Organizational Social Networks in Apache Software Foundation Projects

By

BENJAMIN DAVID MISHKANIAN
B.S (University of California at Davis) 2014

THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

____

Premkumar Devanbu, Chair

____

Vladimir Filkov

____

Cindy Rubio González

Committee in charge

2016

Contents

List of Figures

List of Tables

Benjamin David Mishkanian
September 2016
Computer Science

Organizational Social Networks in Apache Software Foundation Projects

Abstract

put the abstract here

Acknowledgments

xxxxxxx

Preface

Chapter 1

Introduction

Chapter 2

Data Collection

2.1 Initial Research Question and Data Collection

The original research question for this project was to identify differences in commit behavior between paid and volunteer contributors of open source projects. In order to classify contributors as paid or volunteer, a script was written which collects data from various public online profiles of ASF contributors active in 2015 and computes several heuristics which can then be used as features of a classifier. However, two issues made it difficult to identify the differences between the two classes:

1. For a given ASF project and time period, the proportion of active contributors who are paid to work on the project can be very high, sometimes nearing 100

2. A profile can contain certain markers that are highly correlated with paid contributors, but there are no known markers that are highly correlated with volunteer contributors.

Issue #1 was a surprising finding in its own rightwe expected there to be a somewhat even mix of paid and volunteer contributors, but while doing a manual classification of a couple projects, we found these projects to be almost exclusively developed by employees of a few companies during the time period under observation. With such skewed samples, it became difficult to write a classifier sensitive enough to be useful. Issue #2 was an unforeseen limitation of using profile information to identify social group affiliation. Ultimately, there was no reliable way to differentiate between a volunteer and a paid contributor when his/her data does not reveal any obvious link between his/her contributions and his/her employer. Although creating this classifier proved infeasible, we were able to re-purpose our data to do a social network analysis of ASF contributions instead. In particular, the following facets of the dataset were useful for this purpose:

- The dataset associates data from multiple accounts/profiles to one contributor, improving the reliability of finding a given contributors employer.
- Since the contributors employer was one of the data points which the algorithm used, a fair amount of employer data was already available.

2.2 Data Sources

When researching open source software, there are a multitude of possible software repositories and organizations to base the experiment on. For example, some prior work studied SourceForge projects, while others studied GNOME. We chose to study Apache Software Foundation projects because they have a consistent organization structure with respect to how developers collaborate on Git, Github, and JIRA. This structure made it simpler to write a script that can reliably link user identities across these service. To find the employer of a given Git committer, the script attempts to collect data from the Git account, and any associated JIRA and/or Github accounts. It also utilizes the Google Custom Search API to find links to LinkedIn profiles, which can then be inspected manually. Each set of values is stored in a separate entry in a PostgreSQL database, to maximize the amount of information available for finding a contributors employer. The following information is mined from Git, Github, and JIRA accounts:

- Username
- Email address
- Display name

The following additional information is mined from Git accounts:

- Commit count per project

The following additional information is mined from Github accounts:

- Location
- Company name
- List of organizations

Furthermore, to work around the rate limit of the Github API, we utilized a GHTorrent.org database dump as an offline cache of Github data.

2.3 Data Collection Constraints

Since a contributors employer name is information which can only be obtained if the contributor chooses to reveal it, there will typically be some individuals for which we have no employer name. A further constraint is that if this information cannot be mined automatically for most people, it becomes infeasible to fill missing values in the dataset through manual inspection. After performing some case studies, we found that this can be a significant problem for larger projects such as Apache Kafka, which have many individual committers. However, we found that the vast majority of commits to a project typically came from a small group of the committers, for which we could more easily find their employer. For this reason, we limited our data collection to only the prolific contributors of each project, defined as contributors in the minimum size set S such that the sum of the number of commits done by members of S is at least 80% of the number of commits done to the project for the time period under observation. This also had the effect of improving the number of employer names mined automatically, because prolific contributors seem to be more likely to keep their Git, Github, and JIRA profile information up-to-date. An additional constraint was that we only mined data for ASF projects that met the following constraints:

- The project is listed in the GHTorrent.org database
- The project has a JIRA hosted at `issues.apache.org/jira/`
- At least 20 commits were done to the project within the period under observation

These constraints resulted in the exclusion of projects which lacked sufficient data to analyze.

2.4 Data Collection Algorithm

The script jiradb.py performs the majority of the data gathering. Its algorithm can be summarized as follows:

1. Get JIRA account data for contributors to the JIRA of the projects
2. Get Git account and commit data for committers to the projects
3. Get Github account data for Git accounts
4. Associate accounts belonging to a single person under a single contributor ID

The following sections provide more details on these steps.

2.4.1 JIRA Data

The jira Python package was used to query the ASF JIRA at `issues.apache.org/jira/`. JIRA can be queried programatically by using the JQL (JIRA Query Language). The following JQL query was used to

```
project = "{0}" AND created < "{1}" AND (created > "{2}" OR resolved < "{1}")
```

2.4.2 Git Data

2.4.3 Github Data

2.4.4 Identity Merging

Appendices