COMP5313 Assignment 2 Report
Distribution of Popularity and
Effect of Coexisting Languages in Repository Network

Zhenjiang Zhu SID: 420103149

Due: June 4, 2015

1 Introduction

Link Analysis is widely applied on exploring the structure of large scale network. Open source projects are important in computer science and software engineering. Many popular software are originated from open source projects. A successful open source project begins as a small project and gradually evolve into a large and complex one. It is well known that a complex software could not be written by a single person or a small group of programmers. Collaboration and involvement are essential in the open source community. The collaboration, and the use of languages also create a large scale network across the programmers around the world.

There are many repository hosting website for programmers to host their code and collaborate with each other. *GitHub* is one of the most popular hosting websites. This report presents the work of exploring two properties of the datasets on the GitHub. The first one studies the distribution of repositories in terms of popularity and the second one studies the how programming languages having similar purpose impact each other. We use the open dataset from GitHub for the analyses.

2 Power Law Distribution of Open Source Project

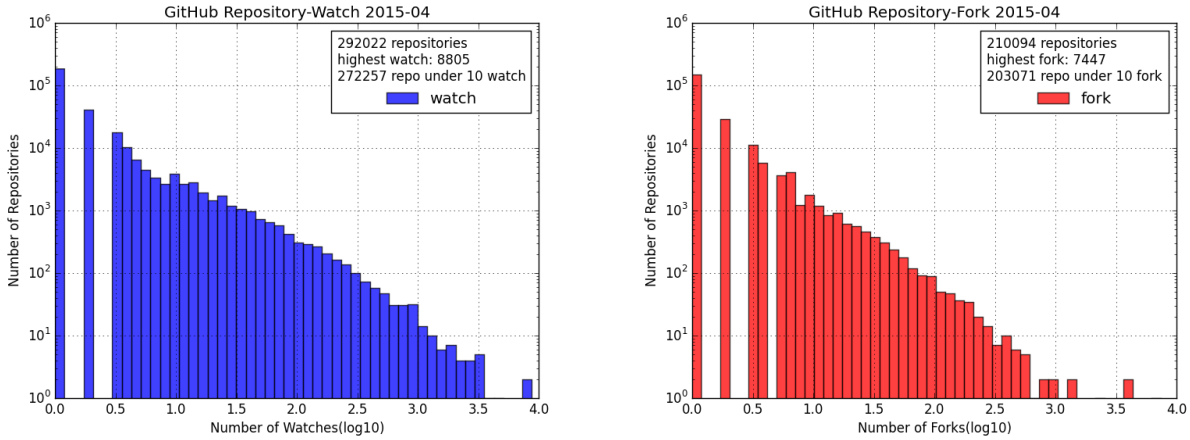
2.1 Introduction

The open source projects and projects of free users on the GitHub are public, which means the code is available for browsing. If a user find a repository interesting, he is able to "star" the project, so that he saves it into favourite list. A user is able fork an entire repository and keep it as his or her own copy as well. It is natural to consider that to fork a repository means the user is a bit more interested in the code than someone who simply "star" a repository.

There is a dedicated page on GitHub called *GitHub Trend*. People are able to find the *trending* repositories on the page. The trending repositories are sorted by the number of "stars" it receives in a day, a week or a month. The trending page is like a news website, where the latest and most popular repositories are listed. We consider that the pattern of popularity of the repositories could be similar to that of news: there are many repositories, but only seldom of which become popular and successful.

2.2 Motivation and Problem Statement

On the music download website, there is an interesting phenomenon that only a few pieces of songs gain extremely high popularity while most of the songs are not very popular. An experiment had been done by the Salganik et al. [4] by setting up a website for music download. They recorded down the number of downloads for each song and displayed it to users. Each user will be redirected into a different version of the web page where the list of popular songs is different. They concluded that due to the "social influence", people would like to download the more popular ones. Though they finally concluded that it is unpredictable whether a song will be popular, and there are many other factors that are not applied in the experiments



a) watch count b) fork count

Figure 1: Histogram

could have impact on it. Mateo et al. [3] concluded that the distribution of network links is a power law distribution, where only a few webpages have extremely high endorsement, and most of the pages have a few. The power law distribution follows the principal of preferential attachment.

GitHub also has the trending page to promote the popularity of the repositories, would the distribution of popularity of repositories on GitHub also follows the power law distribution?

2.3 Experiments

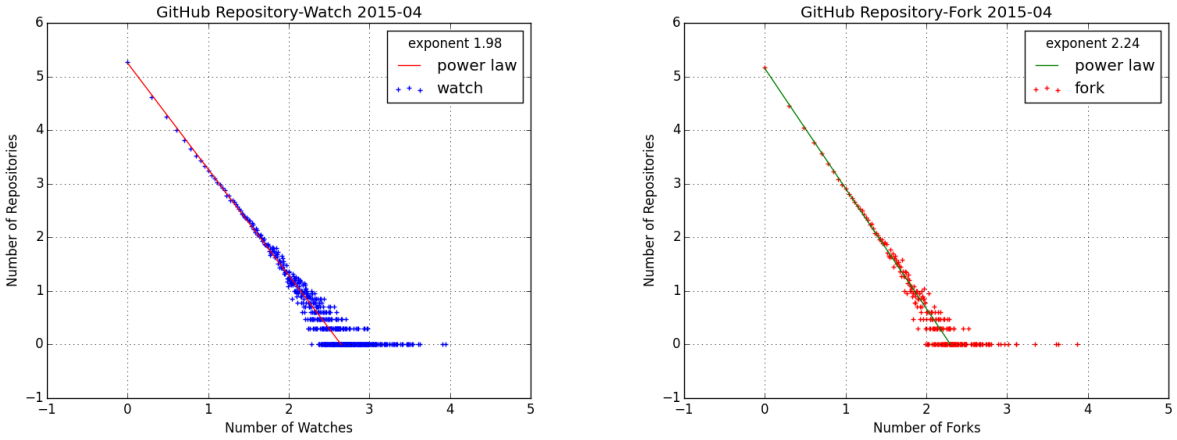
We downloaded data of all events generated from 2015-04-01 to 2015-04-30. The data are in json format, and we use python to process the data. There are 25 different Events recorded by GitHub. A user action will trigger an Event recorded by the system. For example, when a user watches a repository (adds it as favourite), the system will record a *WatchEvent* about the *user* and the *repository*. We analyse the results of *WatchEvent* and *ForkEvent*, because they are the two most typical events representing the popularity of the repositories. Normally when a user browses the trending page, he or she would mark the interesting repository as favourite for further exploration.

We first filter out all the irrelevant Events from the records. Then we count the numbers of WatchEvent or ForkEvent of the repositories from the filtered records respectively. An example result is that there are 1000 being watched by 1 person while there are 4 repositories being watched 108 times. Since the original data is recorded hourly, we will merge the data and get the final result. The raw dataset is approximately 4GB.

2.4 Results

Figure 1 presents the results of histogram of counts received for WatchEvent and ForkEvent respectively. The x-axis and y-axis is log-log plot in this case, which is in the same format in [1, 3] The number of repositories decreases with more number of watches/forks.

Figure 2 presents the distribution of the counts received for two events. We can see that there is an estimated line describing the trend of the data, and since the trend of log-log plot is linear, the trend of the original trend is power law. In the plot b), we can see that the tail is longer, but sparser than the one in a), maybe it is because people are more likely to watch a repository instead of to fork one. The explanation is reasonable because people are more likely to add a repository as a favourite one than to keep a private copy.



a) watch count b) fork count

Figure 2: Distribution

2.5 Summary

From the plot, we can see that the popularity and population of users indeed have the power law distribution. The experiments shows that our conjecture is correct.

3 The Impact of Coexistence of Similar Languages

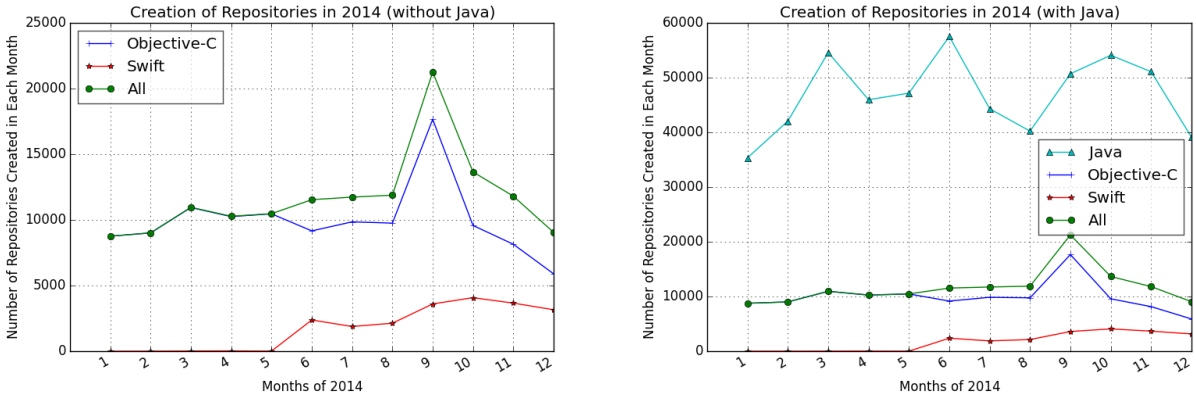
3.1 Background and Motivation

Gramoli [2] presents an example of US supreme court citations. In the lawsuit writings, citations are necessary and serve as ground truth. However, due to the amendment of the law, and new cases based on the new law would come and replace the old ones. The coexistence of old and new cases would impact each other in the certain period of time. There are examples that the number of citations of some cases dropped dramatically before and after the *5th Amendment*. If we treat that a programmer uses a programming languages as the citation, is it possible to find the similar pattern as well? If more programmers endorse a programming language, the language will be actively developed and updated. In return, if a programming languange is being updated with more popular features, it attracts more programmers. We want to see the effect of programming languages that are designed to achieve the similar goal. However, as most of the programming languages are designed for generic usage, it is hard to see if two languages are designed for the same goal. Therefore we use Objective-C and Swift as example.

Objective-C is the primary language for Apple's products and frameworks, ranging from popular iPhone to OSX. In 2014, Apple released a new programming languange called "Swift" for developers. Swift is claimed not be a replacement of the Objective-C, however it is designed for beginners to develop apps on iPhone easily. Many people would regard it as Python like light-weight languange. It is interesting to see how iOS and OSX developers use Objective-C before and after the release of Swift.

3.2 Experiments

In the second task, we also use the GitHub repository data. However due to the fact that we will process hundreds of Gigabyte data, we take advantage of the *Google BigQuery* to accomplish the task. We processed the SQL script for querying count number of each langunages and save them as intermediate results. The only event we use is CreateEvent, because it reflects how users try new stuffs. We analyse the creation of new repositories in each month of 2014. It is noteworthy that Swift had its beta version released in June, and official version in September. The total data being processed is approximately 110GB.



a) without Java

b) with Java

Figure 3: Effect of Swift on Objective-C

3.3 Results

Figure 3 shows the number of repositories created in each month of the year 2014. The red line represents Swift, the blue line represents Objective-C and the green line represents the total amount of the two. In a), we can see that in June, there are some programmers gave a try on Beta version of Swift. In September, there is an obvious spike of Objective-C, and there is also a slight increase of Swift. We think it is because *Apple Worldwide Developers Conference (WWDC)* annual conference was hosted in September. Programmer who use Objective-C would like to try new features, and more people get to know the Swift in that month. The number of new repositories of Swift is very stable, and it does not dramatically since September. After October, we can see that the new repositories reduces, and the number of Objective-C reduces to the lowest point in the year. However, the sum of Swift and Objective-C is not the worst until December.

In b), we add Java to the plot for comparison. We know that Java is a programming language with more generic purposes, there are many different projects using Java. Although Java is the primary language for Android, a competitor mobile OS to iOS, there are many frameworks such as Apache Hadoop are built by Java. We can see there are several ups and downs of Java in a year, and it seems not related to Apple's two languages. It is noteworthy that Google also had its conference Google IO in June 2014, maybe it contributes to the spike of Java in June.

3.4 Summary

We conclude that programming languages having the similar or the same goal will have impact on each other. The existence of the Swift significantly impact the usage of Objective-C. The phenomenon is similar to the example of citation of cases on supreme court as shown by Gramoli [2].

4 Conclusion

Two properties of the large scale repository network are presented: the popularity of the repositories follow the power law distribution; and programming language designed for the similar purpose will impact each other. We use the GitHub datasets, and analyse the properties respectively from the datasets. The results show that our conjecture is correct. In addition, there are some external factors that could impact the structure of the network, such as WWDC os Apple Inc. There are many more interesting properties to be extracted on the repository network, that we can have a better vision of the trending technologies.

Appendix

The *code_and_data* folder contains the code and processed data. The raw data is too large for submission. There are two folders *task1* and *task2* containing the code and data respectively. Due to the changes of the structure, some path in the code must be changed in order to reproduce the results.

References

[1] David Easley and Jon Kleinberg. *Networks , Crowds , and Markets : Reasoning about a Highly Connected World*, volume 81. 2010. ISBN 9780521195331.

[2] Vincent Gramoli. Large-Scale Networks: The Strucutre of the Web, 2015.

[3] San Mateo, San Jose, and Palo Alto. Graph structure in the web. *Systems Research*, 33:1–15, 2006.

[4] Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science (New York, N.Y.)*, 311(5762):854–856, 2006.