BlogPost#2

Contents

## In a way, failure is success

This week’s blog post is going to focus primarily on visualization, with some small attention to inherent network attributes. So let’s get that network graph first!

Load in the data

rm(list=ls())  
  
#MacOSPaths  
redditData <- read.csv("~/Documents/School/DACSS/695N/NetworkAnalysisFinalProject/Data/edgelist.csv")  
nodeAttr <- read.csv("~/Documents/School/DACSS/695N/NetworkAnalysisFinalProject/Data/nodeattr.csv")  
  
#LinuxPaths  
#redditData <- read.csv("~/Developer/NetworkAnalysisFinal-master/Data/edgelist.csv")  
#nodeAttr <- read.csv("~/Developer/NetworkAnalysisFinal-master/Data/nodeattr.csv")

Let’s get some really basic descriptives

library(glue)  
library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':  
  
 filter, lag

The following objects are masked from 'package:base':  
  
 intersect, setdiff, setequal, union

library(igraph)

Attaching package: 'igraph'

The following objects are masked from 'package:dplyr':  
  
 as\_data\_frame, groups, union

The following objects are masked from 'package:stats':  
  
 decompose, spectrum

The following object is masked from 'package:base':  
  
 union

library(GGally)

Loading required package: ggplot2

Registered S3 method overwritten by 'GGally':  
 method from   
 +.gg ggplot2

library(network)

'network' 1.18.2 (2023-12-04), part of the Statnet Project  
\* 'news(package="network")' for changes since last version  
\* 'citation("network")' for citation information  
\* 'https://statnet.org' for help, support, and other information

Attaching package: 'network'

The following objects are masked from 'package:igraph':  
  
 %c%, %s%, add.edges, add.vertices, delete.edges, delete.vertices,  
 get.edge.attribute, get.edges, get.vertex.attribute, is.bipartite,  
 is.directed, list.edge.attributes, list.vertex.attributes,  
 set.edge.attribute, set.vertex.attribute

library(bipartite)

Loading required package: vegan

Loading required package: permute

Attaching package: 'permute'

The following object is masked from 'package:igraph':  
  
 permute

Loading required package: lattice

This is vegan 2.6-8

Attaching package: 'vegan'

The following object is masked from 'package:igraph':  
  
 diversity

Loading required package: sna

Loading required package: statnet.common

Attaching package: 'statnet.common'

The following objects are masked from 'package:base':  
  
 attr, order

sna: Tools for Social Network Analysis  
Version 2.8 created on 2024-09-07.  
copyright (c) 2005, Carter T. Butts, University of California-Irvine  
 For citation information, type citation("sna").  
 Type help(package="sna") to get started.

Attaching package: 'sna'

The following objects are masked from 'package:igraph':  
  
 betweenness, bonpow, closeness, components, degree, dyad.census,  
 evcent, hierarchy, is.connected, neighborhood, triad.census

This is bipartite 2.20.  
 For latest changes see versionlog in ?"bipartite-package". For citation see: citation("bipartite").  
 Have a nice time plotting and analysing two-mode networks.

Attaching package: 'bipartite'

The following object is masked from 'package:vegan':  
  
 nullmodel

The following object is masked from 'package:igraph':  
  
 strength

library(ggraph)  
library(tidygraph)

Attaching package: 'tidygraph'

The following object is masked from 'package:igraph':  
  
 groups

The following object is masked from 'package:stats':  
  
 filter

library(dplyr)

glue('Number of Entries: {dim(redditData)[1]}')

Number of Entries: 36082

glue('Number of Unique Mods: {sum(nodeAttr$Type == "mod")}')

Number of Unique Mods: 25488

glue('Number of Subreddits: {sum(nodeAttr$Type == "sub")}')

Number of Subreddits: 6916

There’s about a couple thousand subreddits that were

Trying the igraph way

reddit.ig <- graph\_from\_data\_frame(  
 redditData,  
 directed = F  
)  
reddit.ig

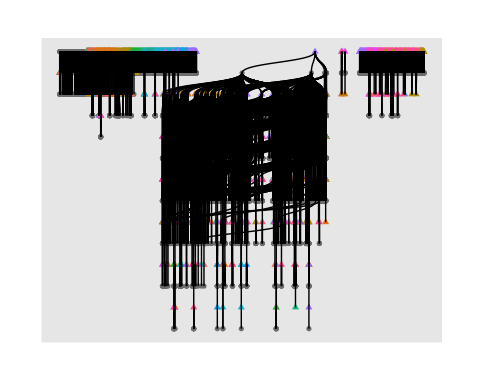
IGRAPH 928488e UN-- 32404 36082 --   
+ attr: name (v/c)  
+ edges from 928488e (vertex names):  
 [1] u/mbround18 --Toyota u/lKANl --Toyota   
 [3] u/dirty\_hooker --Hookit u/CJM8515 --Hookit   
 [5] u/brick-geek --regularcarreviews u/SteveTehTree --regularcarreviews  
 [7] u/skorea2021 --ToyotaTacoma u/stabracadabra --ToyotaTacoma   
 [9] u/PlusZombie5154--ToyotaTacoma u/jpbronco --ToyotaTacoma   
[11] u/Datsoon --Nissan u/raybrant --Nissan   
[13] u/tristinGrind --Nissan u/sarge-m --Nissan   
[15] u/Justice502 --Saturn   
+ ... omitted several edges

nodeAttr <- nodeAttr[match(V(reddit.ig)$name, nodeAttr$Name), ]  
V(reddit.ig)$type <- nodeAttr$Type  
V(reddit.ig)$category <- nodeAttr$Category  
V(reddit.ig)$subcount <- nodeAttr$SubscriberCount  
V(reddit.ig)$nummods <- nodeAttr$NumberOfModerators  
reddit.ig

IGRAPH 928488e UN-B 32404 36082 --   
+ attr: name (v/c), type (v/c), category (v/c), subcount (v/n), nummods  
| (v/n)  
+ edges from 928488e (vertex names):  
 [1] u/mbround18 --Toyota u/lKANl --Toyota   
 [3] u/dirty\_hooker --Hookit u/CJM8515 --Hookit   
 [5] u/brick-geek --regularcarreviews u/SteveTehTree --regularcarreviews  
 [7] u/skorea2021 --ToyotaTacoma u/stabracadabra --ToyotaTacoma   
 [9] u/PlusZombie5154--ToyotaTacoma u/jpbronco --ToyotaTacoma   
[11] u/Datsoon --Nissan u/raybrant --Nissan   
[13] u/tristinGrind --Nissan u/sarge-m --Nissan   
+ ... omitted several edges

V(reddit.ig)$shape <- ifelse(V(reddit.ig)$type == "mod", "circle","triangle")  
V(reddit.ig)$color <- ifelse(V(reddit.ig)$type == "mod", "blue","red")  
V(reddit.ig)$vertexlabel <- ifelse(V(reddit.ig)$type == "mod", "", V(reddit.ig)$name)

#ggraph(reddit.ig, layout = 'fr', niter=1000) +  
ggraph(reddit.ig, 'igraph', algorithm = 'tree') +  
 geom\_node\_point(aes(colour=V(reddit.ig)$category, shape = V(reddit.ig)$shape), show.legend = F)+  
 #geom\_node\_label(data = V(reddit.ig)$vertexlabel)  
 geom\_edge\_diagonal() +  
 theme(  
 plot.margin = unit(c(1, 1, 1, 1), "cm") # Increase plot margins  
 )



ggsave(filename = "plot1.png", width = 8.53, height = 4.8, units = "in", dpi = 300)

#ggraph(reddit.ig, layout = 'fr', niter=1000) +  
ggraph(reddit.ig, 'igraph', algorithm = 'tree') +  
 geom\_node\_point(aes(colour=V(reddit.ig)$category, shape = V(reddit.ig)$shape), show.legend = T)+  
 #geom\_node\_label(data = V(reddit.ig)$vertexlabel)  
 geom\_edge\_diagonal() +  
 theme(  
 plot.margin = unit(c(1, 1, 1, 1), "cm"), # Increase plot margins  
 legend.position = "right",   
 legend.key.size = unit(0.5, "cm")  
 ) +  
 guides(  
 shape = "none"  
 )

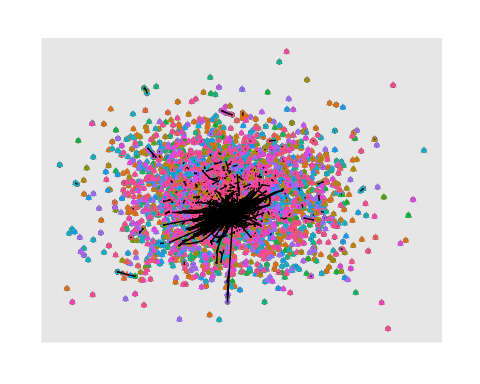


ggsave(filename = "plot.png", width = 8.53, height = 4.8, units = "in", dpi = 300)

#dgreddit.ig <- decompose(reddit.ig)  
#ggraph(graph\_tidy, 'igraph', algorithm = 'tree') +  
# geom\_node\_point(aes(colour=V(graph\_tidy)$category, shape = V(graph\_tidy)$shape), show.legend = F)+  
# #geom\_node\_label(data = V(reddit.ig)$vertexlabel)  
# geom\_edge\_diagonal() +  
# geom\_node\_text(label=V(graph\_tidy)$name) +  
# theme(  
# plot.margin = unit(c(1, 1, 1, 1), "cm") # Increase plot margins  
# )

#graphReddit <- as\_tbl\_graph(reddit.ig)  
  
#target\_nodes <- graphReddit %>%   
# activate(nodes) %>%   
# filter(category == "history")  
  
#target\_ids <- target\_nodes %>%  
# pull(name)  
  
#associated\_nodes <- graphReddit %>%  
# activate(nodes) %>%  
# filter(name %in% neighbors(reddit.ig, target\_ids))  
# #bind\_rows(target\_nodes) # Combine with target nodes if you want to keep them  
  
#associated\_nodes

ggraph(reddit.ig, layout = 'fr', niter=1000) +  
#ggraph(reddit.ig, 'igraph', algorithm = 'tree') +  
 geom\_node\_point(aes(colour=V(reddit.ig)$category, shape = V(reddit.ig)$shape), show.legend = F)+  
 #geom\_node\_label(data = V(reddit.ig)$vertexlabel)  
 geom\_edge\_link() +  
 theme(  
 plot.margin = unit(c(1, 1, 1, 1), "cm") # Increase plot margins  
 )



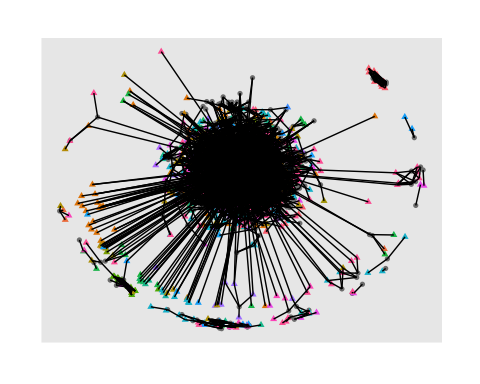
ggsave(filename = "plot2.png", width = 8.53, height = 4.8, units = "in", dpi = 300)

degrees <- igraph::degree(reddit.ig)  
  
# Identify vertices to keep (degree > 2)  
verticesToKeep <- V(reddit.ig)[degrees > 2]  
  
# Create a new graph with only the vertices that have more than 2 connections  
filteredGraph <- subgraph(reddit.ig, verticesToKeep)  
  
filteredGraph <- igraph::delete.vertices(filteredGraph,igraph::degree(filteredGraph) == 0)

Warning: `delete.vertices()` was deprecated in igraph 2.0.0.  
ℹ Please use `delete\_vertices()` instead.

ggraph(filteredGraph, layout = 'kk') +  
#ggraph(reddit.ig, 'igraph', algorithm = 'tree') +  
 geom\_node\_point(aes(colour=V(filteredGraph)$category, shape = V(filteredGraph)$shape), show.legend = F)+  
 #geom\_node\_label(data = V(reddit.ig)$vertexlabel)  
 geom\_edge\_fan(aes(show.legend = F)) +  
 theme(  
 plot.margin = unit(c(1, 1, 1, 1), "cm") # Increase plot margins  
 )

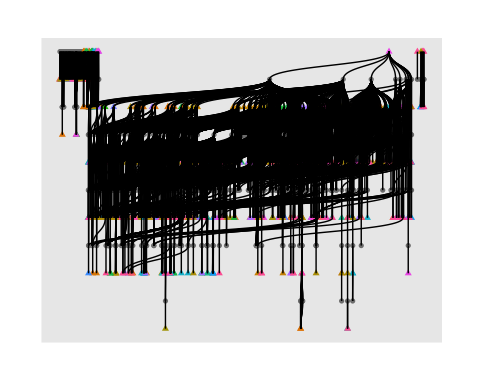
Warning in geom\_edge\_fan(aes(show.legend = F)): Ignoring unknown aesthetics:  
show.legend



ggsave(filename = "plot3.png", width = 8.53, height = 4.8, units = "in", dpi = 300)

As it turns out graphing this network is incredibly difficult. Getting any sort of graphs to work effectively is a slow process, with disappointing results often. Statnet could not load anything into memory.GGnet seems to work fairly well, but the kk layout which normaly isn’t as clustered consumes too much memory (maybe there’s a trick to this?) The good news is after a lot of fiddling I have a cleaned dataset, and have figured out generally how to make all the node attributes and my edgelist play together nicely. Unfortuantely there isn’t a more tangible result for the amount of different layouts and algorithms that were tried so far except some really grouped graphs. An alternative solution would also be to truncate my dataset but this likely wouldn’t be as insightful. Based on my current albeit hard to interpret graphs there does seem to be some tangible patterns that need to be explored. There are definitely some group clusters.

ggraph(filteredGraph, 'igraph', algorithm = 'tree') +  
 geom\_node\_point(aes(colour=V(filteredGraph)$category, shape = V(filteredGraph)$shape), show.legend = F)+  
 #geom\_node\_label(data = V(reddit.ig)$vertexlabel)  
 geom\_edge\_diagonal() +  
 theme(  
 plot.margin = unit(c(1, 1, 1, 1), "cm") # Increase plot margins  
 )



ggsave(filename = "plot4.png", width = 8.53, height = 4.8, units = "in", dpi = 300)