Exploration in Data Visualization

Wenhao Wu, Shuyang Ling, Chuan Qin

June 9, 2015

1 Introduction

In this report, we will explore how to use D3, Jave Script and Shiny to deal with data visualization.

2 Shiny

3 Data Visualization on the Facebook Data

3.1 Data Set Description

In this part of our homework, we focus on the visualization of the Facebook data [1] for community detection in ego networks [2]. The publication and the C++ code related to this work can be found at this webpage [3]. The original data consists of 10 ego networks originated from 10 different root users containing 50 800 users, each labeled into a few (potentially overlapping) "circles". Also, the features of each user in the ego network is available. The goal of [2] is, given a single ego network, to infer the circles and the members corresponding to each circle based on the links and the features of each user in the ego network.

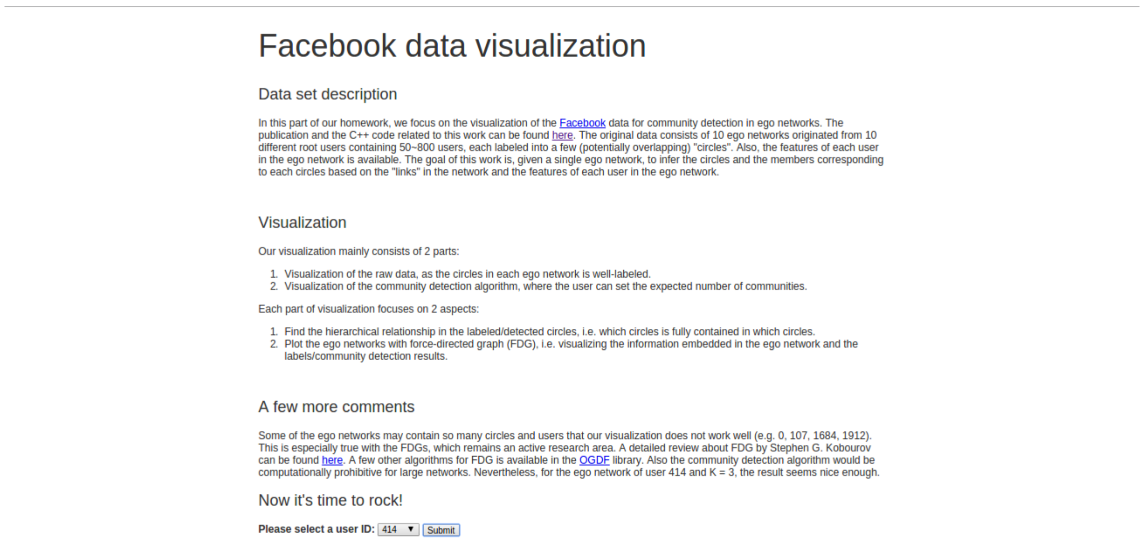
3.2 Framework

To enable a dynamic, interactive visualization of the original dataset and the community detection algorithm in [2] using the source code available at [3], we build a light-weighted Flask application. We process the original data in Python and generate .json files and pass it to the webpages, which contains a piece of javascript code using d3 library to generate .svg figures and some text information.
Our visualization mainly consists of 2 parts:

1. Visualization of the raw data, as the circles in each ego network is well-labeled.
2. Visualization of the community detection algorithm, where the user can set the expected number of communities.

Each part of visualization focuses on 2 aspects:

1. Find the hierarchical relationship in the labeled/deteted circles, i.e. which circles is fully contained in which circles. Since the hierarchical relationship can be represented with a directed acyclic graph (DAG), here we make use of the dagre-d3 library [4] to plot the .svg figures.
2. Plot the ego networks with force-directed graph (FDG), i.e. visualizing the information embedded in the ego network and the labels/community detection results. For this project we use d3's force directed layout functionalities. Though it is easy to implement, we



Facebook data visualization

Data set description

In this part of our homework, we focus on the visualization of the Facebook data for community detection in ego networks. The publication and the C++ code related to this work can be found here. The original data consists of 10 ego networks originated from 10 different root users containing 50~800 users, each labeled into a few (potentially overlapping) "circles". Also, the features of each user in the ego network is available. The goal of this work is, given a single ego network, to infer the circles and the members corresponding to each circles based on the "links" in the network and the features of each user in the ego network.

Visualization

Our visualization mainly consists of 2 parts:

1. Visualization of the raw data, as the circles in each ego network is well-labeled.
2. Visualization of the community detection algorithm, where the user can set the expected number of communities.

Each part of visualization focuses on 2 aspects:

1. Find the hierarchical relationship in the labeled/detected circles, i.e. which circles is fully contained in which circles.
2. Plot the ego networks with force-directed graph (FDG), i.e. visualizing the information embedded in the ego network and the labels/community detection results.

A few more comments

Some of the ego networks may contain so many circles and users that our visualization does not work well (e.g. 0, 107, 1684, 1912). This is especially true with the FDGs, which remains an active research area. A detailed review about FDG by Stephen G. Kobourov can be found here. A few other algorithms for FDG is available in the OGDF library. Also the community detection algorithm would be computationally prohibitive for large networks. Nevertheless, for the ego network of user 414 and K = 3, the result seems nice enough.

Now it's time to rock!

Please select a user ID: 414 Submit

Figure 1: The index page.

noticed that it is extremely computationally intensive for relatively large ego networks. We refer to [5] and [6] for a review and some implementations of more advanced FDG layout algorithms. The integration of more advanced FDG algorithm is left to our future works.

3.3 Examples

Our visualization on the Facebook data starts with an index page where a brief introduction (and a few disclaimers) are presented. Then the user gets to choose which ego network to visualize, as shown in Fig. 1 (Here we pick the ego network corresponding to user 414).
When submitted, a new tab is opened presenting the visualization of the original data (Fig. 2). This page mainly consists of 4 parts:

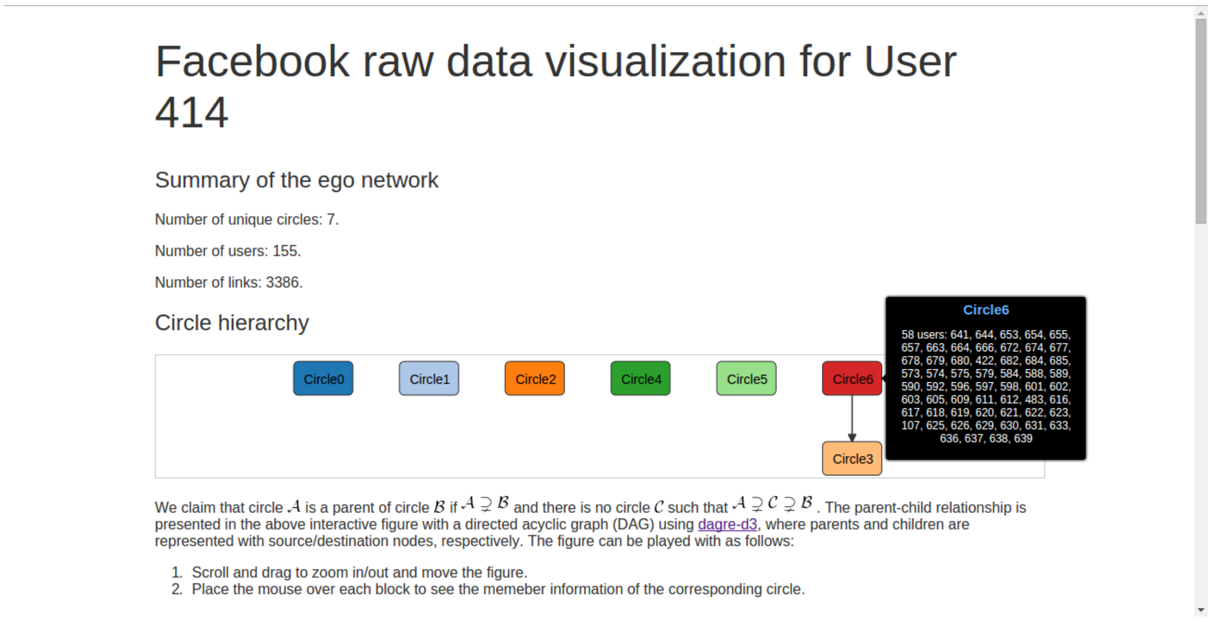
1. A brief summary on the orignal data: this ego network has 7 unique circles, 155 users and 3386 edges.

2. The hierarchy graph of the circles. This figure is interactive in that when placing the mouse over each block, the number/indices of members in the corresponding circles will be shown. Also the figure can be zoomed and moved by scrolling/dragging.

3. The FDG of the ego network. Users in different circles are colored differently (in accordance with the hierarchy graph), and the users in different numbers of circles (due to overlapping) are represented with different types of node as described in the legend. This figure is interactive in that

   - The nodes can be dragged to move under the influence of the forces.
   - When placing the mouse over any node, the corresponding user ID and the circle IDs which contains this user will be shown.
   - For user contained in multiple circles, its color can be toggled among the colors assigned to each of these circles by double clicking the corresponding node.

4. Finally, the user gets to run the community detection algorithm by specifying K, the number of expected circles (Here we pick $K = 3$).



Facebook raw data visualization for User 414

Summary of the ego network

Number of unique circles: 7.

Number of users: 155.

Number of links: 3386.

Circle hierarchy

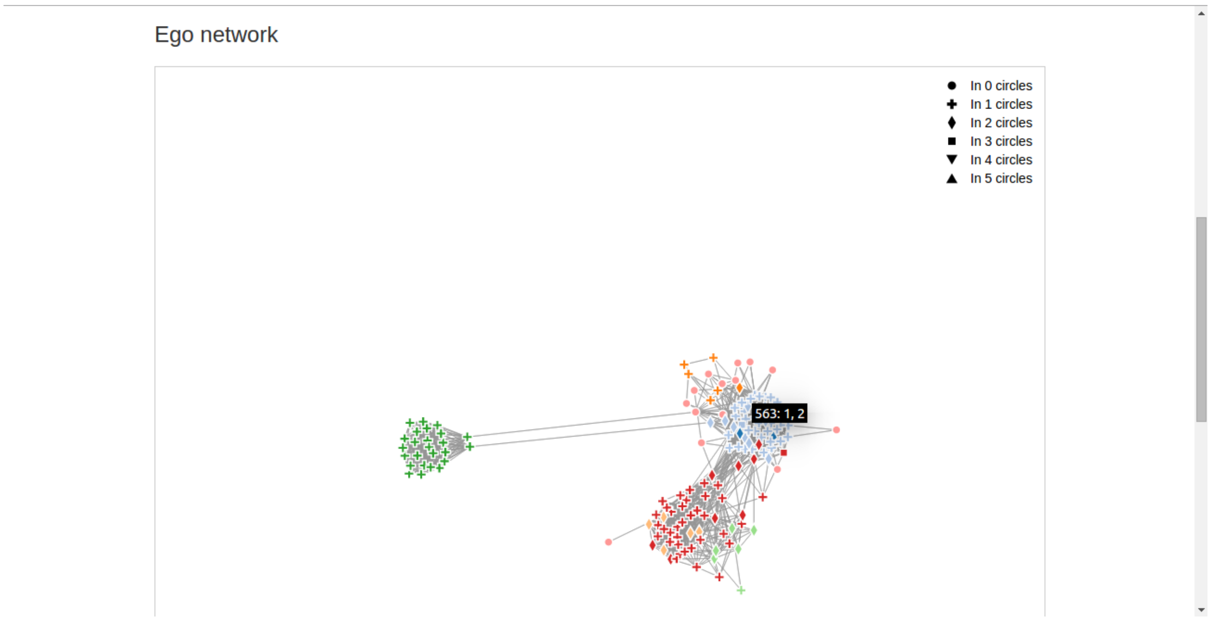
Circle6

58 users: 641, 644, 653, 654, 655, 657, 663, 664, 666, 672, 674, 677, 678, 679, 680, 422, 682, 684, 685, 573, 574, 575, 579, 584, 588, 589, 590, 592, 596, 597, 598, 601, 602, 603, 605, 609, 611, 612, 483, 616, 617, 618, 619, 620, 621, 622, 623, 107, 625, 626, 629, 630, 631, 633, 636, 637, 638, 639

We claim that circle $\mathcal{A}$ is a parent of circle $\mathcal{B}$ if $\mathcal{A} \supsetneq \mathcal{B}$ and there is no circle $\mathcal{C}$ such that $\mathcal{A} \supsetneq \mathcal{C} \supsetneq \mathcal{B}$. The parent-child relationship is presented in the above interactive figure with a directed acyclic graph (DAG) using dagre-d3, where parents and children are represented with source/destination nodes, respectively. The figure can be played with as follows:

1. Scroll and drag to zoom in/out and move the figure.
2. Place the mouse over each block to see the memeber information of the corresponding circle.

(a) Summary and circle hierarchy.



Ego network

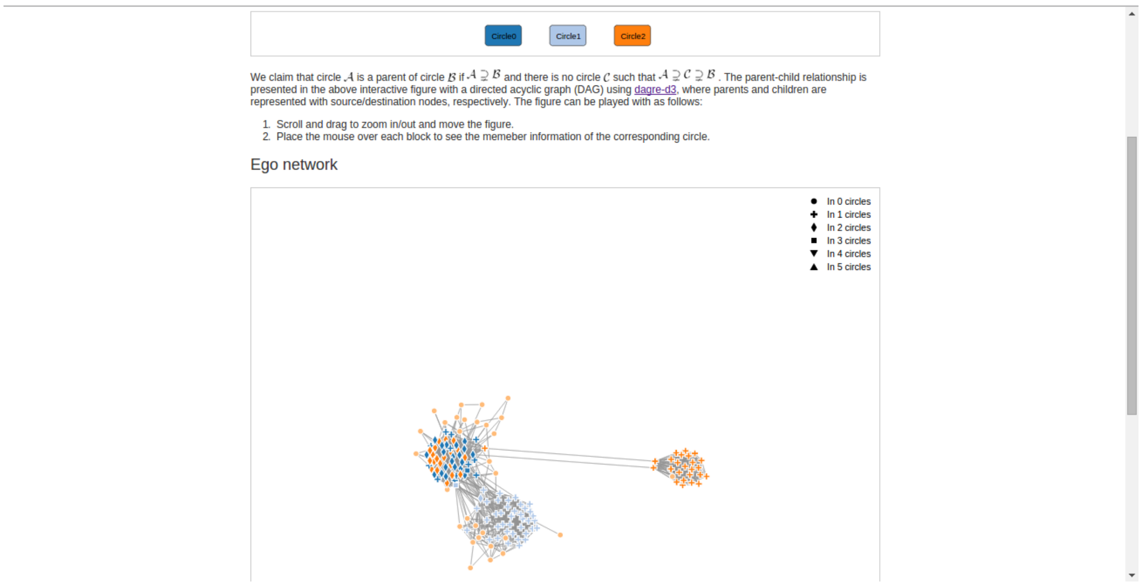
(b) Force Directed Graph (FDG) of the ego network.

Figure 2: Visualization of the original labeled data.

The action in response to the client's submitting a K is to looking for the corresponding community detection results saved in a .out file. If it does not exist, the server will run the community detection algorithm on the fly. Then the detection results are processed into json files and a third page is rendered using the same template as the second page (Fig. 3). Unfortunately, we can see that the detection results results are quite different from the labeled data. In fact, our test show that the algorithm is highly sensitive to its parameters.

4 Shiny

In this section, we will build up an interactive web application to visualize networks by using Shiny. The idea is inspired by the Data Visualization Workshop organized by Duncan Temple



We claim that circle $\mathcal{A}$ is a parent of circle $\mathcal{B}$ if $\mathcal{A} \supsetneq \mathcal{B}$ and there is no circle $\mathcal{C}$ such that $\mathcal{A} \supsetneq \mathcal{C} \supsetneq \mathcal{B}$. The parent-child relationship is presented in the above interactive figure with a directed acyclic graph (DAG) using dagre-d3, where parents and children are represented with source/destination nodes, respectively. The figure can be played with as follows:

1. Scroll and drag to zoom in/out and move the figure.
2. Place the mouse over each block to see the memeber information of the corresponding circle.

Ego network

Figure 3: Visualization of the community detection results.

Lang and
The objective is to present the function network inside a package.

References

[1] Social circles: Facebook. `https://snap.stanford.edu/data/egonets-Facebook.html`. Accessed: 2015-06-09.

[2] Jure Leskovec and Julian J Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

[3] Julian McAuley. `http://cseweb.ucsd.edu/~jmcauley/`. Accessed: 2015-06-09.

[4] Home cpettitt/dagre-d3 Wiki GitHub. `https://github.com/cpettitt/dagre-d3/wiki`. Accessed: 2015-06-09.

[5] Stephen G Kobourov. Force-directed drawing algorithms. 2004.

[6] OGDF - Open Graph Drawing Framework. `http://www.ogdf.net/doku.php`. Accessed: 2015-06-09.