Analyzing the Implicit Social Network from GitHub Activities

Progress Report
Team 081

Ran Tavory, Ruzvidzo Ngulube, Jonathan del Campo

Abstract

We develop a method of mining GitHub event activity to construct an implicit social network between GitHub users encompassing their GitHub repositories and the connections between them. We then utilize this network for interesting visualizations, including shortest path between users and finding the most influential users.

Introduction

GitHub [1] (GH) is a popular platform for hosting open source software projects and many developers around the world are familiar with it, many of them use it for their day to day work. While GH provides excellent tools for software development, it also supports a small set of social interactions such as following other users, discussions and a way to collaborate. What is missing in our opinion is a way to analyze the social interactions between users on the platform. In this report we present a method of mining GH event activity to construct an implicit social graph between GH users. By implicit we mean that one user may not necessarily explicitly follow another user, rather they may have interacted with each other working on the same repository or by commenting on an issue or a Pull Request. We consider such interactions as co-contribution and use them to construct a social network between users.

After having built this network we apply interesting visualizations to it such as finding the shortest path between users or finding the most influential users.

A value we see in such tool is both for businesses and individuals. Businesses may care about community discovery for marketing purposes. Individuals may care about personal branding and achievements.

The project, if successful, would become a popular tool within developers and businesses looking to study the social graph implied by GitHub, in particular finding connections to other thought leader users and identifying communities.

To add to that, it has been claimed by Casalnuovo et al. [5] that there is evidence for socialization as a precursor to joining a project and so we believe that presenting the social graph would open doors to more cooperation between users.

[1] https://github.com/

Problem Definition

GH users as well as business entities can benefit from analyzing the social interactions between users. Several use cases come to mind:

- Finding the most influential users in specific field in order to reach out to them for advice.
- Finding the shortest path between users in order to find the best way to reach out to them.
- Discovering communities focused around specific topics or technology. (not in our scope)
- Finding the most active organizations. (not in our scope)

GH allows social interactions such as co-contributions but does not explicitly provide them in its API[13] or UI. We therefore want to develop a tool that would allow businesses and individuals to discover these implicit social interactions.

We will create an interactive UI with a graph that allows querying the activity of GitHub users and relations between them, in particular, find the shortest path between two users (where edges are defined by co-activity on the same project), Similar to the Erdős number in Mathematics [23], rate users in terms of their distance from some of GitHub "celebrity" user, for example distance from Linus Torvalds and find the most influential users for a certain technology ecosystem , e.g. Mike Bostock for D3.

We have not seen such interactive interface for GH as of today. There are databases with raw GitHub data ([8], [9] and [7]) and there is an API for GitHub's data[13] but again, a tool that structures this data in a social network structure to allow interactive discovery we did not find.

A more formal definition of the problem is stated in the following: Given a set of documented interactions between GH users, such as commits to a project, opening a pull request, following a user or starring a project, construct a social network between users where the

edges are defined by co-activity on the same project or follow/star activity; Nodes are defined as two types, a user node and a repository node. Users may directly connect to each other by following each other or indirectly by co-activity on the same project. Implement two graph algorithms on this graph, namely finding the shortest path between two users and finding the most influential users (by means of PageRank or other). Lastly, display the result in an interactive UI allowing the user to query the graph in order to view only the relevant sub-graph.

Survey

Extracting implicit social graph structure from GH has been suggested before, for example Lima et al. [16] created a followers and contributors graphs to calculate statistical measures such as the rich club coefficient [21]. Thung et al. [20] took a similar approach of mining GitHub's event history to build a graph and compute statistics for this graph, such as graph's connectivity, average shortest path and PageRank[19]

Identifying influential users or opinion leaders has its roots in sociology and in recent years has been implemented in different internet systems. One interesting comparison between some of the methods is the focus of Liu et al. in [17] where China's social network "Sina" and a local cluster of 4k students in Shanghai University are used to compare the accuracy of three different opinion leaders discovery methods, PageRank, HITS[15] and Synthesized Centrality (invented by the authors). The authors find that SC results in similar recall to PR and HITS but has higher precision. This is interesting to our work although the computational costs of SC is higher from other methods and we suspect that at our scale this may present challenges.

Hu et al. [12] through the usage of HITS analyzed graphs that show the influence and relationship between GitHub repositories and users. They studied how specific repositories influence the development of the code of other repositories, and study how repositories rank over time.

From a slightly different angle Batista et al. [4] studied the correlation between the properties that measure the strength of software social coding collaboration on GitHub; Several ways to measure collaboration were presented, for example, the Preferential Attachment (PA) which assumes that the more edges a node has (a user or a project), the more likely it will get more edges.

Hu et al. [11] focused on a very specific workflow of GitHub, namely the Follow-Star-Fork workflow and constructed a graph based on these activities. They implemented several quantitative measures, specifically UserRank (like PageRank), HITS, H-index[24], Betweenness centrality[6], Spearman rank correlation[25], and Borda Count voting[22] to measure the influence of a user.

Badashian and Stroulia [3] measured the influence of GitHub users and specifically defined and quantified what does influence mean in GitHub and whether the measured influence remains siloed to specific domain of content (e.g. technology or programming language).

Badashian et al. [2] combined two software development focused websites, GitHub and StackOverflow, to study the influence and contribution across these two networks. After studying the characteristics of each separate network a combination of the networks is studied as the correlation between the same user's activity in one network to the other.

We have not seen such interactive interface for GH as we are planning to implement as of today. What we have seen is statistical analysis of GitHub's graph data ([16], [12] and [4]) but none of the resources we have surveyed allows for interactive discovery.

There are databases with raw GitHub data ([8], [9] and [7]) and there is an API for GitHub's data[13] but again, a tool that structures this data in a social graph structure to allow interactive discovery we did not find.

Interactive interfaces such as the one we plan to implement have been studies, for example Hansen et al. [10] where a user usability study of the network analysis tool NodeXL is conducted and a set of guidelines of do's an don'ts is presented. We do not intend to use NodeXL specifically, yet user study conclusions are useful.

There are several notable risks, in particular - data may be difficult to obtain at scale. Although there are multiple sources, from our research we know that no single source encompasses all the aspects we require so we will have to merge multiple sources. Merging successfully is risky as well as obtaining a complete view of the data at scale. We've seen reports of missing data, incorrect data, inconsistent data and obfuscated data (for privacy reasons)[14].

Proposed Method

There are three main phased to this project, data collection, data processing and data visualization.

We have identified multiple sources of data from which we may collect our data and and we are currently in the process of collecting data and evaluating them.

The data sources we are considering are:

- GitHub's API [13] provides per user information (such as number of followers, number of repositories, avatar)
- GitHub's Archive [9] provides a drop of the data easily processed (however missing some required details).
- Google BigQuery [7] provides a high level query capability atop that data.
- GHTorrent [8] another aggregation of GitHub's data, perhaps in a more complete form.
- ClickHouse query interface for GH data [1] another aggregation of GH data with fast response time in a web UI.

After collecting the data there would be some preprocessing to be done, for example, we would like to go over all user activity and extract the interaction details to progressively build the social graph. For each activity we note the user that performed the activity, the repository performed on and other users if they were involved. For example opening a pull request activity would note the opener user, the name the the repository and the users assigned to review if there are any.

Data is large so there are scalability concerns including collection, preprocessing and serving. How to properly handle such large data and provide fast enough access for an interactive user experience may be challenging.

After this initial data collection and aggregation we would insert the data into a graph database such as Neo4j[18] in order to be able to query for shortest path. We might also pre-compute the PageRank[19] or HITS[15] for each user.

Lastly we create an interactive UI that allows querying the stored data and presenting a visual representation of the implicit network (sub-graphs of it per the query), for example displaying the shortest path between two users and perhaps coloring them by their PageRank or HITS score.

Our list of innovations is as follows:

- We are the first, to the best of our knowledge, to collect and build GH social graph data and present it in an interactive UI.
- We are the first, to the best of our knowledge, to present a shortest path between two users in a GH's implicit social graph.
- We are the first, to the best of our knowledge, to calculate and present central users of GH encompassing multiple social activity types (commits and pull requests), not just the Follow-Star-Fork flow.

TODO: Detailed description of the design of upcoming experiments / evaluation

Plan of Activities

Table 1 summarizes the plan of work

Our current progress is as follows:

- Data collection: The analysis that has been done up to now is includes feasibility analysis of different solutions to extract data from Github. The one that has been selected to used to extract data is BigQuery.
- Data processing: We have identified the data that it is required for the analysis through Graphs and the content of the data.
- Presentation: An initial web UI is implemented using modern web technologies such as React[2], D3[3], Vite[4] and TypeScript[5].

Experiments/ Evaluation

The following steps that are going to be performed are:

- Code to make the data extraction of commits.
- Code to make the data extraction of contents.
- Code to make the data extraction of files.
- Code to make the data extraction of languages.
- Code to make the data extraction of repositories.

The evaluation of the steps that are performed during the data collection will consist in data integrity checks between the different datasets, data comparison with other methodologies in smaller datasets as tests, and data quality check of the final solution through the UI

[2]https://reactjs.org/
[3]https://d3js.org/
[4]https://vitejs.dev/
[5]https://www.typescriptlang.org/

Table 1: Plan of Work in high level

Work item	Who	Start	Duration
Proposal document	Ran	Oct 8	3 days
Proposal presentation	Jonathan	Oct 10	2 days
Proposal video	Ruzvidzo	Oct 10	2 days
Data Collection	Jonathan and All	Nov 1	4 weeks
Data Augmentation	Ruzvidzo	Nov 7	3 weeks
Web and UI	Ran	Nov 1	4 weeks
Progress report	All	Oct 30	1 week
Final report	All	Nov 20	2 weeks
* All team members contribute a similar amount of effort			

making comparable results with other analysis done in other researches.

References

[1] Milovidov A. 2020. *Everything You Always Wanted To Know About GitHub (But Were Afraid To Ask).* https://ghe.clickhouse.tech/

[2] Ali Sajedi Badashian, Afsaneh Esteki, Ameneh Gholipour, Abram Hindle, and Eleni Stroulia. 2014. Involvement, Contribution and Influence in GitHub and Stack Overflow. In *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering* (Markham, Ontario, Canada) *(CASCON '14)*. IBM Corp., USA, 19–33.

[3] Ali Sajedi Badashian and Eleni Stroulia. 2016. Measuring User Influence in GitHub: The Million Follower Fallacy. In *Proceedings of the 3rd International Workshop on CrowdSourcing in Software Engineering* (Austin, Texas) *(CSI-SE '16)*. Association for Computing Machinery, New York, NY, USA, 15–21. https://doi.org/10.1145/2897659.2897663

[4] Natércia A. Batista, Michele A. Brandão, Gabriela B. Alves, Ana Paula Couto da Silva, and Mirella M. Moro. 2017. Collaboration Strength Metrics and Analyses on GitHub. In *Proceedings of the International Conference on Web Intelligence* (Leipzig, Germany) *(WI '17)*. Association for Computing Machinery, New York, NY, USA, 170–178. https://doi.org/10.1145/3106426.3106480

[5] Casey Casalnuovo, Bogdan Vasilescu, Premkumar Devanbu, and Vladimir Filkov. 2015. Developer Onboarding in GitHub: The Role of Prior Social Links and Language Experience. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (Bergamo, Italy) *(ESEC/FSE 2015)*. Association for Computing Machinery, New York, NY, USA, 817–828. https://doi.org/10.1145/2786805.2786854

[6] Linton C. Freeman. 1977. A Set of Measures of Centrality Based on Betweenness. *Sociometry* 40, 1 (1977), 35–41. http://www.jstor.org/stable/3033543

[7] Google. 2015. *Google Cloud Console BigQuery: GitHub Archive.* https://console.cloud.google.com/bigquery?project=githubarchive&page=project

[8] Georgios Gousios. 2013. The GHTorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories* (San Francisco, CA, USA) *(MSR '13)*.

IEEE Press, Piscataway, NJ, USA, 233–236. http://dl.acm.org/citation.cfm?id=2487085.2487132

[9] Ilya Grigorik. 2015. *GH Archive*. https://www.gharchive.org/

[10] Derek L. Hansen, Dana Rotman, Elizabeth Bonsignore, Nataa Milic-Frayling, Eduarda Mendes Rodrigues, Marc Smith, and Ben Shneiderman. 2012. Do You Know the Way to SNA?: A Process Model for Analyzing and Visualizing Social Media Network Data. In *2012 International Conference on Social Informatics*. 304–313. https://doi.org/10.1109/SocialInformatics.2012.26

[11] Yan Hu, Shanshan Wang, Yizhi Ren, and Kim-Kwang Raymond Choo. 2018. User influence analysis for Github developer social networks. *Expert Systems with Applications* 108 (2018), 108–118. https://doi.org/10.1016/j.eswa.2018.05.002

[12] Yan Hu, Jun Zhang, Xiaomei Bai, Shuo Yu, and Zhuo Yang. 2016. Influence analysis of Github repositories. https://doi.org/10.1186/s40064-016-2897-7

[13] GitHub Inc. 2022. *GitHub REST API - GitHub Docs*. https://docs.github.com/en/rest

[14] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. German, and Daniela Damian. 2014. The Promises and Perils of Mining GitHub. In *Proceedings of the 11th Working Conference on Mining Software Repositories* (Hyderabad, India) *(MSR 2014)*. Association for Computing Machinery, New York, NY, USA, 92–101. https://doi.org/10.1145/2597073.2597074

[15] Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *J. ACM* 46, 5 (sep 1999), 604–632. https://doi.org/10.1145/324133.324140

[16] Antonio Lima, Luca Rossi, and Mirco Musolesi. 2014. Coding Together at Scale: GitHub as a Collaborative Social Network. https://doi.org/10.48550/ARXIV.1407.2535

[17] Huanhuan Liu, Xiaoqing Yu, and Jing Lu. 2013. Identifying TOP-N opinion leaders on local social network. In *IET International Conference on Smart and Sustainable City 2013 (ICSSC 2013)*. 325–328. https://doi.org/10.1049/cp.2013.1970

[18] Neo4j. 2022. *Neo4j Graph Data Platform | Graph Database Management System*. https://neo4j.com/

[19] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66. Stanford InfoLab. http://ilpubs.stanford.edu:8090/422/ Previous number = SIDL-WP-1999-0120.

[20] Ferdian Thung, Tegawendé F. Bissyandé, David Lo, and Lingxiao Jiang. 2013. Network Structure of Social Coding in GitHub. In *2013 17th European Conference on Software Maintenance and Reengineering*. 323–326. https://doi.org/10.1109/CSMR.2013.41

[21] Wikipedia contributors. 2020. Rich-club coefficient — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Rich-club_coefficient&oldid=983063532 [Online; accessed 10-October-2022].

[22] Wikipedia contributors. 2022. Borda count — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Borda_count&oldid=1113980432 [Online; accessed 10-October-2022].

[23] Wikipedia contributors. 2022. Erdős number — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Erd%C5%91s_number&oldid=1107735023 [Online; accessed 10-October-2022].

[24] Wikipedia contributors. 2022. H-index — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=H-index&oldid=1111311697 [Online; accessed 10-October-2022].

[25] Wikipedia contributors. 2022. Spearman's rank correlation coefficient — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Spearman%27s_rank_correlation_coefficient&oldid=1112073400 [Online; accessed 10-October-2022].