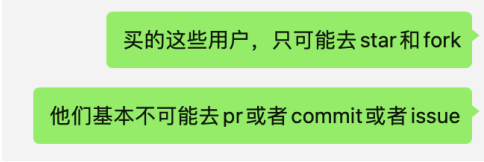
数据集构建初始思路

- CoLA中对公开数据集注入异常的比例，以Cora为例是5%

Dataset	♯ nodes	♯ edges	♯ attributes	♯ anomalies
BlogCatalog	5,196	171,743	8,189	300
Flickr	7,575	239,738	12,407	450
ACM	16,484	71,980	8,337	600
Cora	2,708	5,429	1,433	150
Citeseer	3,327	4,732	3,703	150
Pubmed	19,717	44,338	500	600
ogbn-arxiv	169,343	1,166,243	128	6000

- 罗列并使用Github user和user之间的所有的关系：用户之间一共4个关系，哪四个？follow star fork commit pr issue
- 正常社区构建
  - 从用户的角度出发，合理选取1个适配我自己问题定义具有代表性的流行仓库，基于此构建正常社区
  - 选取途径：open leaderboard，但是数量级太大
  - 比如，结合官方指数和clickhouse数据库筛选出2022年具有代表性，且star、fork、commit等行为都均匀分布的一个仓库，在2022年内
  - **理论依据：选取1个仓库和选取多个仓库是没有区别的，选取的这个仓库仅仅作为一个支撑点，只是为了挖掘存在真实行为的GitHub用户**
- 异常社区构建
  - 引用前人的研究思路，购买157个内部众包异常群体，挖掘基于此构建1个异常社区

    - **需要对提供淘宝异常用户的卖家做出合理要求**
        1. 保证这些用户不同
        2. 这些用户之间尽可能地存在follow关系
        3. 这些用户是正常用户吗？还是说只是为了刷赞存在？他们除了star、fork、watch这些点击一下就能完成的操作，还可以进行issue、pr、commit这些实际开发者才会完成的操作吗？
            - 还是说他们平时也是程序员，但是为了赚外快进行的推广服务？
        4. 我需要尽可能正常且活跃的开发者账户，他们可以在2022年内存在其他的fork、commit、issue事件吗？
        5. 注意时间范围要在2022年内
- 社区内部的邻接关系，附带其中用户的特质说明
    - 给不同的行为赋予不同的权重，这些数据既作为用户的特征，又作为邻接矩阵权重来源，可以基于花费金钱的多少进行理论支撑，也可以说基于直觉
    - 正常社区
        - 共同star、共同fork、共同watch...这些用户之间必然存在交集
        - 挖掘群体内部的follow关系
    - 异常社区
        - 共同star、共同fork、共同watch...这些用户之间必然存在交集
        - 挖掘群体内部的follow关系

    - 

基于这个设定去定义正常用户其实是不准确的，因为购买的异常用户也可能是正常的开发者，所以还是按照上述步骤进行
- 社区之间的邻接关系，**需要将正常社区和异常社区的关系进行融合**，模拟一个现实社区
    - 比如将正常群体内和异常群体内的边进行整体一个整体的正态分布，随机地将正常社区和异常社区内的用户进行边的连接
    - 我的邻接矩阵是一个带权矩阵，这个权重基于什么也是由我设定
- demo
    - 构建的一个正常仓库有10个人，这10个人都是正常的开发者；构建的1个异常仓库有3个人，这3个人是伪装的正常用户？(是否伪装和具体怎么伪装有待考证)
    - 定义的正常用户和异常用户都有不同的特征，目测的话是相似的，他们在2022年都有各种行为
    - 所以如同之前所说，选取的1个正常且活跃仓库和1个故意创建的异常仓库，只是作为用户的邻接关系支撑，选更多仓库，参照的意义其实也一样
- maybe思路都是要基于对照的思维

思路参考

> 大论文

- 概要：1.GitHub社区网络简介 2.获取原始数据的途径和工具 3.数据集节点、边的定义 4.**选取三个阈值参数**缩减数据集规模、**提高连通性**(使用到了超图) 5.**基于三个阈值参数进行对照实验**，对比**保留的信息量**
- 数据获取方式：GH Archive是针对github的开源仓库，X-lab储存于clickhouse中的数据也是基于GH Archive的
- (多看点github研究文献)作弊用户和正常用户区别不大，所以不能以特征去定义正常和异常用户

  为特征等方面与普通用户相差无几。先前的研究[10]中提出，作弊用户通常具备的与普通用户差异较大的特征有：无组织信息，Star 操作数目远高于 Fork 操作数目等。但在本文的研究中发现，如今的作弊用户大多并不具备上述的特征。因此，很难直接通过观察来确定一个项目或用户是否存在作弊行为。
- 蜜罐项目：购买一批用户，然后观察异常用户
- 节点是项目，边是项目之间的相似度
- 项目特征选取：本身属性(编程语言、star、fork等)+统计学特征(事件产生的频率、方差、变化程度)

> 小论文

- github用户和存储库之间存在14种类型的事件，仓库的watchevent和forkevent对应star和fork，是关注的重点
- 作弊仓库的Watchevents和Forkevents频率远高于正常仓库，重点关注这两种事件类型，并选择它们的比例、平均值和变化程度作为库的特征。此外，短时间内产生大量的Watchevents和Forkevents也是需要考虑的。时间间隔设置有不同的粒度：1 分钟、1 小时和 1 天。如下图，为每个节点选择了 28 个特征。其中，平均值可以反映数据的规模，方差和变异系数可以反映数据的变化程度

Feature	Meaning
N_{total}	Number of Star and Fork operations
R_{wf}	Ratio of Star and Fork operations in all operations
$N_i(i \in \{1 \cdots 14\})$	Numbers of different GitHub events
$M_{Avg}, M_{Var}, M_{CoV}$	Average, Variation and Coefficient of variation of Star and Fork operations in one minute
$H_{Avg}, H_{Var}, H_{CoV}$	Average, Variation and Coefficient of variation of Star and Fork operations in one hour
$D_{Avg}, D_{Var}, D_{CoV}$	Average, Variation and Coefficient of variation of Star and Fork operations in one day
N_p	Event number of Star and Fork operations by fraudulent users
$R_p = \frac{N_p}{N_{total}}$	Ratio of S star and Fork operations by fraudulent users in all operations

- **购买异常用户，然后跟踪获取异常仓库，作为异常样本，另外选取流行的存储库作为负样本**
- 数据集缩减思路：对clickhouse数据库中的数据，只保留操作次数在一定阈值内的仓库和用户
- 存在的问题
  - 正样本的定义：通过给你提供刷星服务的账号，去看他们和哪些仓库有联系，然后像这里说的，按事件发生次数排序，就认定次数多的仓库，就是异常仓库吗？然后你最终就得到了200个异常仓库
  - 元路径：项目之间的相似度就是基于被用户的star和fork，两个仓库相似度越高，就越被相同的用户star和fork，以此来构建邻接矩阵