

Srividya Majeti

Assignment 6

CS 532: Introduction to Web Science

Dr. Michael Nelson

Spring 2016

March 18, 2016

Contents

| | | |
|---|-------------------------------|----|
| 1 | Question 1 | 1 |
| 2 | Question 2 | 7 |
| 3 | Extra-Credit Question-3 | 15 |
| | References | 19 |

Question 1

Use D3 to visualize your Twitter followers. Use my twitter account (“@phonedude_mln”) if you do not have ≥ 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-instead-of-just-viewing-their-source>

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

Following are the steps I have taken to solve the problem:

- Using the Twitter API I extracted all my followers and stored ‘screen_name’, ‘name’, ‘profile_image_url’, and ‘index number’ in a JSON file named as ‘nodesData’.
- In function ‘links()’ I stored the source and target nodes in a dictionary which gives the data for directed links. This code is listed in Listing 1.1.
- I took the above generated ‘nodesData’ file as input and obtained all possible pairs for my followers. This is stored in the file ‘sourceTarget’.
- For all these possible pairs I checked the existence of friendship between each other using ‘show_friendship’ API.
- This code is listed in Listing 1.2 .
- The final output Json is stored in ‘finalJsonData’. This data is taken as input for D3 code to generate a graph.
- The Output of the graph is located in URI <http://bl.ocks.org/majetisiri/b0fe8280da899311640c>. The screen shot of the graph is illustrated in Figure 2.2.

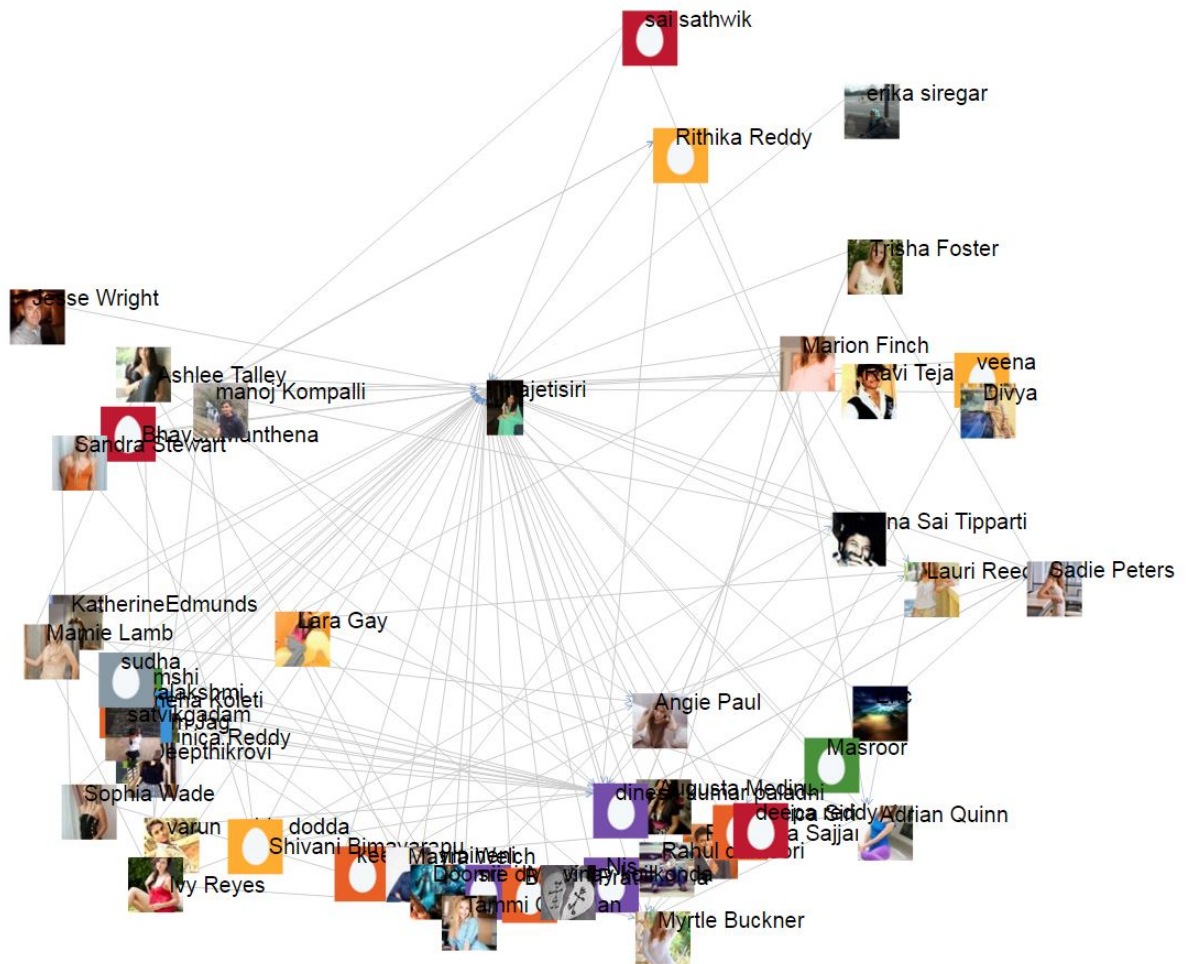


Fig. 1.1. Directed graph between me and my twitter followers including links between my followers if they have any

Code Listing

```

1 import tweepy
2 import sys
3 import json
4 import time
5
6 CONSUMER_KEY = 'wTSsHE3PTA3ZZPiaKHEiQnLtf'
7 CONSUMER_SECRET = '
    UblYYCmNYIEffAY4T4QHGHXwAWMFqiueXdx35xZFhoK3AECp1'

```

```

8 ACCESS_KEY = '157985123-
  WfVzlfDa8KStBZzevMfQBTM7fi8zKHYl2LQpTfGr'
9 ACCESS_SECRET = '
  lSax0XLwIimJ4VVbuU5OY9BpBic4vsSFfi0riAq3DPvTxU'
10
11 auth = tweepy.auth.OAuthHandler(CONSUMER_KEY,
  CONSUMER_SECRET)
12 auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)
13 api = tweepy.API(auth)
14
15 f = open('nodesData', 'w')
16 def nodes():
17     count = 1
18     page_count = 0
19     userData = []
20     for user in tweepy.Cursor(api.followers, screen_name='
      majetisiri').items():
21         usr = {}
22         usr['screenName'] = user.screen_name
23         usr['name'] = user.name
24         usr['id'] = count
25         usr['img'] = user.profile_image_url
26         usr['link'] = "https://twitter.com/"+user.screen_name
27         usr['size'] = 40000
28         page_count += 1
29         userData.append(usr)
30         count += 1
31     f.write(json.dumps(userData)+"\n")
32     f.close()
33
34 f1 = open('linksData', 'w')
35 read = open('nodesData', 'r')
36 def links():
37     userData = []
38     for line in read:
39         data = json.loads(line)
40         for user in data:
41             dict = {}
42             dict['source'] = user['id']
43             dict['target'] = 0
44             userData.append(dict)
45     f1.write(json.dumps(userData)+"\n")
46     f1.close()
47
48 nodes()
49 links()

```

Listing 1.1. Python code for getting my followers data

Code Listing

```

1 import tweepy
2 import commands
3 import json
4 import time
5
6 CONSUMER_KEY = 'wTSsHE3PTA3ZZPiaKHEiQnLtf'
7 CONSUMER_SECRET = '
    UbIYYCmNYIEffAY4T4QHGhXwAWMFqieXdx35xZFhoK3AECp1'
8 ACCESS_KEY = '157985123-
    WFvzlfDa8KStBZzevMfQBTM7fi8zKHY12LQpTfGr'
9 ACCESS_SECRET = '
    lSax0XLwIimJ4VVbuU5OY9BpBic4vsSF0riAq3DPvTxU'
10
11 auth = tweepy.auth.OAuthHandler(CONSUMER_KEY,
    CONSUMER_SECRET)
12 auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)
13 api = tweepy.API(auth)
14
15 f1=open('sourceTarget','w')
16 def getSource():
17     read=open('nodesData','r')
18     data= json.load(read)
19     list= []
20     for user in range(0,len(data)):
21         sourceScreenName= data[user]["screenName"]
22         for user1 in range(user,len(data)-1):
23             targetScreenName = data[user1+1]["screenName"]
24             checkSourceFollowersAndFollowing(sourceScreenName,
                targetScreenName)
25
26 def checkSourceFollowersAndFollowing(sourceScreenName,
    targetScreenName):
27     dict= {}
28     count = 0
29     dict['source']= sourceScreenName
30     dict['target']= targetScreenName
31     f1.write(json.dumps(dict)+"\n")
32
33
34 def getAllLinks():
35     f2 = open('linksData','w')
36     read = open('sourceTarget','r')
37     data= json.load(read)
38     for user in data:
39         dict= {}
40         sourceScreenName= user["source"]
41         targetScreenName= user["target"]

```



```

42     result = api.show_friendship(source_screen_name=
        sourceScreenName, target_screen_name=
        targetScreenName)
43     dict[ 'followed_by' ] =result[0].followed_by
44     dict[ 'following' ] =result[0].following
45     dict[ 'screen_name1' ]= result[0].screen_name
46     dict[ 'screen_name2' ]= result[1].screen_name
47     f2.write(json.dumps(dict)+"\n")
48
49 def getTrueLinks():
50     read = open('linksData','r')
51     f2 = open('trueLinksData','w')
52     data= json.load(read)
53     for user in data:
54         dict = {}
55         if user["following"] == True:
56             dict[ 'source' ]= user["screen_name1"]
57             dict[ 'target' ]= user["screen_name2"]
58             f2.write(json.dumps(dict)+"\n")
59         elif user["followed_by"]== True:
60             dict[ 'source' ]= user["screen_name2"]
61             dict[ 'target' ]= user["screen_name1"]
62             f2.write(json.dumps(dict)+"\n")
63
64 def passTrueLinksSourceAndTarget():
65     read = open('trueLinksData','r')
66     data= json.load(read)
67     for user in data:
68         getIds(user["source"],user["target"])
69
70 def getIds(name1,name2):
71     read = open('nodesData','r')
72     f2 = open('linkIds','a')
73     data= json.load(read)
74     for user in data:
75         dict ={}
76         if name1 == user["screenName"]:
77             id = user["id"]
78             # print name1
79             dict[ 'source' ] = id
80             f2.write(json.dumps(dict)+"\n")
81         elif name2 == user["screenName"]:
82             id = user["id"]
83             # print name2
84             dict[ 'target' ] = id
85             f2.write(json.dumps(dict)+"\n")
86     f2.close()
87     getSource()
88     getAllLinks()

```

```
89 | getTrueLinks ()  
90 | passTrueLinksSourceAndTarget ()
```

Listing 1.2. Python code for checking if friendship exists between my twitter followers

Question 2

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?

Following are the steps I have taken to solve the problem:

- I split the names of each follower into last name, first name and initial.
- By considering the first name of each user I found the gender of each follower using genderize API.
- I updated the JSON data in question 1 by adding gender to each follower.

- The screen shot of updated JSON data is in Figure 2.1

```
[
  "nodes": [
    {
      "id": 0,
      "size": 40000,
      "name": "majetisiri",
      "img": "http://www.cs.odu.edu/~smajeti/website/img/me.JPG",
      "screenName": "majetisiri",
      "link": "https://twitter.com/majetisiri",
      "gender": 0
    },
    {
      "id": 1,
      "size": 40000,
      "name": "varun reddy dodda",
      "img": "http://pbs.twimg.com/profile_images/3072439773/6ba21420364b003029dd9746bc4d313f_normal.jpeg",
      "screenName": "doddavarunreddy",
      "link": "https://twitter.com/doddavarunreddy",
      "gender": 1
    },
    {
      "id": 2,
      "size": 40000,
      "name": "keerthi talipineni",
      "img": "http://abs.twimg.com/sticky/default_profile_images/default_profile_1_normal.png",
      "screenName": "keerthitalip",
      "link": "https://twitter.com/keerthitalip",
      "gender": 1
    },
    {
      "id": 3,
      "size": 40000,
      "name": "vamshi",
      "img": "http://abs.twimg.com/sticky/default_profile_images/default_profile_2_normal.png",
      "screenName": "kolanuvamshi",
      "link": "https://twitter.com/kolanuvamshi",
      "gender": 1
    }
  ]
]
```

Fig. 2.1. Output of updated nodes data

- The code is listed in Listing 2.1.

- This data is taken as input for D3 to generate a graph which differentiates the followers based on gender.
- The table with list of followers and their gender is illustrated in Table 2.1.
- There are 11 followers whose gender is 'null'. When I tried to get my gender with my first name 'majetisiri' it gave a 'null'. If I draw a graph by removing nodes with 'null' gender, it leads to removal of 80 percentage of the links in the graph. So I created a graph including followers who have a 'null' gender.
- All male followers are represented in color 'orange', female followers in color 'green' and those whose gender is not determined are represented in color 'blue'. This graph is illustrated in Figure 2.2
- This graph is located at URI <http://bl.ocks.org/majetisiri/fd87d5725027a5441f78>

Table 2.1. Table with followers and gender

| |
|------------------------|
| Name — Gender |
| Naina — female |
| veena — female |
| Shivani— female |
| Sneha — female |
| erika — female |
| Mayra — female |
| Ivy — female |
| Marion — female |
| Mamie — female |
| Augusta— female |
| Lauri — female |
| Ashlee — female |
| Sadie — female |
| Angie — female |
| Sophia — female |
| Myrtle — female |
| Trisha — female |
| Lara — female |
| Sandra — female |
| Tammi — female |
| Divya — female |
| deepa — female |
| sudha — female |
| varun— male |
| keerthi— male |
| vamshi— male |
| srinivas— male |
| sai— male |
| Jesse— male |
| Prasanna— male |
| Nishant‘— male |
| Ravi— male |
| manoj— male |
| Eric— male |
| Masroor— male |
| Rahul— male |
| Adrian— male |
| vinay— male |
| dinesh— male |
| BhavaniManthena — null |
| Mounica — null |
| Vam — null |
| Mounica— null |
| KatherineEdmunds— null |
| Basani — null |
| Deepthikrovi— null |
| rajyalakshmi— null |
| Rithika — null |
| Doomie — null |
| satvkgadam — null |

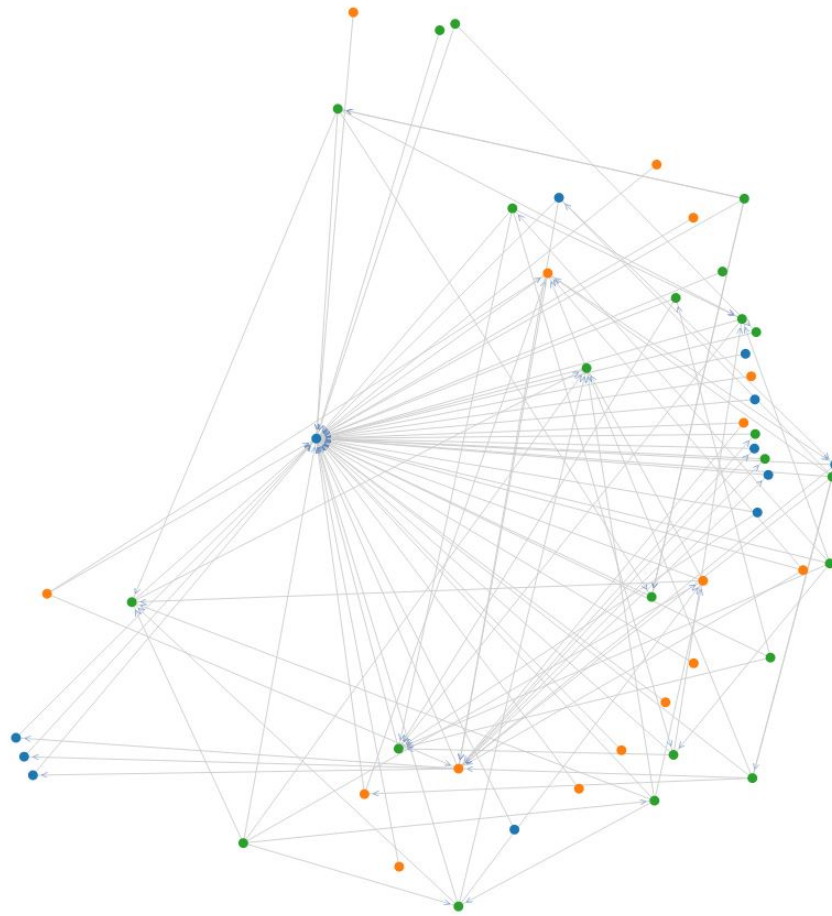


Fig. 2.2. Output Graph with followers distinguished based on gender

Code Listing

```

1 import requests
2 import json
3
4 def function(input):
5     LastName = name.split()
6
7 read=open('nodesData','r')
8 fl=open('gender','w')
9 list = []
10 for line in read:
11     data= json.loads(line)

```

```
12  for user in data:
13      name = (str(user[ 'name' ]))
14      print "\n"+name
15      name = name.partition(" ")
16      firstName = name[0]
17      print "FirstName:"+firstName
18      url = "https://api.genderize.io/?name=" +firstName
19      r = requests.get(url)
20      genderData= r.content
21      print genderData
22      print type(genderData)
23      list.append(r.content)
24  fl.write(str(list))
```

Listing 2.1. Python code for getting gender for followers using their first name

Extra-Credit Question-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 between connected and disconnected.

Following are the steps I have taken to solve the problem:

- I converted the KarateClub graphML data into JSON structure.
- The code is listed in Listing 3.1
- I took the converted karateClub JSON data as input for my D3 code and generated a force directed graph.
- The output of the graph is located at URI <http://bl.ocks.org/majetisiri/316e3a1537b469154779>. The graph appears more accurately in 'Google chrome' than in any other browsers.
- Above the graph there are 2 buttons with captions 'Before split' and 'After split'. If we click on the button 'Before split' it generates a graph with all the nodes in same color. When we hover on the nodes of the graph, it displays the name of the node. This is illustrated in Figure 3.1.

Before Split After Split

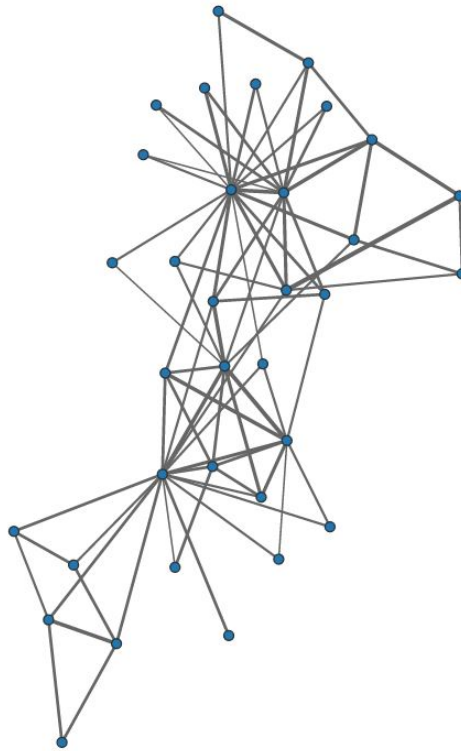


Fig. 3.1. Graph before split

- If we click on the button 'After split' it generates a graph which differentiates 2 groups based on faction. This is illustrated in Figure 3.1.

Before Split After Split

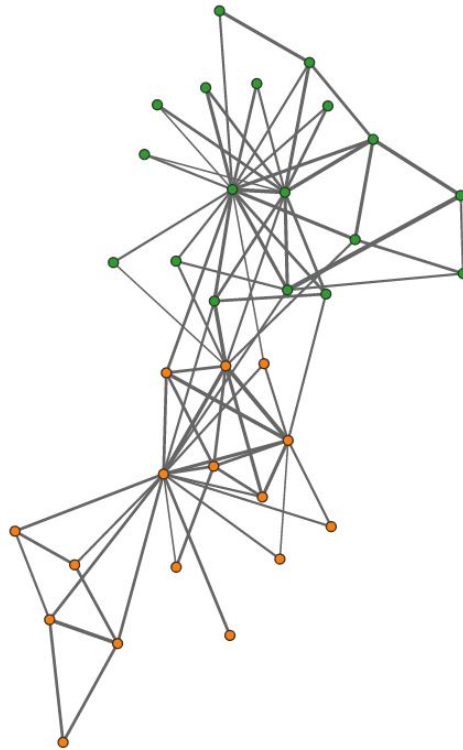


Fig. 3.2. Graph after split which distinguishes two groups based on Faction

Code Listing

```

1 from xml.etree import ElementTree
2 import json
3
4 with open('karate.GraphML', 'rt') as f:
5     fl= open('karateClub.json', 'w')
6     tree = ElementTree.parse(f)
7     root = tree.getroot()
8     print root
9     data = {}
10    data1={}
11    nodes = []

```

```

12 links= []
13 count = 0
14 f1.write('{ ' + '\n' + '"nodes":\n' + '[' + "\n" )
15 for parent in root:
16     for child in parent:
17         if child.tag == ('{http://graphml.graphdrawing.org/
18             xmlns}node'):
19             id = child.get('id')
20             print "id:",id
21             for children in child:
22                 if children.attrib.get('key') == 'name':
23                     name= children.text
24                     print "name:",name
25                     data["name"] = name
26                     f1.write(json.dumps(data)+'\n')
27                 if children.attrib.get('key') == 'Faction':
28                     Faction= children.text
29                     print "Faction:",Faction
30                     data["faction"] = int(Faction)
31                     data["color"] = 1
32 f1.write('], ' + '\n' + '"links":\n' + '[' + "\n" )
33 for parent in root:
34     for child in parent:
35         if child.tag == ('{http://graphml.graphdrawing.org/
36             xmlns}edge'):
37             source = child.get('source')
38             count +=1
39             data1["source"] = int(source)
40             target = child.get('target')
41             data1["target"] = int(target)
42             for children in child:
43                 if children.attrib.get('key') == 'weight':
44                     weight= children.text
45                     data1["weight"] = int(weight)
46                     data1["id"] = "e"+str(count)
47                     f1.write(json.dumps(data1)+"\n")
48 f1.write(']' + '\n' + '}')

```

Listing 3.1. Python code for converting KarateClub XML graph to JSON data

References

1. D3 graph for question 1: <http://bl.ocks.org/mbostock/1153292>, Mike Bostock, 2016
2. D3 graph for question 3: <https://bl.ocks.org/mbostock/4062045>, Mike Bostock, 2016
3. How to change data on click button: <http://bl.ocks.org/d3noob/7030f35b72de721622b8>, d3noob Block, 2015
4. Force directed layout with images: <http://bl.ocks.org/eesur/be2abfb3155a38be4de4>, Sundars Block , 2015
5. How to split a name: <http://stackoverflow.com/questions/1720503/parsing-peoples-first-and-last-name-in-python> , Joel Spolsky, 2008