Kevin R. Clemmons
(oduprogrammer16)
CS432 — Spring 2016 **Assignment 6** March 18, 2016

All files mentioned in this file are uploaded into the *github* repository.

The LaTeXcode invovled in the generation of this document was aided by the example code provided in the links that Dr. Nelson sent out on January 17, 2016[**mohammedaturban2013**].

This document was compiled on `www.sharelatex.com`

Problem 1

Before starting to work on the D3 Graph, a script was written to extract the data from my twitter account.

First a list of all of my followers was created and written to a text file called *kevin-followers.txt*. This file is primarily used for backup. Then the program goes through the list of all of my followers and for each follower, checks the other followers to determine if the follower follows the other followers. The function that determines if a person follows another person is found the tweepy library[5]. This code is found in Listing 1 and the final result is a csv file which describes a list of edges in the format (follower,personFollowing). In addition, this program utilizes a credential library that I developed, to manage my twitter authorization credentials. The code for this credential library is located at my repository via the following link `https://github.com/oduprogrammer16/CredentialManager`.

Listing 1: Determining if a follower follows another follower

```
try:
       result =
          api.show_friendship(source_screen_name=user,target_screen_name=follower)

except tweepy.RateLimitError:
       printEdgeList(edgeList,'kevin_network.csv')
       printEdgeList(unknownEdges,'unknownEdges.csv')
       print("\t\tRate Limit Error Exceeded: sleeping for 15 minutes")
       time.sleep(15 * 60)
       result = api.show_friendship(source_screen_name=user,
       target_screen_name=follower)

except tweepy.TweepError as e:
       pass

if result is not None:
       if result[1].following:
              edgeList.append((follower,user))
else:
       unknownEdges.append((follower,user))
```

It should be noted that this code is not effecient. A better strategy for determining who follows who in my follower list would be to get a follower's screen-name and then determine who they follow in my follower list.

The code that was ran to extract the followers and who they follow in my follower list took exactly 4 hours, 29 minutes and thirty one seconds. This due to the fact that the twitter api only allows 180 requests every 15 minutes. [3]

After the data was extracted, a visual graph was created using D3[2]. The graph was created in an html file called *followergraph.html* and the is a modified version of Mike Bostock's basic directional force layout diagram[1]. Figure 1 shows my network graph.

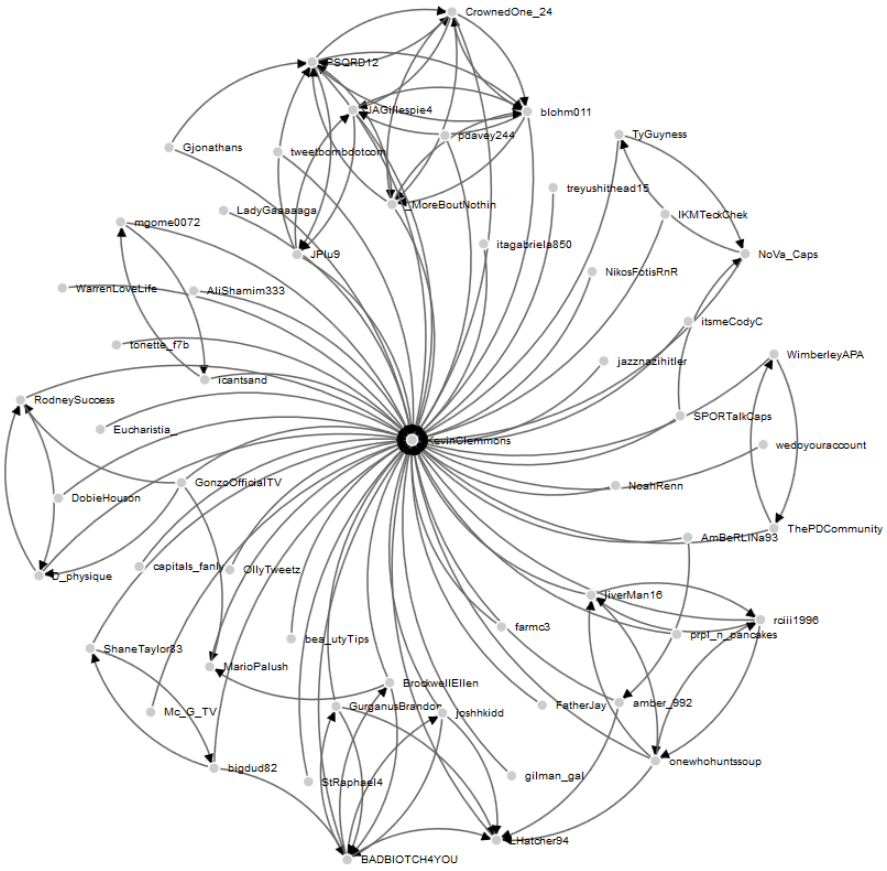

Figure 1: Follower Network

To determine Gender Homophily, a list of followers and their names was obtained from my twitter account using the twitter api[5]. Once all the names were obtained, the genders of each follower was estimated using the the gender-detector library[6]. The genders for all the followers are dumped into a json file called *backup2.json*. The program was able to obtain genders for 29 followers and was unable to obtain genders for 27 followers. Table 1 lists the users in which a gender was successfully estimated and Table 2 lists the users in which a gender was not successfully estimated.

Username	Name	Gender
JPlu9	Jared Pluciniczak	male
StRaphael4	St Raphael	male
joshhkidd	Joshua Reynolds	male
Dphysique	Dynasty For Life Rec	female
beautyTips	Beauty Tips	female
JAGillespie4	Justin Gillespie	male
Gjonathans	Jonathan Gustafsson	male
MarioPalush	MARIO	male
LadyGaaaaaga	Lady Gaga	female
GurganusBrandon	Brandon Gurganus	male
amber992	Amber Schakel	female
KevinClemmons	Kevin Clemmons	male
NoahRenn	Noah Renn	male
rciii1996	Robert RCIII Croson	male
CrownedOne24	Steven	male
itagabriela850	gabriela	female
treyushithead15	trey peters	male
bigdud82	Mike Dudley	male
RodneySuccess	Rodney Pickett II	male
ThePDCommunity	Kevin McCarthy	male
PSQRD12	Paul Porter	male
AliShamim333	Shamim	female
LHatcher94	luke	male
liverMan16	Eric Liverman	male
AmBeRLiNa93	Amber Schakel	female
WarrenLoveLife	Warren Happy	male
prplnpancakes	Daniel Pancake	male
BrockwellEllen	Ellen brockwell	female
DobieHouson	dobie houson	male

Table 1: Followers that a Gender was Obtained

Username	Name
jazznazihitler	jay z
WimberleyAPA	Wimberley APA (WAPA)
tweetbombdotcom	TweetBomb.com
tonettef7b	tonettef7b
pdavey244	Pat
McGTV	Mc G TV
NoVaCaps	NoVa Caps
IKMTeckChek	IKM TeckChek
BADBIOTCH4YOU	.
SPORTalkCaps	Capitals SPORTalk
blohm011	Hunter Blohm
capitalsfanly	Capitals Report
gilmangal	cabbage patch kid
GonzoOfficialTV	GONZO
ShaneTaylor83	Shane Taylor
onewhohuntssoup	Hunter Zupo
TyGuyness	Tyler Humphries
wedoyouraccount	wedoyouraccounting
OllyTweetz	I Follow Back ;) !
NikosFotisRnR	NikosFotisR'nRoller
farmc3	A. P. Parsley
MoreBoutNothin	Scott, Marquis
FatherJay	Jay Wagner
mgome0072	mg
icantsand	Sandy 'man' Wilson
Eucharistia	Real Presence
itsmeCodyC	Cody Crampton

Table 2: Followers that a Gender was not Obtained

Once The genders were obtained, a new csv file called *kevinGenderNetwork* was generated and contains a reduced number of edges. This is due to the fact that the edges who had nodes in which a gender was deemed unknown were eliminated from the graph. Figure 2 shows the network without the nodes in which a gender could not be obtained.

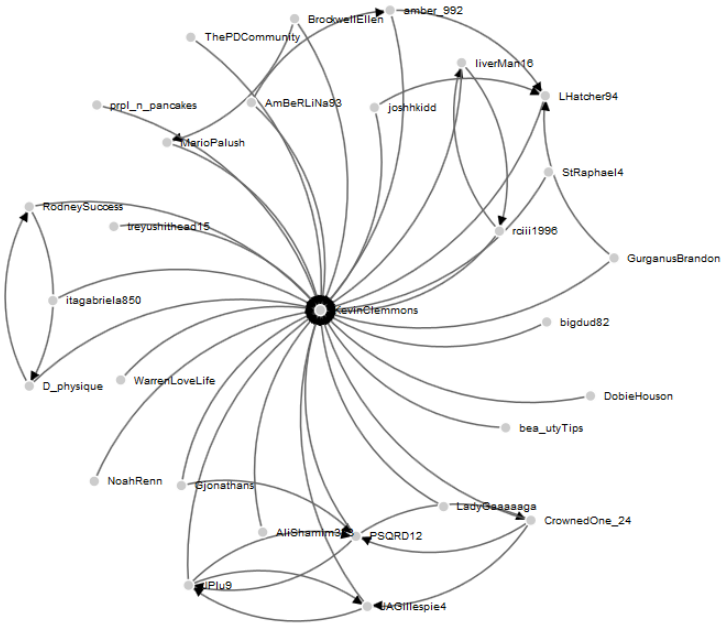

Figure 2: Gender Network

To compute the gender homophlly of the graph, let p be the ratio of male to male edges to the total number of edges, q be the ratio of female to female edges to the total number of edges and r represent be the ratio of cross gender edges to the total number of edges. Gender homophilly of the graph exists if $r < 2pq$[4].

To compute p, q and r, A program called *genderHomophily.py* iterates through the edges, and computes p, q, r and $2pq$ The counts are then divided by the total number of edges.

The following results were obtained below:

- $p = 0.02$
- $q = 0.71$
- $r = 0.26$
- $2pq = 0.031$

Since $r > 2pq$, gender homophily exists for my graph.

References

[1] Mike Bostock. *Basic Directional Force Layout Diagram*. Feb. 2016. URL: `http://bl.ocks.org/d3noob/5141278`.

[2] Mike Bostock. *Data-Driven Documents*. 2015. URL: `https://d3js.org/`.

[3] Twitter Inc. *https://dev.twitter.com/rest/public/rate-limiting*. 2016. URL: `https://dev.twitter.com/rest/public/rate-limiting`.

[4] Michael Nelson. *Selection and Social Influence Slides*. URL: `http://phonedude.github.io/cs532-s16/`.

[5] Joshua Rosslein. *API Reference*. 2009. URL: `http://docs.tweepy.org/en/latest/api.html`.

[6] Marcos Vanetta. *gender-detector 0.0.4*. 2014. URL: `https://pypi.python.org/pypi/gender-detector/0.0.4`.