15.05.2016

Project 2,
Assignment 3

Group 9

Taha Emre, 150120119
Mustafa Safa Özdayı, 150110128
Abdurrahman Namlı, 150110110

Aim

Aim of the assignment was to implement a script that extract and interpret some information from git repositories. For this purpose, we particularly used policy component of Chromium’s repository[1].

Implementation

1) We start by importing required libraries

```
import git
import numpy as np
from collections import defaultdict
import matplotlib.pyplot as plt
```

git[2] allows as us to efficiently parse the commits.
Numpy is used for arrays.
Defaultdict() is an efficient data structure to store the data.
Matplotlib is used to plot graphs.

2) We parse the commits and get the desired data

```
#1. Connect to repo and get files&developers
repo = git.Repo()
devs, files = defaultdict(int), defaultdict(list)
for commit in repo.iter_commits():
    authorName = commit.author.name
    commitedFiles = commit.stats.files.keys()
    for f in commitedFiles:
        files[f].append(authorName)
    devs[authorName] += 1
```

Devs contains developers and their number of commits.
Files contains commited files and name of their respective commiters.

3) We proceed to draw commit distribution of all developers and commit distribution of developers who have %80 of total commits. Though, names appear a bit fuzzy as they don't fit the plot.

```
#2. Identify top devs and plot commit blocks
def plotDict(d, plotName):
    plt.bar(range(len(d)), d.values(), align='center')
    plt.xticks(range(len(d)), d.keys())
    plt.savefig(plotName)
    plt.close()
#2.a Plot commit distribution
plotDict(devs, 'all-commit')
#2.b Identify top devs&plot top dev distribution
eightyCommit = sum(devs.itervalues())*0.8 # %80 of total commits
sortedDevs, topDevs = sorted(devs.items(), key=lambda x:x[1], reverse = True), defaultdict(int)
for dev in sortedDevs:
    if dev[1] < eightyCommit:
        topDevs[dev[0]] = dev[1]
        eightyCommit -= dev[1]
plotDict(topDevs, 'top-commit')
```

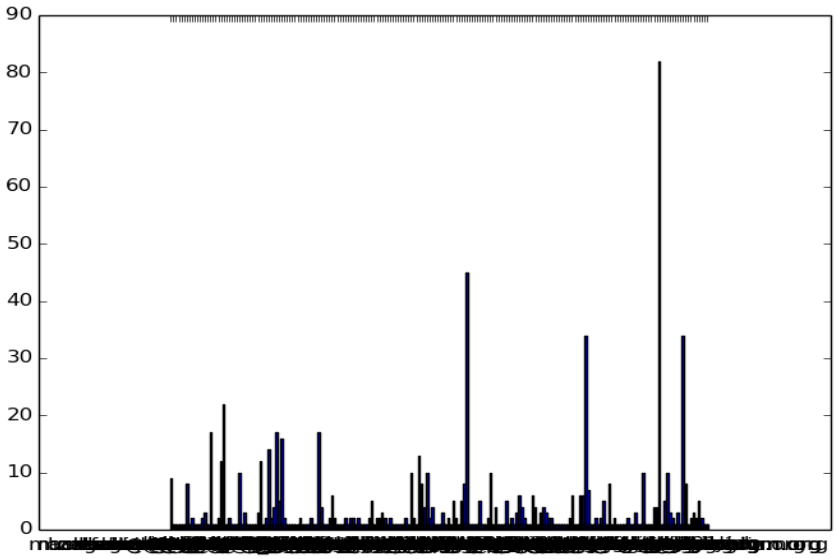


Fig 1) Commit distribution of all developers.

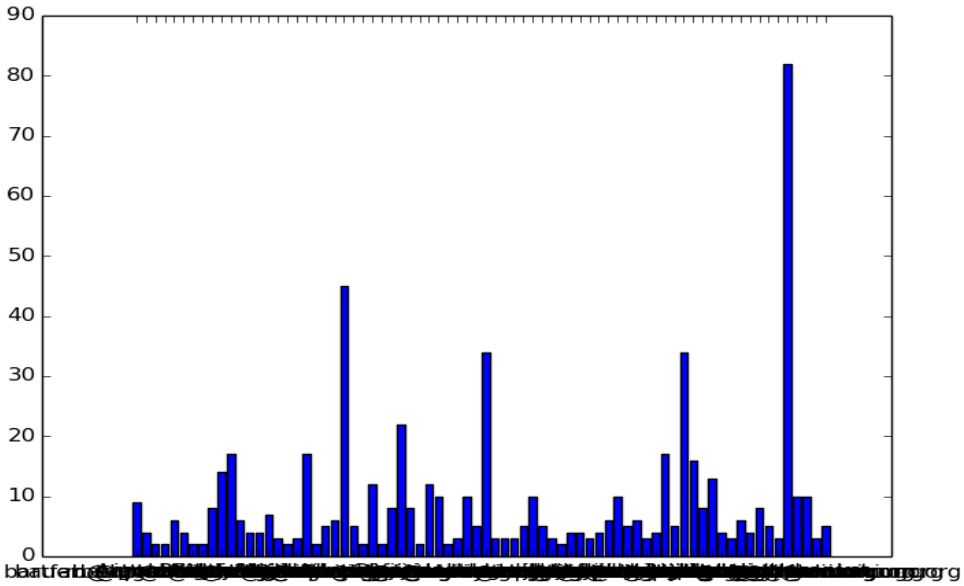


Fig 2) Commit distribution of top developers.

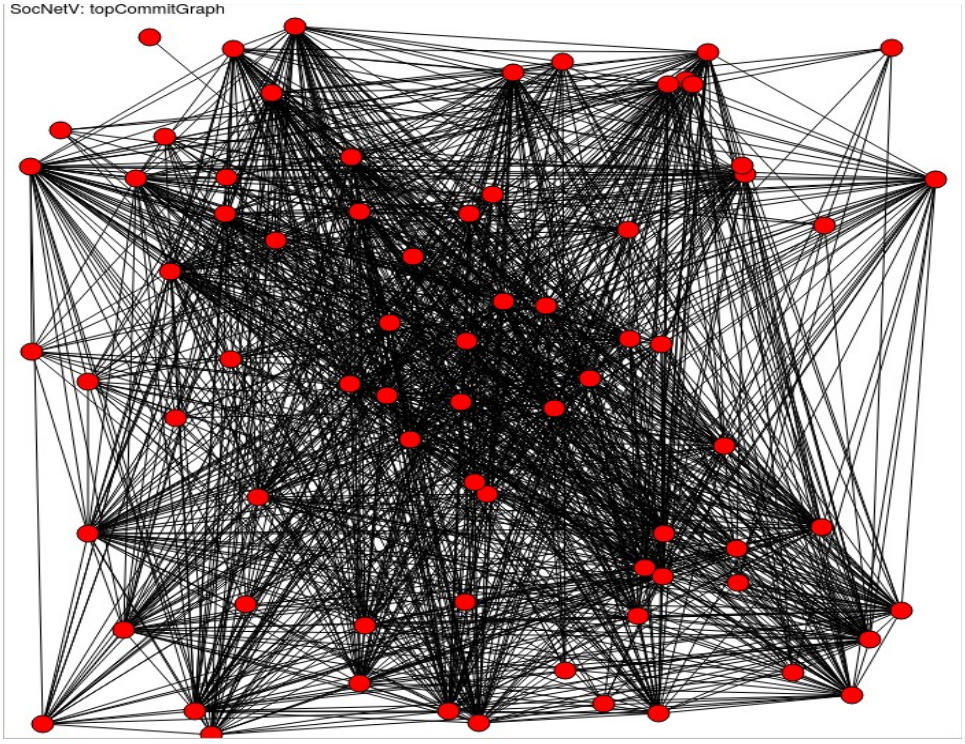
4) We extract the matrix as described in part 3. Rows contain files and columns contain developers. I, j is 1 if developer j committed to file I.

```
#3. Build and save the commit matrix
commitMatrix = np.zeros((len(files), len(devs)), dtype=np.int)
commitM = defaultdict(dict)
i, j = 0, 0
for f in files:
    for dev in devs:
        if dev in files[f]:
            commitMatrix[i][j] = 1
            commitM[f][dev] = 1
        j += 1
    j = 0
    i += 1
np.savetxt('commitMatrix.adj', commitMatrix, fmt='%i', delimiter=' ')
```

5) We extract the matrix required for part 5. It shows relations between top developers.

```
#4.Build and save top devs commit matrix
topCommitM = defaultdict(dict)
for dev in topDevs:
    for f in commitM:
        for dev2 in commitM[f]:
            if dev2 in topDevs and dev in commitM[f] and dev != dev2:
                topCommitM[dev][dev2] = 1
topCommitMatrix = np.zeros((len(topDevs), len(topDevs)), dtype=np.int)
i, j = 0, 0
for dev in topDevs:
    for dev2 in topDevs:
        if topCommitM.get(dev).get(dev2) is not None:
            topCommitMatrix[i][j] = 1
        else:
            topCommitMatrix[i][j] = 0
        j += 1
    j = 0
    i += 1
np.savetxt('topCommitMatrix.adj', topCommitMatrix, fmt='%i', delimiter=' ')
```

The graph that arises from topCommitMatrix is as follows and it is drew by using SocNet[3]:



Results

Files, plots etc. can be found in the GitHub[4].
To run the script simply get it from the repo, put it under a git folder and run it by: python chrome_mine.py

References

[1]https://chromium.googlesource.com/chromium/src/components/policy/
[2]http://gitpython.readthedocs.io/en/stable/
[3]http://socnetv.sourceforge.net/
[4]https://github.com/TinfoilHat0/MineGit.git