Using Formal Concept Analysis to study social coding in GitHub

Otmar M. Pereira Junior\textsuperscript{1}, Luis E. Zárate\textsuperscript{1}, Humberto T. Marques-Neto\textsuperscript{1}, and Mark A. J. Song\textsuperscript{1}

\textsuperscript{1}Department of Computer Science, Pontifical Catholic University of Minas Gerais
Belo Horizonte, Minas Gerais Brazil

Abstract—Currently, the world watches the expansion of the social networks phenomenon in the form of several different networks: Facebook, Twitter, Instagram, GitHub and many others. Such sites join together a user base that can reach millions of people connected from places everywhere in the planet. These impressive marks motivate questions about the characteristics of the members of these networks, due to opportunities present in them that can generate economical gains, augment the influence and popularity of an organization or product.

This paper shows some of the main attributes of the contributors of the 20 most popular projects in GitHub using Formal Concept Analysis to uncover the most relevant characteristics. Despite the global success of the projects, we see geographical clusters of developers behind the most popular projects, what can serve as motivation to engage developers of other regions and increase the cultural variety in popular open source projects.

Keywords: Social Coding, Formal Concept Analysis

1. Introduction

The recent expansion of the social networks reached several different domains, ranging from music preferences to professional networking. Among these platforms, one can find social networks devoted to the development of open-source software. One of the first instances of this kind of social network is SourceForge, still very popular.

More recently, in 2008, GitHub was founded bringing several social features, what made it experiment a great success since then and take the leadership of its category after only three years of existence, having even more affiliated users than the popular SourceForge [1].

Such growth made GitHub object of several works in the community, as observed in [16], [17], that describes the behavior of GitHub members on the Internet. On the other hand, the authors of this paper could not find any work that tries to explain the causes of the formation of the developer networks of the main projects in GitHub in the same sense of the papers of Yu [18], Singh [12] and Gao [19], that describe the communities formed in SourceForge.

Previous works, like [21], show that some companies are already paying attention to platforms like GitHub to more accurately evaluate technical skills and accomplishments of IT professionals. Nonetheless, GitHub does not provide tools to reduce the efforts of Human Resources professionals willing to understand the typical profile of developers that have skills similar to the ones of a popular project of a given technology. We believe that GitHub data can be valuable indicator of developers possessing certain skills and proven experience.

This paper aims to study the developer networks around the most popular software repositories in GitHub by applying Formal Concept Analysis (FCA) to discover the most relevant common attributes of the developers contributing to GitHub’s most successful open source projects using public information of the developers available via GitHub API.

2. Social Coding

The evolution of information and communications technology in the second half of twentieth century, specially in the last 15 years, made the contemporary society highly connected through complex networks, much like in the form described in [15]. As part of this process of increasing interconnection between people all around the world, social networks linking individuals with common interests of several different kinds arose: Facebook reuniting friends, professional networking in LinkedIn, photos and video sharing in YouTube and Instagram and open source software development in SourceForge, GitHub, Google Code and many others.

Among the categories of social networks, the category that brings together software developers is very popular and has SourceForge as one of its main examples since 1999. Still in the current days, SourceForge has over three million registered members. Many popular open source software are hosted in SourceForge, that continues to be a friendlier place to end users than GitHub. While we see many applications redirecting users to download installation binaries in SourceForge, we cannot observe a similar pattern in GitHub, that seems to be more targeted to developers. This may be one of the reasons that GitHub does not provide via its API a downloads counter of the project binaries and even the Downloads API has been deprecated.

Other social networks, such as Facebook, MySpace and Orkut have popularized Web 2.0 by giving a special emphasis on social aspects of the applications used by people on the Internet. In addition to the characteristics found in SourceForge, GitHub put in the spotlight social features of its developers’ networks such as user feeds, following users and projects, collaboration to project’s wiki pages and visualization of graphs showing the connections of members.
Another facet of collaborative development that has been promoted by GitHub is the use of distributed version control systems instead of centralized ones, like Subversion and CVS. Developed by the famous and acclaimed Linus Torvalds to help in the development of Linux kernel, Git was one of the first distributed version control systems, where different versions of the source code are not in a single server. Each development machine contains all the previous versions of the software, instead of only the workspace version, as is the case when working with centralized version control systems. This paradigm reduces significantly common issues in concurrent software development, like resolving conflicts when merging files before a release. GitHub promotes Git as its main version control system and some of its features to promote social interactions, like forking repositories and sending pull requests to other developers.

This paper studies a network formed by developers that have been tagged as contributor to at least one of the 20 projects in GitHub that had received most stars as of September 2013 in GitHub. We aim to unveil the main characteristics shared between the people that work on open source software projects by using Formal Concept Analysis (FCA) to discover association rules and implication rules between the attributes exposed by the GitHub API [7], that exposes some geographical and personal data about each developer.

### 2.1 Selected Dataset

In the quest for understanding the popularity of projects in GitHub, we concentrated our efforts on the data of the 20 projects that had been starred the most by GitHub members as of September 2013. The selection of the projects was based upon data available in GitHub Archive [3] and can be seen in Table 1.

<table>
<thead>
<tr>
<th>Project name</th>
<th>Stars received</th>
</tr>
</thead>
<tbody>
<tr>
<td>bootstrap</td>
<td>60355</td>
</tr>
<tr>
<td>jquery</td>
<td>25750</td>
</tr>
<tr>
<td>node</td>
<td>25541</td>
</tr>
<tr>
<td>html-boilerplate</td>
<td>22924</td>
</tr>
<tr>
<td>rails</td>
<td>19987</td>
</tr>
<tr>
<td>d3</td>
<td>19685</td>
</tr>
<tr>
<td>Font-Awesome</td>
<td>18594</td>
</tr>
<tr>
<td>impress.js</td>
<td>18256</td>
</tr>
<tr>
<td>angular.js</td>
<td>16310</td>
</tr>
<tr>
<td>backbone</td>
<td>16142</td>
</tr>
<tr>
<td>chosen</td>
<td>13297</td>
</tr>
<tr>
<td>jQuery-File-Upload</td>
<td>13777</td>
</tr>
<tr>
<td>three.js</td>
<td>13249</td>
</tr>
<tr>
<td>jekyll</td>
<td>12906</td>
</tr>
<tr>
<td>brackets</td>
<td>12900</td>
</tr>
<tr>
<td>gitignore</td>
<td>12467</td>
</tr>
<tr>
<td>gitlabhq</td>
<td>11023</td>
</tr>
<tr>
<td>meteor</td>
<td>10265</td>
</tr>
<tr>
<td>textmate</td>
<td>9221</td>
</tr>
<tr>
<td>select</td>
<td>8750</td>
</tr>
</tbody>
</table>

Table 1: List of the projects and stars given by GitHub users as of September 2013.

The criteria of received stars has been chosen due to the belief in it as an indicator of the project popularity and interest upon the project. According to GitHub API documentation [8], stars are used to bookmark a repository and show an approximate level of interest, what makes it a good indication of project popularity. It is similar to the metric employed by [12] to measure the commercial success of an open-source project in SourceForge platform. Moreover, the authors consider this metric the most appropriate among the ones available in GitHub API to evaluate the success of project.

The gathering of the top projects’ ranking was conducted using the tool Google Big Query [2], that imposes some restrictions on the volume of data processed by the submitted queries. Such restriction directed our efforts on the analysis of the 20 most popular projects. Because of that, in this work its expected to observe trends applicable only to the most prominent projects, and attributes that set these projects apart of the less popular ones should be object of other papers devoted to analysis of much large scale, including hundreds or thousands of projects. In spite of the limited number of projects analyzed, Table 1 shows some indications of the rich gets richer effect described by [15]: very few projects concentrate most of the characteristics and in general, these very few instances are the ones that continue to grow. We see that in a universe of millions of repositories, the bootstrap repository has almost ten times more stars received than the 20th repository of the ranking.

### 2.2 Network construction

After selecting the projects to analyze data from, further data was retrieved to build contributors’ network of the projects. The co-participation of two developers in a project serves as link to this network that was built as an undirected graph. The information about the contribution to a project is available via GitHub API [7] and in order to consolidate the data, an Extract-Transform-and-Load process has been implemented in the tool Pentaho [9]. The list of user attributes retrieved from GitHub API is shown in the section 3.1 and the data gathering process is depicted in Figure 1.

After having the 20 project’s contributors data collected, another Pentaho transformation has been implemented to resolve geographic information - continent and country - for each developer using the Microsoft Bing Maps API [10]. Since the location information for each developer is filled in GitHub in natural language form in many different ways - some inform just their country or province while many others give only their city name - a more powerful geographical resolution API was put in place to improve the information about the location of each member of the network.

Finally, after collecting all the attributes about each developer, a Java program has been implemented to generate a
file in the format of the PAJEK software. This format is very popular and can be imported by a great number of network analysis tools.

In the the network constructed after applying all these steps, one can observe the small-world phenomenon, originally studied by Stanley Milgram [11]: the graph diameter is only 5, what means that every developer of the 5,385 contributors in this network knows each other through partaking in the same project with at most four other developers. One of the factors that contribute to the low diameter of this graph is certainly the significant number of partaking between pairs of developers: 3,857,471 were found, making this network a fairly dense one.

3. Formal Concept Analysis

Formal Concept Analysis (FCA) is a recent knowledge area that has been in expansion since the 1980s and interesting results have been obtained by its application to hierarchize and discover concepts that were not immediately evident in the original form of the given problem.

The studies in this area involve the definition of a formal context, relating the elements being analyzed and the attributes of these objects.

A formal context is represented by a triple \((G, M, I)\), where \(G\) is the set of the objects, \(M\) is the set of attributes corresponding to some of the elements’ properties under analysis and \(I \subseteq G \times M\) is an incidence relation that indicates which objects contain which attributes. In some cases, the attributes have a single valued domain, where the attribute is whether true or false. It is also possible to model attributes corresponding to domains of higher cardinality, like colors or size - small, medium, large. In this work, for the sake of simplicity, we model a formal context where all the attributes are of the former kind.

A concept of a formal context is a pair \((A, B)\), where \(A \subseteq G\), is defined as the extent of the concept and consists of all the objects pertaining to the concept. The set \(B\), where \(B \subseteq M\), is the intent of the concept is formed by all the common attributes shared between the objects of the extent set.

The construction of a formal context as defined before enables one to build a conceptual lattice that acts as a hierarchy of concepts and enables the discovery of new concepts - unexpected combinations of objects and attributes - and the relations between the objects and attributes.

Some algorithms can be applied to find interesting patterns of attributes that appear together in association rules and implication sets that are not trivial neither redundant. In general, algorithms studied by FCA academics and available in some tools are used to mine logical dependencies between the attributes and groups of instances that given a subset of attributes are frequently found together.

We recommend readers interested in more details about Formal Concept Analysis to refer to [4] for a thorough explanation of the mathematical foundations around the Formal Concept Analysis.

3.1 Formal Context definition

In this work, the set of objects \(G\) has been defined as the set of developers who contributed to the 20 selected projects. These developers have several properties describing them in their profile and 36 of them were used to construct the set of binary attributes, \(M\), which elements are shown in the following list with a description of each one of the selected properties:

1) Indicator of whether or not the registered user account is personal. When an account is not registered as a personal one, it means that it has been registered in the name of an organization.
2) Flag that indicates if the user administrates a project in GitHub.
3) Profile information indicating member availability for hiring.
4) The contributor has more than 50 public repositories in GitHub.
5) The user has already shared more than 50 gists - code snippets used in discussion with other members of
GitHub.

6) The biographical information on the user profile is very long. As inspection of some profiles having this characteristic showed that the biographical information, when very long, is used by developers to publish their curriculum vitae.

7) The member follows more than 1,000 other GitHub users.

8) The developer has more than 1,000 followers.

9) The user account was created more than three years ago.

10) Member location is in North America.

11) Person location is South America.

12) Developer is based in Europe.

13) User’s location is Africa.

14) Asia is the informed location by the user.

15) The developer has informed his or her location as somewhere in Oceania.

16) The last 20 attributes indicate the participation in project $i$, where $i$ indicates the participation of the member in the project indexed by $i$ in the list of projects sorted alphabetically. For example: a developer who contributed to the first project in the list, angular.js, will have the Participated$_1$ attribute set to true, while someone who has not contributed to the last project in the list, textmate.js, will present the attribute Participated$_{20}$ indicating false for such participation.

In order to represent and analyze the formal context, a specific tool designed to build concept lattices from formal contexts and calculate implication sets, association rules and new concepts. Conexp [5] software was the tool elected to help on the analysis of the formal context.

All the attributes mentioned above were retrieved from the network built previously and imported into Conexp [5]. Figure 2 shows part of the formal context defined in this section. The rows correspond to the login of GitHub members and the columns are marked with an X when an attribute of the list 3.1 is true and is left blank otherwise. More details about this tool can be found in its documentation page [6].

4. Results

The analysis of the formal context was conducted using Conexp software. In this section, the most relevant rules will be discussed. Conexp computed a total of 1,822 concepts, 758 implication sets and 872 association rules involving different combinations of the attributes found on the input data. The most prominent combinations are discussed next.

Implication sets contain logical implications involving the attributes of the given formal context. Although it is possible to analyze all the combinations of the finite set of attributes and objects of the formal context, clearly, many of them would be trivial or redundant. It is thus necessary to restrict the set of implications that should be analyzed. Conexp was used to compute the minimal set of rules and the most relevant characteristics of these implications - frequent attributes - are discussed below.

- The implication User account is personal $\rightarrow$ user account created more than three years ago is true for 5,365 of 5,385 developers. This means that in general, companies and other institutions do not contribute to the popular open source projects in GitHub. Furthermore, the developer is an old member of GitHub. Popular projects in GitHub seem to be an space where newcomers may not be welcome: in the vast majority of the cases analyzed, the attribute that indicates the user as having an user account registered more than three years ago prevailed.

- Results regarding user’s geographical information: 421 rules involve North American members; 342 developers based in Europe; 204 have developers with undetermined location; 187 relate South American developers; 133 rules involve Asian members of GitHub, 115 from the Oceanic continent and African developers are part of just 34 implication rules. Despite the fact of being a platform spread across the world, there is a great deal of concentration of users in North America and Europe. Surprisingly, in the Asian continent we find less contributors than in South America, a continent having approximately 10 times less inhabitants.

- The implication available for hiring $\rightarrow$ User account is personal AND user account created more than three years ago that is true for 1,277 members indicate that long time contributors are also paying attention to new professional opportunities in GitHub.

5. Conclusions

In this paper we have seen that the Formal Concept Analysis can be a valuable tool to analyze very large scale social networks, like GitHub, which user base reaches marks around millions of users and projects. The theoretical and practical framework in Formal Concept Analysis field enables the application of techniques to study social network characteristics. This was the case with GitHub, where
relevant knowledge related to the geographical location and affiliation time to the network was gathered by analyzing the output of the FCA algorithms.

It was possible to realize that despite the possibility of connecting people around the world, some clusters are geared towards the most developed locations, namely North America and Europe, contrasting with a very low number of users in the African continent. This brings the question for future works that investigate the causes of such clustering: do they appear due to social-economic factors or is the social network built around people from the same region? The prevalence of longtime user accounts in GitHub as contributors of the most popular projects is also a factor to be taken into account, given the possible importance users seem to give to the past contributions of the developers to the community.

Besides all these questions around attributes of the members, there is a difficulty that is present when trying to analyze social networks: the manipulation of an ultra-large volume of data that create several technical challenges to manipulate the information and execute algorithms to analyze input data.

Due to the huge amount of information, there is no tool capable of accessing very recent GitHub data, and this is aggravated by the addition of a higher volume of information in GitHUb than the amount of data that can be collected by a single machine during the same period of time - GitHubArchive reports peaks of 50,000 events per hour in GitHub, while the GitHub API imposes a restriction of 5,000 requests per hour for authenticated users that are querying its database. Initiatives like an open platform available to manipulate the information and execute algorithms to analyze input data.

Acknowledgements

The authors acknowledge the financial support received from the Foundation for Research Support of Minas Gerais State, FAPEMIG, through Project CEX PPM 107/12, and the National Council for Scientific and Technological Development, CNPq, Brazil.

References