GitHub开发者影响力分析

费锡通 孔思萱

1. 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。

1. 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。

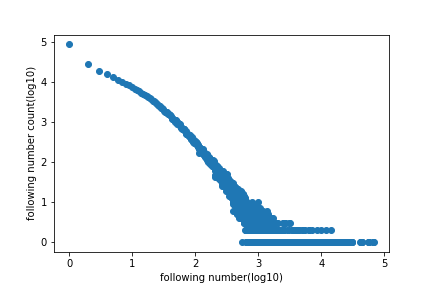
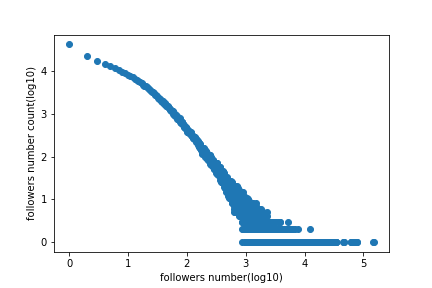
用到的Github数据集中主要有ods层的两张表ods\_github\_log和ods\_github\_users和dim层dim\_github\_actor, ods\_github\_log为2015年至今的全部日志数据，ods\_github\_users每周过去七天全域产生日志数据量较多的账号的用户信息, dim\_github\_actor的作用是根据用户id查询到用户名从而可以访问其github主页。

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 |

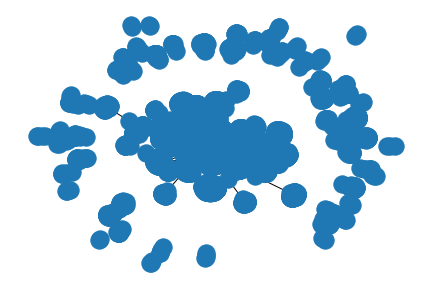
绘制散点图，横坐标为用户有多少followers或following，纵坐标表示有某个数量的followers或following的用户人数，为了显示清楚，横纵坐标均取log10，其中横坐标的值先加了1，防止log0的出现。比如左图左上角的点表示约有10的4.6次方（即40000左右）的用户没有任何人关注。



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。

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。



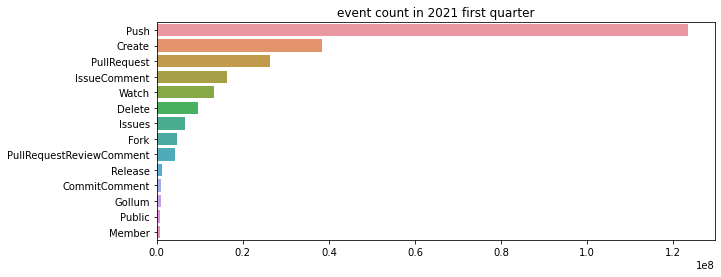
3.日志数据分析

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。

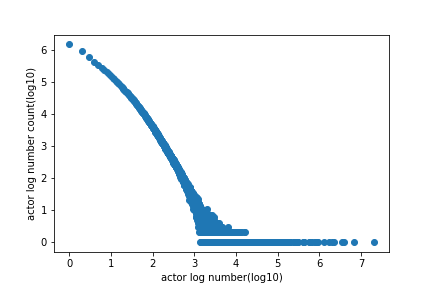


4.日志数量的幂律分布分析

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。



1. 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

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

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，，其中和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,考虑到了开发者和项目的交互行为。用矩阵描述这个图，表示开发者和项目之间关联强度。用突发性初始化权重矩阵，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
6. 评价
7. 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，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。

1. 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 |

1. 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 |

1. 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]。

1. 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/sixuanKong/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