The Strength of Social Coding Collaboration on GitHub

**Gabriela B. Alves, Michele A. Brandão, Diogo M. Santana,
Ana Paula C. da Silva, Mirella M. Moro**

Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brazil

{gabrielabrant,micheleabrandao,diogo.marques,ana.coutosilva,mirella}@dcc.ufmg.br

Abstract. *Social coding is an approach of software development that enables cooperation among developers. Specially, GitHub can be modeled as a social coding network and its study allows the discovery of relevant patterns, e.g., the collaborations strength. Finding such patterns may help to improve the recommendation of developers and the evaluation of team formation. Here, our goal is to analyze the correlation between network properties and such strength.*

1. Introduction

Social coding is a software development approach that provides a collaborative environment to developers, encouraging them to share and discuss new ideas and knowledge [Dabbish et al. 2012]. Such approach is changing the software development process because developers can contribute to a project independently of their location, coordinate projects considering the information available from others and easily find projects to contribute to. In this context, data from social coding websites may help to define social networks (SN) that connect developers. Examples of social coding websites include Google Code[1] and GitHub[2]. Here, we study properties of the collaboration network that may represent the strength of the relationship between developers on GitHub, a website with 14 million users and 35 million projects (April 2016)[3].

Specifically, our goal is to use correlation analysis to identify the relationship between semantic and topological properties that measure the strength of social coding collaboration. Our hypothesis is that higher correlated properties can be used in a model to measure the strength of collaborations. There are studies measuring the strength of social connections with different goals on GitHub [Bartusiak et al. 2016; Casalnuovo et al. 2015], but none evaluates how different SN properties relate to the collaborations strength and what the best way to measure such strength is. Moreover, an appropriate model that measures the collaborations' strength can be used to predict cooperation [Bartusiak et al. 2016], and to evaluate team formation and productivity [Casalnuovo et al. 2015].

We model the GitHub SN as a graph in which the nodes are the developers and the edges between them represent their contribution on the same project. We then propose to measure the strength of social coding collaboration by considering three *new* semantic properties: the number of shared repositories (a directory where users can store their development projects), the jointly developers contribution to shared repositories and the jointly developers commits to shared repositories. In this work, the collaboration weight is assumed as a measure of collaboration strength. We also calculate topological properties

[1]Google Code: `code.google.com`

[2]GitHub: `github.com`

[3]The largest code host on the planet: `github.com/about/press` and `github.com/features`

of the SN and analyze the correlation with such strength. Finally, we combine some of these topological properties with the semantic ones. Overall, our research questions are: (*i*) which features can be used to define the collaborations strength on GitHub? and (*ii*) how are these features related to each other?

2. Related Work

The study of social networks is a powerful tool for discovering how individuals establish relationships, how they cluster themselves into communities as well as for predicting events and processes in the network [Brandão et al. 2013; Brandão et al. 2014; de Oliveira et al. 2015; Silva et al. 2016]. The social coding collaboration is a kind of interaction among individuals in a SN. Here, we focus on GitHub to study how to measure the strength of such interaction.

Many studies address different properties and behaviors on GitHub. For instance, Dabbish et al. [2012] investigate how the transparency on such website influences the way that individuals interpret and make use of information from others actions. The strength of collaborations on GitHub is also discussed to achieve distinct goals. For example, Tsay et al. [2014] show that project managers consider technical contribution practice and the strength of the social connection when deciding to accept or not a pull request. Likewise, Casalnuovo et al. [2015] explore the relationship among developers and find that they usually participate in projects in which they have prior connections.

These studies measure the collaborations strength considering different aspects from the developers' interaction: Tsay et al. [2014] use the social distance and prior interaction; and Casalnuovo et al. [2015] consider the sum of all prior shared projects normalized by the number of developers in the project prior to the join time. However, these studies do not investigate which SN properties influence such strength and how they affect it. Then, our goal is to fill such a fundamental gap.

3. Methodology

We now present the GitHub data, the network model, and the strength of their relationships. Finally, we also overview the topological properties considered in our analyses.

Data Description. We extract data from the GHTorrent database, an open project that collects data from public repositories on GitHub [Gousios 2013]. This project monitors GitHub public events timeline (e.g. CreateEvent). Initially, we consider a complete dataset collected on September 15, 2015, with 1,987,760 projects (32 GB of data). From those projects, 1,204,212 were forked[4]. Then, we remove the forked repositories because the changes made on forked repositories must be approved by the base repositories (done through a *pull request*)[5], resulting in a 529,405 non-forked projects. As our goal is to build the collaboration network established by the developers, and some network metrics have high computation cost, we prune the projects to consider only those developed using a particular programming language: JavaScript (the one with the largest number of projects and average number of changes pushed per repository[6]), which accounts for

[4]Copy of a repository and allow users to experiment with changes without affecting the original project.
[5]Changes committed to an external repository must be approved in the base repository
[6]GitHub Info: `githut.info`

Table 1: Topological properties: Given two nodes u and v, let $\mathcal{N}(u)$ and $\mathcal{N}(v)$ be the set of neighbors of u and v, $w(u)$ and $w(v)$ be the weighted degree (a sum of the weight of each edge connected to the node), and $w(u,v)$ be the weight of the edge between u and v.

Metric	Definition	Interpretation
Clustering Coefficient (CC)	Let $T(u)$ be the total number of triangles that u belongs to and $deg(u)$ the degree (number of connected edges) of u, then CC is: $CC(u) = \frac{2T(u)}{deg(u)(deg(u)-1)}$.	It shows the tendency of the nodes to cluster together. We do not use CC to investigate the correlation with the strength of collaborations, because it is a measure of the nodes no of the edges.
Neighborhood Overlap (NO)	It measures the neighborhood similarity for any two pair of nodes and is computed as: $NO(u,v) = \frac{\lvert\mathcal{N}(u)\rvert\cap\lvert\mathcal{N}(v)\rvert}{\lvert\mathcal{N}(u)\rvert\cup\lvert\mathcal{N}(v)\rvert}$.	According to Easley and Kleinberg 2010, NO can be used to compute the strength of the links. The higher the value of NO, the stronger the relationship.
Adamic-Adar Coefficient (AA)	This metric in network context is customized as: $AA(u,v) = \frac{\sum \forall z \in \lvert\mathcal{N}(u)\rvert\cap\lvert\mathcal{N}(v)\rvert}{log\lvert\mathcal{N}(z)\rvert}$.	Neighbors that are not shared with many others receive more weight.
Preferential Attachment (PA)	The greater the number of neighbors of a node, the higher the value of preferential attachment, defined as $PA(u,v) = \lvert\mathcal{N}(u)\rvert\lvert\mathcal{N}(v)\rvert$.	According to Barabási and Albert 1999, there is a linear relationship between the number of neighbors of a node and the probability of attachment (i.e.,"the rich get richer"). Thus, we investigate such claim in GitHub.
Resource Allocation (RA) – Also known as Propagation Coefficient (PC)	Let $w(u,z)$, $w(z,v)$, $w(u,v)$ be the weight of links (u,z), (z,v) and (u,v), respectively. Then $RA(u,v) = \frac{w(u,v)}{w(u)} + \sum_{z\in\mathcal{N}(u)\cap\mathcal{N}(v)} \frac{w(u,z)w(z,v)}{w(u)w(z)}$.	Considering GitHub SN, a developer u is viewed as sharing repositories and/or committing code with all of his/her contributors, which has a secondary effect to all of the contributors of a developer v who is influenced by it. No studies have used RA to measure the strength of social coding collaboration.
Tieness (T)*	$T(u,v) = \frac{\lvert(\mathcal{N}(u)\cap\mathcal{N}(v))+1\rvert}{\lvert\mathcal{N}(u)\cup\mathcal{N}(v)\rvert} w(u,v)$. Note that, in the denominator, v is not counted as neighbor of u and vice-versa.	It was evaluated measuring the strength of co-authorship ties [Brandão et al. 2016]. Here, we study tieness in the context of social coding collaboration.

**We calculate RA and T three times since we consider the semantic properties ($SR_{(A,B)}$, $JCSR_{(A,B)}$ and $JCOSR_{(A,B)}$) as a weight in the SN at a time.*

90,363 repositories (17% of non-forked projects) and 37,691 developers[7].

Network Model. The GitHub social coding collaboration SN is mathematically represented by a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ the set of nodes and $\mathcal{E}$ the set of non-directed links. Nodes are developers, a link between any two developers exists if both of them contributed to the same repository, and the link weight measures the strength.

New Semantic Properties. We introduce three semantic properties to measure collaboration strengh (edge weight).

1. Number of shared repositories ($SR_{(A,B)}$): Given any two developers A and B, the set of repositories they shared is given by $\mathcal{R}$. The metric $R_{(A,B)}$ is the total number of repositories that they both worked at, and is given by the cardinality of $\mathcal{R}$ set (i.e., $|\mathcal{R}|$).

2. Jointly developers contribution to shared repositories ($JCSR_{(A,B)}$): Consider two repositories r_1 and r_2. The r_1 repository is only shared by developers A and B. So, their jointly contribution to r_1 ($JCSR_{(A,B,r_1)}$) is equal to 1. Instead, r_2 is shared by developers A, B and C. Then, the jointly contribution given by A and B to r_2 ($JCSR_{(A,B,r_2)}$) is 0.66. If A and B share only r_1 and r_2, the jointly contribution given by them to these repositories ($JCSR_{(A,B)}$) is 0.83. Formally, the jointly contribution is:

$$JCSR_{(A,B)} = \frac{\sum\limits_{\forall r_i \in \mathcal{R}} JCSR_{(A,B,r_i)}}{|\mathcal{R}|}.$$

3. Jointly developers commits to shared repositories ($JCOSR_{(A,B)}$): Given $NC_{(A,r_j)}$ as the total number of commits by A into repository r_j, $NC_{(B,r_j)}$ as the total number of commits by B into repository r_j, and $NC_{(r_j)}$ the total number of commits by any

[7]Dataset publicly available at `github.com/lab-csx-ufmg/GitHub-SN`

developer into repository r_j. $JCOSR_{(A,B)}$ is defined as:

$$JCOSR_{(A,B)} = \sum_{\forall r_i \in \mathcal{R}} \frac{(NC_{(A,r_i)}+NC_{(B,r_i)})}{NC_{(r_i)}}.$$

The semantic properties capture the amount of interaction between two developers. Then, the higher their values, the stronger the interaction between those two.

Existing Topological Properties. We characterize the GitHub SN using topological properties in a symmetric way, which capture the patterns of individuals' interaction [Brandão et al. 2016]. Besides the characterization, we study how these properties are correlated with the strength degree. Table 1 summarizes the properties used in our analyses.

4. Results

We first evaluate the intensity of connections among developers on GitHub by analyzing the average clustering coefficient (CC) and the average neighborhood overlap (NO) of the SN. The average CC is 0.735 and NO is 0.897, i.e., both high values. Such high values reflect the way the network is built: for repositories with a large number of collaborators, all of them will be connected among themselves. Therefore, the developers in the social network tend to form clusters and share a large number of neighbors.

In this work, we propose three semantic properties to measure the strength of a collaboration on GitHub (Section 3). Figure 1 presents the distribution of SR, JCSR and JCOSR for pairs of developers in the SN. In general, most pairs have a small value for such properties. Hence, when the strength of social coding collaboration is measured by one of the proposed properties, most pairs of developers have a *weak* relationship (then agreeing with [Newman 2001], the seminal work that evaluates collaboration in four distinct networks and discovers patterns similar to our findings).

Now, our goal is to analyze the correlation between semantic and topological properties that can be used to measure the strength of social coding collaborations. This study helps to develop a new model to measure such strength that may better represent the real level of collaboration and improve, for example, algorithms that recommend developers to work in a project. Indeed, we correlate our semantic properties with all the topological properties defined in Section 3 by using Pearson and Spearman correlation coefficients[8]. After calculating both coefficients, we analyze the results following Cohen 1988's convention to interpret correlation coefficients (ρ): very large ($\rho \geq 0.7$), large ($\rho \in [0.5; 0.7)$), moderate ($\rho \in [0.3; 0.5)$), small ($\rho \in [0.1; 0.3)$), and insubstantial ($\rho < 0.1$)[9].

Figure 2 presents the correlation matrix for the properties in GitHub SN. Due lack of space, we analyze only the unexpected correlations. The least correlated property is SR, with values near zero for both correlation coefficients. The reason is SR only accounts for the number of repositories to which two developers contributed. Thus, it alone does *not* capture the strength of a collaboration. For the cases in which SR is used as weight for the T_SR measure, a high positively linear correlation with neighborhood overlap is found. As neighborhood overlap has been used to measure the strength of relationships and both metrics are linearly dependent, this result may suggest T_SR is also adequate to measure the strength of collaborations.

[8]Pearson: linear relationship between two sets of data; Spearman: monotonic relationship in data.

[9]In negative correlations, the same ranges are valid)

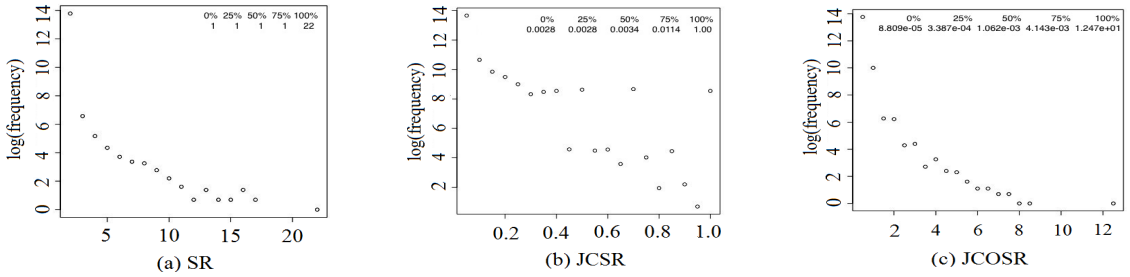

Figure 1: Semantic properties distribution: Large pairs of developers with a small (a) SR, (b) JCSR and (c) JCOSR.

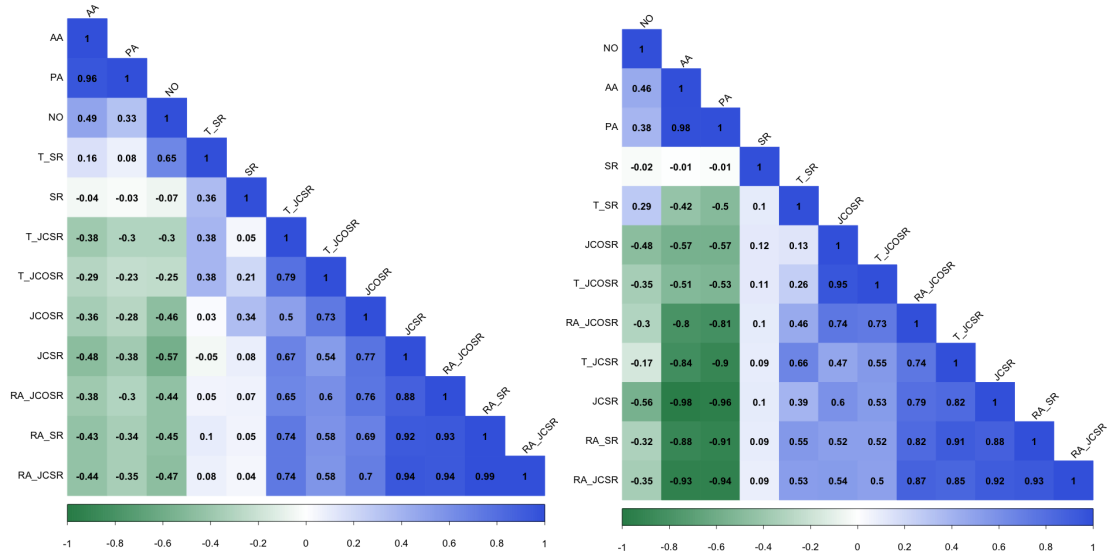

Figure 2: Correlation matrix based on (a) Pearson correlation coefficient and (b) Spearmans rank correlation coefficient for semantic and topological metrics in GitHub SN.

We emphasize the correlation between JCSR and JCOSR since both properties consider different aspects from the relationship between a pair of developers (the former considers the amount of shared projects and the latter, the amount of commits). Thus, the sum of jointly contributions to shared projects is directly related to the sum of jointly commits to shared projects. Such properties should be considered together in a model to measure the collaborations strength. Another interesting result is: RA_JCOSR, T_JCSR, JCSR, RA_SR and RA_JCSR are very negatively monotonic correlated to AA, PA and NO. Note that RA_JCOSR, RA_SR and RA_JCSR have similar behavior regardless the considered weight. Likewise, T_JCSR and JCSR are very negatively monotonic correlated to AA and PA. Thus, it may indicate that contributing to many repositories does not mean to attract many contributors. Further analyses are necessary to conclude that.

JCOSR is large and very large linearly and monotonically correlated with JCSR, T_JCOSR, RA_JCOSR, RA_SR and RA_JCSR. Such result is similar to JCSR. This supports our previous claim that both properties should be considered together. Another important behavior is that the resource allocation with the three weights is large and very

large linearly and monotonically correlated with JCSR and JCOSR. Thus, it is important to consider these properties in the measure of the strength as well.

5. Concluding Remarks

In this paper, we proposed a social coding collaboration network model to GitHub dataset and three new semantic properties to measure the strength of collaborations. Then, we investigated the correlation of these properties with existing topological ones. Our results showed the number of shared repositories is not a significant indicator of the collaborations strength. Furthermore, the JCSR and JCOSR are very large linearly and monotonically correlated with other properties. As future work, we plan to further analyze the correlations among the metrics, consider other semantic aspects from the GitHub SN and define a model to measure the strength of a social coding collaboration.

Acknowledgments. Research partially funded by CAPES, CNPq and FAPEMIG.

References

Barabási, A. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

Bartusiak, R. et al. (2016). Cooperation prediction in github developers network with restricted boltzmann machine. In *ACIIDS*, pages 96–107.

Brandão, M., Moro, M. M., and Almeida, J. M. (2013). Análise de fatores impactantes na recomendação de colaborações acadêmicas utilizando projeto fatorial. In *SBBD Short Papers*, pages 1–6.

Brandão, M. A., Diniz, M. A., and Moro, M. M. (2016). Using topological properties to measure the strength of co-authorship ties. In *BRASNAM/CSBC*, pages 199–210.

Brandão, M. A., Moro, M. M., and Almeida, J. M. (2014). Experimental evaluation of academic collaboration recommendation using factorial design. *JIDM*, 5(1):52.

Casalnuovo, C., Vasilescu, B., Devanbu, P., and Filkov, V. (2015). Developer onboarding in github: The role of prior social links and language experience. In *ESEC/FSE*, pages 817–828.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, second edition.

Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. (2012). Social coding in github: transparency and collaboration in an open software repository. In *CSCW*, pages 1277–1286.

de Oliveira, D. M. et al. (2015). Uma estratégia não supervisionada para previsão de eventos usando redes sociais. In *SBBD*, pages 137–148.

Easley, D. and Kleinberg, J. (2010). *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press.

Gousios, G. (2013). The ghtorrent dataset and tool suite. In *MSR*, pages 233–236.

Newman, M. E. (2001). The structure of scientific collaboration networks. *NAS*, 98(2):404–409.

Silva, T. H. P., Rocha, L. M. A., Silva, A. P. C., and Moro, M. M. (2016). 3c-index: Research contribution across communities as an influence indicator. *JIDM*, 6(3):192–205.

Tsay, J., Dabbishand, L., and Herbsleb, J. (2014). Influence of social and technical factors for evaluating contribution in github. In *ICSE*, pages 356–366.