基于存储库数据挖掘的开源软件成功度量方法

宁德军[1],叶培根[2],刘 琴[1],李 梅[3]
(1. 同济大学软件学院,上海 201210; 2. 上海大学通信与信息工程学院,上海 201210;
3. 中国科学院上海高等研究院智慧城市研究中心,上海 201210)

摘 要: 开源软件已广泛应用于各软件领域,例如操作系统、容器等,但目前尚没有一种能够综合度量开源软件的方法.我们在用户兴趣度量和开发参与度量的基础上,提出了一种能够克服度量维度单一的局限性的度量方法.基于 DM 模型、软件生存力模型等相关文献研究和开源软件存储库数据挖掘,通过对项目过程数据进行聚类、主成分分析、回归分析和对开发过程的思考,本文提出一种基于存储库数据和统计学算法的开源软件成功度度量模型.并通过与用户兴趣度量结果和开发参与度量结果对比分析,证明本文的度量模型能够基于可自动无扰采集的存储库数据,更全面的衡量开源软件项目的成功.度量模型可应用于企业选择优质开源项目、学术研究、智能项目推荐等领域.

关键词: 开源软件; 主成分分析; 成功度量模型; 数据挖掘; 量化分析; 软件工程; DM 模型; 软件生存力模型

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2018)12-2930-06

电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.12.015

Open Source Software Success Measurement Method Based on Mining Software Repository

NING De-jun [1], YE Pei-gen [2], QIN Liu [1], MEI Li [3]
(1. *School of Software Engineering, Tongji University, Shanghai* 201210, *China*;
2. *School of Communication and Information Engineering, Shanghai University, Shanghai* 201210, *China*;
3. *Shanghai Advanced Research Institute, Chinese Academy of Science, Shanghai* 201210, *China*)

Abstract: The open source software (OSS) is widely used in various software fields, such as operating system, container, etc. But there is no method to measure open source software comprehensively. Based on the measurement of user interest and the development of participation metrics, we propose a method that can overcome the single limitation of the metric dimension. Based on DM model, software viability model and other relevant literature research, mining of software repository, we consider the OSS development process and propose an OSS success evaluation model through clustering, principal component analysis and regression analysis. By comparing the metric score of user interest method and development participation method, the evaluation model can be used to measure the success of OSS projects based on the data collected automatically without interference. The evaluation model can be applied to select high quality open source projects, academic research, intelligent project recommendation, etc.

Key words: open source software; principal component analysis; success measurement model; data mining; quantitative analysis; software engineering; DM model; software viability model

收稿日期:2017-11-21;修回日期:2018-04-11 责任编辑:覃怀银
基金项目:中国科学院战略性先导科技专项(No. XDA06000000);国家重点研发计划项目(No. 2016YFB1000805);上海市"科技创新行动计划"基础研究重点科技项目(No. 17511107303,17511110202);上海市扬帆计划(No. 17YF1420500)

1　引言

上世纪末以来开源软件(OSS)发展迅速,极大地影响了软件行业的市场格局,颠覆了一些经典软件工程理论.开源软件项目并没有传统软件工程项目中显式的组织结构和过程安排,也没有传统软件开发团队的过程成熟度等级,但所开发出的软件却可以与传统商业软件相抗衡,这引起了学术界和企业界的广泛关注[1,2].

开源软件的成功有力的驳斥了只有私有化的知识产权能够推动创新的言论[3].然而,开源软件的失败率是较高的[4].并且,对开源软件项目的评估也不同于商业软件系统[5].所以需要一种度量方法能够获知当前开源软件的项目状态和推演开源软件项目的未来发展情况,以便用户选择所需的项目. DeLone 和 McLean [6] 最早提出了度量信息系统的 DM 模型. Polancic 等人提出开源项目质量度量模型[7]. Rajdeep Grewal et al. [8]从技术和商业两个角度来分析项目的成功.

本文试图从 Github 开源社区软件开发演进过程的数据入手,通过存储库数据挖掘的方法找到一种能够对开源软件的成功度进行客观量化度量的、简单易行的新方法,从而使能开源软件开发团队快速了解所开发软件的成功程度和团队状况,明确影响开源软件成功的关键因素,指导开源软件的领导者采取合适的行动.

2　项目成功度量概念模型

开源软件的协作开发依赖于不同风格和技能的开源软件开发人员,每个开发者采用独立并行的协同开发机制,并将修改结果提交给开源项目的核心团队选择最好的代码进入发布[9]. 如图 1 所示,开源软件开发生态系统概念模型(简称 OSSEM)主要由开发者团队,开发协作平台和过程,开源软件和通过社区获取并使用软件的用户组成[10]. 开发者基于社区并行协作开发开源软件,整个软件开发运营过程、交付的代码和工件开放透明,吸引越来越多的用户使用软件并给予反馈,逐步形成闭环螺旋式发展的开源软件生态环境.

通过文献综述,可以看到几乎所有开源软件度量模型都关注在开源软件、开源团队、用户参与和项目的开发过程、技术环境等方面的因素及其相关性的分析.因此,结合开源软件开发生态系统概念模型和团队的软件工程实践经验,本文进行了以下假设:

(1)H1:影响开源软件成功的因素会以软件生命周期数据的方式记录在存储库中;

(2)H2:影响开源软件成功与开源软件质量、开源软件团队及开发过程、用户参与行为等因素相关;

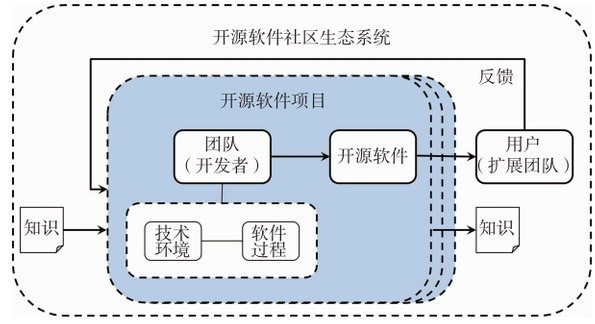

图1　开源软件开发生态系统概念模型 (OSSEM)

如果开源软件成功度 F_{success} 是对开源软件开发生态系统进行总体评价的派生度量元,是指开源软件成功的程度,那么如果假设成立,则开源软件成功度由开源软件开发过程中的团队、开源软件、用户和项目相关的各种因素共同决定的.它们之间的关系可以形式化地表示为式(1):

$$F_{\text{success}} = f(f_{\text{dev}}, f_{\text{pop}}, f_{\text{op}}, f_{\text{vit}}) \tag{1}$$

开源软件成功度的度量,可以通过存储库数据挖掘的方法给出定量的评价,结果可以是具体的得分.

3　模型验证

随着开源软件社区的迅速发展和开源社区协作平台的日益成熟,与传统的软件工程相比,开源软件开发过程中形成了庞大的开源社区存储库,这为我们通过软件存储库数据挖掘技术了解开源软件开发过程和项目状况提供了丰富的数据资源.本章采用的总体分析流程如图 2 所示:

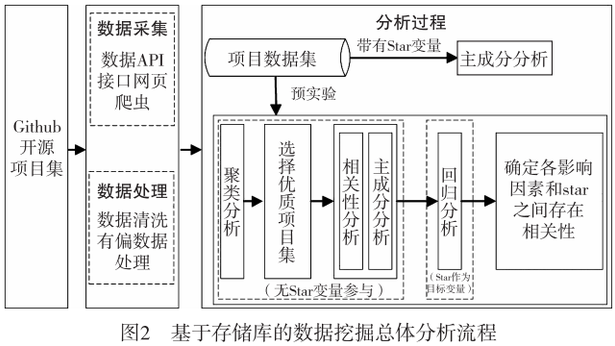

图2　基于存储库的数据挖掘总体分析流程

3.1　数据采集过程

本次采集数据中共包含 27 万个项目的详细数据.其中通过抓取 follower 数排名前 1000 人的项目,共得到约 32000 条记录.随后通过在这 1000 名开发者的关注者群体中随机采样,抽取了约 23 万多个项目.星标数分布情况如表 1 所示.可见此次选取的项目可以较好地反映出 github 社区中较活跃项目的情况.

表 1 Star 数分布情况

	全 github 社区	followers 数前 1000 人		本次抓取的样本集	
		数值	占总体百分比	数值	占总体百分比
总项目数	82,000,000 +	32,438	0.04%	270,287	0.33%
stars 数超过 1000	10,271	1,192	11.61%	8,763	85.32%
stars 数超过 500	20,800	2,027	9.75%	16,875	81.13%
stars 数超过 100	82,646	5,107	6.18%	49,355	59.72%

3.2 数据处理

通过对 GitHub 中软件存储库进行数据爬取,我们获得了27 万条存储库信息. 我们首先通过异常值处理,人工剔除异常记录等处理手段,对原始数据进行数据清洗,保留下约 13 万条存储库记录. 然后,通过数据审查,发现其整体的数据分布是有偏的,并非正态分布,考虑到 Github 开源社区中的开发者在学习使用平台时会建立许多用于入门的练习项目,一些经常参与活跃项目开发的软件工程师也会经常建立新的存储库用于测试,还有一些项目长期疏于管理且无人关注成为了死项目等诸如此类的原因,我们尝试使用聚类方法从数据层面上对项目进行分类. 在聚类结果中去除异常记录,最终获得 90956 条软件存储库的数据集合作为研究对象.

3.3 分析过程

为了验证 H1 和 H2,我们使用经过处理的90956 个项目数据,进行主成分分析,试图从存储库数据中直接得出影响项目的因素,并通过回归拟合处理分析验证主成分的软件工程实践意义. 由于 Github 社区通常用 Star 数作为衡量项目流行度的度量元,而流行度从某种程度上代表了用户参与项目的程度,因此,先不包含 star 进行主成分分析,总方差解释结果如表 2 所示.

表 2 主成分分析总方差解释结果

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	2.615	32.691	32.691	2.615	32.691	32.691	1.987	24.837	24.837
2	1.968	24.595	57.286	1.968	24.595	57.286	1.723	21.536	46.373
3	1.257	15.712	72.998	1.257	15.712	72.998	1.661	20.760	67.133
4	0.877	10.962	83.961	0.877	10.962	83.961	1.346	16.827	83.961
5	0.658	8.227	92.188						
6	0.340	4.252	96.439						
7	0.272	3.397	99.837						
8	0.013	0.163	100.000						

从表 2 的主成分分析的总方差解释结果中可以看到前四项主成分的特征值之和占总方差的 83.96%,说明前四个因子已经对大多数数据给出了充分的解释,信息损失在可接受的范围内. 通过表 3 公因子方差分析结果可以看出,对原始字段信息的保留绝大多数大于 80%,满足主成分分析的要求,因此,通过主成分分析确定了四个主成分.

表 3 主成分分析公因子方差与成分矩阵

	Component					
	Initial	Extraction	1	2	3	4
forks	1.000	.837	.008	.201	.891	-.049
watchers	1.000	.822	.018	.251	.866	-.095
open_issue	1.000	.867	.031	.906	.200	-.066
close_issues	1.000	.857	.018	.887	.260	-.055
codesize	1.000	.993	.996	.029	.015	-.005
mlangsize	1.000	.993	.996	.019	.011	-.003
push_diff_record	1.000	.670	-.001	-.104	-.030	.811
update_diff_record	1.000	.677	-.005	.003	-.090	.818

表3 结果显示:第一主成分主要由 codesize、mlangsize 所解释,主要体现开源软件规模;第二主成分主要由 open-issue、close-issue 所解释,主要体现用户问题的多少和团队对问题的响应速度,它反映了开源软件的质量情况;第三主成分主要由 watchers、folks 所解释,基本代表了开源软件被用户关注和用户参与的程度,体现了用户参与水平;第四主成分主要由 push-diff-record、update-diff-record 所解释,基本体现了项目团队本身的活跃程度. 这个分析结果有力地验证了提出假设 H1 和 H2 的真实性.

一般用户对开源软件参与度越高,用户对开源软件的喜爱程度越高,用户给开源软件的评价等级越好. 目前,开源社区一般用 Star 数来度量开源软件的流行度. 因此,我们把以上主成分分析所得出得四个主成分与 Star 数做了回归拟合分析,用 Star 数作为因变量,回归模型的汇总结果如表 4 所示,调整后的拟合系数为 63.9%,拟合结果较好,其中因子 3 和 Star 具有强相关性.

表 4　回归模型汇总

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.728[a]	.529	.529	805.155
2	.793[b]	.629	.629	714.644
3	.799[c]	.639	.639	705.317
4	.799[d]	.639	.639	705.298

(a). Predictors:(Constant), $ F-因子-3

(b). Predictors:(Constant), $ F-因子-3, $ F-因子-2

(c). Predictors:(Constant), $ F-因子-3, $ F-因子-2, $ F-因子-4

(d). Predictors:(Constant), $ F-因子-3, $ F-因子-2, $ F-因子-4, $ F-因子-1

为了基于存储库数据挖掘找出简单易行的“开源软件成功度”的度量方法,在包含 Star 的情况下,我们再次对 13 万条数据进行主成分分析,并将提取因子的阈值设为 0.8,其主成分分析结果如表 5 所示:

表 5　分析过程主成分分析总方差解释结果

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings	Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Cumulative %	Total	% of Variance	Cumulative %
1	3.294	23.52	23.26	23.526	2.257	16.124	16.124
2	2.493	17.809	41.335	41.335	2.123	15.167	31.290
3	1.725	12.323	53.658	53.658	2.044	14.601	45.892
4	1.195	8.535	62.193	62.193	1.306	9.330	55.221
5	1.048	7.487	69.680	69.680	1.294	9.245	64.466
6	1.009	7.204	76.885	76.885	1.284	9.170	73.636
7	.821	5.864	82.749	82.749	1.276	9.113	82.749
8	.623	4.449	87.198				
9	.426	3.045	90.244				
10	.411	2.938	93.182				
11	.380	2.714	95.895				
12	.353	2.521	98.416				
13	.210	1.502	99.918				
14	.011	.082	100.000				

表5 展示了总方差解释结果. 前七项主成分的累积贡献率达到 82.75%,超过 80%,保留了绝大部分信息,信息损失在可接受的范围内. 以上结果表明此次主成分分析较为成功,可继续分析各主成分的业务含义. 根据表5 中对 13 万条数据进行的改进后的主成分分析,降维后确定为 7 个主成分,形式化表示为 $X_1 \sim X_7$,用 7 个主成分代表原有的 14 个字段,概括原始变量所包含信息的 82.75%. 第一主成分 X_1 主要由 codesize、mlangsize 所解释,它们更多地体现开源软件规模;第二主成分 X_2 主要由 stars、watchers、folks 所解释,表征用户参与度;第三主成分 X_3 主要由 open-issue、close-issue、close-pull 所解释,表征了软件质量;其余四个主成分主要由 create-diff-record、push-diff-record、update-diff-record、Contributors、mainbranch _ commits、forks、open-pull 所解释,更多地与团队活跃度和开发过程有关.

4 基于统计学的评估模型

通过对主成分分析结果的解读,并结合软件开发过程中的实践,如表 6 所示,主成分分析得出的七个主成分完全可以映射到本文第 3 章所提出的假设 2 和假设 3 中所包含的开源软件开发过程的四个维度,结果与第一次主成分分析的结果基本一致.

表 6 开源项目成功要素

维度	指标	形式化	主成分构成
软件维度	软件规模	f_{dev}	X_1(codesize、mlangsize)
用户维度	用户参与度	f_{pop}	X_2(stars、watchers、folks)
运营维度	软件质量	f_{op}	X_3(open-issue、close-issue、close-pull)
团队维度	团队活性	f_{vit}	X_4(create-diff-record、push-diff-record) X_5(push-diff-record、update-diff-record) X_6(contributors、mainbranch_commits) X_7(forks、open-pull)

因此,通过以上分析可知不同的数据能够从不同的维度衡量开源项目的相应指标,由此我们可以得出一种由以上四种成功关键因素共同构成开源软件成功度量方法,针对当前数据集,计算公式如式(2):

$$F_{success} = W_{dev} \cdot f_{dev} + W_{pop} \cdot f_{pop} + W_{op} \cdot f_{op} + W_{vit} \cdot f_{vit} \tag{2}$$

其中,软件规模由第一主成分得分 X_1 表征,能够衡量开发绩效,计算公式如(3)所示:

$$f_{dev} = X_1 \tag{3}$$

用户参与度由第二主成分得分 X_2 构成,能够衡量项目流行度,计算公式如(4)所示:

$$f_{pop} = X_2 \tag{4}$$

运营过程中的软件质量由第三主成分得分 X_3 构成,能够衡量项目绩效,计算公式如(5)所示:

$$f_{op} = X_3 \tag{5}$$

团队绩效由第四到第七主成分构成,能够衡量团队的活跃度,计算公式如式(6)所示:

$$f_{vit} = \frac{W_4}{\sum W_i} \cdot X_4 + \frac{W_5}{\sum W_i} \cdot X_5 + \frac{W_6}{\sum W_i} \cdot X_6 + \frac{W_7}{\sum W_i} \cdot X_7, i = 4,5,6,7 \tag{6}$$

W_i 为各主成分权重,其具体数值的确定是由各字段对降维后总信息量的贡献比率所确定的,权重计算方法为:W_i = 贡献度/累计贡献度.

针对当前数据集分析结果所得权重,可得出开源软件成功度量模型,如式(7)所示:

$$F_{success} = 0.28 \cdot f_{qos} + 0.22 \cdot f_{user} + 0.15 \cdot f_{proj} + 0.35 \cdot f_{team} \tag{7}$$

5 度量结果分析对比

目前,许多研究者对项目成功的研究从以下两个维度分析:用户兴趣度量和开发参与度量[11,12]. Star 数和 Watcher 数是 Github 社区对项目的用户关注行为进行记录的重要指标. 我们使用这两个指标作为用户兴趣度量的依据. 针对于本次采样的数据集,对比三种度量方法结果(如图 3)的差异有以下两种情况:

(1)某项目都有出现在三种度量结果的前十名,但在各方法中具体排名不同. 以 bootstrap 为例:从用户兴趣的角度对项目进行排名,bootstrap 排名第二,如图 3 中(*b*)所示;从开发参与的角度进行排名,bootstrap 排名第三,如图 3 中(*c*)所示;本文所提方法由于综合了用户兴趣和开发参与等因素,避免了单一因素的片面性,其项目排名为第一,体现了 bootstrap 项目在用户兴趣和开发参与等方面的综合成功.

Ranking	repository	metric
1	bootstrap	1
2	Spoon-Knife	0.994
3	freeCodeCamp	0.983
4	ZF2Package	0.946
5	ProgrammingAssignment2	0.756
6	linux	0.673
7	angular.js	0.605
8	android_kernel_oneplus_msm8996	0.53
9	PerspicuOS	0.529
10	legacy-homebrew	0.526

(*a*) 本文度量方法结果

Ranking	repository	Star + Watcher
1	freeCodeCamp	1.995
2	bootstrap	1.452
3	linux	0.964
4	angular.js	0.888
5	react	0.875
6	tensorflow	0.863
7	d3	0.698
8	jquery	0.687
9	laravel	0.676
10	You-Dont-Know-JS	0.668

(*b*) 用户兴趣度量结果

Ranking	repository	folks
1	ProgrammingAssignment2	99512
2	Spoon-Knife	89483
3	bootstrap	48877
4	angular.js	27324
5	courses	23692
6	frontend-nanodegree-resume	23147
7	tensorflow	22422
8	SmartThingsPublic	16239
9	d3	16165
10	linux	16132

(*c*) 开发参与度量结果

图3 三种度量方法的对比

(2) 某项目部分出现在其他两种度量方法中,在本文度量方法排名前列. 以 ProgrammingAssignment2 为例,在开发参与度量结果上排名第一,但却在用户兴趣度量上未进入前十名,经过对该项目的分析发现,该项目为一个编程任务项目,有较为固定的一群算法爱好者不断提交代码,其用户量没有过多的增长,但项目一直较为活跃. 本文方法由于综合了多种度量维度,并且开发要素是软件项目中最为重要的度量元,此项目成为排名靠前的优质项目.

综合以上差异情况的分析,我们发现本文所提出的度量方法能够有效的综合多种维度,避免了从单一角度对项目进行评价所产生的片面性,并且随着对开

源软件理解的不断深入,可以增加更多的维度到此度量方法中使得结果更加客观、准确.

6 总结与展望

根据以上分析,本文基于存储库数据挖掘方法,给出了一种能够方便评价开源软件成功度的量化度量方法.该方法基于开源社区存储库的实时数据,完全客观地描述了一个开源软件的质量、用户参与度、团队活性和软件规模等关键成功要素,使开源软件团队能够方便地了解所开发软件的成功程度和水平.在未来的研究工作中,我们将对影响开源软件成功的具体因素进行分析,为开源软件项目开发提供故障诊断和最佳实践推荐;同时,可以对开源软件进行分类,并基于本文提供的成功度度量方法获取开源软件的分类成功度排名,实现面向不同分类的智能项目推荐.

参考文献

[1] Krogh G V,E V Hippel. Special issue on open source software development[J]. Research Policy,2003,32(7):1149 – 1157.

[2] Yamauchi Y, Yokozawa M, Shinohara T. Collaboration with lean media:how open-source software succeeds[A]. ACM Conference on Computer Supported Cooperative Work[C]. New York,USA:ACM,2000. 329 – 338.

[3] Osterloh M, Rota S. Trust and community in open source software production[J]. Analyse & Kritik, 2016, 26(1): 279 – 301.

[4] Krishnamurthy S. Cave or community?:an empirical examination of 100 mature open source projects[J]. Social Science Electronic Publishing,2002,7(6):960 – 972.

[5] Perens B. The Open Source Definition[M]. USA: Open Sources:Voices from the Open Source Revolution, 2013. 171 – 188.

[6] Petter S, Delone W, Mclean E. Measuring information systems success: models, dimensions, measures, and interrelationships[J]. European Journal of Information Systems, 2008,17(3):236 – 263.

[7] Polancic G, Horvat R V, Rozman T. Comparative assessment of open source software using easy accessible data [A]. International Conference on Information Technology Interfaces[C]. Cavtat, Croatia: IEEE, 2004. 673 – 678.

[8] Grewal R, Lilien G L, Mallapragada G. Location, location, location: How network embeddedness affects project success in open source systems[J]. Management Science, 2006,52(7):1043 – 1056.

[9] Margan D, Candrlic S. The success of open source software:a review[A]. International Convention on Information and Communication Technology, Electronics and Microelectronics[C]. Irbid, Jordan: IEEE, 2015. 1463 – 1468.

[10] 张得光,李兵,何鹏. 基于软件生态系统的开源社区特性研究[J]. 计算机工程,2015,41(11):106 – 113.
Zhang De-guang, Li Bing, He Peng. Characteristic study of open-source community based on software ecosystem [J]. Computer Engineering, 2015, 41(11): 106 – 113. (in Chinese)

[11] Subramaniam C, Sen R, Nelson M L. Determinants of Open Source Software Project Success: A Longitudinal Study [M]. Amsterdam, Netherlands: Elsevier Science Publishers B V, 2009.

[12] Neamtiu I, Xie G, Chen J. TowardsA Better Understanding of Software Evolution: An Empirical Study on Open Source Software[M]. Hoboken, New Jersey, USA: John Wiley & Sons, Inc, 2013.

作者简介

宁德军　男,1972 年 7 月出生于黑龙江省.教授级工程师,CCF 会员.长期从事下一代软件工程技术、大数据智能技术和海云协同计算研究.
E-mail:ningdj@ sari. ac. cn

叶培根　男,1992 年 2 月出生于河南省濮阳市.研究生,主要研究方向为软工数据智能和海云计算.
E-mail:yepg@ sari. ac. cn

刘　琴　女,同济大学软件学院教授、博士生导师,CCF 会员.主要研究方向为软件与数据工程、异构数据相关性分析、基于历史数据的预测模型与应用.
E-mail:qin. liu@ tongji. edu. cn

李　梅　女,1993 年 10 月出生于江苏东台.研究生.主要研究方向为软工数据智能、数据挖掘、智能推荐.
E-mail:limei@ sari. ac. cn