Analyzing Collaboration Patterns in GitHub using Co-Authorship Metrics
Literature Review

Eldan Cohen

I. Introduction

The purpose of my research is to investigate the patterns of collaboration in GitHub network using co-authorship metrics. Co-authorship metrics have been developed and applied to investigate collaboration patterns in Digital Libraries such as academic publications databases and online libraries like Wikipedia. Collaborations patterns in GitHub were the subject of many researches in the recent years, using different techniques. However, Using co-authorship metrics to measure collaboration, developed originaly for a different purpose, might yield interesting results. These results can then be used to compare collaboration patterns with other DL databases. This literature review covers papers on co-authorsip, general work on mining GitHub, and other attempts to analyze collaboration patterns in GitHub or other developers networks.

II. Papers Search Methodology

The papers were found using *Google Scholar* engine. First, I selected a set of relevant topics for the research (as explained in the following section). Then, three search methods were used:

1) Searching terms and ideas according to the selected topics
2) For the relevant papers that I found, I searched for more related papers (using ”Cited by” and ”Related articles” features in *Google Scholar*)
3) Finally, I looked for relevant conferences and journals (according to the papers already found) and checked their proceedings/publications.

III. Relationship to Project

The papers are divided to several topics I selected. Following is a short explanation of the relevance of each topic to the project.

Analyzing collaboration in digital libraries - This is the most important topic. My work will be to apply the methods used in digital libraries research to measure collaboration on GitHub. This methods are all general and are all based on a graph representation of the collaboration in the network.

Mining Github - Since our work is based on mining relevant information from GitHub, it is important to understand the process of mining GitHub, the relevant information we can get, and the conclusions from previous attempts to mine GitHub.

Analyzing collaboration patterns in large developers networks - Measuring collaboration in developers networks was the subject of several research works. Although we approach this from a different angle, we can learn from previous attempts, and also compare our results with them. Also, I wanted to make sure that I am not going to repeat an already existing analysis.

Graph Processing - As all the co-authorship metrics are based on a graph representation of the collaboration, it was important to read about existing libraries for graph processing, and specifically scalable distributed graph processing.

IV. Bibliography

A. Analyzing collaboration in digital libraries (co-authorship)

Bird et al. [1] investigated the structure and dynamics of collaboration in computer science literature. They stated by mining DBLP database and built a collaboration graph based on the extracted data. After classifying the papers to different areas in CS, they use different metrics to examine inter-area(i.e., within-area) collaboration patterns, and network-wide patterns (i.e., for CS literature as a whole). For intra-area analysis they use the following metrics: Degree Distribution, Assorativity, Betweenness Centralization, Community Structure. For inter-area analysis they use Area Overlap, Migration, Interdisciplinariness.

Elmacioglu and Lee [2] also investigates collaboration pattern by mining DBLP database and modeling it as a collaboration graph. They begin with a general statistical analysis of the authors and the papers on DBLP. They continue by measuring the giant component of a graphm I.e., the largest subset of interconnected nodes, and calculates the clustering coefficient of the giant component. They also calculates the average geodesic(shortest) distance between any given pairs of authors in the graph. Finally, they also examine centrality by measuring both closeness and betweenness. They consider the case of non-weighted collaboration graph, in which all the edge has a unit cost, and a weighted graph, in which every edge has a weight depending on how strong the collaboration is.

Liu, Bollen, Nelson and Sompel [3] investigate the collaboration (co-authorship) in the research of Digital Libraries(DL). They use data from IEEE, ACM and joint ACM/IEEE Digital Libraries conferences. They used the similar metrics like Closeness, Betweeness and Degrees, and compare them to PageRank and AuthorRank(PageRank for weighted, directional networks). They validate their results against committee members of past conferences and shows that PageRank and AuthorRank provides better estimations.

Laniado and Tasso [4] investigated patterns of collaboration in Wikipedia. They employ the same methods used for researching academic co-authorship (described in previous papers), and used them to analyze the patterns of collaboration in Wikipedia. They present a scalable method of extracting co-authorship network from wikis revision history. They then analyze the network using the same metrics used in academic co-authorship analysis. Their results indicate that the pattern of online co-authorship are significantly different than academic co-authorship. Among the differences: very low values of mean distance and diameter, low clustering coefficient, strong centralization around some star members, etc.

B. Mining Github

Kalliamvakou et al. [5] investigated the quality and properties of the data available on GitHub. The paper presents a set of Perils and Promises that should be taken into consideration while mining Github for research purposes. For example, they show than 2/3 of the projects are personal, most projects are inactive, many active projects do not conduct all their software development in GitHub, etc.

C. Analyzing collaboration patterns in large developers networks

Surian, Lo and Lim [6] investigates collaboration patterns in large developers networks. They use data from SourceForge, and provide a high-level and low-level analysis of collaboration patterns. At the high-level analysis, they extract network-wide statistics for the network. At the low-level analysis, they extract common topological sub-graph patterns. The use a procedure called Mine Collaboration Pattern, that iteratively mine the top-k frequent graph collaboration patterns. An analysis of these pattern aim to answer questions about the connectivity of developers, the common topological patterns (e.g. the size of collaboration patterns, the relevance of the triadic closure principle, etc), and the average degree of separation (6.55).

Lima, Rossi and Musolesi [7] presents a characterization of GitHub as both a social network and collaborative platform. Using GitHubs events dataset, they extract both collaboration patterns and social ties, and perform comparative study. They show that the number of collaborators per project, contributors per project, stargazers per project and user followers per project show a power-law-like shape. They observe, for instance, that very active users do not necessarily have a large number of followers. They also investigate the impact of geography on collaboration and observe, for instance, that short-range communication is more common than long-range.

Thung, Bissyande, Lo and Jiang [8] analyze the network structure of social coding in GitHub. They model the network as a graph and calculate degree distribution, and shortest path between projects and between developers. They then use PageRank to find influental projects and influental developers.

D. Graph Processing

Most of the proposed metric for analyzing Co-Authorship are based on a graph representation of the relationships. Gonzalez et al. [9] present GraphX, a distributed graph processing framework embedded in Apache Spark. GraphX will be used in this work to model the developers network and calculate the Co-Authorship metrics.

References

[1] Christian Bird, Earl T Barr, Andre Nash, Premkumar T Devanbu, Vladimir Filkov, and Zhendong Su. Structure and dynamics of research collaboration in computer science. In *SDM*, pages 826–837. SIAM, 2009.

[2] Ergin Elmacioglu and Dongwon Lee. On six degrees of separation in dblp-db and more. *ACM SIGMOD Record*, 34(2):33–40, 2005.

[3] Xiaoming Liu, Johan Bollen, Michael L Nelson, and Herbert Van de Sompel. Co-authorship networks in the digital library research community. *Information processing & management*, 41(6):1462–1480, 2005.

[4] David Laniado and Riccardo Tasso. Co-authorship 2.0: Patterns of collaboration in wikipedia. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 201–210. ACM, 2011.

[5] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M German, and Daniela Damian. The promises and perils of mining github. In *Proceedings of the 11th working conference on mining software repositories*, pages 92–101. ACM, 2014.

[6] Didi Surian, David Lo, and Ee-Peng Lim. Mining collaboration patterns from a large developer network. In *Reverse Engineering (WCRE), 2010 17th Working Conference on*, pages 269–273. IEEE, 2010.

[7] Antonio Lima, Luca Rossi, and Mirco Musolesi. Coding together at scale: Github as a collaborative social network. *arXiv preprint arXiv:1407.2535*, 2014.

[8] Ferdian Thung, Tegawendé F Bissyandé, Daniel Lo, and Lingxiao Jiang. Network structure of social coding in github. In *Software Maintenance and Reengineering (CSMR), 2013 17th European Conference on*, pages 323–326. IEEE, 2013.

[9] Joseph E Gonzalez, Reynold S Xin, Ankur Dave, Daniel Crankshaw, Michael J Franklin, and Ion Stoica. Graphx: Graph processing in a distributed dataflow framework. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 599–613, 2014.