Geography of Open Source Software Production[*]

Gábor Békés[†], Julian Hinz[‡], Miklós Koren[§], and Aaron Lohmann[¶]

June 2023

Preliminary and incomplete — please do not cite or circulate

Abstract

This paper analyses the production function of open source software and highlights international aspects. The modern digital infrastructure is relying on open source software. This is often produced in geographically scattered teams which coordinate on software projects. Those software projects themselves have connection among each other. We study the geographical features of connections between and within software projects. We find significant effects on collabortio

Keywords: Software industry, open source, gravity, global production networks
JEL Classification: F10, F14, L86, F23

[*] Funded by the European Union. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

[†] Central European University, KRTK and CEPR.

[‡] Bielefeld University and Kiel Institute for the World Economy.

[§] Central European University, KRTK, CEPR and Cesifo.

[¶] University Bielefeld and Kiel Institute for the World Economy.

1 Introduction

The modern digital infrastructure relies on open source software (OSS). Maching Learning in Python is often performed using `TensorFlow`; Developing web applications in Python is facilitated by `Flask` and `React` helps develop User Interfaces (UI) in `JavaScript`. Though, OSS is often an integral part of software supply chains, the underlying production networks are not well studied. We posit that the production of OSS can be understood as the outcome of team production functions similiar to the approach in Hsieh et al. (2018). This let´s us model the production of software as bipartite network where one set represents the different software projects and the other disjoint set is the set of developers. Further, we allow the software projects to depend.

2 Background

In this section, we discuss more background information on the development of open source software.

2.1 Open source software in the mordern infrastrucutre

2.2 Github - social media for collaborative efforts

JavaScript and NPM `JavaScript` (`JS`) is one of the most popular programming languages commonly used in the development of web applications and mobile apps. x percent of webpages employ `JS` in one form the other; x percent of mobile apps rely to some extent on it. Part of the `JS` infrastructure is `NPM` which is its dominant package manager. A package maanger in the realm of software development can be thought of as a library. It organises, categorizes and provides access to different software projects. For our purpose, `NPM` lends itself nicely to identify a set of relevant software projects.

A team production function The production of software can be thought of as the composition of two related networks. Throughout this paper, we will call the *within*-network the bi-partite network formed by collaborators on different projects. The *between*-network refers to connections of software projects irrespective of their contributors. Software projects and their collaborators can be thought of as a bipartite network. This approach follows Hsieh et al. (2018) who study the production of knowledge goods, more specifically academic research.

Let $\mathcal{G}$ denote the bipartite network which considers collaborators $i = 1, \dots I$ and software projects P_j for $j = 1, \dots J$. Note, that $\mathcal{G}$ can be represented by a $J \times I$ matrix such that projects are in the rows and collaborators are in the columns. Let a single element of $\mathcal{G}$ be denoted by $g_{i,j} \in [0, 1]$ and refers to whether contribtutor i works jointly on project j.

Writing this into a simple production function:

$$P_j = \sum_{i=1}^{I} g_{i,j} \tag{1}$$

, where now a software project would simply be determined by the sum of contributors participating in it.

Dependencies - intermediary inputs We extend the previously proposed team production function by intermediary inputs. This is captured by another network that can in turn again be represetned as a matrix. Let $\mathcal{D}$ be the *between*-network of software projects. This matrix has the dimensions of available software projects $J \times J$. Note, that this network is in fact a directed network since one software project may import another while the other typically will not import this. Curiously, this is not excluded. Let $d_{m,k}$ be the single elements of $\mathcal{D}$, where $m = k = 1, \dots J$. Further, $d_{mk} = 1$ then signals that P_m declares a dependency on P_k.

Augmenting the production function further allows us to write:

$$P_j = \sum_{i=1}^{I} g_{ij} + \sum_{k \backslash j}^{J} d_{jk} \tag{2}$$

Utility of developers The collaboration network is formed by developers. Each of the developers maximises their individual utility which is symmetric across I. Let the utility of a single developer be written as:

$$U_i = \sum_{j}^{J} g_{ij} P_j - \sum_{j}^{J} g_{ij} \tau_{ij} \tag{3}$$

where $\tau_{i,j}$ captures how inclined developer i is to work on Project P_j.

3 Data

We combine a number of different data sources to construct a sample which allows us to estimate the determinants of network formation as discussed in the previous section. We use `Libraries.io`, and the API endpoints of `Github`, `NPM` and `Nominatim`.

Libraries.io

Nominatim

GIthub

NPM

4 Descriptive statistics

We first use the data described in section 3 to plot some simple descriptive statistics about the composition of collaboration and dependencies. Later in this section, we use the collected data to build synthetic locations for software projects to put them on the geographical landscape.

5 Empirical approach

To assess the geographical dimension of both aspects of the production function, we run two different gravity equation. One relates to the *within* project perspective and the other relates to the *between* project perspective.

Gravity equation

Between gravity

Within gravity

6 Results

7 Robustness

8 Conclusion

To be added.

References

Hsieh, Chih-Sheng, Michael D Konig, Xiaodong Liu, and Christian Zimmermann, "Superstar Economists: Coauthorship networks and research output," 2018.

A Appendix

B Data processing