

文章编号: 1000-5641(2024)05-0011-09

## OpenRank 贡献度评估方法及其在开源课程中的实证研究

王 婕<sup>1</sup>, 黄温瑞<sup>1</sup>, 赵生字<sup>2</sup>, 夏小雅<sup>1</sup>, 韩凡宇<sup>1</sup>, 王 伟<sup>1</sup>, 张琰彬<sup>1</sup>

(1. 华东师范大学 数据科学与工程学院, 上海 200062; 2. 同济大学 电子与信息工程学院, 上海 201804)

**摘要:** 提出了一种基于 OpenRank 的开源贡献度评价方法, 旨在解决开源项目实践中学生的贡献度量化评估问题. 以“开源软件设计与开发”课程中的软件开发实践教学为背景, 设计了开源实践教学模式中学生的贡献度评价方法, 提出了一种基于开发者协作网络的 OpenRank 算法, 以评估学生在项目讨论、问题解决、代码编写等方面的贡献和价值. 实验结果表明, OpenRank 与传统评分方法在评价学生表现方面具有一致性, 更能全面展示学生在开源项目中各个方面的贡献. 因此, OpenRank 与传统评分方法的结合, 能更加科学、全面地评价学生在开源项目中的贡献度与综合技能.

**关键词:** 开源协作; 开源贡献度; 贡献度评价; 实证研究

**中图分类号:** TP391 **文献标志码:** A **DOI:** 10.3969/j.issn.1000-5641.2024.05.002

### OpenRank contribution evaluation method and empirical study in open-source course

WANG Jie<sup>1</sup>, HUANG Wenrui<sup>1</sup>, ZHAO Shengyu<sup>2</sup>, XIA Xiaoya<sup>1</sup>, HAN Fanyu<sup>1</sup>,  
WANG Wei<sup>1</sup>, ZHANG Yanbin<sup>1</sup>

(1. School of Data Science and Engineering, East China Normal University, Shanghai 200062, China;

2. School of Electronic and Information Engineering, Tongji University, Shanghai 201804, China)

**Abstract:** This study presents an OpenRank-based method for evaluating open-source contributions, designed to address the challenge of quantifying student contributions in open-source projects. Taking the “Open-Source Software Design and Development” course as a case study, we developed a method to assess student contributions in open-source practice. The OpenRank algorithm, which is based on developer collaboration networks, evaluates student contributions in discussions, problem-solving, and coding. Experimental results indicate that OpenRank not only aligns with traditional grading methods but also provides a more comprehensive view of student contributions. Combining OpenRank with traditional grading offers a more scientific and thorough evaluation of student contributions and skills in open-source projects.

**Keywords:** open-source collaboration; open-source contribution; contribution evaluation; empirical research

## 0 引 言

随着信息技术的飞速发展和社会对于软件人才需求的日益增长, 开源软件开发已经成为软件工

收稿日期: 2024-07-08

基金项目: 国家自然科学基金 (62137001, 62277017)

通信作者: 王 伟, 男, 教授, 博士生导师, 研究方向为开源治理. E-mail: wwang@dase.ecnu.edu.cn

程教育的重要组成部分. 开源软件开发不仅鼓励开发者汇集资源和专业知 识, 以创新解决问题, 而且能确保开发人员长期参与项目<sup>[1]</sup>. 尽管开源软件开发为学生提供了实际参与项目开发的机会, 但如何量化评估学生在开源项目中的贡献, 仍然是一个亟待解决的问题.

在过去的 20 年里, 国内的院校在开源教育领域进行了许多实践探索, 开源软件开发已经成为培养软件人才和推动软件产业发展的重要途径<sup>[2-3]</sup>. 然而, 我国在开源人才培养方面存在挑战, 对学生的开源贡献缺乏有效的评价手段<sup>[4]</sup>. 近年来, 国内外学者对学生在开源项目中的贡献度评估进行了诸多探索, 现有的研究主要集中在项目代码工作量的评估上, 缺乏对开源社区中的社会因素的考虑<sup>[5-8]</sup>.

本文主要通过开源教育的实践性教学来评估学生在开源软件开发中的贡献度. 以华东师范大学数据科学与工程学院开设的“开源软件设计与开发”课程为案例, 探讨了实践性教学模式在开源软件教育中的应用, 包括开源项目实践、团队协作、分布式代码提交、问题解决和沟通技巧等方面的训练, 并创新性提出了基于协作网络中心性的 OpenRank 算法, 以评估开发人员在开源仓库中的贡献与价值.

本文的主要贡献包括以下部分.

1) 将软件工程教育与开源平台相结合, 开设了以开放性学习和项目实践为主线的“开源软件设计与开发”课程, 课程教学中的各个环节都在开源平台中展开.

2) 提出了一种基于开发者协作网络的开源贡献度评价模型, 以评估学生在课程仓库中的贡献度, 并通过与课程传统评价方式进行对比, 从而验证了这种方法的有效性.

## 1 相关工作

### 1.1 开源软件与教育

开源软件一直被视为教育的理想选择, 其鼓励开发者协作汇集资源和专业知 识, 以解决问题<sup>[9]</sup>. 而没有经验的学生参与开源软件开发可能会带来一些问题. 有研究总结了开源软件中的常见错误, 发现这些错误经常出现在学生的贡献中, 如不遵循现有设计、提交混乱的拉取请求等<sup>[10]</sup>. 尽管参与开源软件开发项目对于学生来说具有一定的挑战性, 但通过参与这些项目, 学生能够获得实际的软件工程实践经验, 从而提升专业水平<sup>[11]</sup>. 此外, 有研究分析了激励学生参与开源软件开发的因素<sup>[12]</sup>, 研究发现参与开源软件开发有助于学生在职业生涯中取得成功. 维护和开发开源软件可以提高学生的社交能力, 丰富项目经历<sup>[9]</sup>. 谷歌、雅虎和 Facebook 等大型互联网公司也指出, 相对于普通开发者, 参与开源社区的开发者更具有竞争力<sup>[13]</sup>.

### 1.2 开源贡献评估

我国在开源人才培养方面面临巨大挑战, 缺乏有效的抓手. 学生的开源贡献缺乏有效的评价手段, 高校难以在开源工作上制定相关的鼓励政策<sup>[4]</sup>. 因此, 学校除了开展开源课程和实践教育外, 还应探索学生在开源贡献方面的评估手段, 并基于评估结果给出配套政策措施, 推动开源教育教学的发展. 目前, 对开源贡献度的研究主要集中在项目代码工作量的研究上<sup>[5-6]</sup>, 在开源项目领域, Gousios 等<sup>[7]</sup>提出了一个线性模型, 该模型考虑了来自开发和社交活动的多个行动轨迹来评估贡献. Xia 等<sup>[8]</sup>研究了通过挖掘存储库中的活动来识别关键贡献者, 提出线性模型不足以反映开源软件社区的社会因素. 文献 [14] 提供了明确的证据, 表明社会因素在拉取请求审查期间与维护者的贡献评估高度相关. 文献 [15] 和文献 [16] 分别提出了基于网络和基于用户活动的方法研究开发人员的贡献, 这些研究的重点是将开发人员分类, 而非量化他们的贡献. 本研究采用了一种基于协作网络的方法, 评估开发人员的贡献和价值.

PageRank<sup>[17]</sup> 是一种利用搜索引擎链接相关性计算页面重要性的方法, 当前, 不少学者将 PageRank 的衍生算法应用到文献相关度、用户价值计算、电影推荐等不同领域中. Li 等<sup>[18]</sup> 提出一种用于衡量期

刊论文影响力的排名算法 ArticleRank, 该方法考虑了被引用论文的论文类别, 以区分具有相同引用次数的论文的排名结果. Li 等<sup>[19]</sup> 基于复杂网络理论提出了一种电影排名算法 MovieRank, 该算法考虑了电影参与者的不同身份对电影排名产生的影响. MovieRank 虽然使用了异质网络, 但没有将算法扩展到带初值异质网络的收敛性证明. 本文提出的 OpenRank 将 PageRank 算法拓展到了带初值的任意有向高维异质图, 应用到开源项目贡献度量领域中.

## 2 研究内容

### 2.1 开源软件实践课程

本研究以“开源软件设计与开发”课程为研究对象, 探讨基于社交网络的学生贡献度评估方法. 该课程旨在通过开源软件项目的实际参与和贡献, 培养学生在软件工程领域的专业技能, 课程内容涵盖了从拥抱开源、贡献开源到发展开源的三大板块, 强调理论与实践的结合, 课程内容如图 1 所示.

S1 拥抱开源	S2 贡献开源	S3 发展开源
<p>第一讲 开源世界漫游</p> <p>L01.A 开源简史与概览 L01.B 你身边的开源项目 L01.C 探索开源世界 L01.D 开放源代码运动与黑客文化</p> <p>第二讲 多学科视角下探索开源</p> <p>L02.A 多学科视角下的开源 L02.B Wuhan2020 开源项目 L02.C 初识 GitHub L02.D 开源项目的结构</p> <p>第三讲 软件产业与开源软件商业化</p> <p>L03.A 软件、软件学科与软件产业 L03.B 开源软件的商业化 L03.C Hypercrx 探索开源社区 L03.D 自由软件与商业开源</p> <p>第四讲 全球开源软件生态发展</p> <p>L04.A 认识开源软件生态 L04.B 全球开源软件生态面面观 L04.C OpenGalaxy (开源星系) L04.D 开始你的个人开源项目 L04.E 中国参与全球开源生态建设</p> <p>第五讲 软件工程与数据科学视角看开源</p> <p>L05.A 软件工程视角看开源 L05.B 数据科学视角看开源 L05.C OpenDigger 数据挖掘平台 L05.D OpenDigger 应用案例分析</p>	<p>第六讲 软件开发流程与工具</p> <p>L06.A 开源协作与工程全流程 L06.B 个人研发实践工具 L06.C The Missing Lectures L06.D Git 的特点和设计思路</p> <p>第七讲 个人开源软件开发</p> <p>L07.A 贡献开源项目三板斧 L07.B 个人研发与开源实践流程 L07.C 贡献一个实际的开源项目1 L07.D 贡献一个实际的开源项目2</p> <p>第八讲 团队开源协作模式</p> <p>L08.A 管理并参与大型开源项目 L08.B 团队协作模式 L08.C 代码审查 L08.D OpenDigger 贡献指南</p> <p>第九讲 开源、内源与 DevOps</p> <p>L09.A 开源的 DevOps 流程 L09.B 开源与精益研发管理 L09.C 开源社区的高效协作 L09.D 内部开源的艺术 L09.E GitHub Action 实践指南</p> <p>第十讲 深入参与开源社区</p> <p>L10.A 开源社区数据分析 L10.B GitHub 上的开源生态 L10.C 开源社区项目实践案例 L10.D 开源领域知识图谱</p>	<p>第十一讲 企业开源治理</p> <p>L11.A 开源治理基础 L11.B 开源知识产权基础 L11.C 安全可信的开源体系</p> <p>第十二讲 开源社区运营</p> <p>L12.A 开源社区的数字化分析 L12.B 成为优秀的 Committer L12.C Apache 的开源之道</p> <p>第十三讲 开源职业发展</p> <p>L13.A 全职做开源项目的体验 L13.B 工程师如何面对开源 L13.C 从 OSPO 看开源人才</p> <p>第十四讲 开源项目案例分析</p> <p>L14.A 案例分析1 L14.B 案例分析2 L14.C 案例分析3</p> <p>第十五讲 开源生态分析作品</p> <p>L15.A 作品分析1 L15.B 作品分析2 L15.C 作品分析3</p>

图 1 “开源软件设计与开发”课程内容

Fig. 1 Course content for “Open-Source Software Design and Development”

“开源软件设计与开发”课程特别强调开放式学习过程, 采用电子课本和多样化的教学模式, 满足不同学生的个性化学习需求. 课程以 GitHub 仓库 X-lab2017/OSS101 作为课程仓库, 所有学习资料都在课程仓库中, 学生可以自由获取. 此外, 课程的内容安排、任务布置、问题讨论都在课程仓库中进行, 这样可以促进学生之间的合作和交流, 增强学习过程的参与度, 课程设计如图 2 所示.

课程将开源项目实践作为作业, 学生需在课程仓库中完成项目选择、任务分配、问题讨论、代码提交等一系列开源项目实训任务. 教师和课程助教会筛选适合学生水平的开源项目以供学生选择. 在项目实践过程中, 学生可以通过 GitHub 的 Issue 进行任务分配, 讨论技术问题, 提出反馈和建议. 此外, 学生需要在本地环境中根据分配的任务编写代码, 并学习使用 Git 进行代码版本管理, 包括分支创建、代码提交、与他人合作进行代码合并等. 与此同时, 学生还需要完成代码审查、测试以及必要的

文档编写工作, 确保代码质量符合项目标准, 最终将代码提交到 GitHub 仓库并向上游仓库发起拉取请求 (pull request, PR). 学生发起 PR 后, 项目维护者将会对代码进行审查, 如果提出了反馈和改进建议, 学生还需要及时响应这些反馈, 对代码进行相应的修改和优化, 以提高代码质量, 满足功能需求.

这一系列的开源项目实训任务, 在课程仓库中公开透明进行, 旨在使学生获得软件开发经验的同时, 还能理解开源社区的运作机制, 协作文化和项目管理方法, 能够培养学生的团队协作精神、问题解决能力和沟通技巧.

## 2.2 学生贡献评价

在设计学生开源项目实践的评价体系时, 本文借鉴了 Apache Way 的理念, 考虑了以下 3 个原则.



图 2 “开源软件设计与开发”课程设计

Fig. 2 Course design for “Open-Source Software Design and Development”

1) 开放沟通原则: 未在开源社区中记录的通信被视为未发生. 在课程实践中, 这一原则被应用到学生的交流与协作中. 本文要求所有讨论和协作都必须在 GitHub 仓库内进行, 未在平台上公开记录的交流 (如私人聊天) 不计为有效贡献, 以此确保所有的沟通和协作都能透明地被记录和评估.

2) 贡献度量原则: 学生贡献度量不考虑学生的个人背景, 只有通过长期深入开源项目, 积极参与讨论或代码贡献, 学生的贡献度才能提高. 而学生的贡献度越高, 其参与的 Issue、PR 节点的价值就越高. 这反映了影响力需要通过实际贡献来获得, 长期贡献将提高学生的决策影响力.

3) 社区优先原则: 一个有成熟社区并长期活跃的项目, 总是优于一个初始质量良好但缺乏长期维护的项目. 在课程实践中, 本文通过学生之间的社交协作关系来衡量学生贡献度. 对学生贡献度的衡量基于学生之间的协作程度、贡献的数量与质量.

## 2.3 研究过程

研究过程分为以下 4 个关键步骤.

1) 数据收集: 通过 GitHub 的日志数据收集学生在课程项目中的参与和贡献指标数据, 如 PR 打开/关闭、Issue 打开/关闭、PR Review 等.

2) 模型构建: 基于 GitHub 日志数据, 构建网络模型来描述学生在开源项目中, 通过 Issue 和 PR 的协作和贡献情况.

3) 贡献度评估: 基于协作网络, 使用 OpenRank 算法计算学生在课程项目中的 OpenRank 值作为其在课程仓库中的贡献度.

4) 比较: 将所得到的 OpenRank 贡献度与多个角度的教师评分进行对比. 验证利用 OpenRank 算法对学生进行贡献度评价的合理性.

### 3 OpenRank 算法

#### 3.1 OpenRank 算法思想

OpenRank 算法是 PageRank 算法的一种变形与推广, 使 OpenRank 算法可以应用于异质信息网络, 并推广到高维节点中. OpenRank 算法的主张与 PageRank、HITS 等特征向量中心性算法类似, 即一个节点的中心性由指向该节点的其他节点的中心性所决定, 指向该节点的其他节点的中心性越高, 则该节点的中心性也越高. 与 PageRank 不同, OpenRank 算法中每个节点的中心性不仅基于协作网络的结构, 还考虑了节点的内在价值. 推广到开源协作场景中, 开发者节点的中心性由其参与的 Issue/PR 节点的中心性决定, 其参与的 Issue/PR 节点的中心性越高, 开发者节点的中心性就越高.

#### 3.2 网络模型

OpenRank 算法专门为开源项目的协作单元 (如 Issue 和 Pull Request) 定制. 基本的协作网络模型如图 3 所示, 描绘了开发者在 GitHub 上的一个仓库中, 围绕 Issue 和 PR 节点, 通过不同协作行为进行协作的情景. 其中, Issue 节点和 PR 节点的初始值会受到开发者在 Issue 和 PR 上添加的表情的影响.

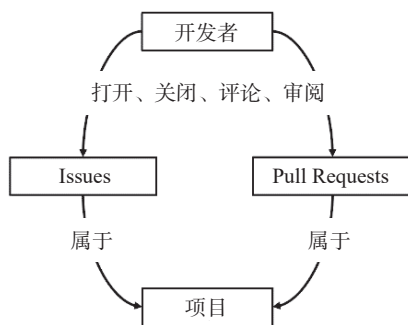


图 3 基本协作网络结构

Fig. 3 Basic collaborative network model

#### 3.3 OpenRank 算法介绍

在 OpenRank 算法中, 每个节点  $v_i$  在每次迭代中的 OpenRank 值的计算公式为

$$v_i = (1 - a_i) \sum_{j=1}^{|V|} \frac{w_{ji}}{d_j^{\text{out}}} v_j + a_i v_0. \quad (1)$$

式 (1) 中:  $v_0$  代表节点的初始值;  $a_i$  表示节点对其初始值的依赖程度;  $d_j^{\text{out}}$  是节点  $j$  的加权出度;  $|V|$  是所有指向  $v_j$  的节点的集合;  $w_{ji}$  是从节点  $j$  到节点  $i$  的边的权重. 如果将归一化权重  $w_{ji}/d_j^{\text{out}}$  组织成矩阵  $S$ , 并将  $a_i$  作为对角值组织成矩阵  $A$ , 根据收敛性证明, 所有节点的 OpenRank 值将收敛到一个向量, 计算公式为

$$\begin{aligned} v &= \lim_{k \rightarrow \infty} v^{(k)} \\ &= \lim_{k \rightarrow \infty} \left[ ASv^{(k-1)} + (I - A)v^{(0)} \right] \\ &= \lim_{k \rightarrow \infty} \left[ (AS)^k v^{(0)} + \sum_{t=0}^{k-1} (AS)^t (I - A)v^{(0)} \right] \\ &= (I - AS)^{-1} (I - A)v^{(0)}. \end{aligned} \quad (2)$$

式(2)中:  $v$  为所有节点的 OpenRank 值组成的向量;  $k$  表示迭代次数;  $v^{(0)}$  表示初始向量, 由所有节点的初值组成;  $v^{(k)}$  表示第  $k$  次迭代后的 OpenRank 向量;  $I$  为单位矩阵.

开发人员节点、Issue 节点和 PR 节点通过包括开放、评论、审查和关闭在内的不同协作行为边连接在一起. 由于这些行为代表了来自开发人员的不同努力, 因此应以不同的权重对其进行加权, 即

$$w_{ij} = \sum_{k=0}^n w_{ijk} c_k. \quad (3)$$

式(3)中:  $n$  代表边的种数;  $w_{ijk}$  是从节点  $i$  到节点  $j$  的第  $k$  类边的权重;  $c_k$  代表第  $k$  类边在所有类型的边中所占的权重比例.

### 3.4 节点初值的选择

对于真实场景中的图网络, 节点通常具有一些先验的信息或特征可以反映该节点的初值. OpenRank 算法在计算中心性时考虑了每种类型节点的初始值. 由于 OpenRank 基于每个月数据进行计算, 如果节点参与了上个月的计算并已经获得了中心性值, 其初始值将从上个月起继承. 如果一个节点没有参与上个月的计算, 其初始值将被设置为 1. 对于 Issue/PR 节点, 初始值会根据开发者点赞数量的增加而增大. 如果 PR 节点在当月被合并, 则其初始值额外增加 50%.

### 3.5 对初始值的依赖程度

在确定网络中不同类型节点的中心性时, 将考虑节点的初始值 (节点自身的特征) 以及它们与其他节点的互动获得的值 (即网络价值转移), 需要确定节点对初始值的依赖程度.

本研究参考了 PageRank 的默认阻尼因子, 对于网络中的所有节点, 其中心性值的 15% 基于初始值, 而中心性值的 85% 则取决于网络中的价值转移.

## 4 实验结果与分析

### 4.1 数据收集

本文的实验数据来源于华东师范大学 2023 年春季学期的“开源软件设计与开发”课程, 对数据科学与工程学院的 85 名研究生在课程学习期间的开源贡献进行了研究. 为获取详细和准确的学生活动数据, 本文使用了 OpenDigger 工具来收集学生从 2023 年 3 月至 2023 年 5 月期间 Issue Comment、Issue Open、Issue Close、PR Review、PR Open、PR Close、PR merge 这 7 种类型的 GitHub 日志数据, 并构建协作网络.

### 4.2 网络构建

课程从 2023 年 3—5 月这 3 个月期间各类型节点数量如表 1 所示.

表 1 网络节点类型统计  
Tab. 1 Network node type

时间	节点数量/个		
	用户	Issue	PR
2023-03	75	34	2
2023-04	94	58	42
2023-05	117	54	124

随着课程逐渐深入, 在课程仓库中活跃的开发者逐渐增加 (仓库开发者数量超过课程学生数量是





从表 3 可知, 在 4 组对比中, Pearson 相关系数从 0.25 到 0.53 不等, Spearman 相关系数则在 0.28 到 0.60 之间波动. 这表明尽管相关性系数存在变化, 但整体上学生 OpenRank 值与教师评分之间存在一定程度的关联. 所有实验的 Pearson 相关性和 Spearman 相关性的  $p$  值都小于 0.01, 表明 OpenRank 与教师评分之间的相关性具有显著性.

OpenRank 评分与教师评分存在一定程度的关联, 尽管 OpenRank 和传统评分方法的考量的角度有所不同, 但它们在评价学生表现方面存在一定的一致性. 值得注意的是, OpenRank 评分与期中作业成绩、期末作业成绩的相关性相对较弱, 与平时分、总成绩的相关性较高. 这反映了传统评分方法主要依赖于特定作业的表现, 而 OpenRank 则更加全面考量学生在开源项目中各个方面的贡献. OpenRank 值与课程总成绩回归图如图 5 所示.

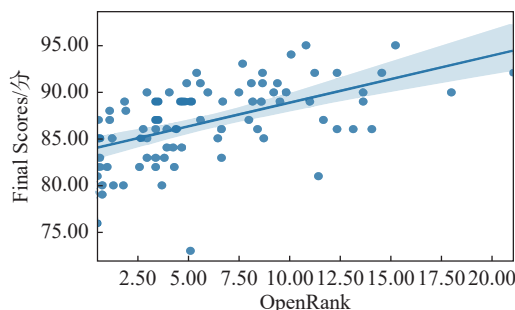


图 5 OpenRank 值与课程总成绩回归图

Fig. 5 Regression plot of OpenRank vs. final scores

为了进一步探索 OpenRank 指标与传统教师评分的差异, 本文对个别学生进行了案例分析. 例如, Surefour 同学在课程中的 OpenRank 得分为 5.12, 处于班级中游水平, 但其课程成绩只有 73.00 分, 为全班最低分. 深入分析其在课程仓库中的表现发现, 这名同学课堂出勤率较低, 期中和期末作业的完成质量也不佳. 但他在课程的 GitHub 仓库上较为活跃, 经常参与课程问题的讨论中, 贡献了大量的代码和解决方案, 因此他的 OpenRank 分数较高. 这表明 Surefour 同学在课程仓库中有较高的活跃度和贡献度, 说明他在课程实践中有一定的成果, 然而仅利用传统评分方式难以衡量这部分贡献.

另一名同学 RTEEnzyme, 他的 OpenRank 得分只有 2.99 分, 但他的总成绩却高达 90.00 分. 查阅他在课程仓库中的协作贡献记录发现, 这位同学期末小组作业的评分较高. 他在课程仓库中的交流与贡献次数较少, 难以展现他在小组作业中发挥的价值, 但传统评分方式对他作出了很高的评价. 这一案例说明, OpenRank 能够洞察学生在合作中的实际贡献.

OpenRank 通过对学生在 GitHub 上的活动进行监测和评价, 能够更准确反映他们在开源项目中的活跃度和贡献度. 这包括代码提交、拉取请求、问题讨论以及文档编写等方面. 与传统评分方法相比, OpenRank 能更全面展示学生的沟通能力、团队合作能力和问题解决能力. 这些是在实际工作环境中非常重要的技能, 都可以通过 OpenRank 被评价和反馈.

将 OpenRank 与传统评分方法相结合, 可以更全面、公正地评价学生的综合能力和实际贡献. 这样的综合评价方法能更好反映学生不同方面的表现, 促进其全面发展. 在教育实践中, 教师可以结合 OpenRank 评价和传统评分方法, 为学生提供更精准的反馈和指导. 这有助于学生更好地了解自己的优势和不足, 并在学习过程中不断提升.

## 5 结 论

本文设计了以开放式学习过程和项目实践为主题的“开源软件设计与开发”课程, 提出了一种基于协作网络中心性的 OpenRank 指标来评价学生在开源项目中的贡献度. 利用学生在课程仓库中的



活跃数据构建协作网络,计算学生的 OpenRank 评分,从而评估学生在开源项目中的贡献度。研究发现,OpenRank 对学生在开源实践的评分与教师评分存在一致性。同时,OpenRank 能够综合反映学生日常的学习行为。相比于教师仅通过作业成果进行评分,OpenRank 能够更加全面展示学生在问题交流、小组协作、任务分工等各个方面的贡献。通过结合 OpenRank 与传统评分方法对学生进行贡献度评价,能够更加全面、公正地评价学生的综合能力和实际贡献,培养学生的团队合作和实际应用能力。

尽管本文验证了 OpenRank 指标在评价学生开源项目贡献度方面的有效性,但仍存在不足之处。首先,研究样本规模相对较小,且学生背景和课程内容较为单一;其次,本研究主要关注课程期间的学生贡献和表现,尚未考察 OpenRank 在长期学习和职业发展中的影响;最后,对学生实际学习过程中的定性分析(如协作细节、激励分析)的研究仍不够深入。在未来的工作中,研究对象将扩展到更多课程和样本以验证其适用性。此外,未来工作将跟踪课程结束后学生的表现,考察 OpenRank 对学生学习和职业发展的长期影响,并结合深入访谈和行为观察,以更全面理解 OpenRank 评分的生成机制和作用。

### [参 考 文 献]

- [1] ELLIS H J C, HISLOP G W, JACKSON S, et al. Team project experiences in humanitarian free and open source software (HFOSS) [J]. *ACM Transactions on Computing Education*, 2015, 15(4): 18.
- [2] 朝乐门. 开源课程及数据科学导论的开源 [J]. *计算机科学*, 2020, 47(12): 114-118.
- [3] 荆琦, 冯惠. 产教融合下的双轨制开源教学模式探索——以北京大学“开源软件开发基础及实践”课程为例 [J]. *高等工程教育研究*, 2023, 41(1): 14-19.
- [4] 黄启春. 基于特色项目制与开源人才培养的软件工程专业学位研究生创新能力培养实践 [J]. *中国信息界*, 2024, 22(2): 25-27.
- [5] KAN S H. *Metrics and Models in Software Quality Engineering* [M]. Boston: Addison-Wesley Professional, 2003.
- [6] WALSTON C E, FELIX C P. A method of programming measurement and estimation [J]. *IBM Systems Journal*, 1977, 16(1): 54-73.
- [7] GOUSIOS G, KALLIAMVAKOU E, SPINELLIS D. Measuring developer contribution from software repository data [C]// *Proceedings of the 2008 International Working Conference on Mining Software Repositories*. 2008: 129-132.
- [8] XIA X, WENG Z, WANG W, et al. Exploring activity and contributors on GitHub: Who, what, when, and where [C]// *Proceedings of the 2022 29th Asia-Pacific Software Engineering Conference*. 2022: 11-20.
- [9] PINTO G, FERREIRA C, SOUZA C, et al. Training software engineers using open-source software: The students' perspective [C]// *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training*. 2019: 147-157.
- [10] SILVA J, WIESE I, GERMAN D M, et al. A theory of the engagement in open source projects via summer of code programs [C]// *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2020: 421-431.
- [11] HU Z, SONG Y, GEHRINGER E F. Open-source software in class: Students' common mistakes [C]// *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*. 2018: 40-48.
- [12] HOLMES R, ALLEN M, CRAIG M. Dimensions of experientialism for software engineering education [C]// *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*. 2018: 31-39.
- [13] DEKOENIGSBERG G. How successful open source projects work, and how and why to introduce students to the open source world [C]// *Proceedings of the 2008 21st Conference on Software Engineering Education and Training*. 2008: 274-276.
- [14] TSAY J, DABBISH L, HERBSLEB J. Influence of social and technical factors for evaluating contribution in GitHub [C]// *Proceedings of the 36th International Conference on Software Engineering*. 2014: 356-366.
- [15] JOBLIN M, APEL S, HUMSEN C, et al. Classifying developers into core and peripheral: An empirical study on count and network metrics [C]// *Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering*. 2017: 164-174.
- [16] CHENG J, GUO J L C. Activity-based analysis of open source software contributors: Roles and dynamics [C]// *Proceedings of the 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering*. 2019: 11-18.
- [17] PAGE L, BRIN S, MOTWANI R, et al. The PageRank citation ranking: Bring order to the web [C]// *Proceedings of the 7th International World Wide Web Conference*. 1998: 1-17.
- [18] LI J, WILLETT P. ArticleRank: A PageRank-based alternative to numbers of citations for analysing citation networks [J]. *Aslib Proceedings*, 2009, 61(6): 605-618.
- [19] LI Y, LI C, CHEN W. Research on influence ranking of Chinese movie heterogeneous network based on PageRank algorithm [C]// *Proceedings of the Web Information Systems and Applications: 15th International Conference*. 2018: 344-356.

(责任编辑: 陈丽贞)