

Web Science cs532-s16

ASSIGNMENT 6 REPORT

DR. MICHAEL L. NELSON

BY: HUAN HUANG

03/17/2016

Problem 1

Use D3 to visualize your Twitter followers. Use my twitter account if you do not have more than 50 followers. For example, @hvdsonp 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-ins`

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

Answer

To solve this problem, I first dug up my old program from assignment 4 and made some modification to get the screen names of the followers. Also, to make this problem easier, I used one of Mr. Nelson's follower "jcdl2012" as the source user. He has 70 followers which is much better than Mr. Nelson's 492 followers.

```
1 #!/usr/bin/python
2 import time
3 import tweepy
4 from tweepy import OAuthHandler
5
6
7 savefile = open('mainfollowers.txt', 'a')
8
9 # load my API credentials
10
11 accesstoken = "110800099-ZiXf6imxd24OVqw5nnm4nkHXmVwhX46QQI0Wb3Zt"
12 accesstokensecret = "F1DFz2OI035FONaiL9uTzfqRauVArVwwVMuxiHDWQsmwo"
13 consumerkey = "1N9TkDHB9g4UIbjVj5j6idqnn"
14 consumersecret = "5Yjw1hRViWd5TloKLmEibPoregSltxO1bY7YJckjnecuSw4n9B"
15
16 # create twitter API object
17
18 auth = tweepy.OAuthHandler(consumerkey, consumersecret)
19 auth.set_access_token(accesstoken, accesstokensecret)
```

```

20
21
22 api = tweepy.API(auth, wait_on_rate_limit=True)
23 mainuser = api.get_user('jcdl2012')
24 userName = mainuser.screen_name
25 #followersCounts = mainuser.followers_count
26 print userName
27 savefile.write(userName)
28 savefile.write('\n')
29
30 users = tweepy.Cursor(api.followers, userName, count=200).items()
31
32 for follower in users:
33     try:
34         #fcount = follower.followers_count
35         fname = follower.screen_name
36         print fname
37         savefile.write(fname)
38         savefile.write('\n')
39     except tweepy.error.RateLimitError:
40         monitor_rate_limit=True, wait_on_rate_limit=True
41         continue
42     except StopIteration:
43         break

```

```

63 plbogen
64 hcayless
65 webternity
66 Galsondor
67 martlnkleln
68 kboughida
69 ibnesayeed
70 yasmina_anwar
71 phonedude_mln

```

Figure 1: Sample of the data file mainfollowers.txt

Next, using these screen names, I need to get the relation between each one of them.

```

1 import time
2 import re
3 import tweepy
4 from tweepy import OAuthHandler
5
6
7 readfile = open('mainfollowers.txt', 'r').read().split('\n')
8 savefile = open('relations.txt', 'a')
9 savefile2 = open('unauthorized.txt', 'a')
10
11 # load my API credentials
12
13 accesstoken = "110800099-ZiXf6imxd24OVqw5nnm4nkHXmVwhX46QQI0Wb3Zt"
14 accesstokensecret = "FIDFz2OI035FONaiL9uTzfqRauVArVwwVMuxiHDWQsmwo"
15 consumerkey = "1N9TkDHB9g4UIbjVj5j6idqnn"

```

```
16 consumersecret = "5Yjw1hRvWd5TloKLMeibPoregSltxO1bY7YJckjnecuSw4n9B"
17
18 # create twitter API object
19
20 auth = tweepy.OAuthHandler(consumerkey, consumersecret)
21 auth.set_access_token(accesstoken, accesstokensecret)
22 api = tweepy.API(auth, wait_on_rate_limit=True)
23
24 i = 0
25 j = 0
26
27 #relation types:
28 #0: A and B are not following eachother
29 #1: A is following B
30 #2: B is following A
31 #3: A and B are following eachother
32 while i < 70:
33     try:
34         relation = api.show_friendship(source_screen_name = readfile[i],
35         target_screen_name = readfile[j+1])
36         if relation[0].following is False and relation[1].following is False:
37             print relation[0].screen_name, relation[1].screen_name, 0
38             savefile.write(str(relation[0].screen_name))
39             savefile.write(',')
40             savefile.write(str(relation[1].screen_name))
41             savefile.write(',')
42             savefile.write(str(0))
43         elif relation[0].following is True and relation[1].following is False:
44             print relation[0].screen_name, relation[1].screen_name, 1
45             savefile.write(str(relation[0].screen_name))
46             savefile.write(',')
47             savefile.write(str(relation[1].screen_name))
48             savefile.write(',')
49             savefile.write(str(1))
50         elif relation[0].following is False and relation[1].following is True:
51             print relation[0].screen_name, relation[1].screen_name, 2
52             savefile.write(str(relation[0].screen_name))
53             savefile.write(',')
54             savefile.write(str(relation[1].screen_name))
55             savefile.write(',')
56             savefile.write(str(2))
57         elif relation[0].following is True and relation[1].following is True:
58             print relation[0].screen_name, relation[1].screen_name, 3
59             savefile.write(str(relation[0].screen_name))
60             savefile.write(',')
61             savefile.write(str(relation[1].screen_name))
62             savefile.write(',')
63             savefile.write(str(3))
64         savefile.write('\n')
65         j = j+1
66         if j == 70:
67             i = i+1
68             j = i
69     except BaseException as e:
```

```

69     if BaseException is "Failed to send request: ('Connection aborted.', error
    (SysCallError(104, 'ECONNRESET')))":
70         time.sleep(5)
71         continue
72     elif BaseException is "Failed to send request: ('Connection aborted.',
    error(104, 'Connection reset by peer')))":
73         time.sleep(5)
74         continue
75     elif BaseException is "Failed to send request: HTTPSConnectionPool(host='
    api.twitter.com', port=443): Read timed out. (read timeout=60)":
76         time.sleep(5)
77         continue
78     else:
79         print readfile[i], ' ', str(e)
80         savefile2.write(str(readfile[i]))
81         savefile2.write(' ')
82         savefile2.write(str(readfile[j]))
83         savefile2.write(' ')
84         savefile2.write(str(e))
85         savefile2.write('\n')
86         j = j+1
87         if j == 70:
88             i = i+1
89             j = i
90         continue

```

Relationship types:

0 means A and B are not following eachother

1 means A is following B

2 means B is following A

3 means A and B are following eachother

```

2464 martinkle1n,ibnesayeed,2
2465 martinkle1n,yasmina_anwar,2
2466 martinkle1n,phonedude_mln,3
2467 kboughida,ibnesayeed,0
2468 kboughida,yasmina_anwar,2
2469 kboughida,phonedude_mln,1
2470 ibnesayeed,yasmina_anwar,3
2471 ibnesayeed,phonedude_mln,3
2472 yasmina_anwar,phonedude_mln,3

```

Figure 2: Sample of the data file relations.txt

Now, I need convert the data into json file for it to be used in the graph.

```

67     {"name": "webternity", "group": 1},
68     {"name": "Galsondor", "group": 1},
69     {"name": "martinklein", "group": 1},
70     {"name": "kboughida", "group": 1},
71     {"name": "ibnesayeed", "group": 1},
72     {"name": "yasmina_anwar", "group": 1},
73     {"name": "phonedude_mln", "group": 1}
74 ],
75 "links": [
76     {"source": "jcdl2012", "target": "SF_BayArea_ACM", "value": 2},
77     {"source": "jcdl2012", "target": "skyworldgo", "value": 2},
78     {"source": "jcdl2012", "target": "jcdl2015", "value": 2},

```

Figure 3: Sample of the data file sourceData.txt

Lastly, write the html file to plot the graph with D3.

```

1 <!DOCTYPE html>
2 <meta charset="utf-8">
3 <style>
4
5 .node {
6     stroke: #fff;
7     stroke-width: 1.5px;
8 }
9
10 .link {
11     stroke: #999;
12     stroke-opacity: .6;
13 }
14
15 .gnodes text {
16     pointer-events: none;
17     font: 6px sans-serif;
18 }
19
20 </style>
21 <body>
22 <script src="d3.min.js"></script>
23 <script>
24
25 var width = 1560,
26     height = 900;
27
28 var color = d3.scale.category20();
29
30 var force = d3.layout.force()
31     .charge(-1)
32     .linkDistance(500)
33     .size([width, height]);
34
35 var svg = d3.select("body").append("svg")
36     .attr("width", width)
37     .attr("height", height);

```

```
38
39 d3.json("sourceData.json", function(error, graph) {
40   if (error) throw error;
41
42   var nodeByname = d3.map();
43
44   graph.nodes.forEach(function(node) {
45     nodeByname.set(node.name, node);
46   });
47
48   graph.links.forEach(function(link) {
49     link.source = nodeByname.get(link.source);
50     link.target = nodeByname.get(link.target);
51   });
52
53   force
54     .nodes(graph.nodes)
55     .links(graph.links)
56     .start();
57
58   var link = svg.selectAll(".link")
59     .data(graph.links)
60     .enter().append("line")
61     .attr("class", "link")
62     .style("stroke-width", function(d) { return Math.sqrt(d.value); });
63
64   var gnodes = svg.selectAll('g.gnode')
65     .data(graph.nodes)
66     .enter()
67     .append('g')
68     .classed('gnode', true);
69
70   var node = gnodes.append("circle")
71     .attr("class", "node")
72     .attr("r", 5)
73     .style("fill", function(d) { return color(d.group); })
74     .call(force.drag);
75
76   var labels = gnodes.append("text")
77     .text(function(d) { return d.name; });
78
79   console.log(labels);
80
81   force.on("tick", function() {
82     link.attr("x1", function(d) { return d.source.x; })
83       .attr("y1", function(d) { return d.source.y; })
84       .attr("x2", function(d) { return d.target.x; })
85       .attr("y2", function(d) { return d.target.y; });
86
87     gnodes.attr("transform", function(d) {
88       return 'translate(' + [d.x, d.y] + ')';
89     });
90   });
91 });
```

92

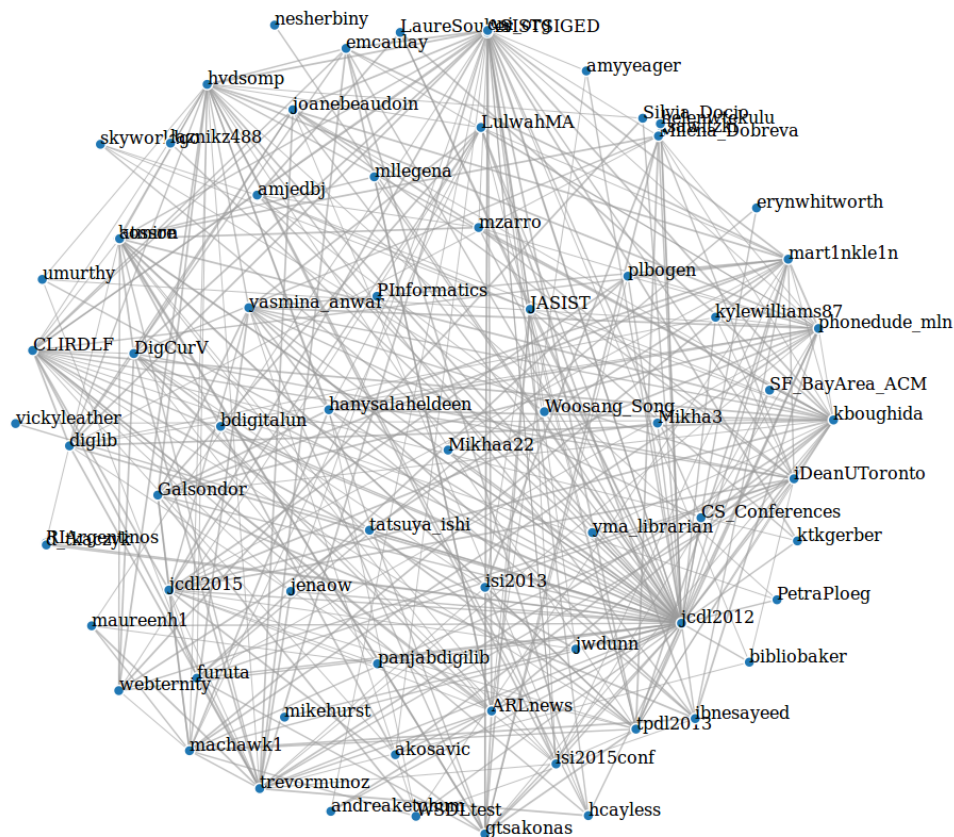
93 `</script>`

Figure 4: Graph of the relationship between jcdl2012 and its followers

Problem 2

Gender homophily in your Twitter graph (5 points)

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 11-15, Week 7. Does your Twitter graph exhibit gender homophily?

Answer