

开源软件关键开发者类型及协作网络鲁棒性分析



卢冬冬 吴洁 刘鹏 盛永祥

江苏科技大学经济管理学院 江苏 镇江 212003

(741799656@qq.com)

摘要 文中以开源软件 AngularJS 项目为例,探究关键开发者类型和协作网络鲁棒性。通过抽取代码修订关系构建开发者协作网络,分析网络的结构和功能。综合开发者的结构与功能属性进行类型划分,探究不同类型开发者流失后网络的结构和功能鲁棒性,以此识别出关键开发者类型。最后模拟新开发者的加入机制,探讨网络鲁棒性的提升策略。研究发现:开发者的结构和功能属性的不对称性导致了开发者协作网络的结构和功能鲁棒性的不一致性;与传统方法相比,对开发者进行类型划分能够更有效地识别关键开发者类型;在社团内部较活跃、与其他社团之间存在密切联系并且拥有大量贡献度的中央核心型开发者对网络鲁棒性影响最大;拥有较大初始度且选择倾向性连接的新开发者加入机制能够有效提高网络鲁棒性。

关键词: 开源软件;AngularJS;开发者协作网络;关键开发者类型;鲁棒性

中图法分类号 TP391

Analysis of Key Developer Type and Robustness of Collaboration Network in Open Source Software

LU Dong-dong, WU Jie, LIU Peng and SHENG Yong-xiang

School of Economics and Management, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003, China

Abstract Taking the open source software project AngularJS as an example, this paper studies the key developer type and the robustness of collaboration network in open source software. The network is constructed by the project code-collaboration relationships to analyze the structure and function. The nodes in the network are classified into different types according to their structure features and function features. Then, the impact of different developers turnover on the network structure and functional robustness is explored to identify the key developer type. What's more, the promotion strategy on robustness of the network is promoted by simulating the joining mechanism for new developers. The study shows that the developer's structure and function features are asymmetrical, which is the reason for the structure and function robustness on the network are inconsistent. Compared with traditional methods, the type division of developers can more effectively identify the key developer type. Central core developers who are active in the community and have connections with other communities also have a large number of contributions, and this type of developers have the greatest impact on the robustness of the network. New developers with higher initial degree and choosing a preference connection mechanism can effectively improve the robustness of collaboration network.

Keywords Open source software, AngularJS, Developer collaboration network, Key developer type, Robustness

近年来,以 GitHub, SourceForge.net 为典型代表的开源社区孕育了一大批优秀的开源软件,如:OpenStack, Apache Spark, TensorFlow 等。具有不同经历和背景、来自世界各地的开发者因为兴趣爱好、就业需求等原因自发通过开源软件社区对感兴趣的软件项目贡献自己的力量,促进软件项目的实现、更新与迭代^[1-3]。与传统商业软件的开发模式不同的是:Raymond^[4]将传统商业软件的开发模式和开源软件的开发模式分别比作“大教堂”(Cathedral)和“集市”(Bazaar),即开源软件社区具有极大的开放性,这使得开发者可以自由地进入或退出项目。但有研究表明,软件开发是一项高度复杂的知识创造活动^[5]，“集市”模式虽然形象生动,但不能充分解

释其工作机理,如社区中的 20/80 法则(社区中少于 20% 的开发者完成了超过 80% 的工作^[3])。在这种自组织模式下,开发者的流动特性,特别是关键类型的开发者流失会对项目的开发进程造成巨大影响。因此,探究关键类型开发者和协作网络鲁棒性具有重要意义,一方面通过分析协作网络面临不同类型开发者流失时的鲁棒性可以得出关键开发者类型,分析其行为特点能够帮助项目管理者更有效地厘清开源软件社区中开发者的协作机制;另一方面,当网络面临关键类型开发者流失时也可以通过探究鲁棒性的提升策略,来更好地为项目管理者提供对策建议。

本文以当前较为流行的开源前端 JS 框架 AngularJS 项

到稿日期:2020-03-24 返修日期:2020-07-10 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金项目(71871108);国家社会科学基金项目(19FGLB029)

This work was supported by the National Natural Science Foundation of China(71871108) and National Social Science Foundation of China(19FGLB029).

通信作者:吴洁(0511wujie@163.com)

目为例,抓取其在 GitHub 托管的代码提交记录并构建开发者协作网络。通过划分开发者类型,探究不同类型开发者流失对网络鲁棒性的影响,从而识别出关键开发者类型,并探究网络面临关键类型开发者流失时的鲁棒性提升策略。

1 相关研究

当前,越来越多的研究者投身开源软件社区的相关研究中,涉及计算机科学、知识创新、管理学等诸多领域^[6-9]。众多学者从网络视角构建开发者协作网络进行相关研究,这使我们对开源软件开发者间的合作模式、协调机制有了更深刻的认识^[6,10-11]。在开发者类型划分方面,早期 Mockus 等^[12]通过计量代码提交量区分核心和边缘开发者。后来 Nakakoji 等^[13]提出了“洋葱模型”将开发者划分为 8 类。目前已有众多基于复杂网络的研究。Wang 等^[14]通过构建开发者合作网络,发现了项目演化过程中的 5 种开发者类型。Xia 等^[7]通过节点度大小将开发者划分为核心和边缘开发者。Zhou 等^[15]通过度、介数的大小对开源社区成员进行重要性识别。

在网络鲁棒性方面,目前开源软件社区中对其研究相对较少。Osterloh 等^[16]研究发现社区成员的技能背景差异化和关注点多样化能够更好地维持网络鲁棒性。Li 等^[17]在研究产学研创新网络鲁棒性时发现要重点保护中度节点。Yu 等^[18]通过研究协同创新网络的鲁棒性发现了最优化模型。Wood^[19]研究发现犯罪合作网络是异配型网络,面对攻击时具有较差的鲁棒性。Zhang 等^[20]基于生命周期模型发现不同阶段开源社区的知识协作网络的鲁棒性不同。

从上述研究可以看出:目前关于鲁棒性的研究多从结构角度进行衡量,但作为复杂密集型知识创造系统的开源软件社区,仅探究其网络结构鲁棒性具有一定的局限性;另外,网络中对于节点的类型划分也只是基于节点的结构属性(如度、介数等)进行重要性排序。因此,有必要结合功能角度对开发者进行类型划分,并探究网络的功能鲁棒性。

2 研究框架与网络分析

2.1 研究框架

图 1 是本文的研究框架。本文首先从 GitHub 抓取 AngularJS 项目的代码提交记录;随后,依据这些记录抽取开发者协作关系来构建网络,分析网络的结构和功能;再从结构和功能视角对开发者类型进行划分;最后仿真模拟不同类型开发者的流失,探究网络的鲁棒性,识别出关键开发者类型;在此基础上仿真模拟新开发者的加入行为,提出网络鲁棒性的提升策略。

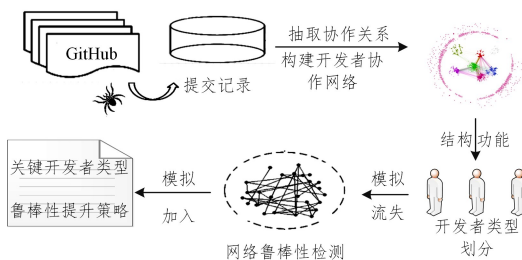


图 1 研究框架

Fig. 1 Research framework

2.2 开发者协作网络的构建

AngularJS 通过 GIT 版本控制系统在 Github 托管其项目库,目前主要包括 Angular, Angular CLI, Angular Material, AngularFire 等多个子项目。本文通过 Git 指令收集了 AngularJS 的 7 个主要子项目库从 2010 年 8 月(项目初始期)到 2019 年 1 月的所有代码提交记录,共计 38764 条。这些记录包括开发者注册邮箱、昵称、提交日期、修改文件信息以及代码增减行数。在此基础上,通过 Python 语言实现开发者协作关系的抽取,将同一版本时间段内针对同一文件存在代码提交行为的开发者视为存在协作关系。为了排除噪声,将开发者邮箱作为识别开发者的唯一标识,将其抽象成网络中的节点,开发者之间的协作关系被抽象成网络中的连边,进而构建开发者协作网络。代码提交记录和网络构建方法如图 2 所示。

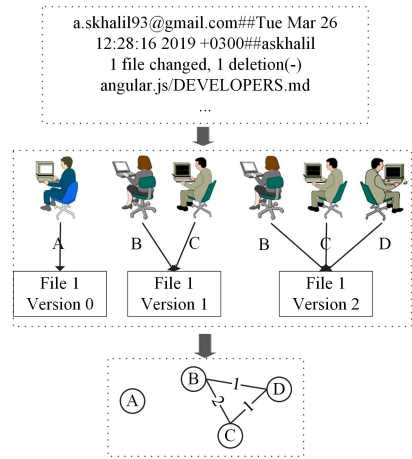


图 2 开发者协作网络的构建流程

Fig. 2 Process of developer collaboration network construction

2.3 开发者协作网络的结构分析

采用社会网络分析的方法(Social Network Analysis, SNA)对网络结构进行分析。网络规模用节点数量表示;平均度为网络中所有节点度的平均值;节点度为网络中与目标节点直接相连的节点数量;平均加权度为网络中所有节点加权度的平均值;节点加权度为与目标节点相连的所有边的权重之和;网络直径为网络中任意两节点之间连边数量的最大值;平均聚类系数为网络中三角形量与三元组量的比值;平均路径长度为网络中任意两节点需要最小连边数的均值;模块度为衡量网络社团结构的指标,本文采用 Louvain 算法^[21]进行计算。

表 1 列出了网络结构指标,可以发现截至 2019 年 1 月已有 3982 名开发者参与到 AngularJS 的开发过程,充分体现了开源软件社区的集体智慧。表中平均聚类系数为 0.66,平均路径长度为 4.135,说明该网络符合小世界特性。模块度为 0.593,说明网络中存在明显的社团结构。

表 1 开发者协作网络拓扑参数

Table 1 Developer collaboration network topology parameters

网络规模 (节点数)	平均度	平均 加权度	网络 直径	平均 聚类系数	平均路径 长度	模块度
3 982	4.35	9.264	12	0.656	4.135	0.593

图 3 为双对数坐标下的度分布拟合,其符合幂律分布,拟合优度较高 $R^2=0.94907$,这说明网络存在着明显的核心边缘结构^[22]。

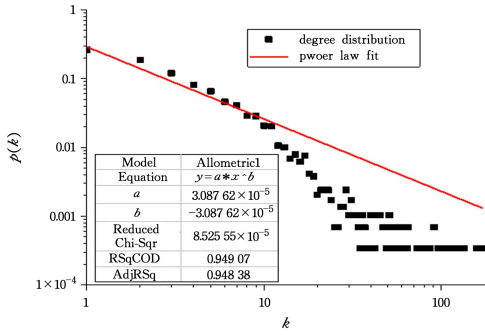


图3 度分布

Fig. 3 Degree distribution

2.4 开发者协作网络功能分析

AngularJS 包含多个用于存放各自子项目的代码库。开发者的代码提交行为均需得到项目管理者挑选,只有最优的内容才会被 merge 到项目中,因此被记录下来的提交行为都是具有价值的。若将各开发者的提交次数视为贡献度,以开发者 `pete@bacondarwin.com` 为例(见图4),其因在 `angular.js` 子项目有累计 838 次提交行为,所以在该子项目的贡献度为 838。以此类推,该开发者的总贡献度为

1 910,共参与 5 个子项目。

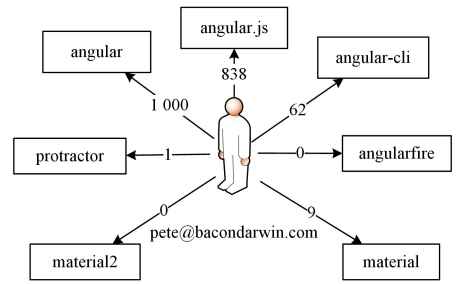


图4 开发者的贡献度度量方法

Fig. 4 Method of developer's contribution

因此,可用二维列联表展示各开发者对子项目的贡献度(见表2)。开发者会根据自己的喜好参与到其中一个或多个子项目的开发进程中,各子项目间的贡献量也因其自身特点和代码工作量而差异较大。社区的总贡献度为 38764,计算可得人均贡献度为 9.73,而仅仅表2列出的 Id 编号为 2,3,4 的开发者的贡献度都已经分别高达 340,1 266,1 910,均远高于人均贡献度,由此也可判定社区中少数的开发者承担了项目的大部分工作量。

表2 开发者与子项目的二维列联表

Table 2 Developer and subproject two-dimensional contingency table

Id	angular	angular.js	angular-cli	angularfire	material	material2	protractor	总计
1	2	15	4	1	2	5	3	32
2	118	65	21	130	1	0	5	340
3	145	0	1	0	97	1 022	1	1 266
4	1 000	838	62	0	9	0	1	1 910
...
总计	13 685	8 896	3 850	653	4 549	5 349	1 782	38 764

3 开发者类型划分

已有的大多数研究对于开发者的类型是按照节点度的大小排序,或按照提交次数进行排序。但开发者类型不应该仅由其在网络中的位置或按工作量进行简单划分,有必要将开发者的结构和功能属性综合起来进行类型划分。

3.1 结构角度

网络结构中,传统意义上的节点的结构重要性指标仅仅是度或介数等简单指标。但开发者协作网络中存在着明显的社团结构,有必要参考 Guimera^[23]提出的考虑社团结构的节点类型划分算法,该算法提出了模块内参与度 Z 和参与系数 P 。

点呈现出高度的聚集性(B区)。节点模块内参与度 Z 值差异较大,高 Z 值节点数量少,低 Z 值节点数量多。而节点参与系数 P 值分布相对均匀。因此,为使开发者类型划分结果更符合开发者协作网络的中央边缘结构,可将 A 区内的开发者节点视为中央开发者($Z \geq 2, P \neq 0$),这类开发者往往在其社团内部参与度较高,又与外界其他社团保持着相对较好的联系。其余各区域的开发者节点则被视为边缘开发者。

$$Z_i = \frac{\kappa_i - \bar{k}_{s_i}}{\sigma_{k_{s_i}}} \tag{1}$$

式(1)用于度量节点 i 与其所在社团内部其他节点的联系。其中, κ_i 表示节点 i 在社团内的连边数量, \bar{k}_{s_i} 表示节点 i 所在社团的平均度, $\sigma_{k_{s_i}}$ 表示该社团的度标准差。 Z_i 越大表示节点 i 与其所在社团内的其他节点联系越密切。

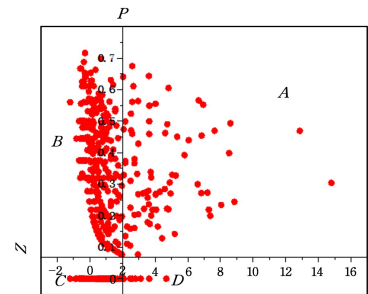


图5 模块内参与度和参与系数分布图

Fig. 5 Z and P distribution

$$P_i = 1 - \sum_{s=1}^{N_M} \left(\frac{k_{is}}{k_i} \right)^2 \tag{2}$$

式(2)用于度量节点 i 与其他社团的联系。其中, N_M 表示整个网络中社团最大编号, k_{is} 表示节点 i 在社团 s 内的连边数量, k_i 表示节点 i 的度。如果一个节点与所有社团均有联系,那么它的参与系数接近于 1;倘若它的所有连接均在自己的社团内部,那么它的参与系数为 0。

3.2 功能角度

从功能角度出发,可以将开发者的贡献度作为划分的指标。图6给出了双对数下整个项目社区中的所有开发者的贡献度分布,可见开发者的贡献度具有较大的差异,高贡献度的开发者相对较少,完全符合“二八原则”。为方便起见,用 100 作为划分核心贡献者与一般贡献者的临界值。贡献度不低于 100 的成员共有 65 人,占社区开发者数量的 1.6%,视为核心

图5为模块内参与度和参与系数分布图。可以看出,节

开发者节点。贡献度低于 100 的成员则为一般开发者节点。

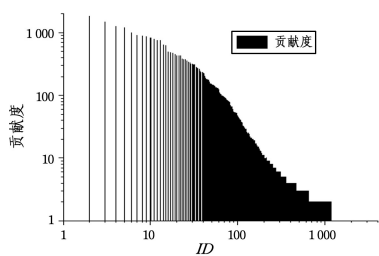


图 6 社区开发者的贡献度

Fig. 6 Contribution of developers

综上,从结构角度可将开发者划分为中央开发者和边缘开发者,从功能角度亦可将开发者划分为核心开发者和一般开发者。若将开发者的结构和功能属性综合考虑,则可采用集合交集的思想,将开发者划分为以下 4 种类型:中央核心开发者(40 人)、中央一般开发者(41 人)、边缘核心开发者(25 人)、边缘一般开发者(3876 人)。

4 网络鲁棒性分析

4.1 鲁棒性指标

通常用于衡量网络性能的指标有:网络聚集系数、平均路径长度、最大连通图、网络效率等^[24]。网络结构中最大连通图能够衡量节点间关系的紧密性,当最大连通图瓦解甚至崩塌时表示网络遭受重创。因此,本文拟采用最大连通图相对大小 S 来衡量网络的结构鲁棒性。

$$S = \frac{N'}{N} \quad (3)$$

其中, N 为网络中的节点总数, N' 为最大连通图中的节点数量。开发者协作网络最大连通图相对大小表示开发者间的紧密联系程度, S 越大代表开发者间协作度越高,网络结构相对稳定,开发者流失不会对系统内的协作关系造成太大的影响。

另一方面,可将项目的贡献度作为开发者协作网络的功能鲁棒性指标。倘若开发者流失后,项目的贡献度骤减,则说明项目的开发进程受到严重的影响。

4.2 鲁棒性分析

基于对开发者类型的划分,引入一个问题:不同指标下开发者的排序对网络结构和功能鲁棒性的影响是否具有 consistency? 本文拟通过重复移除不同重要性排序下的前 50 个开发者,模拟该开源软件社区中开发者的流失,探究不同指标重要性排序下开发者对网络结构和功能鲁棒性的影响。

图 7 给出不同指标重要性排序下开发者对网络鲁棒性的影响。总体而言,贡献度和 Z 值大小排序下的节点流失对网络鲁棒性的影响明显大于 P 值大小排序下的节点流失。当流失 1000 个开发者时,网络鲁棒性遭受重创,均逐渐趋向于 0。

在结构鲁棒性方面:按 Z 值和贡献度排序策略,移除前 250 个节点时,最大连通子图相对大小的下降幅度基本保持一致。当移除超过 250 个节点时, Z 值大小排序对网络结构鲁棒性的影响明显比同数量的贡献度大小排序移除策略对网络结构鲁棒性影响大。而 P 值大小排序下在移除 500~750 个节点时造成了最大连通图相对大小的骤降,说明占据网络重要位置的节点可能并没有最大的 P 值。

在功能鲁棒性方面:按不同策略移除节点时,项目贡献度下降速度最快的始终是按贡献度大小排序策略。按 Z 值大

小排序移除节点时,项目贡献度的下降速度先快后慢。而按 P 大小排序移除节点时,项目贡献度的下降速度整体呈现出先快后慢的趋势。

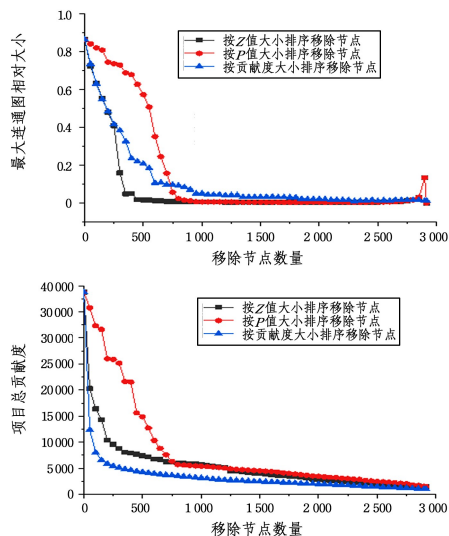


图 7 不同重要性排序下节点移除对网络鲁棒性的影响

Fig. 7 Impact of node removal on robustness of network under different indicators

由此可见,相较于连接多个社团,开发者与其社团内部开发者间的联系更紧密,对网络结构鲁棒性的影响更大,更能维持网络的稳定结构,占据网络的中央位置。而从功能的角度来看,占据网络结构中央位置的开发者并不一定是贡献度最大的,说明开源软件开发者的结构与功能属性具有不对称性,导致协作网络的结构和功能鲁棒性具有不一致性。因此,有必要综合考虑开发者的结构和功能属性进行类型划分。

4.3 不同类型开发者流失的鲁棒性分析

本文基于开发者的结构和功能属性,对开发者类型进行了划分。因此,不同类型开发者的流失对网络鲁棒性的影响和关键开发者类型值得我们进一步探索。本节重复移除各类重要性排序前 5 的开发者节点,模拟不同类型开发者的流失,并将结果与已有研究进行对比。

已有研究通常以节点的度、介数,或将二者结合起来探究网络中节点的重要性。其中节点的度代表了开发者在网络中的影响范围,度越大代表开发者的协作者越多,影响范围越广。节点的介数代表了开发者在网络中人与人之间的“桥梁”作用,介数越大说明开发者占据的“桥梁”位置越重要。

图 8 为不同类型开发者流失对网络结构和功能鲁棒性的影响。由图 8 可见,已有的 3 种指标识别方法下的开发者节点重要性排序对于网络的结构鲁棒性和功能鲁棒性的影响始终不是最大的,说明本文的节点类型划分方法具有一定的意义,能够有效识别出开源软件社区中的关键开发者类型。

4 种类型中,中央核心型开发者不管是在结构上还是功能上对于网络的鲁棒性影响始终是最大的,这类开发者拥有较大的模块内参与度 Z 值、不为 0 的参与系数 P 值和较大的贡献度,说明其在各自社团内部较为活跃,能够成为各自社团的中心,并且与其他社团之间存在着一定联系,拥有相对较多贡献量的开发者往往能处于社区协作网络的中央位置,维系社区内部众多开发者间的协作关系。其因不可替代性往往比网络中的“桥梁”位置的开发者或网络中影响范围广的开发者

更重要,并且这类开发者在贡献度上远远高于度或介数较大的开发者,为社区创造了较大的贡献。因此,可将该类型开发者视为社区中的关键开发者。

中央一般型开发者的流失对网络结构鲁棒性的影响和边缘类开发者相差不大。这类开发者拥有较大的模块内参与度 Z 值和不太大的贡献度,这说明虽然有些开发者在各自社团内部联系较多,能够处于协作网络的中央部分,但由于其较少的贡献度,存在着较大的可替代性。因此,这类开发者的流失并不会对网络的结构和功能鲁棒性造成较大的影响。

边缘核心型开发者的流失对网络功能鲁棒性的影响虽然不是最大的,但与边缘一般类型的开发者相比对网络鲁棒性的影响存在较大差距。这类开发者虽然在各自社团内部不是特别活跃,拥有不太大的模块内参与度 Z 值,但因其较多的贡献度,对网络功能鲁棒性的影响较大,说明社区中存在着一一些不善交际、默默贡献的开发者,这类开发者虽然处在网络的边缘位置,但也为项目贡献了自己大量的智慧。

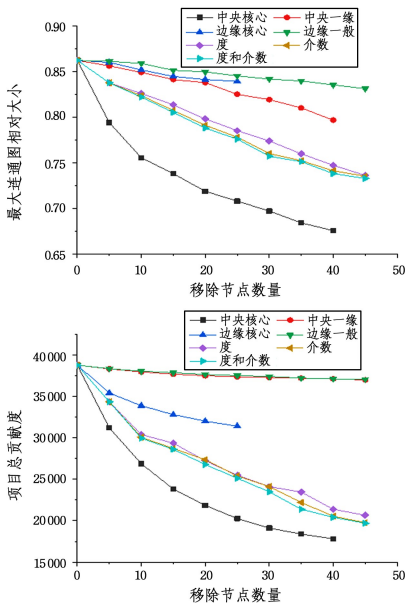


图8 不同类型开发者对网络鲁棒性影响

Fig. 8 Impact of different types of developers on robustness of network

综上所述,并非如以往研究中认为的度越大、介数越大的节点对网络鲁棒性影响越大,在网络中越重要,因此有必要对开发者进行类型划分。不同类型开发者对网络稳定性影响不同,拥有高模块内参与度 Z 值、不为0的参与系数 P 值和较大贡献度的中央核心型开发者是社区中的关键开发者类型,这类开发者的流失对网络结构和功能鲁棒性的影响始终是最大的。社区中还存在着一些处在网络的边缘位置,但默默为项目贡献了大量智慧的边缘核心型开发者。

4.4 鲁棒性提升策略分析

中央核心型开发者作为开源软件社区中的关键类型开发者,对网络鲁棒性影响最大,一旦流失网络将遭受重创。因此,面对这种情况时,网络的鲁棒性提升策略值得进一步探究。

在开源软件社区流动性的特点下,每天都有新开发者的加入。已有研究指出新加入的开发者存在偏好连接行为,大部分新加入的开发者选择与新开发者形成协作关系,但少部分与已有开发者间的协作关系却更频繁^[6]。基于此,本文拟

通过仿真模拟新加入开发者的协作行为,探究网络在面临关键开发者类型流失后鲁棒性的提升策略。

本文拟选用倾向性连接和随机性连接两种策略探究鲁棒性的提升策略,此时网络平均度为3.6,可设新加入节点度 $k=3$,同时设置对照组 $k=6$ 。倾向性连接策略:通过计算网络中节点拓扑参数,选取度较大的节点作为倾向性连接的目标源节点,每个新加入的开发者必须与任意目标源节点形成一条连边,剩余连边则与网络中任意节点相连。随机性连接策略:新加入开发者与网络中的任意节点形成连边。

由于网络功能鲁棒性指标(贡献度)在仿真分析时需要人为进行赋值,因为主观性较大,故本文仅用结构鲁棒性评价指标(最大连通图的相对大小)评价新加入开发者协作行为的网络鲁棒性提升策略。如图9所示,当新加入开发者的节点数量在400人左右时,倾向性连接 $k=6$ 策略下的网络鲁棒性最先恢复到原先水平,而其他策略则需要更多的新开发者加入才能恢复。可见关键开发者类型流失后往往需要项目管理者消耗巨大的成本才能将协作网络修复到原先的鲁棒性。拥有较大初始度的新开发者选择倾向性连接策略能够显著促进网络鲁棒性的恢复,说明新加入开发者越活跃、与网络的中央开发者联系越紧密,往往能更好地提升网络的鲁棒性。与此同时,原先网络中未被移除的中央型开发者也随着新开发者的加入与之形成了广泛的协作关系,进而转变为网络中的新核心。

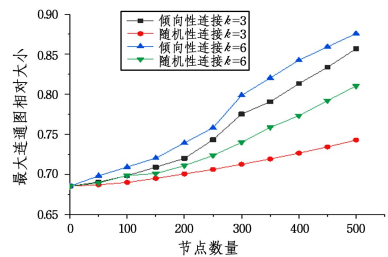


图9 不同策略下加入行为对网络鲁棒性的影响

Fig. 9 Impact of different joining strategies on network robustness

综上,可对项目管理者提出如下建议:首先,必须重点保护网络中的关键类型开发者;其次,可通过相关手段大力吸引新开发者的加入,引导新开发者广泛参与项目的开发进程;最后,在社区建设过程中,可通过相应手段对社区新加入开发者进行引流,促进社区形成多核结构,以此提高网络面对关键类型开发者流失的鲁棒性。

结束语 本文以开源项目 AngularJS 为例提取其在 GitHub 社区的开发者代码提交记录,时间段是从项目初始期至本研究开始时期。通过抽取各版本时期的代码协作关系构建开发者协作网络,据此探究网络的结构和功能。从结构和功能视角对开发者类型进行划分,以网络结构和功能鲁棒性分析方法识别出关键开发者类型。最后通过仿真新开发者的加入行为分析网络鲁棒性提升策略。本文得到以下结论:

(1) 开发者的结构和功能属性具有不对称性,导致开发者协作网络的结构鲁棒性和功能鲁棒性具有不一致性。占据网络结构中心位置、能较好地维持网络中协作关系的开发者并不一定是贡献度最大的开发者。度、介数最大的开发者也并不是网络中最重要开发者。

(2) 开源软件社区开发者类型可根据结构和功能属性分为以下几种类型:在社团内部中心同时也与其他社团保持一定联系,并且贡献度也相对较大的中央核心型;在社团内作为

中心开发者但贡献度较小的中央一般型;贡献度较大但不是社团内部中心的边缘核心型;贡献度较小同时也不是社团内部中心的边缘一般型。

(3)不同类型开发者流失对网络鲁棒性的影响不同,社区中的关键开发者类型是中央核心型,这类开发者既占据网络中的重要位置,又对项目贡献了大量的智慧。与此同时,社区内部也存在着一些虽然处在网络的边缘位置,但仍然默默为项目贡献了大量智慧的边缘核心型开发者。

(4)在社区的发展过程中,项目管理者应综合结构和功能视角关注网络的鲁棒性,重点关注中央核心型开发者;通过宣传手段大力吸引新开发者的加入,鼓励新开发者积极参与项目的开发进程;注重对新加入开发者的引导作用,努力让社区形成多核结构,提高网络面对关键类型开发者流失的鲁棒性。

本文仍然存在一些局限:首先,在开发者类型划分部分,本文虽然已经将开发者的结构属性和功能属性进行综合考虑,并在结构指标上进行创新,将复杂网络领域的社团结构引入研究,但在功能指标上仍然只考虑了提交次数,未来有待进一步完善;其次,本研究目前仍然是单层网络下的研究,未来是否可以建立多层复杂网络进行相关研究也是一大热点,最后,在鲁棒性研究过程中本文虽创新性地加入功能鲁棒性,也通过分析新开发者的加入机制探究鲁棒性提升策略,但未来是否可以从动态视角探究鲁棒性(例如开发者流失后造成的级联失效问题)以及新开发者加入机制的复杂化,均值得进一步深入研究。

参 考 文 献

- [1] O'MAHONY S. Guarding the commons: how community managed software projects protect their work[J]. *Research Policy*, 2003, 32(7): 1179-1198.
- [2] RAYMOND E. The cathedral and the bazaar [J]. *Knowledge, Technology & Policy*, 1999, 12(3): 23-49.
- [3] WU Z F, ZHU T T, XUAN Q, et al. Evaluation of Core Developers in Open Source Software by Contribution Allocation[J]. *Journal of Software*, 2018(8): 2272-2282.
- [4] RAYMOND E. The cathedral and the bazaar[J]. *Knowledge, Technology & Policy*, 1999, 12(3): 23-49.
- [5] ISKOUJINA Z, ROBERTS J. Knowledge sharing in open source software communities: motivations and management[J]. *Journal of Knowledge Management*, 2015, 19(4): 791-813.
- [6] HE P, LI B, YANG X H, et al. Research on Developer Preferential Collaboration in Open-source Software Community [J]. *Computer Science*, 2015, 42(2): 161-166.
- [7] XIA H X, ZHANG X, ZHANG X Z. Study on collaborative network of OpenStack OSS developers[J]. *Systems Engineering-Theory & Practice*, 2017, 37(5): 1373-1382.
- [8] GENCER M, OBA B. Taming of "Openness" in software innovation systems[J]. *International Journal of Innovation in the Digital Economy*, 2017, 8(2): 1-15.
- [9] ALJEMABI M A, WANG Z. Empirical study on the evolution of developer social networks [J]. *IEEE Access*, 2018, 6: 51049-51060.
- [10] LIU X, LI B, HE P. Evolution Analysis of Developer Collaboration Network in Open Source Software Community[J]. *Journal of Chinese Computer Systems*, 2015, 36(9): 1921-1926.
- [11] LIAO Z F, LI S J, HE D Y, et al. Analysis of Key User Behavior in GitHub Open Source Software Development[J]. *Journal of Chinese Computer Systems*, 2019, 40(1): 164-168.
- [12] MOCKUS A. Two Case Studies of Open Source Software Development: Apache and Mozilla[J]. *Acm Transactions on Software Engineering & Methodology*, 2002, 11(3): 309-346.
- [13] NAKAKOJI K, YAMAMOTO Y, NISHINAKA Y, et al. Evolution patterns of open-source software systems and communities [C]// *Proceedings of the International Workshop on Principles of Software Evolution*. ACM, 2002: 76-85.
- [14] WANG W J, LI B, HE P. An Analysis of the Evolution of Developers' Role in Open-Source Software Community[J]. *Complex Systems and Complexity Science*, 2015, 12(1): 1-7.
- [15] ZHOU H L, ZHANG X D. Dynamic robustness of knowledge collaboration network of open source product development community[J]. *Physica A: Statistical Mechanics and its Applications*, 2018, 490: 601-612.
- [16] OSTERLOH M, ROTA S. Open source software development—Just another case of collective invention? [J]. *Research Policy*, 2007, 36(2): 157-171.
- [17] LI C G, ZHANG Y A. Percolation on the Structure Heterogeneity and Robustness of the University-Industry Cooperative Network[J]. *Systems Engineering*, 2018, 36(3): 74-84.
- [18] YU G D, YANG Y, LI F, et al. Analysis and optimization on robustness of customer collaborative product innovation systems [J]. *Computer Integrated Manufa System*, 2014, 20(12): 2926-2934.
- [19] WOOD G. The structure and vulnerability of a drug trafficking collaboration network[J]. *Social Networks*, 2017, 48: 1-9.
- [20] ZHANG X D, ZHOU H L, HU Y. Dynamic robustness of knowledge collaborative network under mass collaboration environment [J]. *Computer Integrated Manufacturing Systems*, 2017, 23(11): 2353-2360.
- [21] BLONDEL V D, GUILLAUME J L, LAMBIOTTE R, et al. Fast unfolding of communities in large networks[J]. *Journal of Statistical Mechanics Theory & Experiment*, 2008(10): P10008.
- [22] BONACCORSIA, ROSSI C. Why Open Source software can succeed[J]. *Research Policy*, 2003, 32(7): 1243-1258.
- [23] GUIMERA R, AMARAL L A N. Cartography of complex networks: modules and universal roles [J]. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(2): P02001.
- [24] HAO Y C, LI C B, WEI L. Cascading failure model of complex networks considering overloaded nodes[J]. *Systems Engineering and Electronics*, 2018, 40(10): 2282-2287.



LU Dong-dong, born in 1995, postgraduate, is a student member of China Computer Federation. His main research interests include complex network and knowledge management.



WU Jie, born in 1968, Ph.D, professor, Ph.D supervisor. Her main research interests include complex network and knowledge management.