Assignment 6

CS 532: Introduction to Web Science

Spring 2017

Grant Atkins

Finished on March 23, 2017

1

Question

1. D3 graphing (10 points)

Use D3 to visualize your Twitter followers. Use my twitter account ("@phonedude_mln") if you do not have >= 50 followers. For example, @hvdsomp follows me, as does @mart1nkle1n. They also follow each other, so they would both have links to me and links to each other.

To see if two users follow each other, see:
https://dev.twitter.com/rest/reference/get/friendships/show

Attractiveness of the graph counts! Nodes should be labeled (avatar images are even better), and edge types (follows, following) should be marked.

Note: for getting GitHub to serve HTML (and other media types), see:
http://stackoverflow.com/questions/6551446/
can-i-run-html-files-directly-from-github-instead-of-just-viewing-their-source

Be sure to include the URI(s) for your D3 graph in your report.

Answer

First approaching this problem I decided to use the tweepy library for python 3.6 to retrieve a json formatted list of Dr. Nelson's followers, which for me at the time was 634 followers, which was then saved to a file named **phonedudeFollowers.json** using the python script **getFollowers.py** shown in Listing **??**. Then I decided to reduce the number of friendships to check since checking friendships between all 634 followers would have been fairly time consuming. Therefore I created another script called **chooseUsers.py** which would pick 100 random integers for numbers between 1 and 634, which would act as which line numbers I would choose to sample from the json followers I retrieved earlier. This method actually worked out very well because I ended up getting quite a few members that were actually at Old Dominion University, as shown in Table **??**, without me tampering with it at all - however I may have run it a few times to test it.

I then took the 100 followers I had just chosen, added Dr. Nelson's user name *phonedude_mln*, and continued to use tweepy again to check friendships between this list of users using the python script **findFriendships.py** shown in Listing **??**. It should be noted that I first decided to minimize the number of api calls by reducing the pairs I had to check. This was done by the *minimizeConnections* function, which mapped all users as keys with values of every other user that was not already a key. This reduced the number of api calls from 10100 to 5150, with more logic explained in that function starting on line 14. This script took a several hours due to Twitter's rate limits. When it was completed all the friendships to each other user was saved to **friendships.csv** in the format of:

Source User	**Target User**	**Following Target**	**Followed By Target**

Table 1: Format of friendships.csv, where Following fields are booleans

Finally to utilize D3.js to its utmost, I decided to convert the csv data and json data saved earlier, to create a new formatted json data with two main keys: nodes and links. To accomplish this I created a script named **createGraphJson.py**, so the data could properly represent a force directed graph. The all followers graph shown in Figure **??** and the friendship graph of the 100 users generated shown in Figure **??** were generated using D3 using the *drawGraph* function defined in **twitterFriendship.js** shown in Listing **??**. On hover a side bar should appear showing the user's name, user id, and profile image which links to the associated account. It should be noted that 1 user on the friendship graph is disconnected because when

tweepy checked the account it was actually suspended by the time it got to it, making friendships impossible to establish. These two visualizations can be found here:

`https://cdn.rawgit.com/grantat/cs532-s17/b410d0d7/assignments/A6/src/index.html`

Chosen Followers
Daniel
Ren Voorburg
Brenda Berkelaar
Cassie Findlay
Hussam Hallak
British Library Labs
Space 3D 360 VR
Kh Talha
Angela Woodall
Fredericik Begbeder
Kevin Levrone
Christopher Chyr
Jessica Smith
Zhaohuan Wang
Michael Stevenson
Stephanie A Kingsley
Andrea Goethals
TEI Landmark
DBpedia
Hung V. Do
Gardner Campbell
Jacquelyn Clements
dinesh kumar paladhi
Alex Dolan-Mescal
psleeman
lyn maccorkle
Larry Wayne Wilson
WOSP Workshop
NetLab
Huan Huang
Karl Blumenthal
Archivportal-D

Kees Teszelszky
Mike Amundsen
OTL National Event
Shira Peltzman
Daniel Rehn
Matthew Weber
Kayla Fox
Alex Wade
IJDL
JCDL 2016
Edward McCain
Naima
Kenneth E. Chapman
rudolf
Lam Mai ?
Lana Kinnie
Shamim
Bertram Lyons
J. Stoddert
Peter Murray
Jennifer Serventi
MD SHAJID
Paul Wheatley
?????? ?????
Jessica Ogden
gaurav
Stanford Libraries
Karen Hanson
DOWN at ODU
Jonathan Greenberg
Melissa Durham
Benn Joseph
Boaty McBoatface
Bergis Jules
CRISTIANO PLAMEN
B. Danette Allen
Ray Matthews

Peter Burnhill
Klaus Berberich
Nancy Alajarmeh
anaas
Mohamed Aturban
WS-DL Group, ODU CS
Gerhard Gossen
Robin C. Pike
ISRJ
Peter Webster
Yasmina Anwar
Robert H. McDonald
tpdl2017
Juliane Stiller
Konrad Lischka
Baptiste Fluzin
Jak Akdemir
Aleph Archives
Mike Zarro
Ben LeBrun
Gina Jones
Mat Kelly
Frank McCown
Andy Jackson
Joan Smith
Scott Ainsworth
Moustafa Emara
Kaitlin L. Costello
Hany SalahEldeen
William J Nixon
Michele Weigle

Table 2: Users generated from **chooseUsers.py**

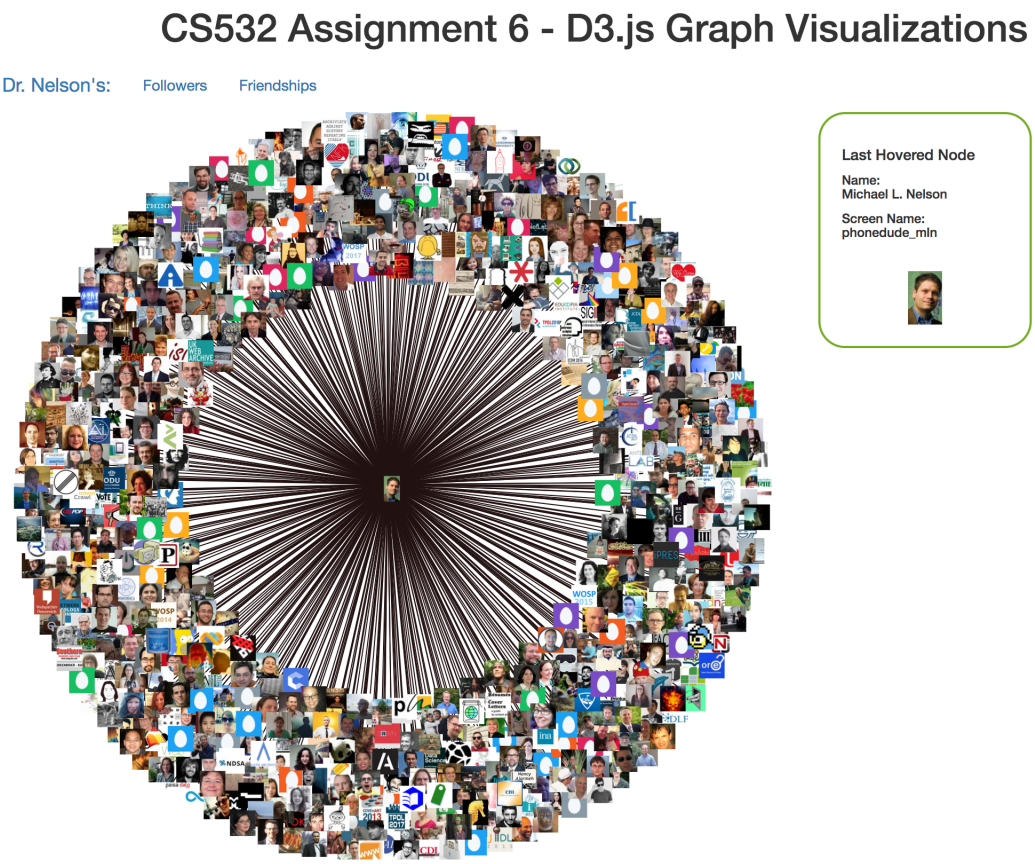

Figure 1: All of Dr. Nelson's followers in D3 force directed graph

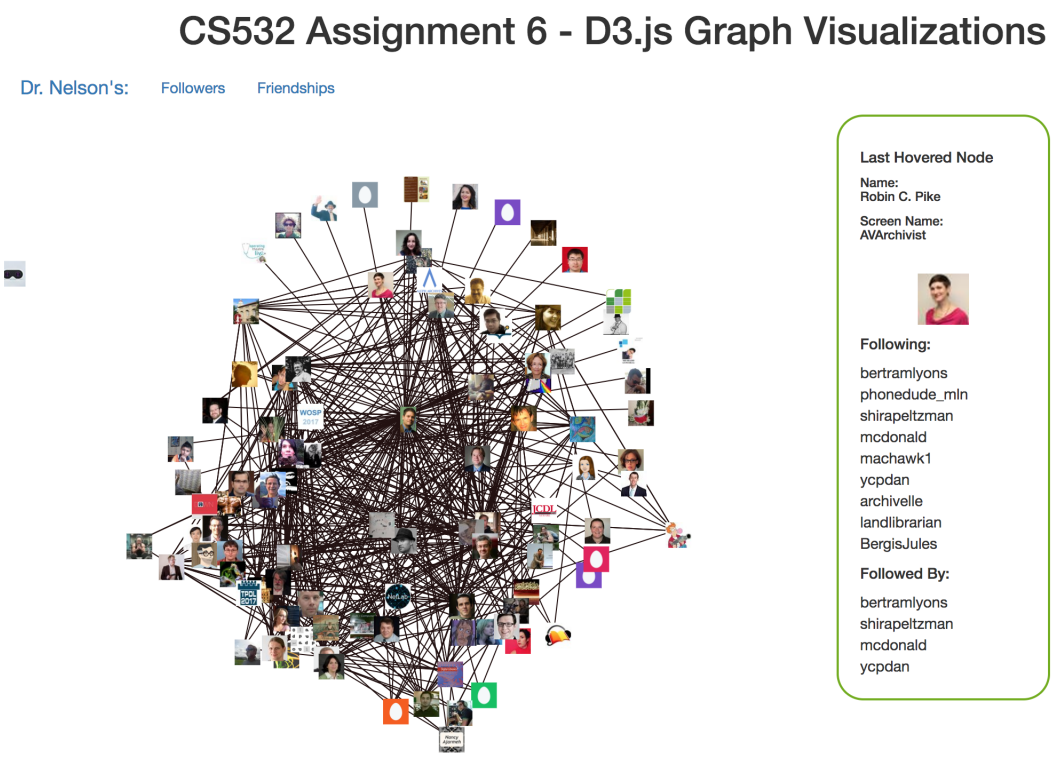

Figure 2: Friendships graph of 100 selected users

```
1  import tweepy
2  import json
3
4  # Variables that contains the user credentials to access Twitter
       API
5  access_token = "821042028800802816-
       E7SvwPXZKJRzazLctidudXhD0X0SgDZ"
6  access_token_secret = "
       hfEMDTkVBX6Kf7x8FddjBZi7joxKZIYYJztq1QFQcF8cp"
7  consumer_key = "RigRve4McsZdYXNpz2rwPRZfx"
8  consumer_secret = "
       EuFivjFeWCBmG205shXMjTPb0u56wTXJgRDRhqaWPRQU1CxYjW"
9
10
11 def getFollowers(api):
```

```
12      # limit by 200, used pages instead of items since its less
            likely to get
13      # timed out.
14      for p in tweepy.Cursor(api.followers,
15                             screen_name="phonedude_mln", count
                                   =200).pages():
16          # used extend since json would break for each page
17          # print(p)
18          for user_obj in p:
19              print(json.dumps(user_obj._json))
20
21
22  if __name__ == "__main__":
23      auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
24      auth.set_access_token(access_token, access_token_secret)
25      api = tweepy.API(auth, wait_on_rate_limit=True,
26                       wait_on_rate_limit_notify=True)
27      try:
28          getFollowers(api)
29      except KeyboardInterrupt:
30          print()
```

Listing 1: Python script to retrieve Dr. Nelson's followers

```
1   import tweepy
2   import json
3   import csv
4   from pprint import pprint as pp
5
6   # Variables that contains the user credentials to access Twitter
        API
7   access_token = "821042028800802816-
        E7SvwPXZKJRzazLctidudXhD0X0SgDZ"
8   access_token_secret = "
        hfEMDTkVBX6Kf7x8FddjBZi7joxKZIYYJztq1QFQcF8cp"
9   consumer_key = "RigRve4McsZdYXNpz2rwPRZfx"
10  consumer_secret = "
        EuFivjFeWCBmG205shXMjTPb0u56wTXJgRDRhqaWPRQU1CxYjW"
11
12
13  def minimizeConnections():
14      '''
15      Meant to minimize the number of connection calls required to
            determine
16      friendship. Creates a dictionary of 100 users mapped to
            other users
17      who aren't already keys.
18      For each Key the formula for number of connections to make:
19      for 1 in total:
```

```
20          pairs_to_check = total - iteration_count
21          total += pairs_to_check
22
23      New total = 5151, vs 10100 to check
24      '''
25      data = []
26      with open('data/chosenFollowers.csv','r') as f:
27          reader = csv.reader(f, delimiter=',')
28          for row in reader:
29              data.append(row[1])
30
31      # add phonedude_mln to this list as well. Makes 101 users
32      data = ["phonedude_mln"]+data
33
34      fDict = {}
35      for i in data:
36
37          temp = []
38          notInDict = True
39          for j in data:
40              if i != j:
41                  if j in fDict:
42                      # skip user if already in dictionary. Since
                            that user
43                      # is already a key inside the dictionary, it
                             means they
44                      # already check every other necessary user
45                      continue
46                  else:
47                      temp.append(j)
48                      fDict[i] = temp
49
50      return fDict
51
52
53  def findFriendships(api,data):
54      # Will take considerable time
55      for key,val in data.items():
56          for i in val:
57              tempDict = {}
58              try:
59                  print(key,"vs",i)
60                  sf = api.show_friendship(source_screen_name=key,
                        target_screen_name=i)
61                  tempDict["isFollowing"] = sf[0].following
62                  tempDict["isFollowedBy"] = sf[0].followed_by
63                  tempDict["source"] = key
64                  tempDict["target"] = i
65                  writeCSV(tempDict,"data/friendships.csv")
```

```
66              except KeyboardInterrupt:
67                  print()
68                  exit()
69              except:
70                  print("Failed to connect friendship for: ",key,"
                        and",i)
71                  tempDict["isFollowing"] = ""
72                  tempDict["isFollowedBy"] = ""
73                  tempDict["source"] = key
74                  tempDict["target"] = i
75                  writeCSV(tempDict,"data/friendships.csv")
76
77
78  def writeCSV(data, filename):
79      with open(filename, 'a', newline='') as file:
80          writer = csv.writer(file, delimiter=',')
81          row = [data["source"], data["target"], data["isFollowing
                "],data["isFollowedBy"]]
82          writer.writerow(row)
83
84
85  if __name__ == "__main__":
86      auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
87      auth.set_access_token(access_token, access_token_secret)
88      api = tweepy.API(auth, wait_on_rate_limit=True,
89                       wait_on_rate_limit_notify=True)
90      try:
91          # findFriendships(api)
92          fDict = minimizeConnections()
93          findFriendships(api,fDict)
94          pp(fDict)
95          # print(fDict)
96      except KeyboardInterrupt:
97          print()
```

Listing 2: Python script to check twitter friendships

```
1
2  function drawGraph(filename,checkFriendships){
3
4          var width = $(".svg-section").width(),
5              height = 700;
6
7          var svg = d3.select(".svg-section").append("svg")
8              .attr("width", width)
9              .attr("height", height);
10
11
12         var force = d3.layout.force()
```

```
13              .gravity(.05)
14              .distance(100)
15              .charge(-100)
16              .size([width, height]);
17
18          d3.json(filename, function(error, json) {
19                  var edges = [];
20              json.links.forEach(function(e) {
21                      var sourceNode = json.nodes.filter(function(
                            n) { return n.id === e.source; })[0],
22                      targetNode = json.nodes.filter(function(n) {
                            return n.id === e.target; })[0];
23
24                      edges.push({source: sourceNode, target:
                            targetNode, value: e.Value});
25              });
26
27            force
28                .nodes(json.nodes)
29                .links(edges)
30                .linkDistance(200)
31                .start();
32
33            var link = svg.selectAll(".link")
34                .data(edges)
35                    .enter().append("line")
36                .attr("class", "link");
37
38            var node = svg.selectAll(".node")
39                .data(json.nodes)
40              .enter().append("g")
41                .attr("class", "node")
42                .on("mouseover", function(d){
43                            d3.select(this).select("circle").
                                transition()
44                                .duration(750)
45                                .attr("r", 16);
46
47                            var content = '<h5>Last Hovered Node</
                                h5>'
48                             content += '<h6>Name:</br> ' + d.name
                                    + '</span></h6>';
49                             content += '<h6>Screen Name:</br> ' +
                                    d.id + '</span></h6></br>';
50                             content += '<div class="row"><div
                                    class="col-sm-12"><a target="
                                    _blank" href="https://twitter.com
                                    /'+d.id+'"><img src=' + d.image +
51                              ' alt="" class="center-element" style
```

```
                                    ="max-width:100%;max-height:50px
                                    ;"></a></div>';
52                                  if(checkFriendships){
53                                      content += '<div class="col-sm
                                            -12"><h5>Following:</h5>';
54
55                                  var followedBy = [];
56                                  for(var i in edges){
57                                var tempID = edges[i]["source"]["id"];
58                                var tempTargetID = edges[i]["target"]["
                                      id"];
59                                if(tempID == d.id){
60                                      content+= edges[i]["target"]["id
                                            "]+'</br>';
61                                }
62                                if(tempTargetID == d.id){
63                                      followedBy.push(edges[i]["source
                                            "]["id"])
64                                }
65                              }
66
67                                  content+= '</div><div class="col-sm
                                        -12"><h5>Followed By:</h5>';
68                                  for(var i in followedBy){
69                                      content+=followedBy[i]+'</br>';
70                                  }
71
72                                  content += '</div></div>'
73                                }else{
74                                      content += '</div>';
75                                }
76                                  $("#user-popup").html(content);
77                                  $("#user-popup").show();
78                  })
79                      .on("mouseout", mouseout)
80                  .call(force.drag);
81
82              node.append("image")
83          .attr("xlink:href", function(d){return d.image;})
84          .attr("x", -8)
85          .attr("y", -8)
86          .attr("width", 24)
87          .attr("height", 24);
88
89              // node.append("svg:p")
90              //      .append("text")
91              //      .attr("dx", 12)
92              //      .attr("dy", ".35em")
93              //      .text(function(d) { return d.name})
```

```
94
95
96              force.on("tick", function() {
97                link.attr("x1", function(d) { return d.source.x; })
98                    .attr("y1", function(d) { return d.source.y; })
99                    .attr("x2", function(d) { return d.target.x; })
100                   .attr("y2", function(d) { return d.target.y; });
101
102
103             node.attr("transform", function(d) { return "
                    translate(" + d.x + "," + d.y + ")"; });
104           });
105
106
107         function mouseout() {
108           d3.select(this).select("circle").transition()
109               .duration(750)
110               .attr("r", 8);
111         }
112
113         });
114 }
```

Listing 3: Javascript to create force directed graphs

2

Question

Extra credit: (5 points)

2. Gender homophily in your Twitter graph

Take the Twitter graph you generated in question #1 and test for male-female homophily. For the purposes of this question you can consider the graph as undirected (i.e., no distinction between "follows" and "following"). Use the twitter name (not "screen name"; for example "Michael L. Nelson" and not "@phonedude_mln") and programatically determine if the user is male or female. Some sites that might be useful:

https://genderize.io/
https://pypi.python.org/pypi/gender-detector/0.0.4

Create a table of Twitter users and their likely gender. List any accounts that can't be determined and remove them from the graph.

Perform the homophily test as described in slides 11-16, Week 8.

Does your Twitter graph exhibit gender homophily?

Answer

NOT ATTEMPTED

3

Question

```
Extra credit: (3 points)

3.  Using D3, create a graph of the Karate club before and after
the split.

- Weight the edges with the data from:
http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/zachary.dat

- Have the transition from before/after the split occur on a mouse
click.  This is a toggle, so the graph will go back and forth beween
connected and disconnected.
```

Answer

NOT ATTEMPTED

References

[1] Csardi, Gabor. "Package 'graphdata' " iGraphData. Cran-R-Project, 13 July 2015. Web. 16 March 2017.https://cran.r-project.org/web/packages/igraphdata/igraphdata.pdf.

[2] Csardi, Gabor, "Package 'igraph' " iGraph. Cran-R-Project, 13 July 2015. Web. 16 March 2017. http://igraph.org/r/doc/igraph.pdf.

[3] Rodrigues, David."Finding Communities in networks with R and igraph" N.p., n.d. Web. 16 March 2017. http://www.sixhat.net/finding-communities-in-networks-with-r-and-igraph.html.

[4] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33, 452-473 (1977).