

社群发现模型

背景介绍:

在直播互动中，有黑产会来直播间进行刷人气、抢红包和抽奖等方式来薅平台羊毛。

- 目前线上的规则通常只能通过用户的行为是否正常，而不能发现用户与用户之间的团体关系。
- 黑产往往不是个体，而是团体，可能是采用多个真人也有可能是多个虚假账号。
- eg: 直播间黑产薅羊毛的行为: 多个用户刷直播间抢红包→用户送礼给指定的主播→主播提现; 因此我们希望能够利用图聚类算法来找出异常的团伙。

在黑产薅羊毛这个链路中，其中总共可以构建两个社区图：

- 在第1步抢红包：该批用户去不同直播间抢红包薅羊毛形成的**抢红包图**。（需要使用异构图，有2种以上的节点，有问题的团伙之间通常是没有连接的），这种现象也被称为lockstep行为模式，lockstep行为模式：在整个社区中找到的最致密的社区，该社区中具有一种同步的、大量的关联行为模式，这种行为模式称之为lockstep，而且往往比较契合现实场景中欺诈团伙的本质特点。 应用场景：例如电商刷单、团伙欺诈检测、团伙聚集性去干坏事的情况。-研究的方向:高密子图挖掘算法

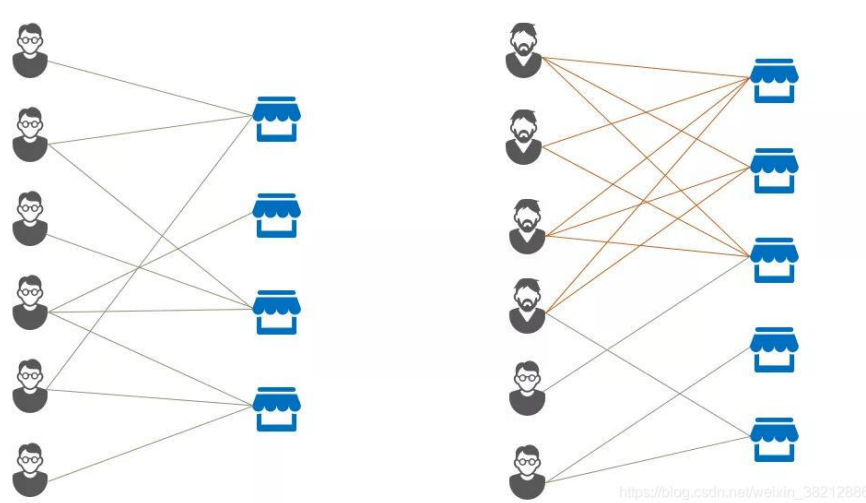


图1 正常行为 vs lockstep行为模式

- 在第2步送礼：该批用户将抢到的薯币送给主播形成的**送礼图**。（可以看成同构图，寻找密集社区，筛选有问题的社区）。-研究的方向:社区发现算法

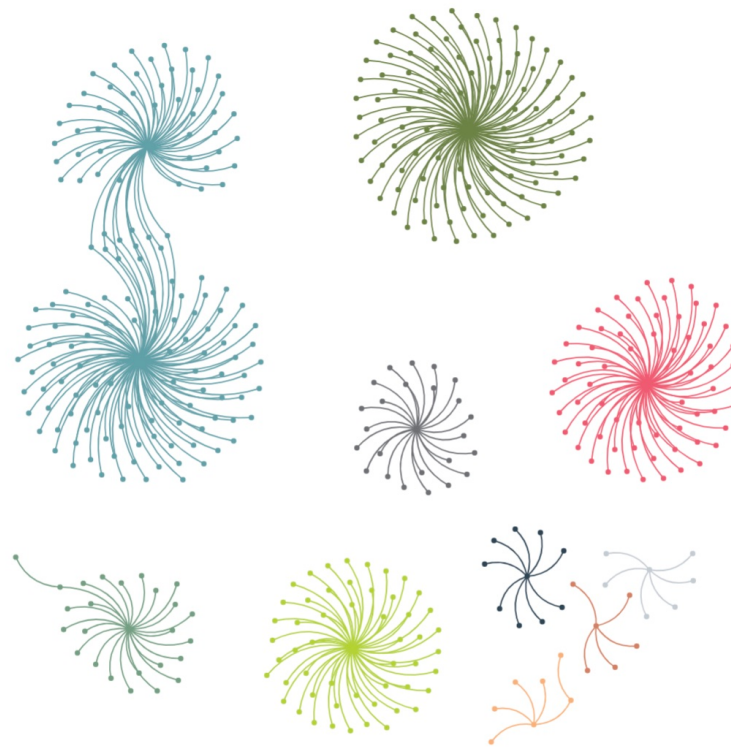


图2 社区聚类算法

社区发现算法vs高密子图挖掘算法的区别:

- 不同点:
 - 社区发现处理的对象: 同构图, 需要对整张图的节点进行社区划分迭代。
 - 高密子图处理的对象: 异构图(节点的类型不是单一的, 同类型节点之间一般没有边); 只关心最紧密的那个社区;
- 相同点:
 - 定义一个衡量密集度的指标, 启发式地不断优化这个值。

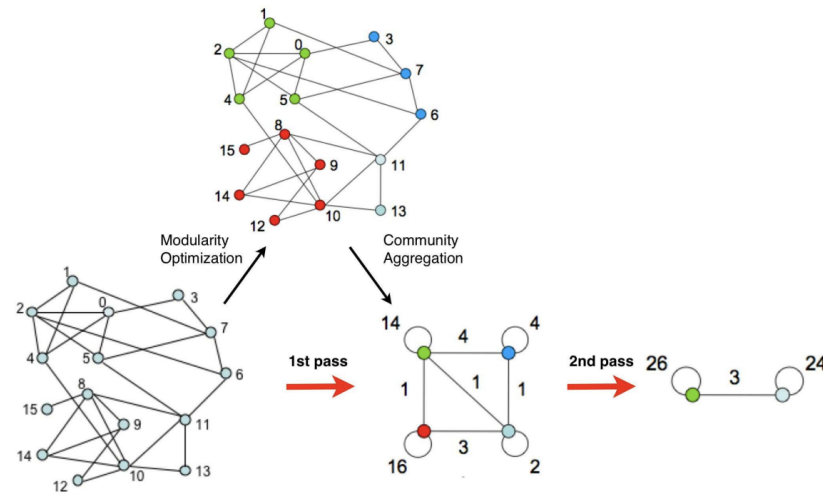
社区发现常用算法:

- louvain算法 → infomap算法

louvain算法流程:

模块度是评估一个社区网络划分好坏的度量方法, 它的物理含义是**社区内节点的连边的权重之和与随机情况下的连边的权重之和的差**, 它的取值范围是 $[1/2, 1)$ 。

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right]$$



- 初始化，将图中的每个节点看成一个独立的社区；
- 对每个节点，依次尝试把节点*i*分配到其邻居节点所在的这个社区计算分配之前与分配之后的模块度变化 ΔQ ，并记录 ΔQ 最大的社区，如果 $\text{Max} \Delta Q > 0$ ，则将该节点分配到该社区；
- 重复第二步，直到所有节点的所属社区不再变化。

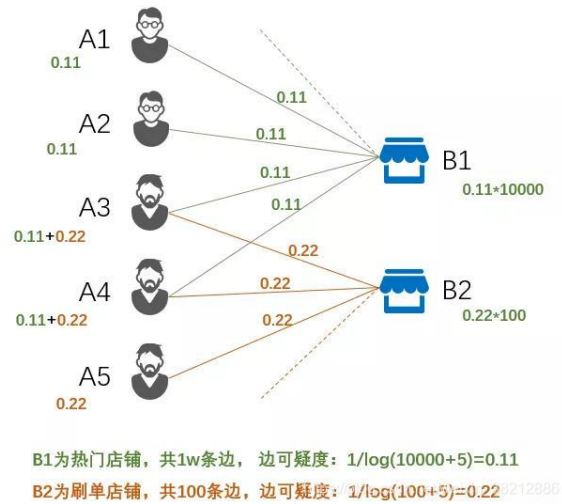
高密度图挖掘常用算法：

- fraudar算法

fraudar算法流程图：Fraudar定义全局度量 $G(S) = F(S)/|S| = (F(A) + F(B))/(|A| + |B|)$ ；其中 $F(\cdot)$ 是结点可疑度的总合， $|S|$ 是当前的结点数，所以 $G(s)$ 可以理解成网络结构中每个结点的平均可疑程度。 $G(\cdot)$ 用于表达结点的平均可疑度，在逐步贪心移除可疑度最小结点的迭代过程中，使 $G(\cdot)$ 达到最大的留存结点组成了可疑度最高的致密子图。

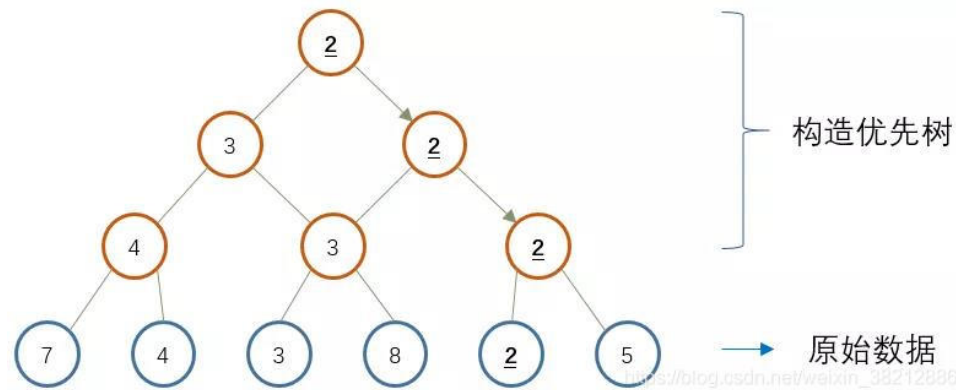
1. 定义全局度量

- 当网络中增加一个高可疑度的结点时，其带来的 $F(S)$ 增加百分比大于 $|s|$ 带来的增加， $G(S)$ 增大；
- 当增加一个低可疑度的结点时，其带来的 $F(S)$ 增加百分比小于 $|s|$ 带来的增加， $G(S)$ 减小；
- 结点总可疑度 $F(s) = F(A) + F(B)$ ，是两类结点可疑度的总和。而结点可疑度又是其所连接的边的可疑度的总和。
- Fraudar对边可疑度的定义准则是“与B类结点连接的边越多，其可疑程度越小，即根据连接数降权。
- 一个有效的边可疑度计算公式为 $1/\log(x+5)$ ，其中x是边的数量，实际含义是交易量越大的店铺，其交易可疑程度越小，因为大概率是真热门店铺。



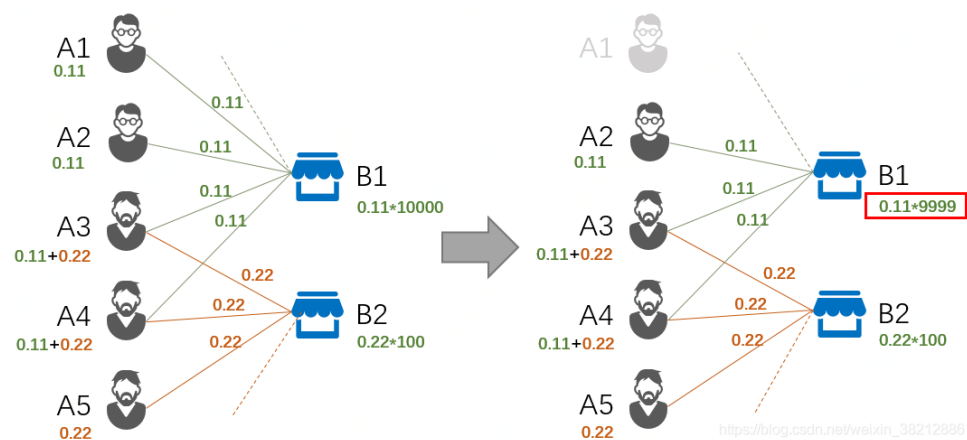
1. 结点的迭代移除

- 以空间换时间，为两类结点分别构建了用于快速搜索的优先树。优先树是一棵二叉树，以二部图中的A类或B类结点作为叶结点，并让父结点记录其子结点中的最小值。这样从根结点记录的全局最小值出发，可以快速定位到该最小值所对应的叶结点，然后将其从二部图中删除，并更新网络可疑度、更新优先树。



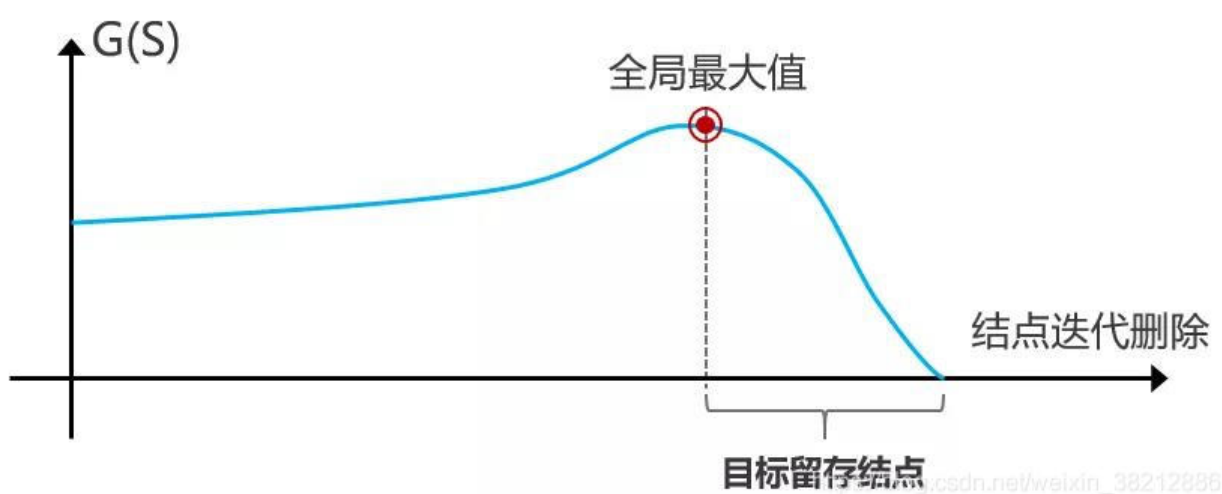
2. 结点的可疑度更新

- 迭代过程对结点的删除改变了网络结构，每次迭代后都需要在保持边可疑度不变的前提下，更新剩余有关结点的可疑度。迭代初始，A类结点中正常的低活消费者因为边连接少、可疑度低而被首先移除，如下图中的A1结点。同时B1商户因与A1关联的边连接被删除，其可疑度也被更新降低。如此往复，低可疑度的A类用户结点逐步减少、正常B类商户结点的可疑度也逐步降低而被删除，网络剩余结点的全局平均可疑度 $G(\cdot)$ 逐步增大，直到最后一个结点被移除而归为0



3. 回溯最大全局度量

- 全局平均可疑度 $G(S)$ 在结点的贪心迭代移除过程中会取得最大值，当全部结点迭代移除完毕时，回溯此过程中使 $G(\cdot)$ 达到最大的迭代，此时对应的留存结点即我们的目标结点，他们之间的关系网络是整个网络的最可疑致密子图。该子图中两类结点之间关系紧密，且都与外部结点连接相对稀疏。
- 全图的一次迭代可以获得一个高危群体，将该群体的结点从图中删除后再运行一次迭代，又可以获得另一个次高危群体，如此往复可以根据业务需要得到多个群体，再进行后续排查分析。



louvain算法 vs fraudar算法的对比:

算法名称	相同点	不同点	应用场景
louvain	定义一个衡量密集度的指标，启发式地不断优化这个值。	1. 应用在无向图中，属于无重叠的社区发现算法，整个社区迭代划分	社区内部关联紧密&&社区与外部关联稀疏

fraudar	2. 主要应用在二部图中，每次寻找是最紧密的那个社区	团伙内部基本无交互&&与整体与外部关联密集
---------	----------------------------	-----------------------

louvain算法花费的时间：

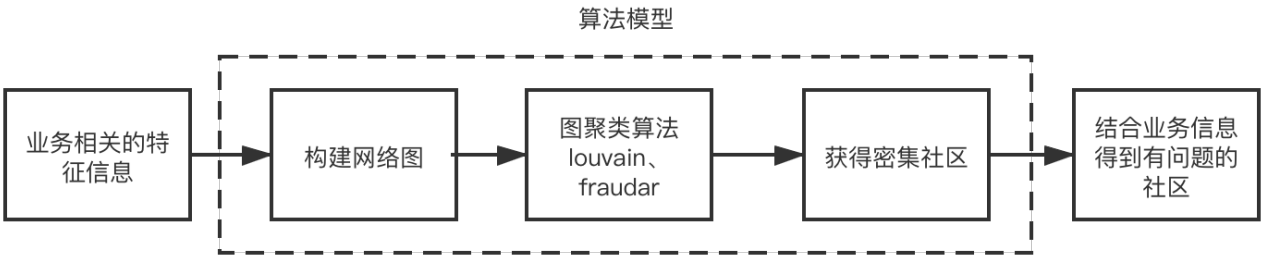
node/links	70K/351K	325K/1M	2. 6M/6. 3M	39M/783M	118M/1B
时间	1s	3s	134s	738s	154min

注:统计结果来自louvain论文。

业界风控相关的图算法：

相关团队	应用场景	使用的算法	解决的问题	达到效果	相关链接
度小满技术团队	信用贷款	Louvain算法	找出欺诈群体	结合用户的群体特征，能找出潜在的一些欺诈群体	https://mp.weixin.qq.com/s/u-Xosttbh-HgSWoH4Kqb6w
中国银联电子支付研究院	转账社区中找出异常转帐的群体	修正的TD-Louvain社团划分算法	找出异常转帐的群体	洗钱模式创新的TD-Louvain算法相对原先的Louvain算法算法有了明显的效果方面的提升。	https://zhuanlan.zhihu.com/p/105575602?utm_source=wechat_session
京东数科风控算法与技术团队	电商刷单	Fraudar算法	找出刷单团伙	-	https://blog.csdn.net/weixin_38212886/article/details/107912403

图聚类算法流程图：



讨论：

- 明确自己的业务场景，以及思考需要那种网络图结构。
- 高效利用业务信息来构建网络图和对应的权重，可以直接决定网络模型的性能。
- 抛砖引玉，听听大家的想法。

相关参考文献：

1. “FRAUDAR: Bounding Graph Fraud in the Face of Camouflage.” Hooi, Bryan , et al. the 22nd ACM SIGKDD International Conference ACM, 2016.

2. Blondel, Vincent D., et al. “Fast unfolding of communities in large networks.” *Journal of statistical mechanics: theory and experiment* 2008.10 (2008): P10008.