

Chapter 3

Visualizing Network Data

Statistical Analysis of Network Data, with R - Eric D. Kolaczyk

Thu Nguyen

28 June, 2019

Contents

| | | |
|----------|----------------------------------------|----------|
| 1 | Introduction | 1 |
| 2 | Elements of Graph Visualization | 1 |
| 3 | Graph Layouts | 2 |
| 4 | Decorating Graph Layouts | 4 |
| 5 | Visualizing Large Networks | 6 |

Libraries

```
library(igraph)
library(igraphdata)
library(sand)
```

1 Introduction

2 Elements of Graph Visualization

3 Graph Layouts

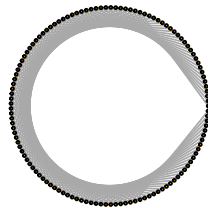
```
g.l <- graph.lattice(c(5,5,5)) # Lattice graph
# data(aidsblog)
summary(aidsblog)
```

```
## IGRAPH NA D--- 146 187 --
```

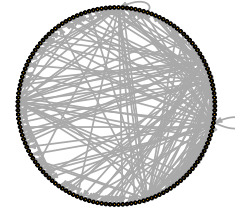
Circular layout: `layout=layout.circle`:

```
igraph.options(vertex.size=3, vertex.label=NA, edge.arrow.size=0.2)
par(mfrow=c(1,2)); par(mar=c(0,0,1,0))
plot(g.l, layout=layout.circle); title("5x5x5 Lattice")
plot(aidsblog, layout=layout.circle); title("Blog Network")
```

5x5x5 Lattice



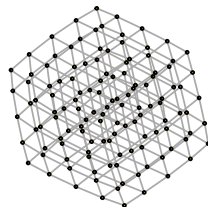
Blog Network



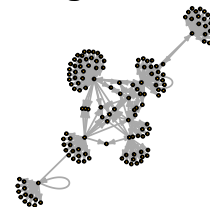
Spring-Embedder layout: `layout=layout.fruchterman.reingold`:

```
par(mfrow=c(1,2)); par(mar=c(0,0,1,0))
plot(g.l, layout=layout.fruchterman.reingold); title("5x5x5 Lattice")
plot(aidsblog, layout=layout.fruchterman.reingold); title("Blog Network")
```

5x5x5 Lattice



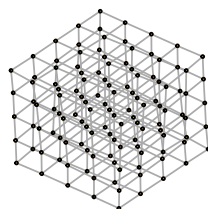
Blog Network



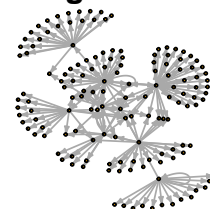
Energy-Placement layout: `layout=layout.kamada.kawai`:

```
par(mfrow=c(1,2)); par(mar=c(0,0,1,0))
plot(g.l, layout=layout.kamada.kawai); title("5x5x5 Lattice")
plot(aidsblog, layout=layout.kamada.kawai); title("Blog Network")
```

5x5x5 Lattice



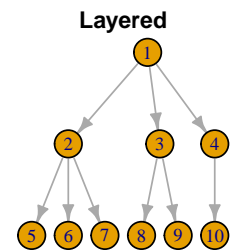
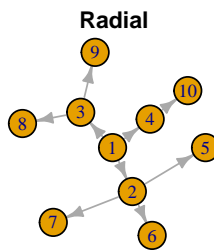
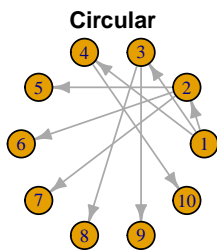
Blog Network



Trees:

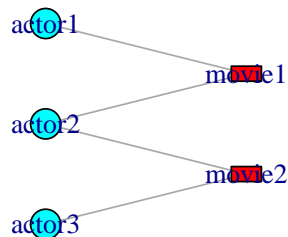
- *circular*: `layout=layout.circle`
- *radial*: `layout=layout.reingold.tilford(circular=T)`
- *layered*: `layout=layout.reingold.tilford`

```
g.tree <- graph.formula(1--2,1--3,1--4,2--5,2--6,2--7,3--8,3--9,4--10)
par(mfrow=c(1, 3)); par(mar=c(0,0,1,0))
igraph.options(vertex.size=30, edge.arrow.size=0.5, vertex.label=NULL)
plot(g.tree, layout=layout.circle); title("Circular")
plot(g.tree, layout=layout.reingold.tilford(g.tree, circular=T)); title("Radial")
plot(g.tree, layout=layout.reingold.tilford); title("Layered")
```



Bipartite graphs: `layout=layout.bipartite`

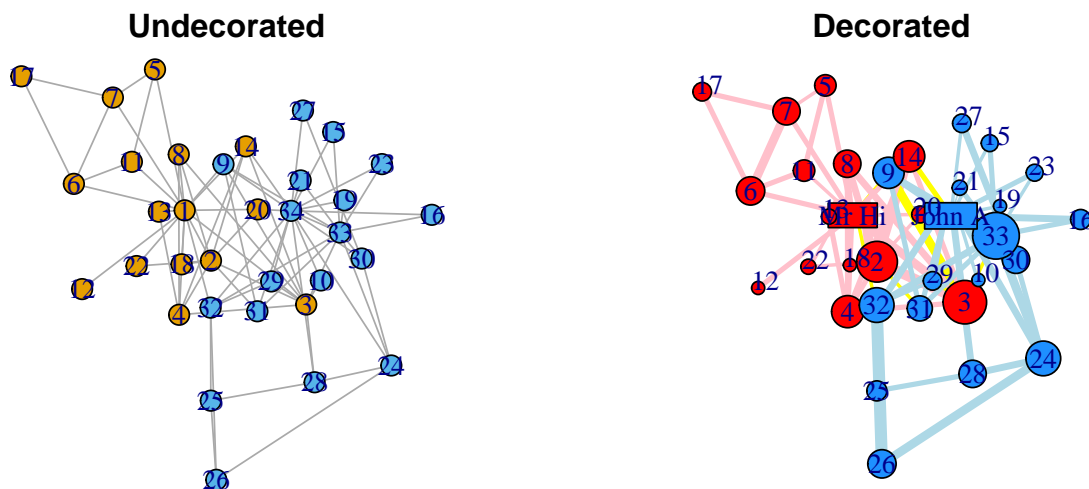
```
par(mar=c(0,0,0,0))
plot(g.bip, layout=layout.bipartite(g.bip)[,2:1],
     vertex.size=30, vertex.shape=ifelse(V(g.bip)$type, "rectangle", "circle"),
     vertex.color=ifelse(V(g.bip)$type, "red", "cyan"))
```



4 Decorating Graph Layouts

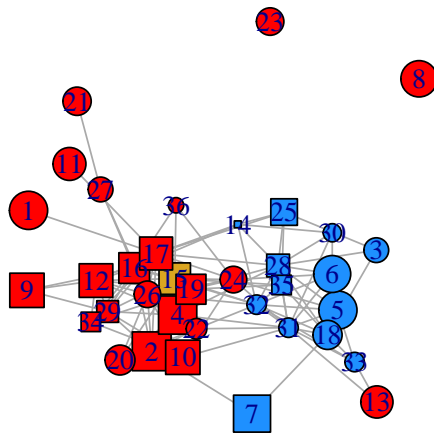
Karate dataset

```
library(igraphdata)
data(karate)
# Reproducible layout
set.seed(42)
l <- layout.kamada.kawai(karate)
# Plot undecorated first.
igraph.options(vertex.size=10)
par(mfrow=c(1,2)); par(mar=c(0,0,1,0))
plot(karate, layout=l, vertex.label=V(karate)); title('Undecorated')
# Now decorate, starting with labels.
V(karate)$label <- sub("Actor ", "", V(karate)$name)
# Two leaders get shapes different from club members.
V(karate)$shape <- "circle"
V(karate)[c("Mr Hi", "John A")]$shape <- "rectangle"
# Differentiate two factions by color.
V(karate)[Faction == 1]$color <- "red"
V(karate)[Faction == 2]$color <- "dodgerblue"
# Vertex area proportional to vertex strength
# (i.e., total weight of incident edges).
V(karate)$size <- 4*sqrt(graph.strength(karate))
V(karate)$size2 <- V(karate)$size * .5
# Weight edges by number of common activities
E(karate)$width <- E(karate)$weight
# Color edges by within/between faction.
F1 <- V(karate)[Faction==1]
F2 <- V(karate)[Faction==2]
E(karate)[ F1 %--% F1 ]$color <- "pink"
E(karate)[ F2 %--% F2 ]$color <- "lightblue"
E(karate)[ F1 %--% F2 ]$color <- "yellow"
# Offset vertex labels for smaller points (default=0).
V(karate)$label.dist <- ifelse(V(karate)$size >= 10, 0, 0.75)
# Plot decorated graph, using same layout.
plot(karate, layout=l); title('Decorated')
```



Lazega dataset

```
library(sand)
data(lazega)
par(mar=c(0,0,1,0))
# Office location indicated by color.
colbar <- c("red", "dodgerblue", "goldenrod")
v.colors <- colbar[V(lazega)$Office]
# Type of practice indicated by vertex shape.
v.shapes <- c("circle", "square") [V(lazega)$Practice]
# Vertex size proportional to years with firm.
v.size <- 3.5*sqrt(V(lazega)$Years)
# Label vertices according to seniority.
v.label <- V(lazega)$Seniority
# Reproducible layout.
set.seed(42)
l <- layout_fruchterman_reingold(lazega)
plot(lazega, layout=l, vertex.color=v.colors,
     vertex.shape=v.shapes, vertex.size=v.size,
     vertex.label=v.label)
```



5 Visualizing Large Networks

fblog dataset

```
# summary(fblog)
party.names <- sort(unique(V(fblog)$PolParty))
party.names

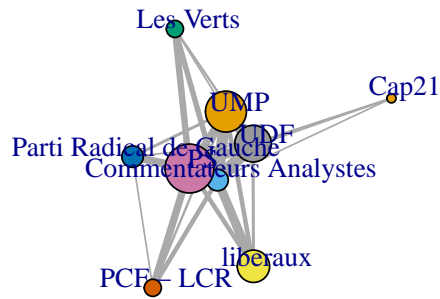
## [1] " Cap21"                " Commentateurs Analystes"
## [3] " Les Verts"            " liberaux"
## [5] " Parti Radical de Gauche" " PCF - LCR"
## [7] " PS"                   " UDF"
## [9] " UMP"
```

```
par(mfrow=c(1,2)); par(mar=c(0,0,1,0))
# Layout: Kamada-Kawai
set.seed(42)
l = layout.kamada.kawai(fblog)
party.num.f <- as.factor(V(fblog)$PolParty)
party.num.s <- as.numeric(party.num.f)
plot(fblog, layout=l, vertex.label=NA, vertex.color=party.num.s, vertex.size=3)
title('Kamada-Kawai')
# Layout: DrL
set.seed(42)
l <- layout.drl(fblog)
plot(fblog, layout=l, vertex.size=5, vertex.label=NA, vertex.color=party.num.s)
title('DrL')
```



Contracting group of vertices into meta-vertices:

```
par(mar=c(0,0,0,0))
fblog.c <- contract.vertices(fblog, party.num.s)
E(fblog.c)$weight <- 1
fblog.c <- simplify(fblog.c)
party.size <- as.vector(table(V(fblog)$PolParty))
plot(fblog.c, vertex.size=5*sqrt(party.size),
     vertex.label=party.names,
     vertex.color=V(fblog.c),
     edge.width=sqrt(E(fblog.c)$weight),
     vertex.label.dist=1.5, edge.arrow.size=0)
```



Further look into `karate` dataset: (1) Extracting neighbors:

```
data(karate)
k.nbhds <- graph.neighborhood(karate, order=1)
sapply(k.nbhds, vcount)
```

```
## [1] 17 10 11 7 4 5 5 5 6 3 4 2 3 6 3 3 3 3 3 4 3 3 3
## [24] 6 4 4 3 5 4 5 5 7 13 18
```

(2) Pulling out 2 largest subnetworks and plotting them

```
par(mfrow=c(1,2)); par(mar=c(0,0,1,0))
k.1 <- k.nbhds[[1]]
k.34 <- k.nbhds[[34]]
plot(k.1, vertex.label=NA,
     vertex.color=c("red", rep("lightblue", 16)))
plot(k.34, vertex.label=NA,
     vertex.color=c(rep("lightblue", 17), "red"))
```

