

HW2-2(h)

Feiran Ji

4/23/2018

```
library(igraph, quietly = TRUE)

##
## Attaching package: 'igraph'
## The following objects are masked from 'package:stats':
##       decompose, spectrum
## The following object is masked from 'package:base':
##       union
library(statnet, quietly = TRUE)

##
## Attaching package: 'statnet.common'
## The following object is masked from 'package:base':
##       order
## network: Classes for Relational Data
## Version 1.13.0.1 created on 2015-08-31.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##                         Mark S. Handcock, University of California -- Los Angeles
##                         David R. Hunter, Penn State University
##                         Martina Morris, University of Washington
##                         Skye Bender-deMoll, University of Washington
## For citation information, type citation("network").
## Type help("network-package") to get started.

##
## 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

##
## ergm: version 3.8.0, created on 2017-08-18
## Copyright (c) 2017, Mark S. Handcock, University of California -- Los Angeles
##                         David R. Hunter, Penn State University
##                         Carter T. Butts, University of California -- Irvine
##                         Steven M. Goodreau, University of Washington
##                         Pavel N. Krivitsky, University of Wollongong
##                         Martina Morris, University of Washington
```

```

##           with contributions from
##           Li Wang
##           Kirk Li, University of Washington
##           Skye Bender-deMoll, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("ergm").

## NOTE: Versions before 3.6.1 had a bug in the implementation of the
## bd() constraint which distorted the sampled distribution somewhat.
## In addition, Sampson's Monks datasets had mislabeled vertices. See
## the NEWS and the documentation for more details.

##
## networkDynamic: version 0.9.0, created on 2016-01-12
## Copyright (c) 2016, Carter T. Butts, University of California -- Irvine
##           Ayn Leslie-Cook, University of Washington
##           Pavel N. Krivitsky, University of Wollongong
##           Skye Bender-deMoll, University of Washington
##           with contributions from
##           Zack Almquist, University of California -- Irvine
##           David R. Hunter, Penn State University
##           Li Wang
##           Kirk Li, University of Washington
##           Steven M. Goodreau, University of Washington
##           Jeffrey Horner
##           Martina Morris, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("networkDynamic").

##
## tergm: version 3.4.1, created on 2017-09-12
## Copyright (c) 2017, Pavel N. Krivitsky, University of Wollongong
##           Mark S. Handcock, University of California -- Los Angeles
##           with contributions from
##           David R. Hunter, Penn State University
##           Steven M. Goodreau, University of Washington
##           Martina Morris, University of Washington
##           Nicole Bohme Carnegie, New York University
##           Carter T. Butts, University of California -- Irvine
##           Ayn Leslie-Cook, University of Washington
##           Skye Bender-deMoll
##           Li Wang
##           Kirk Li, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("tergm").

##
## ergm.count: version 3.2.2, created on 2016-03-29
## Copyright (c) 2016, Pavel N. Krivitsky, University of Wollongong
##           with contributions from
##           Mark S. Handcock, University of California -- Los Angeles
##           David R. Hunter, Penn State University
## Based on "statnet" project software (statnet.org).

```

```

## For license and citation information see statnet.org/attribution
## or type citation("ergm.count").

## NOTE: The form of the term 'CMP' has been changed in version 3.2
## of 'ergm.count'. See the news or help('CMP') for more information.

## sna: Tools for Social Network Analysis
## Version 2.4 created on 2016-07-23.
## 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

##
## statnet: version 2016.9, created on 2016-08-29
## Copyright (c) 2016, Mark S. Handcock, University of California -- Los Angeles
##                         David R. Hunter, Penn State University
##                         Carter T. Butts, University of California -- Irvine
##                         Steven M. Goodreau, University of Washington
##                         Pavel N. Krivitsky, University of Wollongong
##                         Skye Bender-deMoll
##                         Martina Morris, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("statnet").

## unable to reach CRAN

library(Matrix, quietly = TRUE)

# download dataset
link <- "https://snap.stanford.edu/data/facebook_combined.txt.gz"
download.file(link, destfile = "web-fb.txt.gz")
fb.data <- read.table("web-fb.txt.gz", sep=" ", stringsAsFactors = FALSE)

# create igraph from edgelist
fb.igraph <- simplify(graph.data.frame(fb.data, directed = FALSE))
# adjacency matrix from igraph
fb.adjacency <- as_adj(fb.igraph)
fb.edgelist <- as.matrix(fb.data) + 1
fb.network <- network(fb.edgelist)

```

Shortest Paths

First, we will identify the shortest paths are between each pair of nodes.

```
shortest.distances <- distances(fb.igraph, mode = "all")
```

```

#network diameter: the longest shortest path that is not infinite
max(shortest.distances[which(shortest.distances < Inf)])
```

```

## [1] 8
```

```

#what is the max shortest path?
max(shortest.distances)
```

```

## [1] 8
```

```

#infinite tells us that there are isolates!
```

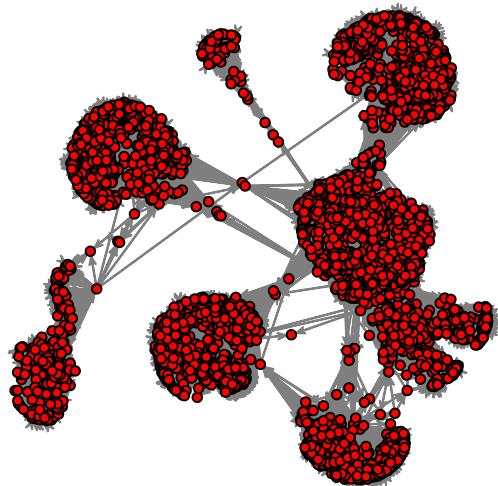
There is no isolate account in this network.

Network Centrality Measures and Visualization

Plot the data

```
fb.x <- plot(fb.network, main=paste("Facebook Network"), edge.col = "grey50")
```

Facebook Network



Calculate In-Degree, Out-Degree, Eigenvector, Betweenness, and Closeness Centralities

```

#in degree centrality
in.degree.centrality <- colSums(as.matrix(fb.adjacency))

#out degree centrality
out.degree.centrality <- rowSums(as.matrix(fb.adjacency))

#eigenvector centrality
eigenvector.centrality <- eigen_centrality(fb.igraph, directed = FALSE)$vector

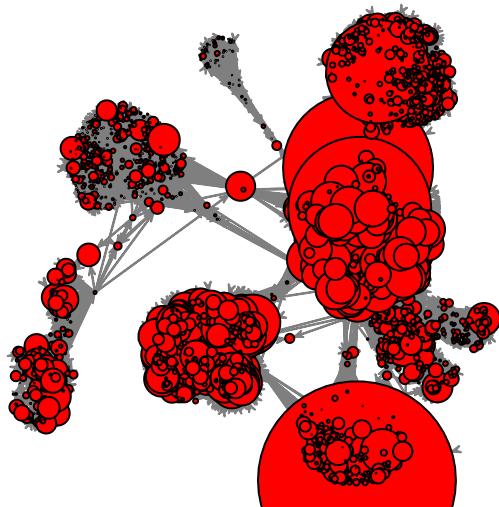
#normalized betweenness centrality
betweenness.centrality <- estimate_betweenness(fb.igraph, directed = TRUE, cutoff = 10)
```

```
#closeness centrality  
closeness.centrality <- estimate_closeness(fb.igraph, mode = "total", normalized = FALSE, cutoff = 10)
```

Plot Network According to Centrality Measures

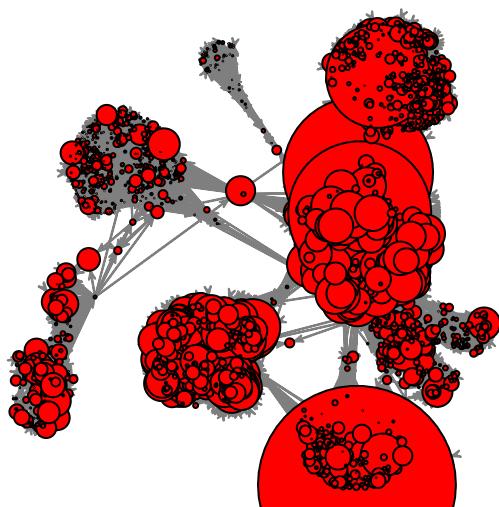
```
#plot in-degree  
plot(fb.network, main = paste("In-Degree Centrality"), usearrows = TRUE, vertex.cex = in.degree.centrality)
```

In-Degree Centrality



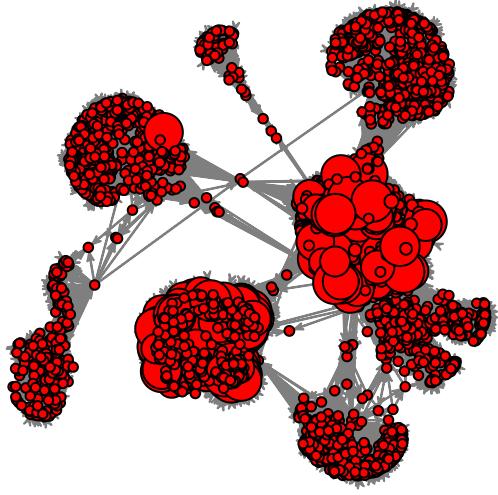
```
#plot out-degree  
plot(fb.network, main = paste("Out-Degree Centrality"), usearrows = TRUE, vertex.cex = out.degree.centrality)
```

Out-Degree Centrality



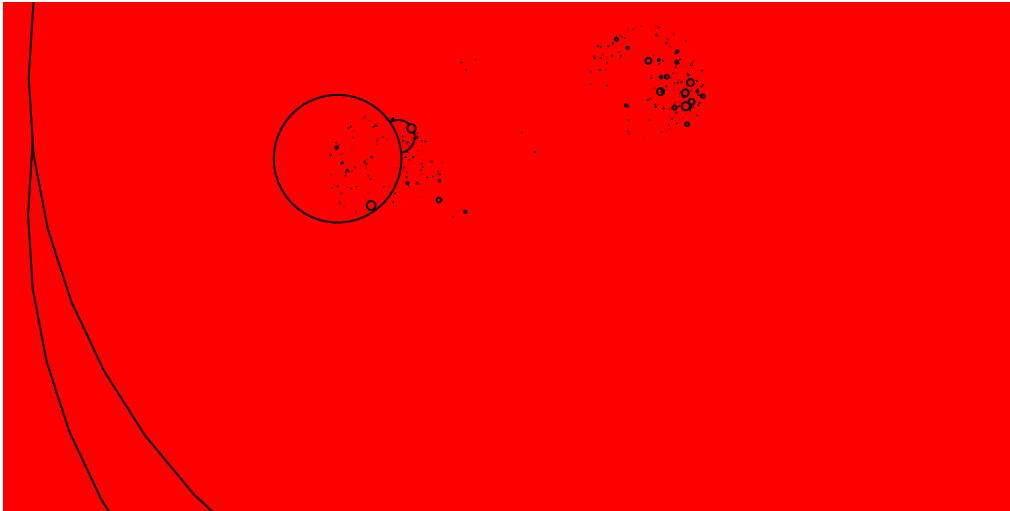
```
#plot eigenvector  
plot(fb.network, main = paste("Eigenvector Centrality"), usearrows = TRUE, vertex.cex = eigenvector.centrality)
```

Eigenvector Centrality



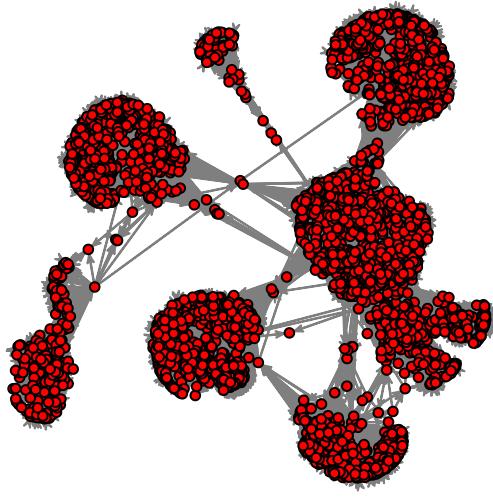
```
#plot betweenness
plot(fb.network, main = paste("Betweenness Centrality"), usearrows = TRUE, vertex.cex = betweenness.cen
```

Betweenness Centrality



```
#plot closeness
plot(fb.network, main = paste("Closeness Centrality"), usearrows = TRUE, vertex.cex = closeness.central
```

Closeness Centrality



Community Detection

We ran 5 different community detection algorithms on Facebook network and compare the results that we get from each method.

```
fb.igraph2 = simplify(fb.igraph)
#Running 5 different community detection methods and compare
f_g <- cluster_fast_greedy(fb.igraph2)
info.clusters <- cluster_infomap(fb.igraph2)
l_p <- cluster_label_prop(fb.igraph2)
louvain <- cluster_louvain(fb.igraph2)
walktrap <- cluster_walktrap(fb.igraph2)

#size of communities for each method
table(f_g$membership)

##
##   1   2   3   4   5   6   7   8   9   10  11  12  13
## 816 982 543 372 548 206 208 59  37  25  18  219 6
table(info.clusters$membership)

##
##   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18
## 426 322 282 236 193 192 186 165 148 157 131 145 93  97  73  57  64  52
## 19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
## 48  38  60  38  39  25  36  31  21  29  30  21  22  33  29  19  19  22
## 37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
## 19  23  16  19  14  22  13  15  16  12  10  10  16  12  10  15  11  9
## 55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
## 10  13  10  8   8   9   10  8   7   7   6   6   7   4   6   6   5   6
## 73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
## 4   6   4   4   3   3   3   3   3   3   3   3   2   2   2   2   2   2
## 91  92  93  94
## 2   2   2   2
```

```



```

