

面向 GitHub 开源社区的开源软件流行度分析研究

杨柳 刘柳皓 欧阳海飞 郑美光 龙军 廖志芳

(中南大学计算机学院 长沙 410083)

(zfliao@csu.edu.cn)

Popularity Analysis of Open Source Software on GitHub

Yang Liu, Liu Liuhao, Ouyang Haifei, Zheng Meiguang, Long Jun, and Liao Zhifang

(School of Computer Science and Engineering, Central South University, Changsha 410083)

Abstract The popularity of open source software (OSS) is important for OSS quality assessment and OSS recommendations. This paper presents the influence factors of OSS on GitHub. Nine attribute factors for GitHub are analyzed in this paper, including *watches*, *stars*, *forks*, *questions*, *submissions*, *age*, *size*, *release*, and *contributors*, and then the concept model of OSS popularity is built. According to user behavior attributes and other attributes, a popularity correlation analysis method based on Spearman rank correlation coefficient test is proposed in this paper. From the correlation statistics analysis between user behavior attributes (*watch*, *star*, *fork*) and other attributes, the attribute of *watch* is an efficient measure for the popularity of OSS on GitHub.

Key words open source software; software popularity; popularity analysis; GitHub

摘要 开源软件流行度对于开源软件社区的质量评估与开源软件推荐具有重要意义。本文研究了 GitHub 上开源软件流行度影响因子, 分析了 GitHub 中的 *watch*、*star*、*fork*、*issue*、*commit*、*age*、*size*、*release*、*contributor* 九个属性因子, 构建了开源软件流行度概念模型。基于用户行为属性与其他维度属性, 提出了基于斯皮尔曼等级相关系数检验的流行度相关性分析方法, 通过基于用户行为属性 (*watch*、*star*、*fork*) 与其他属性之间的相关性分析, 统计分析结果表明 *watch* 属性是衡量 GitHub 开源软件流行度的度量因子。

关键词 开源软件; 软件流行度; 流行度分析; GitHub

1 引言

开源软件^[1] (Open Source Software, OSS) 指的是用户或者组织机构的源代码能够被公众免费开放使用的软件, 也就是开放源代码, 并且用户可以对这些源代码免费使用、修改以及不受限制的发布。截止 2018 年, GitHub^[2] 已成为世界上最大的开源软件集

合, 拥有大约 3100 万的用户和 9600 万个存储库。随着越来越多的开发者参与到开源社区研究中, 开源软件的流行度成为了用户或是企业越来越关注度的焦点, 想参与的潜在开发者或是已经参与其中的开发人员等都关注着项目存储库的流行度, 他们想知道自己的存储库创建后是否会受到关注, 也想

解自己参与的项目是否是值得参与和贡献的流行项目。因此,对于开源软件流行度的评估及预测显得尤为重要。

当前已经提出来评估开源软件流行度的模型有多种,但是这类模型的基本功能主要针对开源软件项目的单个属性特征,如代码行、软件文档,然后通过分析从开源社区上抓取到的大量数据集来度量开源软件的流行度和成功性。由于这类模型忽略了影响软件开发过程的属性与属性之间的相关关系,并且没有对未来的软件流行度做出量化分析,不能给潜在的开发者提供相应的指导意见,因此,本文基于 GitHub 社区开发的开源软件自身的九个项目属性特征,将其划分为四个维度:用户维度(watch、star、fork)、软件维度(size、release、age)、运营维度(issue、commit)、团队维度(contributor),建立一个可以通过四个维度来反映开源软件流行度的评估模型,能帮助推演预测开源软件项目的未来流行度的发展。开源软件的这四个维度贯彻整个开源软件的生存周期,并且不同的实验数据能够从这些维度的不同方面来衡量开源项目的指标,避免了单一指标因素的片面性,更能体现开源软件项目的流行度,以及做出准确的预测分析。

本文根据从 GitHub 上划分的四个维度、九个属性指标深入,来分析开源软件的流行度。通过分析开源软件流行度相关维度与属性,构建开源软件流行度的概念模型,再基于用户行为进行属性关联性分析,确定开源软件流行度度量因子,再通过项目流行度度量因子在固定时间内的增长趋势,确定开源软件的四增长模式。论文接下来的章节安排如下:第 2 节介绍相关研究工作,第 3 节分析了 GitHub 中开源软件流行度影响因子,构建了开源软件流行度概念模型,包含用户维度、运营维度、团队维度以及软件维度等四个维度和九个相关属性。第 4 节分析了用户维度三种行为属性(watch、star、fork)与其它维度属性的相关性,第 5 节提出了基于斯皮尔曼等级相关系数检验的流行度相关性分析方法,通过基于用户行为属性(watch、star、fork)与其他属性之间的相关性分析,watch 属性是衡量 GitHub 开源软件流行度

的度量因子。第 6 节是总结与展望。

2 相关研究

开源软件的流行度分析对开发人员来说具有重要的意义,开发人员都想知道自己参与开发的项目或者自己创建的项目是不是在整个开发过程中吸引新的潜在用户,以及社区中的用户是否能接受该开源项目后续发布的新版本,或者是满足相关用户的期待。通常评价开源软件的流行度都是借助于这些平台内部的项目属性(如贡献者解决问题 issue 的效率)以及开发者的用户属性,如基于 GitHub 平台的社交指标 watch、fork、star 等,通过这些社交属性因子的相关度来衡量开源软件的流行度。现有的流行度预测方法,包括基于特征分类的流行度预测方法与基于时间序列建模的流行度预测方法,以及能够得到流行度具体数值的基于回归分析的预测方法。

(1) 基于特征分类的预测方法是使用一个的标准化的模型来预测所有的消息。Ma 等人^[3]为了预测 Twitter 未来哪些主题将是流行性主题,在 Twitter 上收集了包含来自 200 万用户的 3100 万条推文数据集,针对数据集识别用于描述新主题标签的有效特征,并从主题标签字符串与包含主题标签的推文中提取 7 个内容特征,以及从采用该主题标签的用户形成的社交图中提取 11 个上下文特征。利用 5 种标准分类模型即贝叶斯、k-means、决策树、支持向量机与逻辑回归进行预测分析,实验结果表明,使用特征提取的标准分类器明显优于不使用这些特征的基线方法,再根据突发主题标签与连续主题标签的预测准确性,表明逻辑回归模型比连续主题标签能更有效地预测突发主题标签的流行度。

(2) 基于时间序列建模的预测方法主要是利用所得的模型对用户生成的内容进行流行度预测。Yang 等人^[4]基于 K-means 聚类算法与逻辑回归模型,对获取到的上亿条推文与上亿篇博客数据集进行随时间变化的聚类分析,挖掘生成了内容流行度随着时间变化的消涨模式。Figueiredo 等人^[5]利用随机森林对时间序列进行分类,以此预测内

容信息的流行度。Hu 等人^[6]则是研究短时间内爆火的热门话题的流行度时间序列,分析得到这些时间序列存在高度相似性,并以此定义流行度的平均值、走势,根据这些时序特征空间来分析流行度随时间变化存在的趋势。最终研究结果表明,利用少部分历史数据,便可准确预测对应的话题流行度。

(3) 基于回归分析的流行度预测方法是最常用的预测方式。Tatar 等人^[7]为了了解一个项目比另一些项目更流行的原因,使用 Web 页面新闻早期的评论数据训练线性回归模型参数,最后使用该线性回归模型来预测 Web 最终的新闻评论数量,通过构建文章类型和发布时间为自变量的预测模型,来预测 Web 页面新闻受欢迎程度。实验结果表明,通过构建的预测模型能最终体现出其流行度的性能没有显著性提高。

3 开源软件流行度影响因子分析

开源软件的流行度最直接体现在该存

储库项目的受用户关注的程度,以及被用户收藏的次数,或者为此项目真正做出贡献的开发人员的数量,亦或是体现在整个项目的规模上,即相应的代码的数量。本节主要通过分析 GitHub 上影响开源软件流行度的属性因子,构建相应的开源软件流行度概念模型,利用 python 爬取的 GitHub 的数据集,研究开源软件的用户行为属性与其他维度属性之间的相关性,并分析确定流行度度量因子。

3.1 开源软件流行度属性特征提取

本文研究了 GitHub 上的诸多项目与用户属性,最终抽取了能够反映项目流行度特性等的九个属性,包括用户的 watch、star、fork、issue、commit、release、contributor 等相关属性,在 GitHub 上的具体体现如图 1 所示。

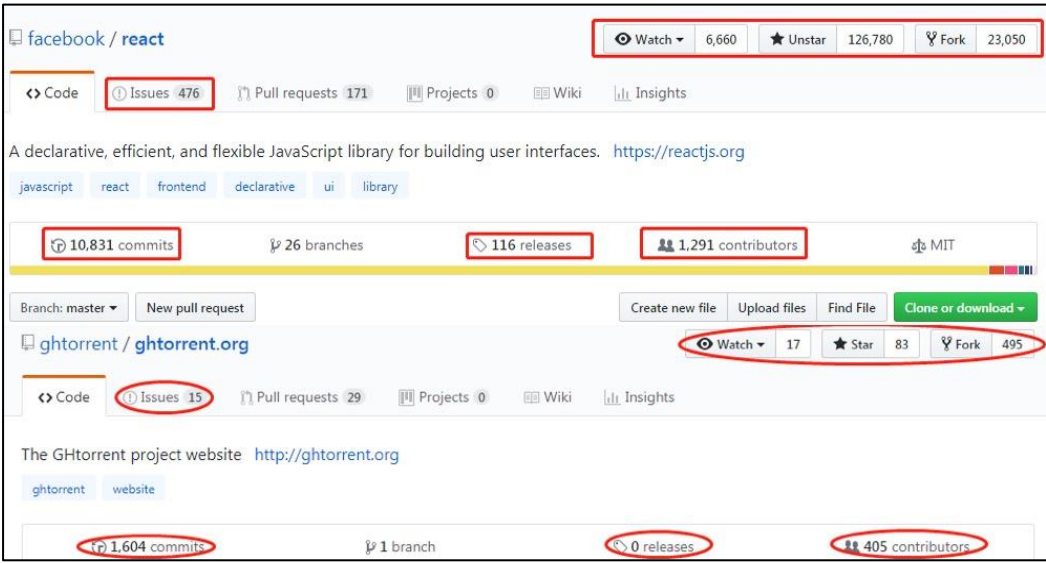


图 1 GitHub 项目页面属性图

图 1 是 GitHub 上两个开源项目的相关属性对比,GitHub 开发的社交属性功能包含 watch、star、fork,且图中 facebook/react 项目与 ghtorrent/ghtorrent.org 项目对应的属性数量有较大的差异,facebook/react 作为当前受用户关注的流行性项目,其对应的 watch、star、fork 数都有较高值,如 watch 数为 6660,star 数为 1267800,fork 数为 23050,并且均高于 ghtorrent/ghtorrent.org 项目对应的属性

数,其次,受欢迎项目的 issue、commit、contributor、release 的数量也均高于不受欢迎的项目。

3.2 开源软件流行度属性建模

本节基于这九个属性分析,根据对应的特征联系将其划分为四个维度,分别是用户维度、软件维度、运营维度、团队维度,构建了如图 2 所示的开源软件概念模型。

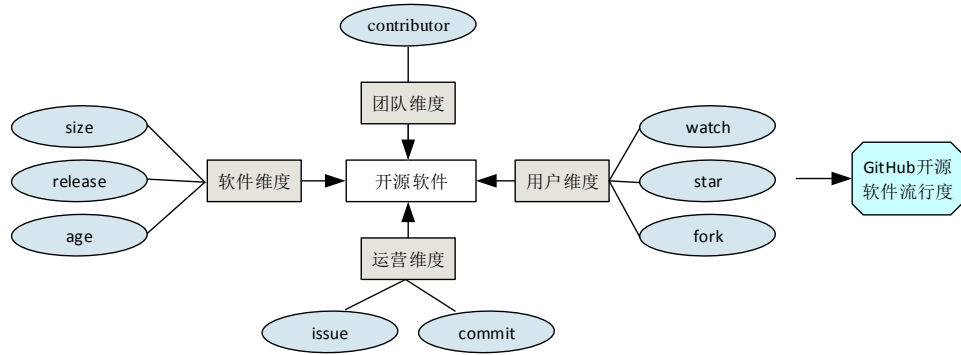


图 2 开源软件概念模型

图 2 的开源软件概念模型中的各维度包含的属性因子及含义的描述统计如表 1 所示。

表 1 属性因子统计表

维度	属性	意义
用户维度	watch	watch 的数量
	star	star 的数量
	fork	fork 的数量
软件维度	size	项目的代码行数
	age	项目持续的时间
	release	项目存储库的版本数
运营维度	issue	issue 的数量
	commit	项目提交更改的次数
团队维度	contributor	贡献者的数量

由表 1 可知，本节将 GitHub 的九个属性特征划分为四大维度：（1）用户维度包含三个属性，分别为 watch、star、fork，这三个属性主要描述的是用户对于开源项目的一种用户行为关注，表现出用户希望以此关注项目的后续进展。（2）软件维度所包含的 size、age、release 属性体现了项目的代码行数、项目演化发展周期以及迭代更新的版本次数。（3）运营维度则是从项目运作周期考虑，包含 issue 问题属性与项目提交更改次数 commit 属性。（4）参与项目贡献的团队维度，由 contributor 表示其贡献者的数量。

4 基于用户行为属性的关联结果分析

本文设计实现了一个开源软件爬虫算法，从 GitHub 中爬取了 30000 个存储库项目的数据集，并进行数据清理后选择了满足

要求的 5000 个数据集，本文基于四个维度进行属性之间的相关性分析，通过分析这些属性维度之间存在的关联性，来帮助研究分析影响开源软件流行度的属性因子。

4.1 用户维度行为属性分析

本节首先对用户维度三种行为属性（watch、star、fork）进行分析统计，根据散点图分析算法，得到的分布图如图 3 所示。图 3(a)的纵轴代表用户的关注数量，横坐标表示用户 fork 项目的数量，用户关注数 watch 约在[0,5000)的范围内变化；fork 数大约在[0,1200)的范围变化。大部分用户的 watch 数与 fork 数值都相对较少，大约集中在 200 附近。这一数据说明，相对受用户关注的项目是较少的。

图 3(b)的纵坐标代表用户关注的数量，用户关注的数量与图 3(a)中的纵坐标及变化

范围保持一致,横坐标表示用户的 star 数量,star 数量的变化范围大致为[0,5500)。大部分

的用户关注 watch 数与 star 数都集中在 [250,500]之间。

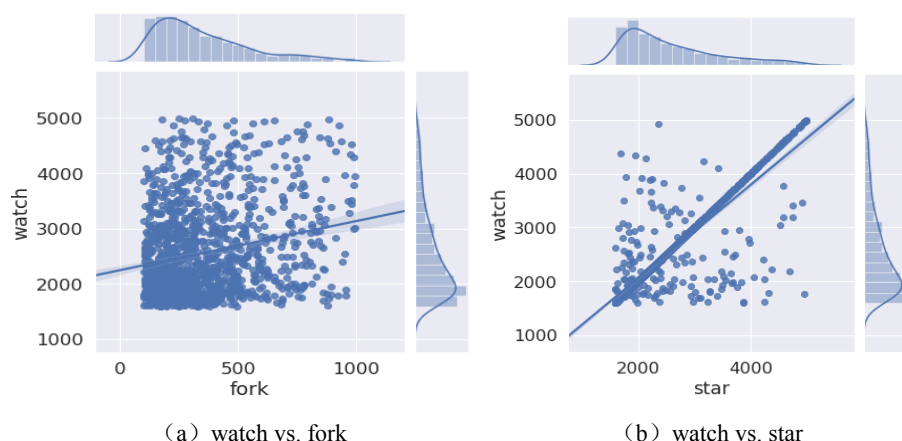


图3 watch VS. star、fork

从图 3(a)还可以观察到: 用户的关注数 watch 与 fork 数有一个正相关关系(趋势直线所示), 即随着 fork 数的增加, 用户关注数 watch 也增加, 但是增加变化的幅度较小。从图 3(b)可以分析得出, 用户的 watch 数与 star 数存在正相关关系, 即随着 star 数的增加, watch 数也增加, 并且两者的增加速度幅度较大且保持一致。

4.2 用户行为与运营维度

本节分析考虑运营维度的 issue、commit 属性与用户行为属性之间的相关关系, issue 作为整个开发过程中的问题跟踪系统域, commit 对应的项目修改提交的次数, 两者影响整个开源软件的开发周期, 用户根据

GitHub 上的 watch、star 等操作行为, 会对项目相关问题进行关注追踪。

(1) watch、star、fork 和 issue

用户创建的 issue 的数量, 是影响项目发展的主要特征。本节首先统计分析 watch、star、fork 与 issue 的相关关系, 根据论文的散点图分析算法, 得到的分布图如图 4 所示。图 4 中三个小图中的横坐标代表 issue 数, 纵坐标分别表示 fork、star、watch 数, 其中 issue 数的变化区间为[0,150]。图 4(a)中 fork 数大约在[0,1000) 的范围内变化, 而图 4(b)中的 star 数与图 4(c)中的变化的范围分别为 [0,5000), [0,5000)。这些数据表明, issue 数的数值与这三个属性数值之间的相对差度较大。

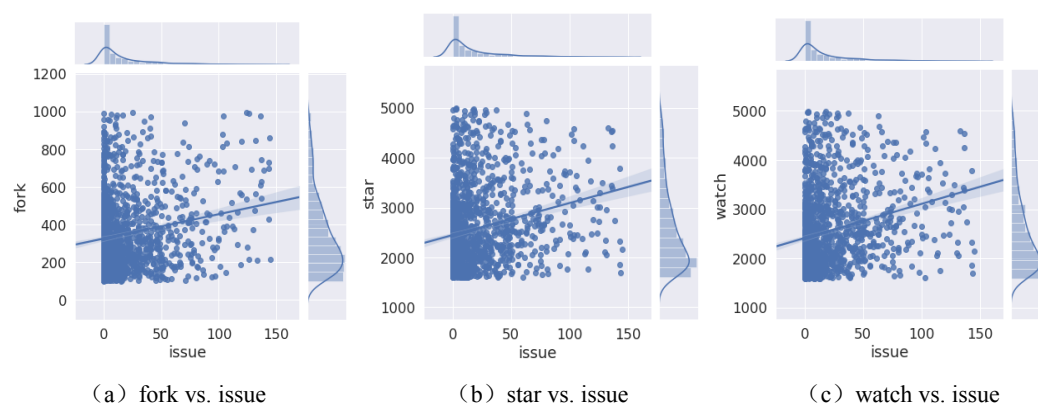


图4 watch、star、fork VS. issue

由图 4 可知, 用户行为属性的三个因子与 issue 之间存在一定的相关关系。根据图

4(a), 随着 issue 数增加, fork 数也随之增加, 两者之间存在一种正相关关系。对于图 4(b)

与图 4(c)而言, issue 与 star、watch 之间不仅存在和图 4(a)一样的正相相关关系, 且他们之间的相关增加速度幅度更高。由此可以表明, 用户的 star、watch 数的增加趋势, 会随着开源项目流行周期中所创建的 issue 数的增加而受较大的影响。

(2) watch、star、fork 和 commit

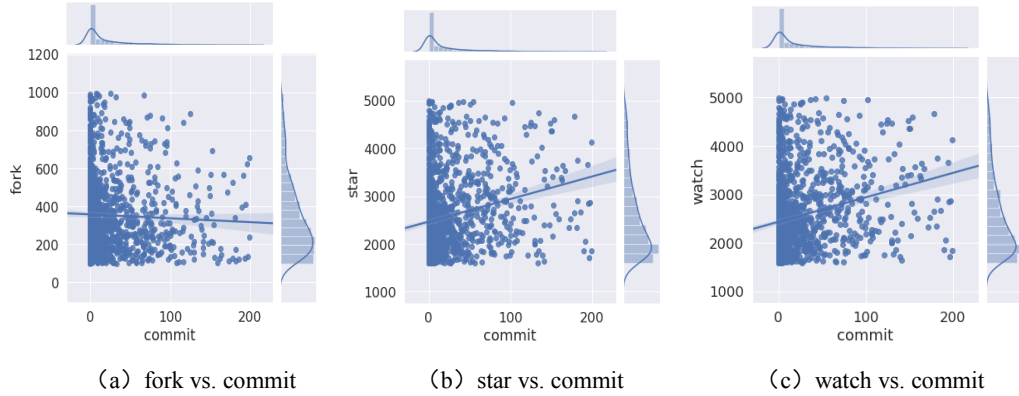


图 5 watch、star、fork VS. commit

由图 5(a)可知, 与上述 5(a)中 issue 与 fork 之间不同的是, 图 5(a)中 commit 与 fork 之间出现了这样一个趋势, 随着 commit 数的增加, fork 数呈现出减少的情况, 即两者之间存在一个负相关关系。这种情况可以解释为, 当开发者致力于在某个项目中增加修改贡献次数时, 相应的 fork 其他项目在本地存储库的次数会随之减少。而图 5 (b)、图 5(c) 与其不同, 这两个属性的增加趋势保持一致, 且随着 commit 数的增加, star 与 watch 的数量相应的增加。也就是说, 对于更改提交次数较多的项目, 用户会希望继续关注其后续的开发进展, 并可能促进项目的流行。

论文分析研究 commit 属性对 watch、star、fork 的影响, 具体的影响结果统计表现在图 5 中。图 5 中的三个小图的纵轴表示的是 fork、star、watch 的数量, 横坐标代表 commit 提交的次数, 且 commit 数变化的范围为[0,200)。

4.3 用户行为与团队维度

本节分析在 GitHub 上作出相关贡献的团队的影响力, 分析其与用户行为属性存在的相关关系, 包括 watch、star、fork 和 contributor 之间的关系。

通过关注项目的贡献者数 contributor, 利用散点图分析了其与 watch、star、fork 之间相关关系, 对应的分析结果如图 6 所示。图 6 中三个小图的横坐标表示的是 contributor 数, 三个小图对应的纵坐标分别为 fork、star、watch 数, 其中 contributor 的变化区间为[0,20), 是一个相对变化较小的区间值。

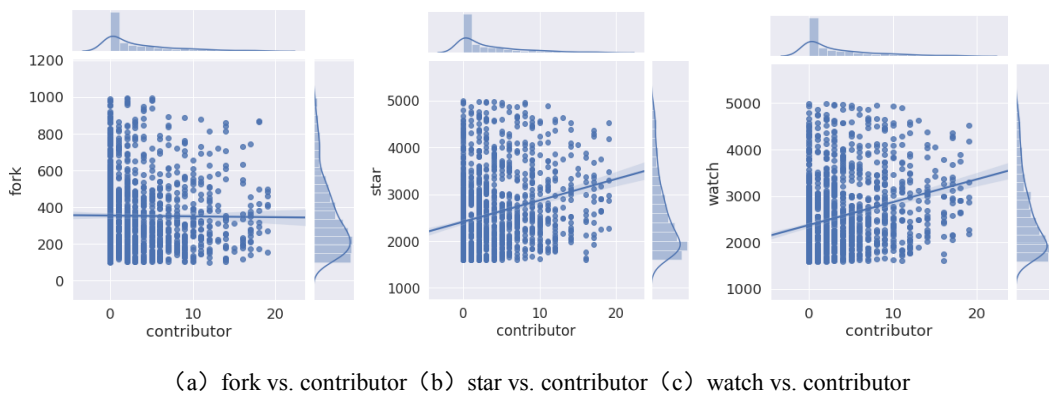


图 6 watch、star、fork VS. contributor

由图 6(a)可知, contributor 与 fork 数之间的相关性趋势又不同于图 4(a)、图 5 (a), 图中随着 contributor 数的增加, fork 数量没有呈现出明显的变化, 换句话说, contributor 数量的增加与否对 fork 的影响力极小。而 contributor 与 star、watch 之间存在一种正向相关性, 且 star 集中变化的范围为 [2000,3000], 而 watch 集中在[1500,1700]之

间, 且从这两小图中的趋势直线可知, 两者的相关程度较为接近。出现这种情况的可能原因是, GitHub 平台上参与到项目开发中的贡献者的人数, 能够影响开源软件的活跃性, 且已有研究显示大众贡献者^[8]更多是会选择活跃度与流行度更高的开源软件。

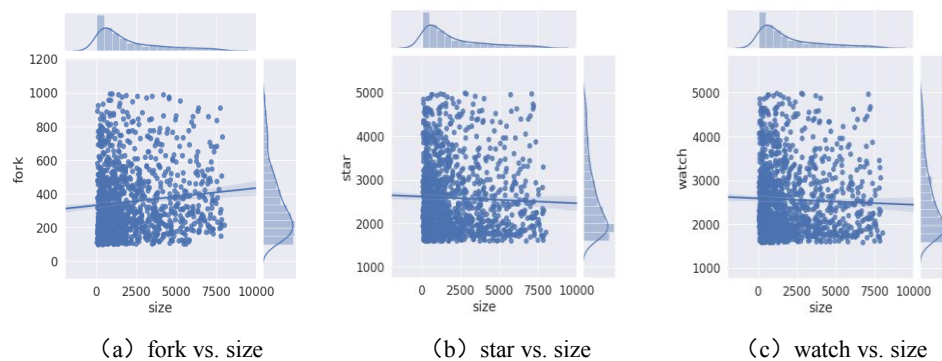


图 7 watch、star、fork VS. size

4.4 用户行为与软件维度

此节分析软件维度, 主要是考虑开源软件整体的代码量、存在的时间周期以及它迭代的版本次数对于项目流行的影响, 用户为了了解这些项目属性的进展, 会通过 watch、star 等行为操作来跟进后续的发展。

(1) watch、star、fork 和 size

论文基于项目的代码行数 size, 分析用户行为属性与其关联度, 得到的分析结果散点图如 7 所示。图 7 中的三个小图的横坐标表示的是 size, 整体数据对应的数值波动范围为[0,10000], 而较多 size 数密集分布在 [0,2500]这个区间。

根据图 7(a)可知, fork 数与 size 之间存在一种微弱的正相关关系, 即随着 size 数增加, 相应用户的 fork 数也会随之增加, 只是增加的幅度较小。而图 7(b)与图 7(c)呈现的趋势图与图 7(a)正相反, 这两个小图中的 star、watch 数, 随着 size 数值的变大, 有减少的趋势。

(2) watch、star、fork 和 age

考虑 age 项目演化生存的时间周期, 分析其与用户行为属性 watch、star、fork 之间的关联关系, 并通过散点图分析算法得到如图 8 的分析结果。图 8 中三个小图的对应的横坐标均为 age, 相应的纵坐标分别为 fork、star、watch。图 8(a)中 fork 数据主要集中在分布在[0,500)附近, age 整体变化的范围在[0,450)之间, 且 watch 与 age 之间的关系趋势图与 fork、star 和 age 之间的关系走势有明显不同。

根据图 8(c)可知, age 与 watch 之间有一个较弱的负向相关关系, 即随着 age 数的变大, fork 数呈现出一定的减少趋势, 但是减少的幅度较小。而图 8(a)与图 8(b)中 fork、star 与 age 之间关系走势和图 8(c)不同, 他们分别与 age 呈正相关关系, 只是对应的增加幅度较小。由此, 也可体现出用户关注开源项目的重点可能并非此项目的生存存在周期, 而是更多关注 GitHub 中新兴的流行项目。

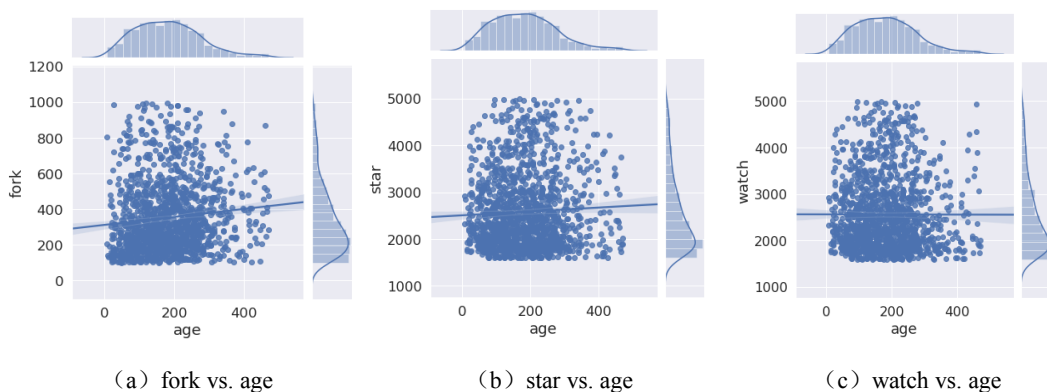


图 8 watch、star、fork VS. age

(3) watch、star、fork 和 release

基于 release 版本属性的特征,论文分析统计它对用户行为属性产生的影响,通过对取得的数据分析,得到的散点图如图 9 所

示。release 对应的项目的版本迭代次数,在图 9 中分别是对应的横坐标,且其数值变化的范围在 $[0,20)$ 之间。图 9 中三个小图中的纵坐标分别表示为 fork、star、watch。

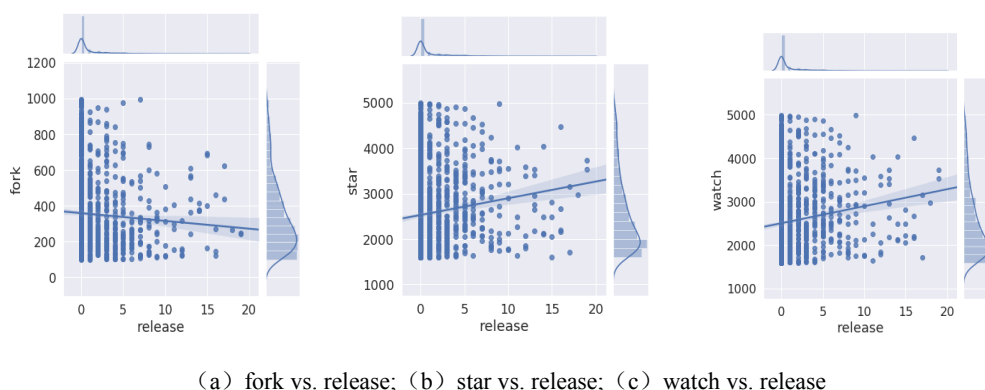


图 9 watch、star、fork VS. release

由图 9(b)与 9(c)可知, release 与 star、watch 数之间正向线性相关关系,即随着项目迭代的版本次数增加,相应的 star 与 watch 数都会增加。而图 9(a)体现的正是一种相反的关系,也就是随着版本次数的增加,相应的 fork 数会减少。根据这三个图的分析结果,项目的版本次数会影响 watch 与 star 属性,可能与 GitHub 中 watch 操作中包含关注的“release only”相关,促使用户跟进开源项目的新版本,并关注该项目。

5 Watch 流行性属性相关性分析

根据上一节基于用户行为的属性关联分析,可以初步判断开源软件流行度量因子 watch、star、fork 与其他维度的属性因子存在一种线性相关关系,为了量化属性之间具体的相关性,论文使用斯皮尔曼等级相关系数检验他们之间的相关关系,具体的线性相关统计表如表 2 所示。

表 2 影响因子与属性的相关性

属性		watch	fork	star	contributor	issue	commit	size	age	release
Spearman 的 ρ	watch	1.000	0.559**	0.774**	0.495**	0.520**	0.451	-0.149	-0.050	0.296
	fork	0.559**	1.000	0.415	0.320	0.359	0.276	0.355	0.271	0.087
	star	0.774**	0.415	1.000	0.383	0.409	0.354	-0.117	-0.041	0.239

根据上表 2 可知,基于用户行为属性利用斯皮尔曼等级系数 ρ 分析确定属性之间的关系。

(1) watch 与 fork、star 属性对应的 Spearman 的 ρ 值已经均高于 0.500,分别为 0.559(p 值 <0.001)与 0.774(p 值 <0.001),表明这三者之间是正向的显著相关关系,且任意用户行为属性的操作都会影响另两个属性。watch、fork、star 与 contributor 之间的斯皮尔曼 ρ 值分别为 0.495, 0.320, 0.380 (p 值 <0.001),由此可知,用户属性的三种行为与贡献者数量存在正向相关关系,且 watch 属性与 contributor 的相关度最高,以及两者之间属正向相关关系。

(2)对于三种属性行为与 issue、commit 之间的关系,watch 属性与 issue、commit 的相关度 ρ 值最高为 0.520、0.451(p 值 <0.001),并体现的是一种正向相关关系。对于 size、age 这两个属性,他们与用户行为属性 watch、star 之间存在着负相关关系,但相关系数略低,仅为-0.149、-0.117,这有可能和用户关注新兴流行度的小规模流行项目数相关联。由于 watch 与 release 对应的相关度 $\rho=0.296$ (p 值 <0.001)也高于 star、

fork,也为论文确定流行度度量因子提供了数据支撑。

因此,根据上述用户行为属性 watch、fork、star 与其他属性之间的相关性关系,并利用斯皮尔曼等级系数 ρ 分析确定,本节确定开源软件流行度度量因子为 watch,且 watch 数越多,对应的 GitHub 开源软件将越流行。

6 总结与展望

本文分析了开源社区 GitHub 的软件流行度相关四个维度与九个属性,构建了开源软件流行度的概念模型,分析开源社区 GitHub 中软件流行度的影响因子,并基于用户行为属性与其他维度属性的进行斯皮尔曼等级相关性检验度,通过基于用户行为属性(watch、star、fork)与其他属性之间的相关性分析,统计分析确定 watch 属性是衡量 GitHub 开源软件流行度的度量因子。论文下一步的工作是通过流行度度量因子针对 GitHub 开源软件进行开源软件流行度的趋势预测。

Data Mining, ACM,2011.

参考文献

- [1] Chris DiBona, Sam Ockman, Mark Stone. Open Sources: Voices from the Open Source Revolution[C].O'Reilly Open Source,1999.
- [2] GitHub.[2019-3-30].<https://github.com/>.
- [3] Z. Ma, A. Sun, G. Cong. On predicting the popularity of newly emerging hashtags in Twitter[J].Journal of the American Society for Information Science and Technology,2013,64(7):1399-1410.
- [4] Yang J, Leskovec J. Patterns of Temporal Variation in Online Media[C]//ACM International Conference on Web Search &
- [5] F Figueiredo, JM Almedia. The tube over time: charactering popularity growth of youtube videos[C].Proceedings of the Forth Conference on WSDM.2011:9-12.
- [6] Hu Y, Hu C, Fu S, et al. Predicting the popularity of viral topics based on time series forecasting [J].Neurocomputing, 2016, 210:55.
- [7] Tatar A, Amorim M D D, Fdida S, et al. A survey on predicting the popularity of web content[J].Journal of Internet Services & Applications,2014,5(1):8.
- [8] Zhou, Yu ming, Bao wen. Predicting the Maintainability of Open Source Software Using Design Metrics[J].Wuhan

-
- University Journal of Natural Sciences,2008, 13(1):14-20.
- [9] 王涛. 面向软件复用的大规模开源资源定位技术研究[D]. 国防科学技术大学, 2014.
- [10] Yang C, Fan Q, Wang T. et al. RepoLike: a multi-feature-based personalized recommendation approach for open-source repositories [J].Frontiers Information Technology & Electronic Engineering, 2019, 20: 222
- [11] Laura Dabbish, Colleen Stuart. Social coding in GitHub: transparency and collaboration in an open software repository[J].Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work.2012:1277-1286.
- [12] 周明辉, 张伟, 尹刚. 开源软件的量化分析[M].中国计算机学会通讯,2016,12(2):24-29.
- [13] M. Ahmed, S. Spagna, F. Huici,S. Niccolini. A peek into the future: predicting the evolution of popularity in user generated content[J].6th International Conference on Web Search and Data Mining (WSDM),2013:607-616.
- [14] H. Sajnani, V. Saini, J. Ossher, C. V. Lopes, Is popularity a measure of quality? an analysis of Maven components[J].30th Software Maintenance and Evolution (ICSME),2014:231-240.
- [15] B. Vasilescu, Y. Yu, H. Wang, P. Devanbu, V. Filkov. Quality and productivity outcomes relating to continuous integration in GitHub[C].Proc. 2015 10th Jt. Meet. Found. Softw. Eng. - ESEC/FSE 2015.
- [16] Krovetz R, Ugurel S, Giles C L. Classification of source code archives [C].In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval.2003:425-426.
- [17] Nora McDonald, Sean Goggins, Performance and participation in open source software on GitHub[J].CHI'13 Extended Abstracts on Human Factors in Computing Systems,2013.
- [18] Fragkiskos Chatziasimidis, Ioannis Stamelos. Data collection and analysis of GitHub repositories and users[J] .International Conference on Information, Intelligence, Systems and Applications (IISA),2015.