Beginning Steps in Network Data Analysis: Understanding Crossposting between Reddit Communities

Carlyle Morgan

Introduction

The Stanford Network Analysis Project(SNAP) hosts a dataset that contains information about reddit 'crossposting', in which a post on one subcommunity(or 'Subreddit') of the site posts a hyperlink to another subcommunity of the site. According to SNAP, the Reddit data is formatted such that "each hyperlink is annotated with three properties: the timestamp, the sentiment of the source community post towards the target community post, and the text property vector of the source post. The network is directed, signed, temporal, and attributed". From these attributes, one can understand

Cleaning

```r
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 4.0.5

##
## Attaching package: 'igraph'

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

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

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:igraph':
##
##     as_data_frame, groups, union

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

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

```r
library(stringr)

rh<-read.csv("C:/Users/sc_mo/Documents/R/Projects/Stats Research/reddithyperlinks.tsv", sep = "\t")
```

```r
g.rh<- graph.data.frame(rh)
is.simple(g.rh)
```

```
## [1] FALSE
```

```r
##Weight according to the number of connected edges
g.rh<-as.undirected(g.rh, mode = 'collapse')
E(g.rh)$weight<-1
g.rh.weighted<-simplify(g.rh)
```

Exploratory Data Analysis

Overall Edge and Vertex Summary Statistics

Having simplified and applying weighting to our network, it might be helpful to explore the relationship between the degrees and strength of our verticies.

Which subreddits have the greatest number of instances of starting and receiving crossposts? This is concerned with how we decided to weight each edge, and thus nodes with greater summed weights have

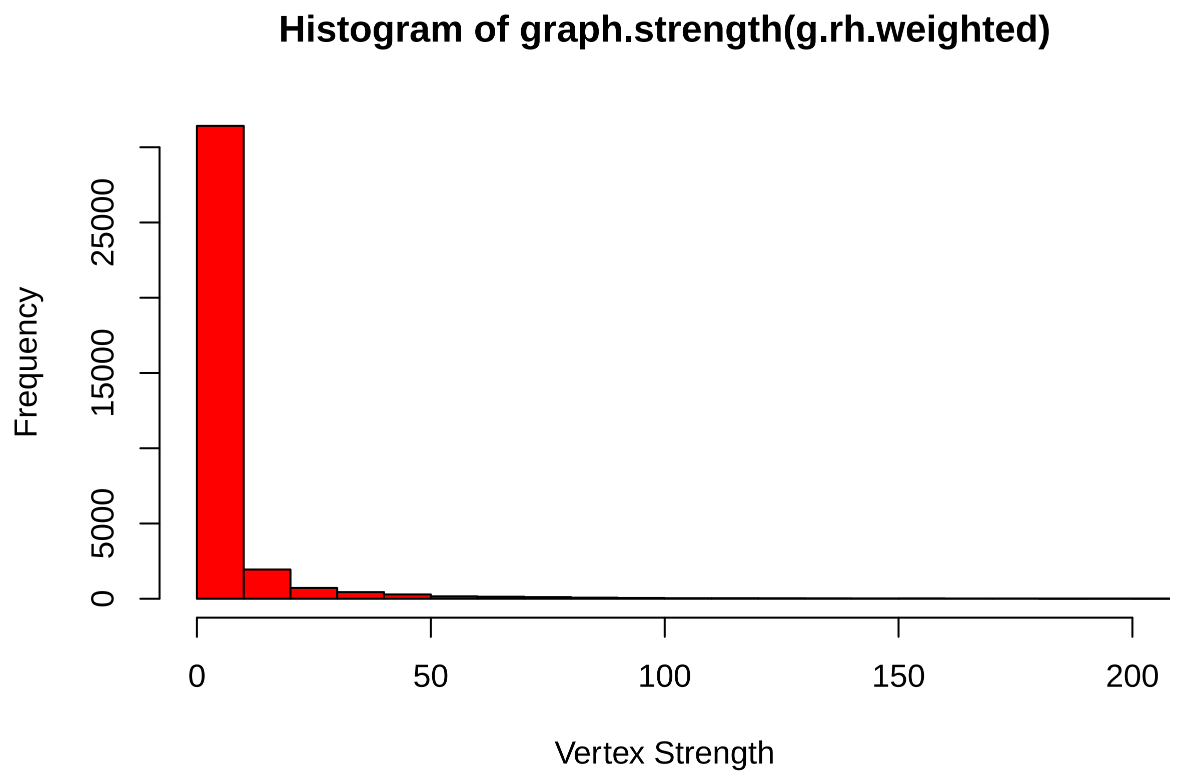
```r
head(sort(graph.strength(g.rh.weighted), decreasing=TRUE),25)
```

```
##            askreddit        todayilearned       writingprompts                 iama
##                 3079                 2371                 2242                 2093
##       subredditdrama         outoftheloop                 pics       leagueoflegends
##                 1887                 1232                 1196                 1117
##         adviceanimals         tipofmypenis            copypasta               videos
##                 1117                 1096                 1031                  959
##                  wtf                funny     explainlikeimfive                 mhoc
##                  924                  897                  863                  857
##          circlebroke            worldnews           conspiracy               gaming
##                  807                  806                  779                  763
##                drama        hailcorporate circlejerkcopypasta     shitliberalssay
##                  709                  701                  700                  698
##             dogecoin
##                  638
```

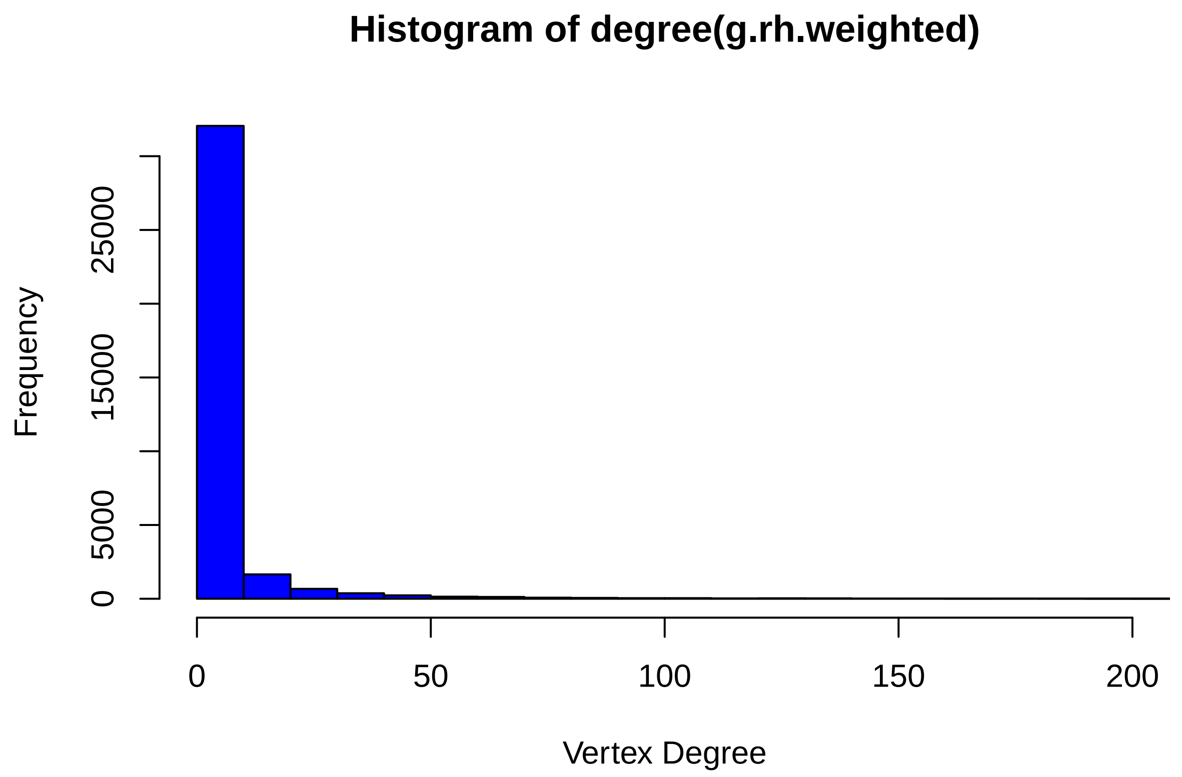
```r
head(sort(degree(g.rh.weighted), decreasing=TRUE),25)
```

```
##          askreddit               iama     subredditdrama    writingprompts
##               2336               1839               1598              1043
##       outoftheloop               pics             videos     todayilearned
##                986                956                881               816
##              funny             gaming    leagueoflegends         copypasta
##                783                719                708               669
##         conspiracy          worldnews              drama explainlikeimfive
##                666                665                642               617
##               mhoc        legaladvice               news subredditoftheday
##                607                590                587               573
##         the_donald       pcmasterrace            bitcoin      tipofmypenis
##                565                500                487               485
##           dogecoin
##                484
```

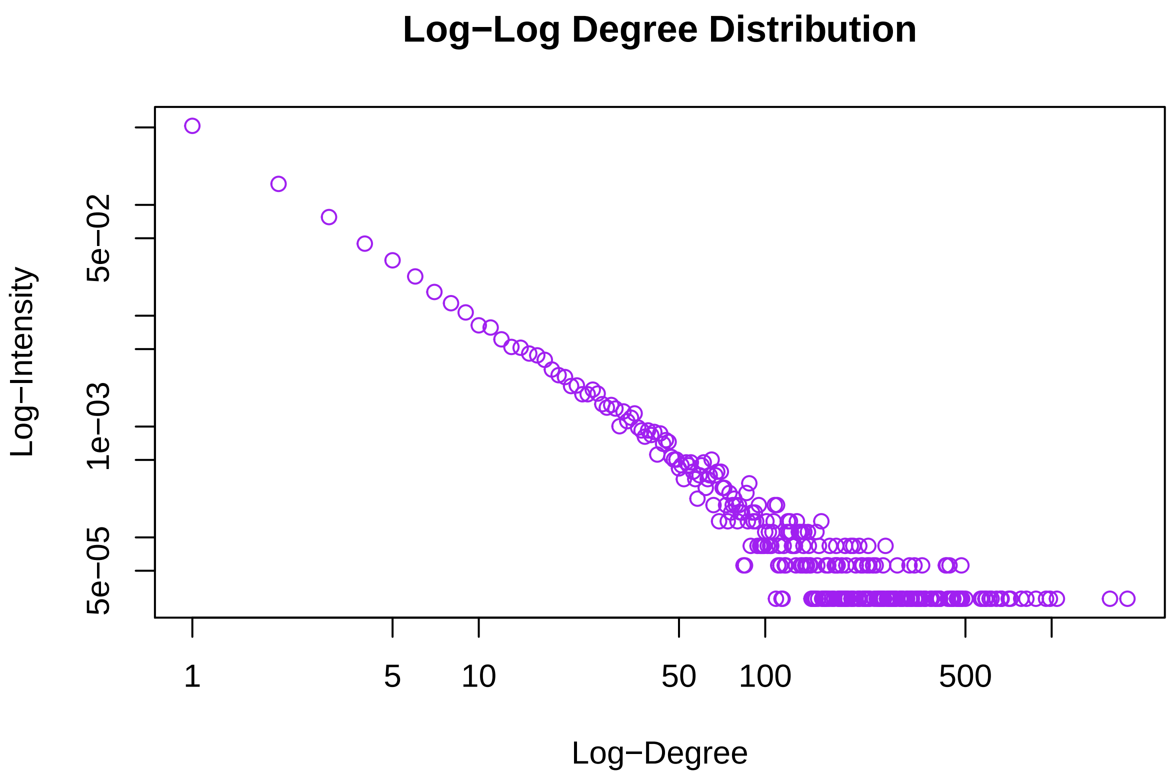
```r
hist(graph.strength(g.rh.weighted), col="red", xlab="Vertex Strength",
     ylab="Frequency",
     xlim = c(0,200), breaks = seq(0,10000,by=10))
```



```
hist(degree(g.rh.weighted), col="blue", xlab="Vertex Degree",
     ylab="Frequency", xlim = c(0,200), breaks = seq(0,10000,by=10))
```



```
d.rh <- degree(g.rh.weighted)
dd.rh <- degree.distribution(g.rh.weighted)
d <- 1:max(d.rh)-1
ind <- (dd.rh != 0)
plot(d[ind], dd.rh[ind], log="xy", col="purple", xlab=c("Log-Degree"), ylab=c("Log-Intensity"), main="Lo
```



Component Analysis

```
is.connected(g.rh.weighted)
```

```
## [1] FALSE
```

```
parts<-decompose.graph(g.rh.weighted)
table(sapply(parts, vcount))
```

```
##
##      2      3      4      5      8      9     42  34671
##    450     33      3      7      1      1      1      1
```

There is clearly the presence of a giant component in this data.

```
vcount(g.rh.weighted)
```

```
## [1] 35776
```

```
g.rh.weighted<-decompose.graph(g.rh.weighted)[[1]]
vcount(g.rh.weighted)
```

```
## [1] 34671
```

```
is.connected(g.rh.weighted)
```

```
## [1] TRUE
```

```
g.rh.weighted<-delete_vertices(g.rh.weighted, degree(g.rh.weighted) <= 100)
```

```r
average.path.length(g.rh.weighted)
```

```
## [1] 1.80745
```

With such a small average path length,

Hierarchical Clustering and Graph Partitioning

```r
greedy_rh<-fastgreedy.community(g.rh.weighted)
sizes(greedy_rh)
```

```
## Community sizes
##   1   2   3   4   5   6   7
##  10   9 137  13  37 142   2
```

```r
plot(greedy_rh,g.rh.weighted, asp = 0, vertex.size = 5, vertex.label.cex = 0.8, vertex.label=NA)
```