GitHub 开发者影响力分析

费锡通 孔思萱

一. 引言

近年来，随着互联网行业的飞速发展，开发者成为被社会关注的热门群体之一，开发者社交网络应运而生。GitHub 作为目前最大的 git 仓库托管商，吸引了众多开发者的目光，许多开发者使用该平台进行开源项目的托管，议题追踪和代码评审……

而众多的开发者和项目使得用户难以快速且直接寻找到其中高水平，高热度，高价值的开发者。因此，我们通过分析 GitHub 用户项目以及各种交互数据，衡量开发者的影响力，找到具有较高价值的用户。

二. 问题描述

本次实验针对 GitHub 用户数据和日志数据，通过三种不同算法对数据进行处理计算和分析，量化获得不同用户的影响力值，从而发现高价值用户。

用到的 Github 数据集中主要有 ods 层的两张表 ods_github_log 和 ods_github_users 和 dim 层 dim_github_actor，ods_github_log 为 2015 年至今的全部日志数据，ods_github_users 每周过去七天全域产生日志数据量较多的账号的用户信息，dim_github_actor 的作用是根据用户 id 查询到用户名从而可以访问其 github 主页。

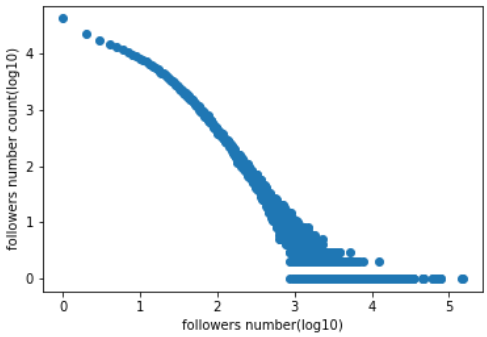
1.followers 和 following 数量的幂律分布分析

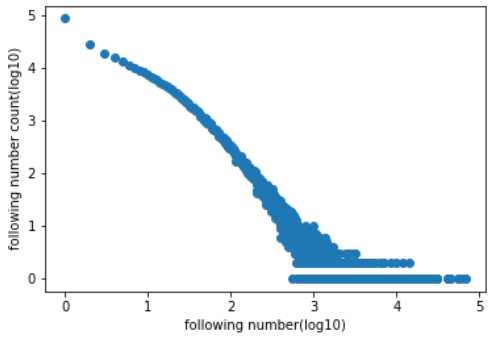
ods_github_users 共有 618242 条数据，去重后剩下 373093 条数据。其中有用的数据主要是 followers 和 following，followers 表示该用户被那些人关注，following 表示该用户关注了哪些人，统计其中的用户 followers 和 following 数量。

	mean	std	min	25%	50%	75%	max
following count	44.89	460.49	0	1	7	26	67221
followers count	85.93	732.24	0	3	14	45	1488982

绘制散点图，横坐标为用户有多少 followers 或 following，纵坐标表示有某个数量的 followers 或 following 的用户人数，为了显示清楚，横纵坐标均取 log10，其中横坐

标的值先加了 1，防止 log0 的出现。比如左图左上角的点表示约有 10 的 4.6 次方（即 40 000 左右）的用户没有任何人关注。





用户的 followers 数量和 following 数量基本符合幂律分布，即分布函数符合$y = cx^{-\alpha}$的形式,因为对 x 和 y 都去了对数，所以是一个斜率为负数的线性函数形式,即$logy = -\alpha logx + logc$。

2. 网络的稀疏性分析

随机挑选 200 条用户的数据，根据 follow 的关系绘制网络链接图，一个点表示一个用户，两个点之间有变表示用户之间存在 follow 关系，随机挑选的 200 条数据中共有 3852 个点，4248 条边。

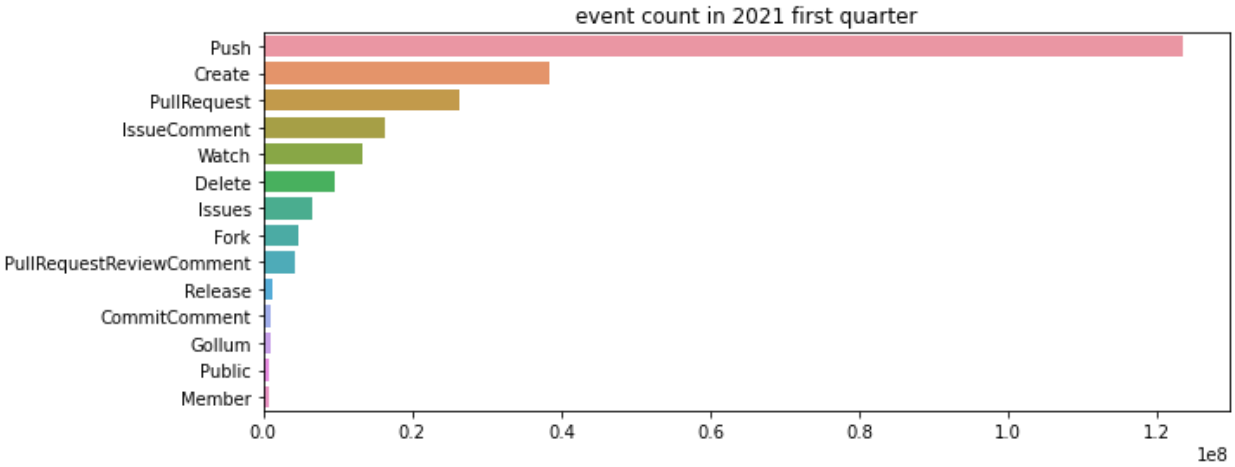
3. 日志数据分析

统计每一年的 log 数据量

year	2015	2016	2017	2018	2019	2020	2021
count (million)	212.22	320.07	412.94	479.18	605.53	863.41	998.56

因为 log 数据量太大，后续的分析和计算都是基于 2021 年的。其中分析时用的是 2021 年第一季度的数据，计算时会做数据清洗，所以用的是 2021 年全部的数据。

查看 2021 年第一季度各种日志事件的数量。

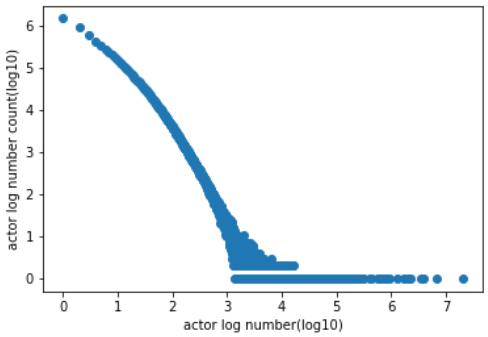


4. 日志数量的幂律分布分析

统计 2021 年第一季度各个用户的日志数量

mean	std	min	25%	50%	75%	max
34	8.2	1	2	5	17	1991606

绘制散点图，横坐标为用户产生日志的数量，纵坐标表示产生某个数量日志开发者人数，为了显示清楚，横纵坐标均取 log10，因为是从日志数据中塞选的用户，所以这部分数据里面所有用户都至少有一次日志产生，横坐标不加 1 也不会有 log0 出现。比如左图左上角的点表示约有 10 的 6 次方（即 100000 左右）的用户在 2021 年第一季度产生了一条日志。



三．方法

根据数据的特征，采用 3 种方法来分析开发者的影响力，分别为传统的网络链接算法 PageRank 和 HITS，以及考虑了用户与代码仓库链接的 BurstBiRank。先给出 PageRank 和 HITS 的算法步骤，再分析 BurstBiRank 相较于这两个的优势，最后给出 BurstBiRank 的算法步骤。

1. PageRank

算法基于以下假设:

- 如果一个网页被很多其他网页所指向，那么说明这个网页比较重要;
- 如果一个网页被重要网页所指向，那么说明这个网页比较重要。

因为用户的影响力也符合类似假设，所以可以尝试用 PageRank 算法计算用户影响力并排序:

- 如果一个用户被很多其他用户 follow 了，那么说明这个用户比较有影响力;
- 如果一个用户被影响力大的用户 follow 了，那么说明这个用户比较有影响力。

节点影响力更新公式如下

$$rank(u) = \frac{1-q}{N} + q * \sum_{(v_i,u)\in E} \frac{rank(v_i)}{N(v)}$$

输入: 用户 Follow 图 G = <V,E>

步骤

（1） 初始化所有节点的 rank 值为 1

（2） 遍历每一个结点，更新 rank 值

（3） 重复步骤(2)，直到收敛

（4） 输出每个节点的 rank 值

2. HITS

HITS 算法基于以下两个假设

- 一个高质量的 Authority 页面会被很多高质量的 Hub 页面所指向;
- 一个高质量的 Hub 页面会指向很多高质量的 Authority 页面。

用户的影响力也可以符合类似假设，Authority 用户是开发能力较强的用户，Hub 用户是喜欢关注 Authority 用户的用户。

- 一个影响力强的 Authority 用户会被很多影响力强的 Hub 用户所 follow;
- 一个影响力强的 Hub 用户会 follow 很多影响力强的 Authority 用户。

节点影响力更新公式如下

$$a(u) = \sum_{(v_i,u)\in E} h(v_i)$$

$$h(u) = \sum_{(w_i,u)\in E} a(w_i)$$

输入： 用户 Follow 图 G = <V,E>

步骤

（1） 初始化：将各节点的 a 值和 h 值均设为 1

（2） 遍历每一个结点，更新节点的 a 值

（3） 遍历每一个结点，更新节点的 h 值

（4） 将 a 值和 h 值归一化

（5） 重复 2-4 步骤，直至最终收敛

（6） 输出每个节点的 a 值和 h 值

3. BurstBiRank

BurstBiRank 思想：

（1）Burstiness，突发性。在许多现实世界或在线系统中，人们的活动通常是间歇性的，在短时间内表现出强烈的活动，然后是长时间的活动减少，甚至没有活动。突发性度量被提出来衡量行为偏离周期性行为的程度，$B = \frac{\sigma_t - m_t}{\sigma_t + m_t}$，其中$\sigma_t$和$m_t$分别代表用户活动时间间隔的标准差和平均值。B 的取值范围为-1 到 1，B 大于 0 表示用户的行为是突发的，越接近 1 突发性越强；B 小于 0 表示用户的行为是频发的，越接近-1 频繁性越强。

（2）Bipartite，两部分组成的$\mathrm{G} = (\mathrm{U} \cup \mathrm{P}, \mathrm{E})$,考虑到了开发者和项目的交互行为。用矩阵$\mathrm{W}\left(\in \mathrm{R}^{|\mathrm{U}|\times|\mathrm{P}|}\right)$描述这个图，$\mathrm{w_{ij}}$表示开发者i和项目j之间关联强度。用突发性初始化权重矩阵$W_{ij} = f(B_{ij}) = -B_{ij} + 1$，所以突发的行为的初始化权重接近 0，而频繁的用户行为则是大于 1 的，跟倾向于将经常经常与项目有交互的用户 rank 值提高。

输入：用户与项目的二部图

步骤：

（1） 数据预处理，塞选出一年 pr 数量大于 20 的用户和一年 pr 数量大于 200 的仓库

（2） 计算一个项目与一个仓库的突发性 B

（3） 用突发性初始化权重 W

（4） 调用 birank 迭代计算 rank 值

（5） 输出用户的 rank 值和仓库的 rank 值

四．评价

1. 相关性分析

提取出 PageRank、HITS、BurstBiRank

	PageRank&HITS	PageRank&BurstBiRank	HITS&BurstBiRank
Pearson 相关系数	0.7682	-0.0016	-0.0043
显著性检验 p 值	0	0.6003	0.1557

Pearson 相关系数表示两组数据的相关性，大于 0 表示正相关，小于 0 表示，在 0 附近表示没有线性相关关系。显著性检验的 p 值表示原假设(即两组数据不相关)成立的条件下，得到观测数据的概率，p 值越小，拒绝原假设的概率越大。如果取显著性水平$\alpha = 0.05$，那么我们可以得出结论，PageRank 和 HITS 是强相关的，且结论是显著的；PageRank 和 BurstBiRank、HITS 和 BurstBiRank 是不相关的，但是结论并不显著。

这是因为 PageRank 和 HITS 都是只用了用户 follow 的网络，而没有考虑用户与代码仓库的交互情况。而 BurstBiRank 不仅考虑了 follow 网络，而且根据用户对代码仓库做出 pr 的行为提取特征作为权重的初始化，所以与前两中方法几乎不相关。

2. 案例分析

选出 PageRank 值排名前 5 的用户，他们的用户名分别为分别为 torvalds、JakeWharton、ruanyf、yyx990803、gaearon,定位到 github 主页。而 HITS 的排名权威值前 5 用户中有 4 位与 PageRank 的结果相同，另一个人是 sindresorhus。尽管在网页的链接分析中 PageRank 和 HITS 有不同的功能和作用，HITS 算法是与用户输入的查询请求密切相关，PageRank 与查询请求无关，但是因为本次实验数据是静态的，而且前面验证了

PageRank 和 HITS 的强相关性，排名的结果也类似，所以后续的分析仅用 PageRank 的结果与 BurstBiRank 的结果相比较。

选出 BurstBiRank 的结果中 rank 值前 5 的用户，发现都是机器人，而前 10 也都是机器人、前 50 中有 44%的机器人、前 100 中有 28%的机器人。将用户名中带有 bot 的机器人全都去掉后，选出排名前五的人，他们的用户名分别为 wannte、YounghoonKwon、TaewanKimmmm、PapimonLikelion、knae11，进入他们的 github 主页，发现他们都不是 PageRank 值高的那种名声在外的用户，他们未必有很多的 followers，但是他们对很多代码仓库做出了比较高的贡献。

比较 PageRank 排名前 5 的用户在 BurstBiRank 中的排名位置，发现他们虽然排名下降了很多，但还是在前 3%左右。说明 BurstBiRank 会倾向于把有很多 followers 但是很少提交 pr 的用户的排名降低。

用户 id	49307266	28701943	50273712	61370901	66905013
PageRank 排名	1	2	3	4	5
BurstBiRank 排名百分比	3.8711	1.4807	1.3291	3.8038	2.8259

比较 BurstBiRank 前 5 的用户在 PageRank 中的排名。其中-1 表示该用户不在 PageRank 的排名中，因为他们不在 ods_github_users 表中。可以看出 BurstBiRank 排名靠前的用户在 PageRank 中的排名也相对靠前，说明 BurstBiRank 排名出的用户还是从 follow 网络来看也是具有一定影响力的。

用户 id	1024025	66577	905434	499550	810438
BurstBiRank 排名	1	2	3	4	5
PageRank 排名百分比	-1	16.4579	-1	7.3521	17.3430

3. 稳定性分析

考虑到会有用户恶意地通过增加 follow 人数或者给一写仓库提交 pr 来提升自己的排名，所以选择了排名分别位于 20%、40%、60%、80%的用户，给他们随机添加 follow，随机往仓库提交 pr（因为提交 pr 有时间属性，为了模拟现实的情况，这些 pr 都集中在一个月的时间中）。从表中可以看出 PageRank 排名越靠后的，通过作弊提升的排名越高，

而 BurstBirank 的排名通过作弊都只提升了很少的百分比。这是因为即使通过作弊添加 pr，这些操作的时间比较接近，所以突发性 B 就接近 1，初始化后对应的权重就接近 0，所以并没有排名的明显提升。

原排名百分比	20	40	60	80
作弊后 PageRank 排名百分比	19.9268	34.2109	39.9983	56.8092
作弊后 BurstBiRank 排名百分比	19.9998	39.9997	59.9995	79.9994

五. 相关工作

1.数据集的下载和处理

GitHub 数据集过大，很难将其全部下载后使用，尝试多种方法，最终选择选取 2021 年一年的数据作为代表进行实验。

2.对本次试验选择的算法进行了解和调研

PageRank 算法是一个链接分析算法，是用来衡量搜索引擎搜索到的结果网页重要程度并进行排序的一种算法。PageRank 算法通过计算网络页面的链接数量和质量来确定网站的重要性[1]。

HITS 算法与 PageRank 算法思想相近，不过每个页面被赋予两个属性：Hub 值和 Authority 值。根据页面两个属性值的大小，网页可以被分为 Hub 页面和 Authority 页面。Hub 页面是指包含了很多指向 Authority 页面的链接的页面；Authority 页面是指包含了较多实质性内容的页面[2]。

PageRank 和 HITS 都只考虑了用户的 follow 网络，没有考虑用户和项目的行为，而 BurstBiRank 算法加入了对用户活动的突发性以及用户和项目的交互行为的考察，更全面更有说服力[3]。

3.基于不同网络对用户影响力的分析

数据集中数据种类丰富，若不同影响因素一概而论或片面使用可能导致算法的表现差，因此分别基于 follow 网络、pr 网络进行分析[4]。

六. 结论

在信息爆炸的时代，数据分析与研究是赋予数据更大价值和意义必不可少且至关重要的。GitHub 作为开发者社交网络平台，其庞大的用户数据集成为一大重要宝藏，通过适当的方法对其进行分析和利用是一项极具意义的工作。GitHub 数据集与普通数据集不同，其数据量庞大，网络关系复杂，涉及个体多样，本次试验通过提取用户和项目特征并进行分析，然后完成用户影响力分析。通过方法的比较，验证了 PageRank 和 HITS 在用户影响力排名中的相关性，BurstBiRank 相较于 PageRank 和 HITS 更关注项目而不是用户的 follow 关系以及其稳定性。

参考文献

[1]韩忠明,陈炎,刘雯,等. 社会网络节点影响力分析研究[J]. 软件学报,2017,28(1):84-104.

[2]喻依,甘若迅,樊锁海,刘庆,邵晴.基于 PageRank 算法和 HITS 算法的期刊评价研究[J].计算机科学,2014,41(S1):110-113.

[3]Yan, D., Shao, Z., Zhang, Y. & Qi, B. 2020, "BurstBiRank: Co-Ranking Developers and Projects in GitHub with Complex Network Structures and Bursty Interactions", *Complexity (New York, N.Y.)*, vol. 2020.

[4]王姗姗. GitHub 用户数据分析与研究[D].大连理工大学,2018.

附录：分工

费锡通：相关工作调研，数据处理，计算相关代码实现，结果分析，报告编写。

孔思萱：相关工作调研，文献查阅，分析相关代码实现，报告编写。

附录：进度

4-6 周：确定选题，相关工作调研

7-10 周：数据下载及预处理，完成 PageRank 和 HITS 的计算

11-13 周：BurstBiRank 的实现，三种方法结果的分析评价

14 周：编写报告

附录：项目地址

项目的 github 链接：https://github.com/fxt2021/social-computing

因为数据及结果文件过大无法 push 到 github 仓库，所以完整的代码、数据及结果保存在百度网盘中：链接：https://pan.baidu.com/s/1co6FR7HWJqy9QnqZ2U2-cA?pwd=f75J

提取码：f75J