The Diversity-Innovation Paradox in Open-Source Software

Mengchen Sam Yong
School of Computer Science
Carnegie Mellon University
myong@andrew.cmu.edu

Lavínia Paganini
Centro de Informática
Federal University of Pernambuco, Brazil
lfp2@cin.ufpe.br

Huilian Sophie Qiu
School of Computer Science
Carnegie Mellon University
hsqq@cmu.edu

José Bayoán Santiago Calderón
Biocomplexity Institute & Initiative
University of Virginia
jbs3hp@virginia.edu

Abstract—Prior studies have shown that, in open-source software (OSS), diversity is a positive indicator of productivity. Yet, code submissions from underrepresented groups are less successful. This mirrors the diversity-innovation paradox found in science—diverse groups produce more innovations, but historically underrepresented people have less successful careers in these groups. In this preliminary research, we want to investigate whether the effect of the diversity-innovation paradox is present in OSS. We define software innovation as a novel co-usage of two packages in the same project. Using World of Code, we identified JavaScript projects' innovations from late 2008 to early 2014. We intend to calculate diversity measures for the authors who produced the innovations and build models to test the presence of the diversity-innovation paradox in OSS.

Index Terms—open-source, innovation, diversity

I. Introduction

In the era where open-source software (OSS) is ubiquitous, making up the digital infrastructure of our society [1], how to sustain OSS projects is crucial to study. Prior work has shown that diversity is beneficial to OSS projects [2]: both gender and tenure diversities are positive predictors of productivity. On the other hand, prior work has identified a diversity-innovation paradox in science [3]: "diversity breeds innovation, yet underrepresented groups that diversify organizations have less successful careers within them."

We see evidence of a negative correlation between underrepresentation and OSS success. One documented example includes a penalty in the acceptance rates of pull requests authored by female contributors [4]. Since OSS maintenance demands regular participation from contributors [5], [6], it is vital to identify factors that might prevent contributors from sustained participation.

Inspired by the work of Hofstra et al. [3] on the diversity-innovation paradox in science, instead of just considering an individual's exposure to diversity, we want to look at how generating OSS innovations might also affect sustained OSS participation. Given limited time during the hackathon event, we focus on the following research questions (RQs):

RQ1. How do innovations present in OSS?

RQ2. Does an individual's exposure to group diversity lead to them producing more OSS innovations?

II. Related work

Vasilescu et al. [2] shows that tenure and gender diversity in an OSS team is a good predictor of productivity. Qiu et al. [7] use social capital theory to explain how collaboration among people with diverse programming language backgrounds is associated with prolonged OSS engagement.

The "diversity-innovation paradox" is defined by Hofstra et al. [3]—a paradox between scientific innovations produced by demographically diverse groups and career success of the minority researchers—after examining a large dataset of papers from US Ph.D. thesis between 1977 to 2015. The study uses topic modeling and co-occurrence graphs to identify scientific innovations. They found that while groups with higher diversity create more innovations than average, innovations by underrepresented scientists are adopted at lower rates.

OSS innovations are not yet well studied, and there is a gap in our knowledge about the effect of producing innovations on an individual's OSS career. Thus, we consider it important to study whether the paradox exists in OSS and how innovations affect sustained OSS participation.

III. Data and methods

We attempt to answer our research questions by computing measures of diversity and innovation using data from World of Code (WoC) [8] and GHTorrent [9]. For each measure, we divide a year into four three-month windows (e.g., January to March, April to June).

A. Diversity

Following Vasilescu et al. [2], we computed projects' gender and tenure diversity. We collected and calculated two levels of diversity measures—on a project level and an individual level. We retrieved contributors' names and the dates of their first commits from GHTorrent.

On the project level, we used Blau's index [10] to capture gender diversity. Blau's index is defined as $1 - \Sigma_{i \in m,f} p_i^2$, where p_i are the fraction of male and female team members for each project. We used the tool Namsor [11] to infer contributors' gender. For tenure diversity, following the practice by Vasilescu et al. [2], we first computed each contributor's commit tenure, which is the number of three-month windows since the contributor's first commit. A project's tenure diversity is then computed using the coefficient of variation, defined as the ratio of the standard deviation and the mean.

The individual-level diversity is the level of diversity that a contributor is exposed to. For each contributor in the project, we computed the average of diversity measures of all the projects the contributor participated in during that three-month window, for gender and tenure diversity respectively, as the exposure to diversity.

B. Innovation

To measure innovation, we adapted the concept from [3], where scientific innovation is defined as a novel co-occurrence of two scientific concepts. We defined software innovation in the open-source JavaScript community as a novel co-usage of two packages in the same project. The novel co-usage establishes a link between the packages, that the two can be useful together in a single project. The first project to establish such a link could be innovative in being among the first software in an area where the packages are applicable or using one of the packages in a new way to complement the other.

We used WoC data maps of JavaScript projects "JSthruMaps/tPaPkgRJS.s" from late 2008 to Mid-March of 2014.[1] The data maps contain entries of when and by whom new packages are introduced into a project. We scanned all entries within the time frame in chronological order to identify the first co-occurrence of any pair of two packages in the entire database. We recorded when, in which project, and by whom a pair of packages are first co-used.

Specifically, first co-occurrences were found with the following process: (1) For each project, suppose up to time T packages P, Q already exist in the project, then when author A introduces a new package R into the project, we consider the co-usage (P,R) and (Q,R). (2) The innovations were tracked across all projects up to time T, and if (P,R) has been an innovation at some previous time, then (P,R) is not considered an innovation by author A; if (P,R) has never been an innovation in any project at any previous time, then (P,R) is now recorded as an innovation. (3) In addition, for every innovation, we track how many times the pair of packages co-occur in any projects at a later time. This is defined as the impact of this innovation, (i.e., number of times the same occurrence is observed).

[1]The code used in this project is available at https://github.com/woc-hack/diversity-innovation

We pruned those innovations with impact only one so that the remaining innovations (with an impact of at least 2) each have some "impact" observed in other projects at a later time. We then divided the time frame from 2008-04-01T00:00:00 UTC to 2013-01-01T00:00:00 UTC into 20 three-month windows. We aggregated, for each author in each window, how many innovations the author produced during that window.

IV. Preliminary results

A. Innovation

In the period from late 2008 to mid-March of 2014, where we measured innovations and impact, we found 6,020 different authors who introduced new co-usage of packages in 11,971 repositories.

The average number of windows where an author introduced innovations was 1.445 ($s = 0.965$), suggesting that authors introduce innovations in only 1 or 2 windows on average. It is unclear whether they dropped the open-source community or became less innovative afterward. The result may also be attributed to the bias of collected data limited to no later than 2014, while these authors could be contributing more innovations later on.

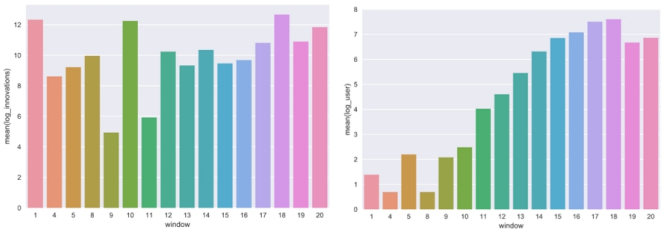

Fig. 1: Natural log of number of innovations (left), natural log of number of innovating authors (right) per window.

No innovations were found in windows 2, 3, 6, and 7, so they are omitted in Figure 1. Preliminary analysis shows that, while the total number of innovations does not change drastically over the windows (left), the number of open-source contributors who are making innovations has dramatically increased over time (right).

B. Diversity

Due to limited time during the hackathon and various challenges we will discuss next, we did not finish measuring diversity exposure for the authors we found.

V. Discussion

A. Using World of Code

We used WoC data maps to scan for innovations. Entries were initially not sorted by time across all projects. Thus, simply scanning through the data maps could not produce the correct innovations, as package co-usage at an earlier time could appear "later" in the data maps. WoC maintainer Professor Audris Mockus then created a new

map with all entries sorted by time so that our code would work correctly.

During the hackathon, we recorded innovations as a dictionary in memory that was transferred into a text file when the script finishes. This approach was sub-optimal, as the script increasingly required a more significant amount of memory. Furthermore, adding new entries to the innovations dictionary required reading in the entire current record and updating in memory. After the hackathon, we migrated it into a relational database to speed up future data collections.

One issue we encountered when aggregating innovations by author and when calculating author diversity is that, multiple versions of the author field may refer to the same person, as sometimes the same developer uses different emails or displayed names. While there are algorithms and maps in this research area to alias these authors, we suggest WoC include such a map and provide a lookup function to find all used aliases of a given author.

B. Future work

There is little prior work on what constitutes reliable OSS innovations. As a next step to identifying the paradox's potential presence, we must validate that our construct for "innovations" is indeed some form of innovation. One way is to survey contributors and users of the projects where innovations are found, on how innovative they perceive the projects to be. Another approach to validate the innovations is to identify them by an orthogonal definition, for example, based on the forking network of JavaScript projects.

We intend to continue our data collection for diversity and innovation measures to include more recent data. Additionally, we would like to consider other measures for the impact of an innovation, such as the number of stars of or forks from the initial repository. Lastly, we plan to build models incorporating author diversity and innovation measures to test the diversity-innovation paradox's presence.

References

[1] N. Eghbal, "Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure," Ford Foundation, Research Reports, 2016. [Online]. Available: https://fordfoundation.org/work/learning/research-reports/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure

[2] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, and V. Filkov, "Gender and Tenure Diversity in GitHub Teams," in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, ser. CHI '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 3789–3798.

[3] B. Hofstra, V. V. Kulkarni, S. Munoz-Najar Galvez, B. He, D. Jurafsky, and D. A. McFarland, "The Diversity–Innovation Paradox in Science," PNAS, vol. 117, no. 17, pp. 9284–9291, 2020.

[4] J. Terrell, A. Kofink, J. Middleton, C. Rainear, E. Murphy-Hill, C. Parnin, and J. Stallings, "Gender differences and bias in open source: pull request acceptance of women versus men," PeerJ Computer Science, vol. 3, 2017.

[5] Y. Fang and D. Neufeld, "Understanding sustained participation in open source software projects," J Manag Inf Syst, vol. 25, no. 4, p. 9–50, Apr. 2009.

[6] J. Coelho and M. T. Valente, "Why Modern Open Source Projects Fail," in Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, ser. ESEC/FSE 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 186–196.

[7] H. S. Qiu, A. Nolte, A. Brown, A. Serebrenik, and B. Vasilescu, "Going farther together: The impact of social capital on sustained participation in open source," in 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), May 2019, pp. 688–699.

[8] Y. Ma, C. Bogart, S. Amreen, R. Zaretzki, and A. Mockus, "World of Code: An Infrastructure for Mining the Universe of Open Source VCS Data," in Proceedings of the 16th International Conference on Mining Software Repositories, ser. MSR '19. IEEE Press, 2019, p. 143–154.

[9] G. Gousios and D. Spinellis, "GHTorrent: Github's data from a firehose," in 2012 9th IEEE Working Conference on Mining Software Repositories (MSR). IEEE, 2012, pp. 12–21.

[10] P. M. Blau, Inequality and Heterogeneity: A Primitive Theory of Social Structure. Free Press, 1977.

[11] NamSor, "Namsor api v2 : classify personal names accurately by gender, country of origin, or ethnicity." [Online]. Available: https://namsor.com