开源项目活跃度模型构建及实证

杨欣捷,田　蜜,江一鸣
(上海浦东发展银行股份有限公司,上海 200000)

摘　要: 随着开源技术不断涌现,开源作为一种治理模式、经济模式和企业生态,变得越来越重要。关注开源项目的活跃程度,分析活跃度的变化情况,在一定层面上能够反映项目发展状态。通过梳理现有的开源项目活跃度模型,发现:第一,现有模型没有考虑项目类别,不能有效反映镜像类项目活跃度;第二,现有模型没有考虑活跃度趋势,不能有效反映阶段性活跃项目的整体活跃度。在此基础上构建和完善了活跃度模型,并以浦发银行开源治理平台为例,对模型进行了验证,为企业在开源项目的选择上提供一定的参考价值。

关键词: 开源;活跃度;开源社区
中图分类号: TP391　　**文献标识码:** A　　**DOI:** 10.19358/j.issn.2096-5133.2021.07.005
引用格式: 杨欣捷,田蜜,江一鸣. 开源项目活跃度模型构建及实证[J].信息技术与网络安全,2021,40(7):27-33,51.

A model and empirical study on the activity of open source projects

Yang Xinjie, Tian Mi, Jiang Yiming
(Shanghai Pudong Development Bank, Shanghai 200000, China)

Abstract: With the emergence of open source technology, open source, as a governance model, economic model and enterprise ecology, is becoming more and more important. Analyzing the activity of open source projects and the changes of activity can reflect the development status of projects to a certain extent. By combing the existing open source project activity models, this paper finds that: first, the existing models do not consider the project category, they can not effectively reflect the activity of mirror projects. Second, the existing models do not consider the trend of activity, and they can not effectively reflect the overall activity of the stage active projects. On this basis, this paper constructs and improves the activity model. At the same time, the open source governance platform of Shanghai Pudong Development Bank is taken as an example to verify the model, which provides a certain reference value for enterprises in the selection of open source projects.

Key words: open source; activity; open source community

0 引言

开源软件(Open Source Software,OSS)是一种源代码开放的软件,作为一种有效的软件开发模式,已经获得了极大的普及[1],为几乎所有领域的应用程序提供了动力。开源已经出现了一个大发展的趋势[2]。代码仓库与活跃用户数都在高速增长;项目覆盖面越来越广,占据着各领域的主要市场份额;参与开源的企业数量保持稳定增长并呈现主动开源趋势[3]。显然开源正以它开放共享、合作共赢的特点吞没着整个世界。

在开源高速发展和国家自主安全的发展战略影响下,不少企业愿意选择开源软件,期望使用开源软件的企业从59%迅速提升至77%,而那些已经使用了开源软件的企业也在加大其使用广度和深度[2]。开源软件以社区形式开展开发工作,社区内的成员可以自由、开放地交流沟通、共享经验、参与协作。分析社区中开源项目的活跃度,挖掘开发者和企业组织在整个开源产业中的表现,有利于剖析国内外开源现状,为企业开源项目选型提供参考,为企业数字化转型赋能。

1 研究现状

围绕开源主题的研究得到了广大学者的关注,成为一大研究热点。学者们分别从开源现状、开源社区、开发者等角度进行了研究。对开源现状的剖析,如刘凯等对开源软件产业的发展现状及趋势进行了研究[4]。对开源社区的探究,如 ECKERT R 等对开源社区的组织间联系进行了研究,发现倾向于控制其所有资源的开源社区属于自主组织[5]。对开发者的研究,如 MENEEIY A 等利用开发者的网络关系成功预测项目可能出现的缺陷[6]。而针对开源项目活跃度的研究尚处于起步阶段。刘雅新、吴高艳等结合复杂网络理论与软件开发实践,以开源社区中开发者合作行为为研究取向,分析开发者在社区中的活跃度变化情况[7]。华东师范大学 Xlab 从社区披露数据出发,考虑了拉取请求(pull request,pr)、问题(issue)和评论(comment)等多个指标,获取开源项目活跃度,剖析开源现状[8]。Grank 项目活跃度分析工具,以 pr、提交暂存(commit)和贡献者(contributor)为指标,计算 GitHub 上开源项目的活跃度。浦发银行利用开源项目活跃度算法,在开源软件选型中提供重要参考。

1.1 浦发银行开源项目活跃度评估算法

浦发银行开源治理平台活跃度模型(式 1)是从开发者 commit 代码的角度去考虑项目的活跃度,但实际上项目的活跃度还体现在 issue 提交数、pr 数、评论数等。以 chartjs/Chart.js 为例,commit 有 3 118 条,pr 有 5 072 条,issue 有 2 184 条,评论有高达 3 万条,所以除了 commit 以外,pr、issue 和评论均可体现活跃度,因此只考虑 commit 提交数存在局限性。

$$A_r=\partial C_{\text{commit\_one\_month}}+\beta C_{\text{commit\_three\_month}}+\delta C_{\text{commit\_six\_month}}+\gamma C_{\text{commit\_twelve\_month}} \quad (1)$$

式中,A_r 为项目活跃度;$C_{\text{commit\_one\_month}}$ 为近一个月的暂存提交数;$C_{\text{commit\_three\_month}}$ 为近三个月的暂存提交数;$C_{\text{commit\_six\_month}}$ 为近半年的暂存提交数;$C_{\text{commit\_twelve\_month}}$ 为近一年的暂存提交数;∂、β、δ、γ 为权重。

1.2 Xlab 开源项目活跃度评估算法

Xlab 活跃度计算模型(式(2)、式(3))是从产生活跃度的开发者角度出发,更为全面地考虑了 pr、issue 和 comment 等多个因素,但未考虑到时间衰减因素,以及 GitHub 镜相类项目关闭了 pr 和 issue 功能。

$$A_u=C_{\text{issue\_comment}}+2C_{\text{open\_issue}}+3C_{\text{open\_pr}}+4C_{\text{revirew\_comment}}+5C_{\text{pr\_merged}} \quad (2)$$

$$A_r=\sum\sqrt{A_u} \quad (3)$$

式中,A_u 为开发者活跃度;$C_{\text{issue\_comment}}$ 为问题评论数;$C_{\text{open\_issue}}$ 为问题数;$C_{\text{open\_pr}}$ 为拉取请求数;$C_{\text{revirew\_comment}}$ 为评审评论数;$_{\text{pr\_merged}}$ 为拉取请求合并数;A_r 为项目活跃度。

1.3 Grank 项目活跃度分析工具

Grank 项目活跃度评估算法将项目的提交数、拉取请求数和贡献者数作为主要因素,分析项目的活跃度变化的趋势和幅度。

综上,当前关于开源项目活跃度评估算法选取的指标各异。同时,不同类别的开源项目活跃度分析角度是否存在差异,一段周期内的活跃度是否受时间因素的影响,目前均尚无结论。本文将从这两个问题出发构建活跃度模型,完善开源项目活跃度研究。

2 模型构建

2.1 构建原理

新模型的构建原理主要是对前文提到的“浦发模型”和“Xlab 模型”各自优点的融合,并且针对前文提到的“镜相类项目”和“时间衰减”两个因素进行了优化。以下针对这两个因素进行具体分析。

2.1.1 镜相类项目的活跃度

以 GitHub 上的项目作为样本,发现存在从私库同步至 GitHub 上的镜像类项目。通过查找 GitHub 上的其他镜像类项目样本并进行统计(如图 1 所示),可以看出镜像类项目的共同点有:第一,pr 和 issue 数量很少;第二,pr、issue 和 comment 贡献者在总贡献者中占比不大;第三,commit 贡献者在总贡献者中占比较大;第四,commit 贡献者进行了多次的代码提交。进一步访问各镜像类项目的 GitHub 地址,发现有些项目明确标注了关闭 pr 和 issue 功能,可见镜像类项目的特点是仅在 GitHub 上同步,进行少量的 pr 和 issue。

Xlab 模型在计算开发者活跃度时仅依托于 pr、issue 和 comment 贡献者,但是产生活跃度的开发者行为不止表现在 pr、issue 和 comment,所以 Xlab 模型无法全面体现镜像类项目的开发者活跃度,故结果会产生偏差。为了能更好地看出差别,引入一些非镜像类项目,基本信息如图 2 所示。

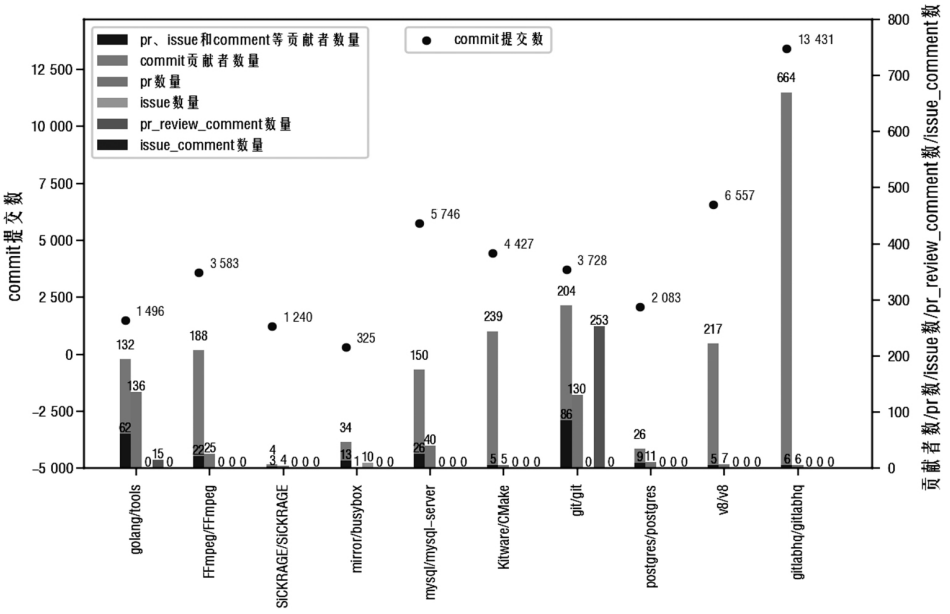

图 1 镜像类项目行为指标汇总

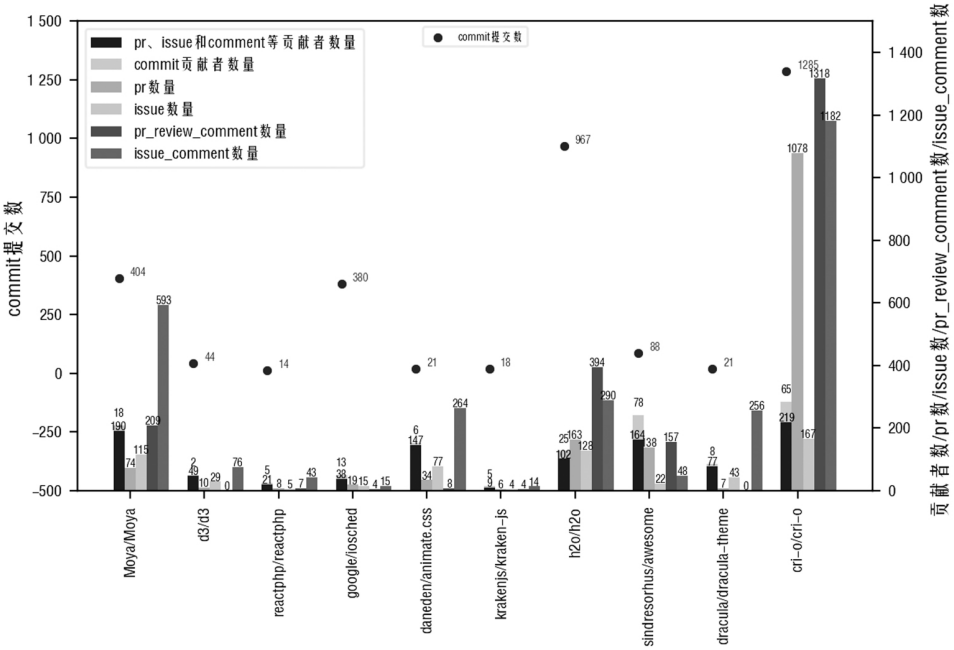

图 2 非镜像类项目行为指标汇总

将镜像类项目和非镜像类项目做两两对比,如图 3 所示,此样本中镜像类项目产生活跃度的贡献者数量基本上比非镜像类项目多,且进行的贡献行为次数也基本上比非镜像类项目多,有理由猜测镜像类项目活跃度并不低。

基于以上猜想,以一个镜像类和一个非镜像类为一个组,共 10 组,每组项目分别利用 Xlab 模型和浦发模型计算活跃度,镜像类和非镜像类活跃度大小关系完全相反,汇总活跃度如图 4 所示。在对于镜相类项目的活跃度评估中,浦发模型考虑了commit因素,显然更能准确反映镜相类项目的活跃度。但由于其未考虑 pr、issue 和 comment 等其他因素,其模型效果也有提高空间。

2.1.2 一段周期内的阶段性活跃度

仅对项目一段周期内总活跃度进行横向对比是不够的,对项目在一段周期内的活跃度变化趋势进行分析则更具意义。对样本项目(2019 年 3 月至 2020 年 3 月)这一年期间每个月的 GitHub 日志进行

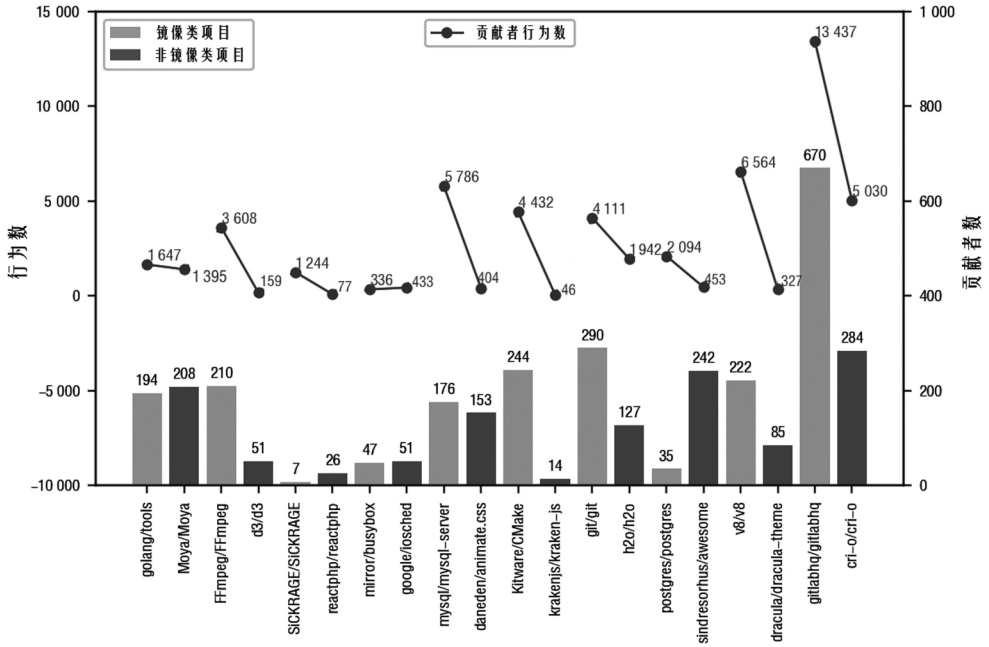

图 3 镜像类、非镜像类项目行为指标汇总

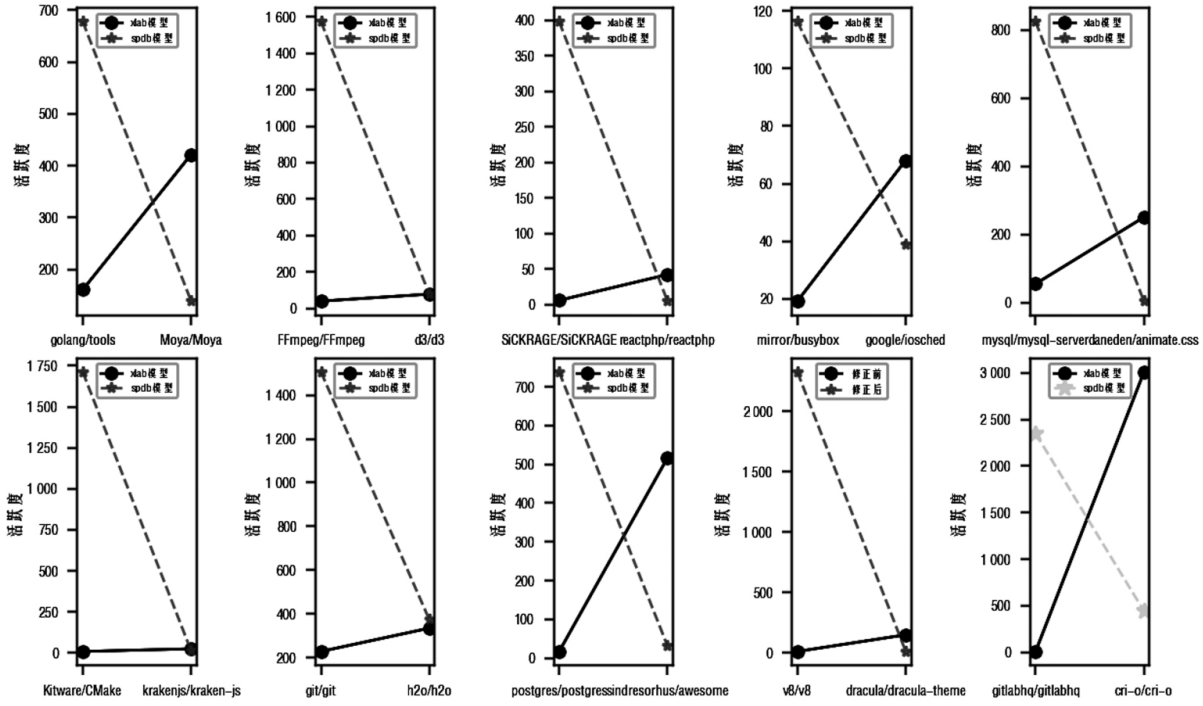

图 4 镜像类、非镜像类项目的 Xlab 和 Spdb 模型活跃度对比

统计,分析各个项目活跃度变化趋势,如图 5 所示,各个项目的活跃度变化趋势各不同,有的呈现波动趋势,如 996icu/996.ICU;有的比较平稳,如 twbs/bootstrap、github /gitignore 和 mrdoob/three.js 等。

现对 996icu/996.ICU 和 mrdoob/three.js 的开发者活跃度进行具体分析,2019 年 3 月至 2020 年 3 月期间 996icu/996.ICU 开发者人数为 1 263 人,mrdoob/three.js 开发者人数为 1 690 人,具体分布见图 6,显然 996icu/996.ICU 这一年来仅在 2019 年 3 月和 4 月开发者较多(488 人、627 人)、贡献较大,而 mrdoob/three.js 开发者较为平稳地活跃。

观测 2019 年 3 月至 2020 年 3 月期间每一个月的项目活跃度变化,如表 1 所示,容易看出,996icu/996.ICU 在 2019 年 3 月、4 月活跃度很高,5 月之后

出现明显降低趋势,而 mrdoob/three.js 呈现持续平稳态势。

对比两个项目这一年总的活跃度(表 2)发现,996icu/996.ICU 活跃度比 mrdoob/three.js 高。同时利用浦发银行开源治理平台活跃度模型(式(3))计算两者的活跃度,结果为:996icu/996.ICU(314.800 000)、mrdoob/three.js(2 056.500 000),996icu/996.ICU 活跃度比 mrdoob/three.js 低。究其原因,Xlab 活跃度模型未考虑时间因素对活跃度的影响,使得活跃度趋势波动较大的项目的整体活跃度有异。

2.2 构建模型

通过前文的模型计算实践,可以得出结论:(1)GitHub 镜像类项目大都关闭提交 issue 的功能,对于 pr 也有限制。Xlab 模型未考虑 commit 变量,因

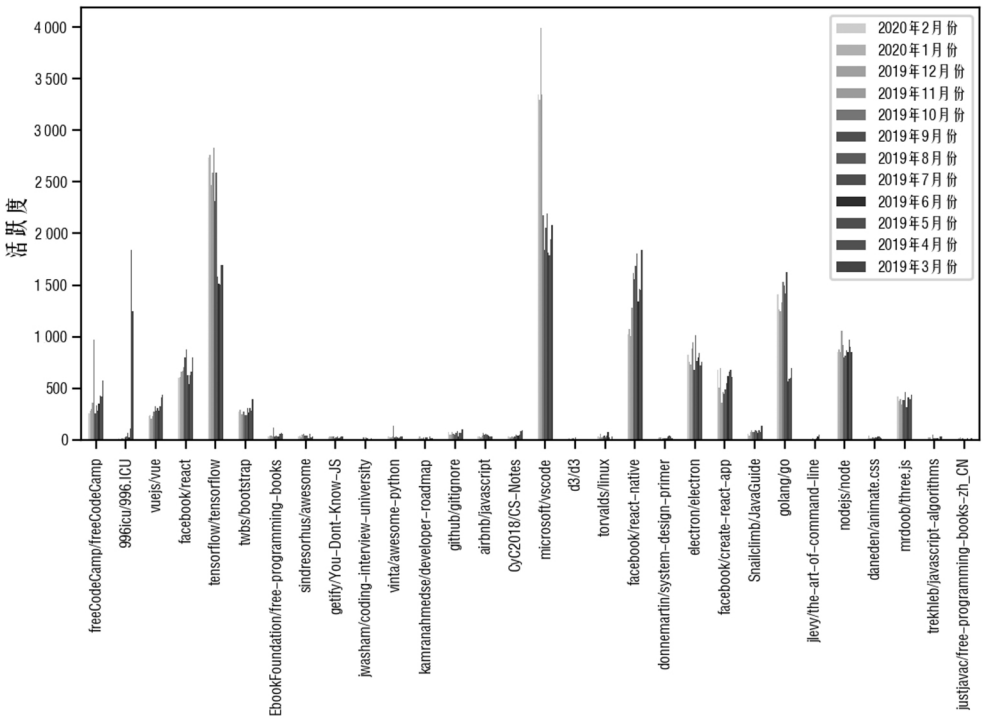

图 5 项目月活跃度汇总

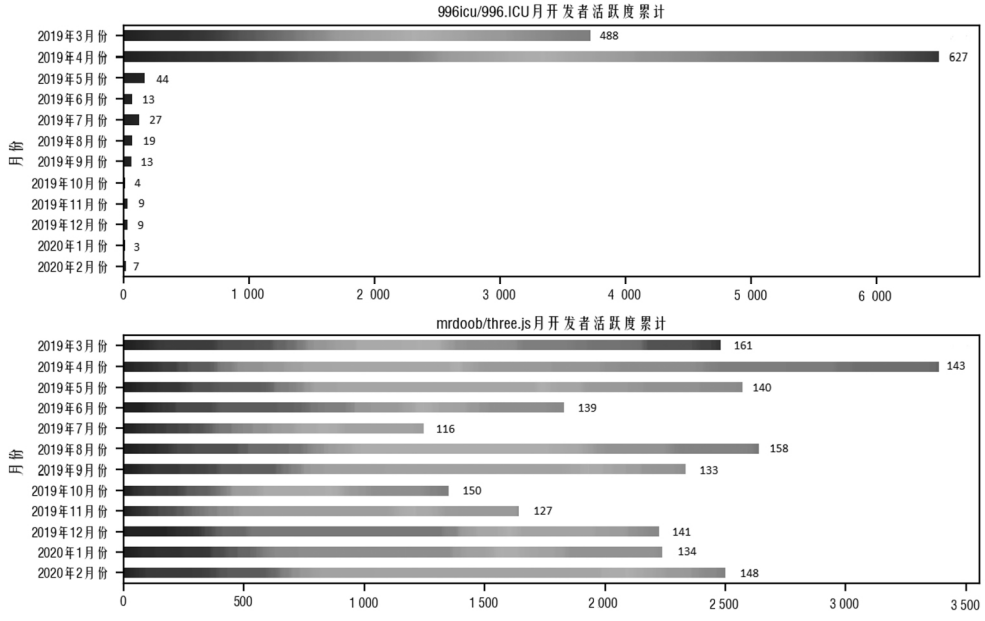

图 6 单个项目活跃度周期内变化规律

表 1 项目活跃度变化汇总

时间	996icu/996.ICU	mrdoob/three.js
2019.3	1 246.336 351	436.882 001
2019.4	1 844.452 699	396.801 997
2019.5	109.462 118	409.965 846
2019.6	27.254 919	412.137 127
2019.7	64.919 957	317.531 28
2019.8	37.061 286	463.676 007
2019.9	26.582 915	384.703 127
2019.10	6.928 204	385.104 179
2019.11	16.305 898	343.555 446
2019.12	17.023 337	393.399 322
2020.1	5.196 153	378.984 304
2020.1	12.124 357	424.303 53

表 2 项目总活跃度

	996icu/996.ICU	mrdoob/three.js
总活跃度	3 285.376 397	3 062.640 067

GitHub 上存在大量关闭 issue 和 PR 的镜像类项目,该模型无法有效体现项目活跃度。(2)Xlab 模型未考虑时间衰减因素,不能很好反映发布时参与者较多,后劲不足的项目的实际活跃情况。(3)浦发模型仅考虑了 commit 变量,未纳入 issue 和 pr 等其他因素,模型结果全面性有待提高。

为解决以上问题,在 Xlab 模型基础上,引入 commit 变量和衰减函数,形成修正后模型(式(4)~式(6))。

$$A_{ut}=\boldsymbol{c}e^{-kt}(C_{issue\_comment}+2C_{open\_issue}+3C_{open\_pr}+4C_{revirew\_comment}+5C_{pr\_merged}+C_{commit}) \quad (4)$$

$$A_u=\sum A_{ut} \quad (5)$$

$$A_r=\sum\sqrt{A_u} \quad (6)$$

式中,$\boldsymbol{c}$ 为初始值(将为 100%)或全权重;e 为自然对数;k 为衰减常数,对于所需的曲线为 0.05;t 为月份。$C_{issue\_comment}$ 为问题评论数;C_{open_issue} 为问题数;C_{open_pr} 为拉取请求数;$C_{revirew\_comment}$ 为评审评论数;C_{pr_merged} 为拉取请求合并数;C_{commit} 为暂存提交数;A_{ut} 是开发者在 t 月的活跃度;A_u 是一段时间内的总开发者活跃度;A_r 是一段时间内的项目活跃度。

3 模型实证

随着金融行业开源生态的日益成熟,加上数字化生态银行转型的需求,越来越多的应用系统构建在了开源软件之上,为此浦发银行探索了一套开源治理体系。为自主、高效、安全地使用开源软件提供技术上和制度上的支持,为提升管理效能,浦发银行自研了开源软件管理平台,覆盖了开源软件治理四个主要的流程,引入评估、使用、安全漏洞持续评估和生命周期持续评估,做到软件使用有迹可循,实现介质来源可控,防范开源软件带来的安全问题。

现以浦发银行开源治理平台为例,选取了平台引入的软件,以前文构建的活跃度模型进行评价。以一个平稳类和一个波动类为一个组,一个镜像类和一个非镜像类为一个组,每组项目分别利用 Xlab 活跃度模型、浦发活跃度模型和修正后活跃度模型计算活跃度。Xlab 模型与修正后模型计算出的每组活跃度大小关系基本完全相反,浦发模型与修正后模型计算出的每组活跃度大小关系基本趋同,如图 7 所示。修改后的模型在一定程度上抹平了活跃度趋势波动较大项目的整体活跃度,同时也提高了镜像类项目的活跃度。

对 Xlab 活跃度模型、浦发活跃度模型与修正后的活跃度模型的计算结果进行相关性分析,如表 3 所示。第一,修正后的活跃度模型与 Xlab 活跃度模型不相关,与浦发活跃度模型相关。说明修正后的活跃度模型与浦发活跃度模型结果趋同,区别只是浦发活跃度模型未考虑 pr、issue 和 comment 等能产生活跃度的因素。第二,Xlab 活跃度模型与浦发活跃度模型不相关,与修正后的活跃度模型也不相关。进一步剔除镜像类项目后,结果如表 4 所示。Xlab 活跃度模型与浦发活跃度模型相关,与修正后的活跃度模型也相关。说明虽然 Xlab 活跃度模型考虑的影响因素很全面,但仍在镜像类项目活跃度的计算上存在偏差。同时结合平稳类和波动类项目来看,Xlab 模型又未考虑时间因素,修正后的活跃度模型考虑了时间权重因素。综上,修正后的活跃度模型同时具备了 Xlab 活跃度模型与浦发活跃度模型的优势,从更全面的角度去反映项目活跃度。

4 结论

目前针对开源项目活跃度的评估算法研究尚处于起步阶段,考虑的影响因素不同,且未考虑镜像类项目特点和时间权重因素对活跃度的影响。本文对浦发银行和 Xlab 活跃度评估算法进行了探索,结合镜像类项目特点、时间权重因素,将 pr、issue、comment 和 commit 等参数和衰减函数融合,形成修正后模型,该模型能更好地反映项目活跃度。针对企业而言,探索开源社区的行为规律,剖析开源项

目活跃度算法,能够在企业开源项目的选择上提供一定的参考价值。对开源社区和开源项目而言,需要提高社区服务质量,确保社区成员来源多样化,提高开发人员的贡献力度,进而提高整个项目的活跃度,促成良性循环。

参考文献

[1] FANG Y, NEUFELD D. Understanding sustained participation in open source software projects[J]. Business Strategy Review, 2009, 25(4): 9-50.

[2] 开源社.2020 年中国开源年度报告[R].开源社, 2020: 1-15.

[3] 智研咨询集团.2021-2027 年中国开源生态行业市场研究分析及投资战略规划报告[R].深圳:智研咨询集团, 2020: 1-198.

[4] 刘凯.开源软件产业在中国的发展现状及趋势研究[D].武汉:华中师范大学, 2013.

[5] ECKERT R, GREER D, JUREK-LOUGHREY A. Alone or together? Inter-organizational affiliations of open source communities[J]. The Journal of Systems and Software, 2019, 149(4): 250-262.

[6] MENEEIY A, WLIIAMS L, SNIPESS W. Predicting failures with developer networks and social network

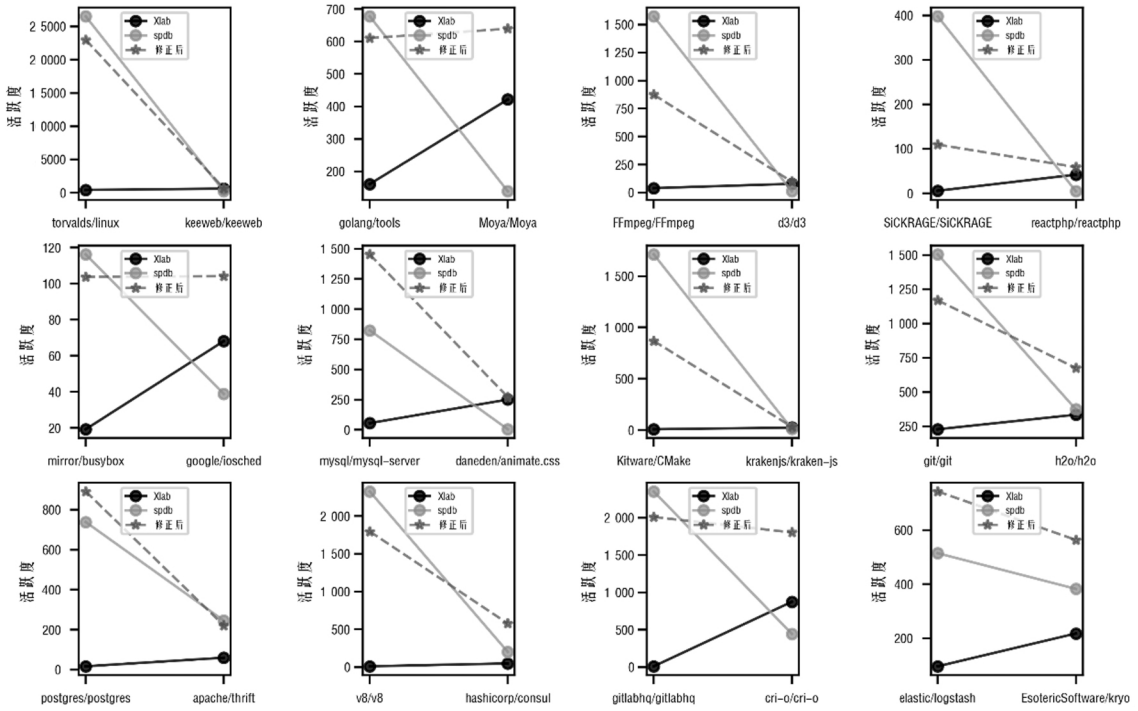

图 7 活跃度结果对比

表 3 模型结果相关性分析

		Xlab 模型	SPDB 模型	修正后模型
Xlab 模型	Pearson 相关性	1	0.009	0.175
	显著性(双侧)		0.967	0.414
	N	24	24	24
SPDB 模型	Pearson 相关性	0.009	1	0.983**
	显著性(双侧)	0.967		0.000
	N	24	24	24
修正后模型	Pearson 相关性	0.175	0.983**	1
	显著性(双侧)	0.414	0.000	
	N	24	24	24

** 在 0.01 置信水平下显著

表 4 模型结果相关性分析(非镜像类)

		Xlab 模型	SPDB 模型	修正后模型
Xlab 模型	Pearson 相关性	1	0.723**	0.919**
	显著性(双侧)		0.005	0.000
	N	13	13	13
SPDB 模型	Pearson 相关性	0.723**	1	0.924**
	显著性(双侧)	0.005		0.000
	N	13	13	13
修正后模型	Pearson 相关性	0.919**	0.924**	1
	显著性(双侧)	0.000	0.000	
	N	13	13	13

** 在 0.01 置信水平下显著

(下转第 51 页)

[9] JUELS A, RIVEST R L. Honeywords: making password-cracking detectable[C]. Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, 2013: 145-160.

[10] PAXTON N C, JANG D, RUSSELL S, et al. Utilizing network science and honeynets for software induced cyber incident analysis[C]. 2015 48th Hawaii International Conference on System Sciences. IEEE, 2015: 5244-5252.

[11] GOH H C. Intrusion deception in defense of computer systems[R]. Naval Postgraduate School Monterey CA, 2007.

[12] JULIAN D P. Delaying-type responses for use by software decoys[R]. Naval Postgraduate School Monterey CA, 2002.

[13] MICHAEL J B, AUGUSTON M, ROWE N C, et al. Software decoys: intrusion detection and countermeasures[R]. Naval Postgraduate School Monterey CA, Dept of Computer Science, 2002.

[14] NOSSITER A, SANGER D E, PERLROTH N. Hackers came, but the French were prepared[N]. New York Times, 2017, 9.

[15] BOWEN B M, KEMERLIS V P, PRABHU P, et al. A system for generating and injecting indistinguishable network decoys[J]. Journal of Computer Security, 2012, 20(2-3): 199-221.

[16] REN J G, ZHANG C M. A differential game method against attacks in heterogeneous honeynet[J]. Computers & Security, 2020, 97: 101870.

[17] CARROLL T E, GROSU D. A game theoretic investigation of deception in network security[J]. Security and Communication Networks, 2011, 4(10): 1162-1172.

[18] 黄羡飞, 王高才, 彭颖. 移动云计算中一种虚拟机迁移的预拷贝传输策略研究[J]. 计算机应用研究, 2018, 35(11): 3356-3360.

[19] 彭颖, 王高才, 黄书强, 等. 移动网络中基于最优停止理论的数据传输能耗优化策略[J]. 计算机学报, 2016, 39(6): 1162-1175.

(收稿日期: 2021-03-18)

作者简介:

吕德龙(1988-), 男, 硕士研究生, 主要研究方向: 网络欺骗防御。

翁溪(1990-), 女, 硕士, 助理研究员, 主要研究方向: 网络防御。

周小为(1982-), 女, 硕士, 工程师, 主要研究方向: 网络防御、信息安全。

(上接第 33 页)

analysis[C]. Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2008: 13-23.

[7] 刘雅新, 吴高艳. 开源软件社区中开发者活跃度特性分析[J]. 软件导刊, 2017(9): 164-170.

[8] 王伟, 周添一. 全球开源生态发展现状研究[J]. 信息通信技术与政策, 2020(5): 38-44.

(收稿日期: 2021-06-13)

作者简介:

杨欣捷(1984-), 男, 硕士, 主要研究方向: 开源软件治理、数据库技术、物联网金融。

田蜜(1992-), 女, 硕士研究生, 主要研究方向: 开源软件治理。

江一鸣(1984-), 男, 本科, 主要研究方向: 开源软件治理、物联网技术、容器云技术。