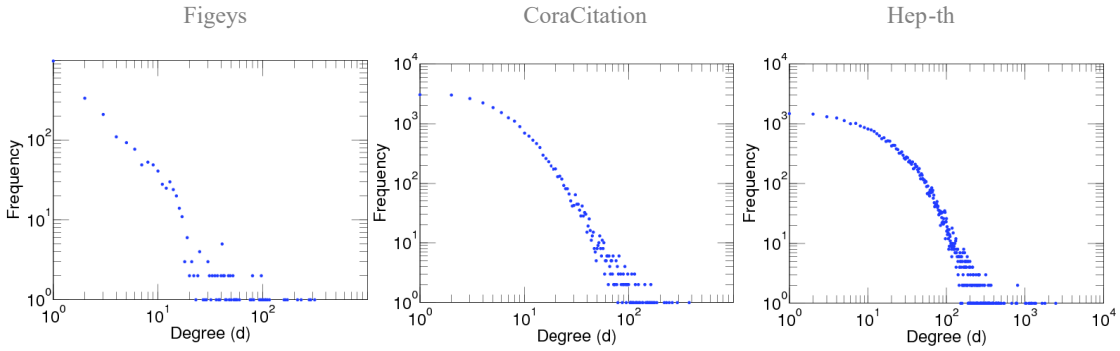
3.1 数据集获取与描述

本次实验使用的网路数据集均来自于与一个关于网络科学研究的大型网络数据集项目：KONECT (the Koblenz Network Collection). 数据均由该项目经过处理。其中，我们对于动态图中的一些时间戳混乱的数据进行了再处理，以使数据符合我们小组的实验要求。

静态网络：我们共选取了 3 张静态网络：Figeys, (phy)hep-th, CoraCitation。相关的网络信息如下表格与图片所示：

数据集	|V|	|E|	聚类系数	平均最短路径长度	直径
Figeys	2239	6452	0.761%	3.98	10
CoraCitation	23166	91500	11.7%	5.74	20
Hep-th	34546	421578	14.6%	4.27	14

度分布：



除了小型网络 Figeys 的聚类系数指标较低外，三张网络图属性与真实世界网络的属性基本相符。

动态网络：动态网络由于计算代价较大，且资源较少，仅选取了两张网络进行实验：

数据集	|V|	静态-|E|	动态-|E|	聚类系数	平均最短路径长度	时间跨度
Digg	30398	86404	87627	0.560%	4.68	08/10/28-08/11/13
Mathoverflow	24818	239,978	506,550	2.3%	3.57	2350 days

3.2 实验环境与参数选择

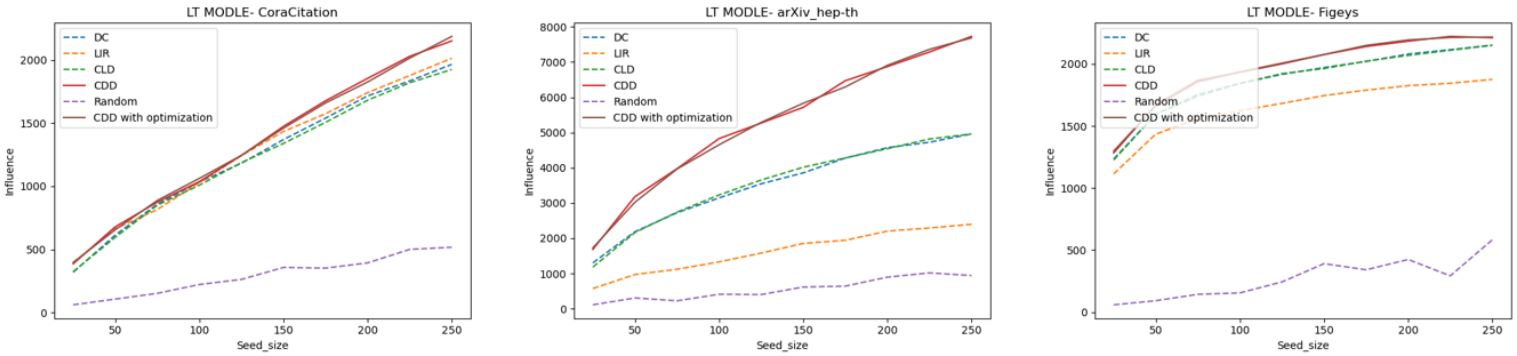
实验使用的语言为 Python 3.7，在操作系统为 Windows 10，CPU 为 Intel® Core™ i7-8750H CPU @ 2.20Ghz 的家用笔记本电脑上运行。

静态图中，我们选择为边随机赋权，距离控制策略选择$d = 3$，在传播过程中$iteration = 10$，即重复 10 次传播实验，取激活节点数的平均以度量影响力。在剪枝优化算法中，取$a = 10$.

在动态图中，我们选择将动态网络分为了 10 张快照，采取 TSN5 的标准模拟动态网络上的影响力传播过程，对于每一张快照，采用与静态图一致的标准。

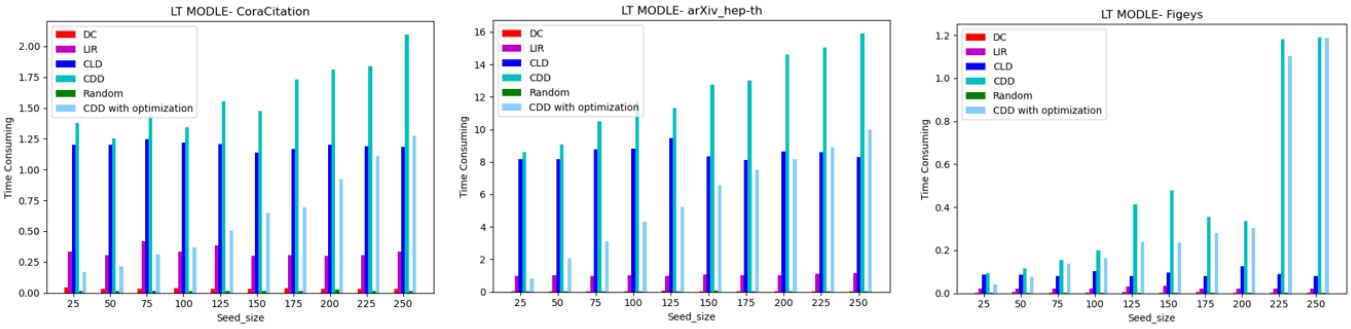
4. 结果

4.1 静态图结果与评估



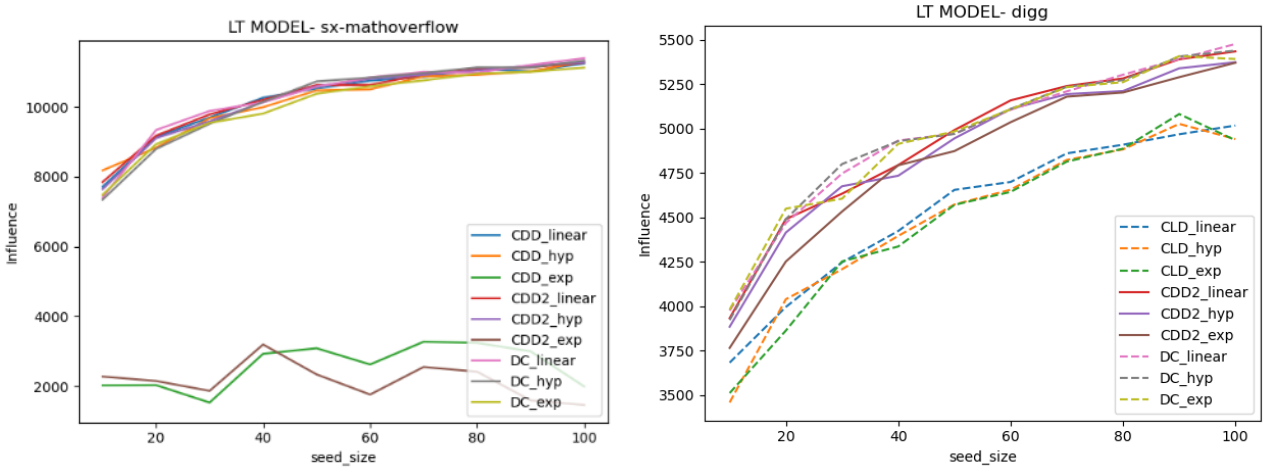
上图为在三张静态网络中，影响力大小随种子节点集大小增长的变化情况，算法 DC, LIR, CLD, CDD 如前所述，Random 为作为对照组的随机取种子节点，CDD with optimization 为经过剪枝优化的 CDD 算法。

在结果中可以看到，CDD 算法在各张图中都表现优异，在各个种子节点集大小中，影响力大小(激活的节点数)都显著高于用于比较的其他算法；CDD 算法与 CDD with optimization 的曲线几乎重合，剪枝优化几乎不会带来影响力下降。



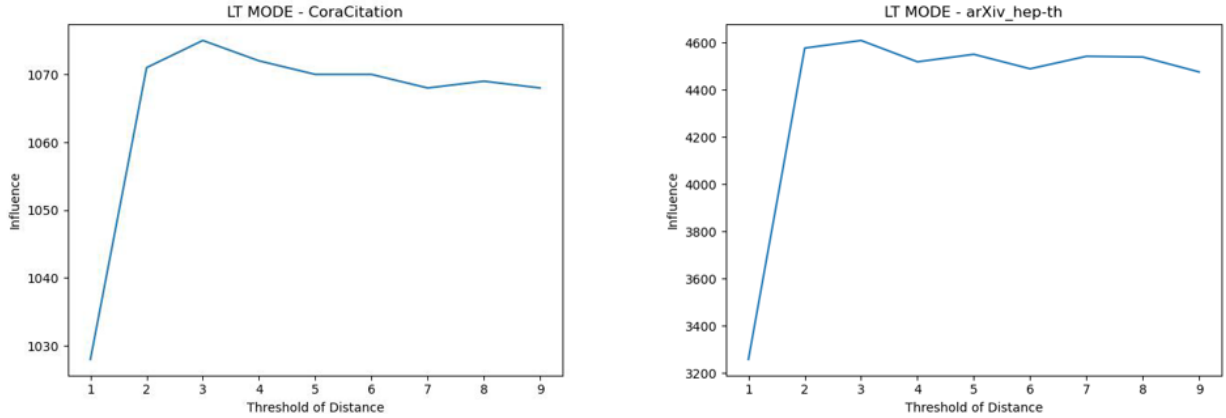
运行时间方面，如前所述，基于拓扑结构，尤其是基于度中心性的算法，计算代价非常小，对于边数 10^4-10^5 的网络可以在数秒到数十秒中完成，CDD 算法的速度自然也非常快。而在经过剪枝优化后，可以观察到时间代价的进一步减少，在 size k 较小时，加速尤为明显。

4.2 动态图结果与评估



在动态图中，可以观察到，权值随时间迅速下降的指数遗忘机制效果表现不佳，而在线性遗忘机制和双曲线遗忘机制中，CDD、CDD2、DC 算法都没有表现出明显的差异，但依旧可以看出 CDD 算法在动态图上也同样具有一定的健壮性，存在进一步研究的价值。

4.3 距离控制参数的影响



不考虑距离 d 的自适应优化，在size = 100时，观察参数 d 的变化对影响力效果的影响。观察到，采用距离控制与否对于结果影响极大；但随距离增大，影响力下降并不明显。因此，对于 CDD 算法，建议的参数 d 取值为 2 或 3。

5. 讨论和总结

5.1 优势与贡献

(1)据我们所知，我们首先提出了作为一种混合策略的 CDD 算法，实现了均优于两种纯策略的影响力效果，在实验中表现良好; (2)对 CDD 算法进行了剪枝优化；(3)将 CDD 算法移植到动态算法上；(4)CDD 算法只有一个超参数 d，且范围很小，易于调参，同时易于并行化

5.2 局限

(1)除了经验数值外，没有为参数选择策略提供理论基础。实际上，在特殊的高度密集网络中，拒绝使用距离控制策略可能是更好的；(2)在动态图的实验中，CDD 算法并没有展现出足够的优越性；(3)由于实验设备的限制，未能运行基于传播过程的算法，并与之比较。

5.3 静态网络与动态网络 Top-K 节点的异同

相同点：均基于网络的拓扑结构，估计节点，以及包含多个节点的节点集的影响力大小

不同点：(1)动态网络需要考虑网络的动态演变特性，对网络不同时间阶段的属性赋权进行综合评估；(2)动态网络关注未来一段过程的总激活数目；(3)动态网络由于是一种对未来演变的“预测”，传统算法效果的稳定性相对较低，可能引入机器学习等方法有助于解决这一问题。

小组分工：

罗子钦：文献查询，数据获取与处理、算法设计与测试、项目报告撰写

张志杰：文献查询，算法设计与测试，项目汇报，项目报告撰写

游涛：CDD 算法设计与测试，数据处理，项目报告撰写

6. 参考文献

[1] Richardson, M. & Domingos, P. Mining knowledge-sharing sites for viral marketing. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 61–70 (2002).

[2] Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2003 Aug 24–27; Washington, DC, USA; 2003. p. 137–46.

[3] R. Michalski, T. Kajdanowicz, P. Br´odka, and P. Kazienko. Seed selection for spread of influence in social networks: Temporal vs. static approach. New Generat. Comput., 32(3-4):213–235, 2014.

[4] Dong Liu, Yun Jing, Jing Zhao, Wenjun Wang, and Guojie Song. A fast and efficient algorithm for mining top-k nodes in complex networks. Scientific Reports, 7(1), 2017.

[5]李梦甜. 复杂网络中重要节点排序及影响力度量研究[D].兰州大学,2018:29-40

[6]李侃,张林,黄河燕.社会影响力分析——模型、方法和评价[J].Engineering,2018,4(01):90-104.

[7]胡怀雄. 基于独立级联模型的社交网络影响力最大化研究[D].深圳大学,2018:28-43.

[8] Chen W,WangY, Yang S. Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2009 Jun 28–Jul 1; Paris, France; 2009. p. 199–208.

[9] *PageRank* inWikiPedia, https://en.wikipedia.org/wiki/PageRank.