Chapter 5
Mathematical Models for Network Graphs

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

Thu Nguyen

24 June, 2019

Contents

Libraries

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

---

1 Introduction

---

2 Classical Random Graph Models

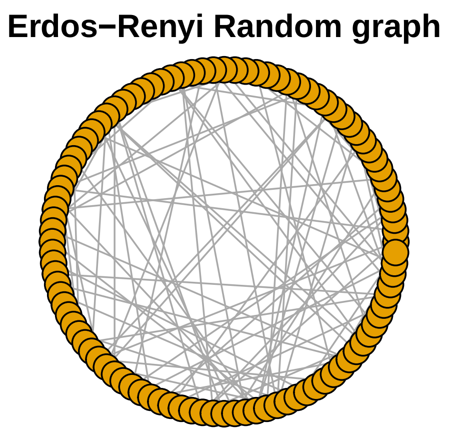
Random graph model refers to model specifying a collection $\mathcal{G}$ and a **uniform probability** $\mathbb{P}$ over $\mathcal{G}$.

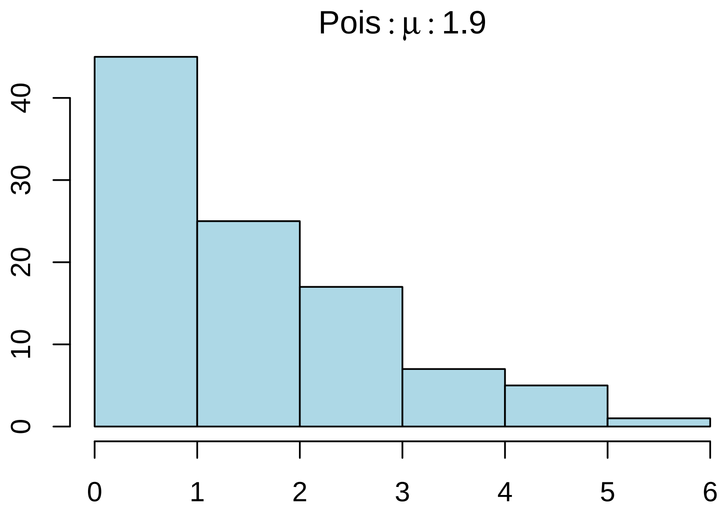
The classical example is **Erdos-Renyi** model: given graph $G = (V, E)$ with n vertices and m edges, with $N = \binom{n}{2}$:

$$\mathbb{P}\left(G\right) = \binom{N}{m}^{-1}$$

A variant of $\mathcal{G}_{n,m}$ is a collection $\mathcal{G}_{n,p}$ with fixed n vertices, and probability $p \in (0, 1)$ for every edge between each pair of vertices *independently*. The model's degree distribution is approximately *Poisson*. In `R`: `erdos.renyi.game()`:

```r
par(mfrow=c(1,2)); par(mar=c(2,0,1,0))
set.seed(42)
g.er <- erdos.renyi.game(100, .02)
plot(g.er, layout = layout.circle, vertex.label = NA, main = 'Erdos-Renyi Random graph')
hist(degree(g.er), col = 'lightblue', xlab = 'Degree', ylab = 'Frequency',
     main = expression(Pois: mu: 1.9))
```





Further look: G is not connected, with 15 isolated vertices and a giant component of 71 vertices:

```r
table(sapply(decompose.graph(g.er), vcount))
```

```
##
##  1  2  3  4 71
## 15  2  2  1  1
```

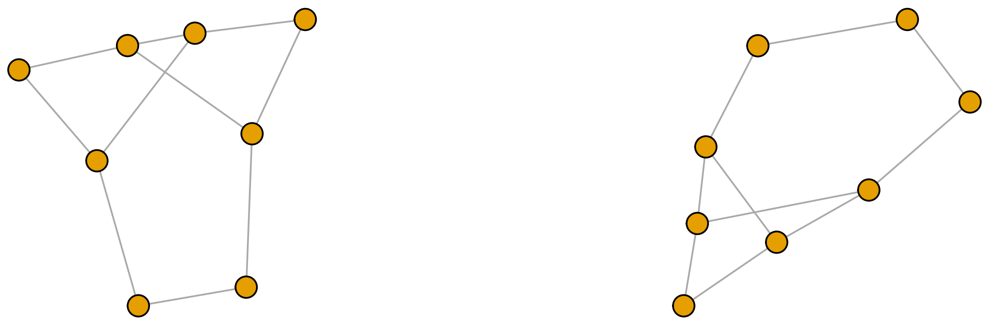
```r
apl <- round(average.path.length(g.er),2)
diam <- diameter(g.er)
trans <- round(transitivity(g.er),2)
print(paste0('Average path length: ', apl, '; Diameter: ', diam, '; Transitivity: ', trans))
```

```
## [1] "Average path length: 5.28; Diameter: 14; Transitivity: 0.02"
```

3 Generalized Random Graph Models

The **Erdos-Renyi** model can be generalized into a collection $\mathcal{G}$ with fixed n vertices and a given characteristics, and a **uniform distribution** for all graph G over $\mathcal{G}$. While **Erdos-Renyi** fixes m edges, the 2^{nd} characteristic is an *ordered* pre-specified degree sequence $\{d_1, d_2, \ldots, d_n\}$. In `R`: `degree.sequence.game()`:

```
par(mfrow=c(1,2)); par(mar=c(0,0,0,0))
degs <- c(2,2,2,2,3,3,3,3)
g1 <- degree.sequence.game(degs, method = 'vl'); g2 <- degree.sequence.game(degs, method = 'vl')
plot(g1, vertex.label = NA); plot(g2, vertex.label = NA)
```



```
iso <- graph.isomorphic(g1, g2); print(paste0('Are they isomorphic? ', iso))
```

```
## [1] "Are they isomorphic? FALSE"
```

```
print(paste0('Edge counts: graph 1: ', ecount(g1), '; graph 2: ', ecount(g2)))
```

```
## [1] "Edge counts: graph 1: 10; graph 2: 10"
```

For example, taking the sequence from the `yeast` dataset:

```
data(yeast)
degs <- degree(yeast)
fake.yeast <- degree.sequence.game(degs, method = 'vl')
print(paste0('Same degree sequence? ', all(degree(yeast) == degree(fake.yeast))))
```

```
## [1] "Same degree sequence? TRUE"
```

```
diam <- round(c(diameter(yeast), diameter(fake.yeast)),0)
trans <- round(c(transitivity(yeast), transitivity(fake.yeast)),4)
data.frame(data = c('Original Yeast', 'New Yeast'),
           diameter = diam, transitivity = trans)
```

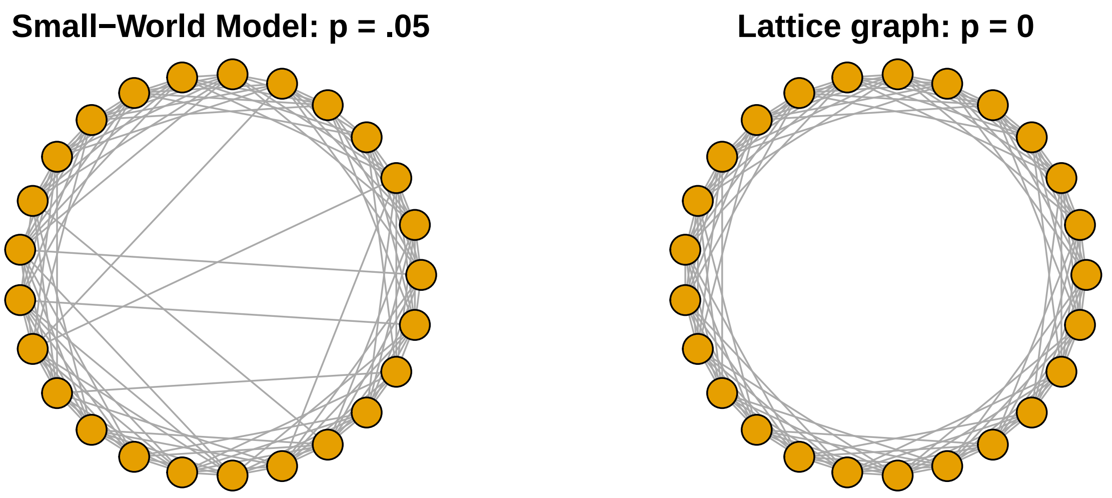
```
##               data diameter transitivity
## 1 Original Yeast       15       0.4686
## 2      New Yeast        8       0.0403
```

4 Network Graph Models Based on Machanisms

4.1 Small-World Models

Small-World models: high clustering but small distances between most nodes, which cannot be produced by *classical random graphs*. In R, `watts.strogatz.game()`:

```
par(mfrow=c(1,2)); par(mar=c(0,0,1,0))
g.ws <- watts.strogatz.game(1, 25, 5, .05)
g.lat100 <- watts.strogatz.game(1, 25, 5, 0)
plot(g.ws, layout = layout.circle, vertex.label = NA, main = 'Small-World Model: p = .05')
plot(g.lat100, layout = layout.circle, vertex.label = NA, main = 'Lattice graph: p = 0')
```

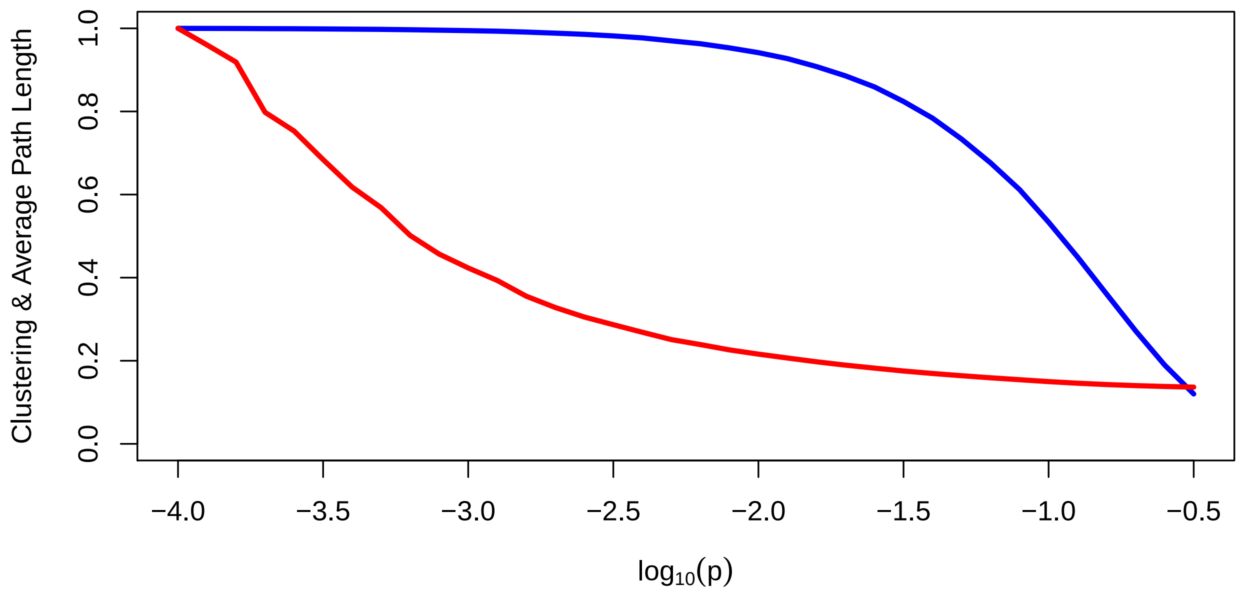


```
apl <- round(c(average.path.length(g.ws), average.path.length(g.lat100)),2)
diam <- round(c(diameter(g.ws), diameter(g.lat100)),0)
trans <- round(c(transitivity(g.ws), transitivity(g.lat100)),4)
data.frame(data = c('Small-World Model', 'Lattice'),
           ave.path = apl, diameter = diam, transitivity = trans)
```

```
##                data ave.path diameter transitivity
## 1 Small-World Model     1.63        3       0.5544
## 2           Lattice     1.75        3       0.6667
```

Simulation of the **Small-world models** as p varies:

```r
steps <- seq(-4, -.5, .1)
len <- length(steps)
cl <- numeric(len)
apl <- numeric(len)
ntrials <- 100
for (i in 1:len) {
  cltemp <- numeric(ntrials)
  apltemp <- numeric(ntrials)
  for (j in 1:ntrials) {
    g <- watts.strogatz.game(1, 1000, 10, 10^steps[i])
    cltemp[j] <- transitivity(g)
    apltemp[j] <- average.path.length(g)
  }
  cl[i] <- mean(cltemp)
  apl[i] <- mean(apltemp)
}
par(mar=c(4,5,2.5,2))
plot(steps, cl/max(cl), ylim = c(0,1), lwd = 3, type = 'l', col = 'blue',
     xlab = expression(log[10](p)), ylab = 'Clustering & Average Path Length')
lines(steps, apl/max(apl), lwd = 3, col = 'red')
```



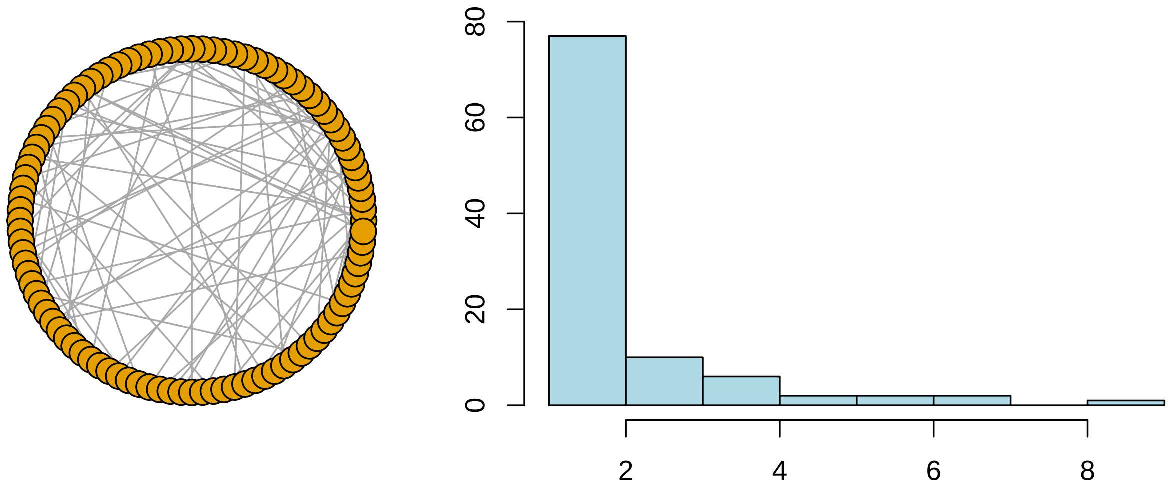
4.2 Preferential Attachment Models

Preferential attachment: models the principle 'the rich get richer', where the probability of the new vertex will be conneced to a given vertex v is proportional to the existing degree of v:

$$p = \frac{d_v}{\sum_{v' \in V} d_{v'}}$$

At each stage, k existing vertices are connected to a new vertex preferential to those with higher degrees. After t iterations, the resulting G^t will have $N_v^t = n + t$ vertices and $N_e^t = m + tk$ edges. In R, barabasi.game():

```
par(mfrow=c(1,2)); par(mar=c(2,0,1,0))
set.seed(42)
g.ba <- barabasi.game(100, directed = FALSE)
plot(g.ba, layout = layout.circle, vertex.label = NA)
hist(degree(g.ba), col = 'lightblue', xlab = 'Degree', ylab = 'Frequency', main = '')
```



Remark: as $t \to \infty$, the graphs G^t have degree distributions that tend to a power-law form $d^{-\alpha}$, with $\alpha = 3$.

```
data.frame(Attributes = c('Ave.Path.Length', 'Diameter', 'Transitivity'),
           Values = c(average.path.length(g.ba), diameter(g.ba), transitivity(g.ba)))
```

```
##        Attributes    Values
## 1 Ave.Path.Length  5.815556
## 2        Diameter 12.000000
## 3    Transitivity  0.000000
```

5 Assessing Significance of Network Graph Characteristics

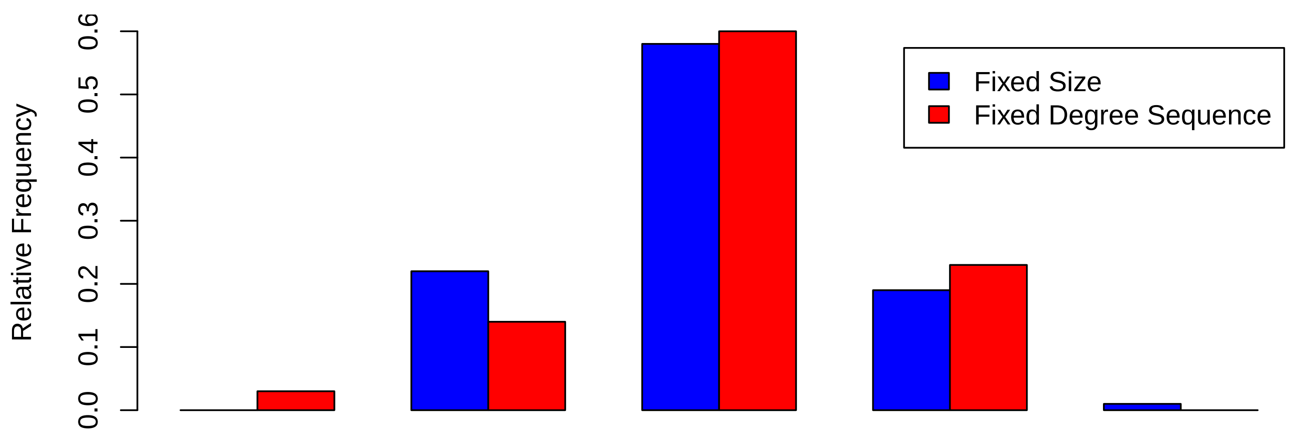
- Question of interest: **is $\eta\left(G^{obs}\right)$ significant?** where $\eta(\cdot)$ is some structural characteristic, under a formal reference distribution, assuming *uniform distribution* of elements in $\mathcal{G}$:

$$\mathbb{P}_{\eta,\mathcal{G}}\left(t\right)=\frac{\#\{G\in\mathcal{G}:\eta(G)\leq t\}}{|\mathcal{G}|}$$

5.1 Assessing the Number of Communities in a Network

Using *Monte Carlo simulation*, with the n vertices, m edges, and degree sequence from `karate` dataset:

```r
data(karate)
# Observed graph statistics
nv <- vcount(karate); ne <- ecount(karate); degs <- degree(karate)
# Fix #(vertices) and #(edges)
num.comm.rg <- numeric(ntrials)
for (i in 1:ntrials) {
  g.rg <- erdos.renyi.game(nv, ne, type = 'gnm')
  c.rg <- fastgreedy.community(g.rg)
  num.comm.rg[i] <- length(c.rg)
}
# Fix degree sequence
num.comm.grg <- numeric(ntrials)
for (i in 1:ntrials) {
  g.grg <- degree.sequence.game(degs, method = 'vl')
  c.grg <- fastgreedy.community(g.grg)
  num.comm.grg[i] <- length(c.grg)
}
par(mar=c(.5,5,.5,0))
results <- c(num.comm.rg, num.comm.grg)
idx <- c(rep(0, ntrials), rep(1, ntrials))
counts <- table(idx, results)/ntrials
barplot(counts, beside = TRUE, col = c('blue', 'red'),
        xlab = 'Number of Communities', ylab = 'Relative Frequency',
        legend = c('Fixed Size', 'Fixed Degree Sequence'))
```



5.2 Assessing Small World Properties

Given a *directed* graph G with adjacency matrix $\mathbf{A}$, and total degree d_v^{tot} of vertex v (in-degree + out-degree), the *clustering coefficient* is:

$$cl(v) = \frac{(\mathbf{A} + \mathbf{A}^T)^3_{vv}}{2\ [\ d_v^{tot}\ (d_v^{tot} - 1) - 2(\mathbf{A}^2)_{vv}\ ]}$$

For example, dataset `macaque` from package `igraphdata`:

```r
library(igraphdata)
data(macaque)
summary(macaque)
```

```
## IGRAPH f7130f3 DN-- 45 463 --
## + attr: Citation (g/c), Author (g/c), shape (v/c), name (v/c)
```

```r
# function to calculate clustering coefficient for directed graph:
clust.coef.dir <- function(graph) {
  A <- as.matrix(get.adjacency(graph))
  S <- A + t(A)
  deg <- degree(graph, mode = 'total')
  numerator <- diag( S %*% S %*% S )
  denom <- diag(A %*% A)
  denominator <- 2 * (deg * (deg-1) - 2*denom)
  cl <- mean(numerator/denominator)
  return(cl)
}

ntrials <- 1000
nv <- vcount(macaque); ne <- ecount(macaque)
cl.rg <- numeric(ntrials); apl.rg <- numeric(ntrials)
for (i in 1:ntrials) {
  g.rg <- erdos.renyi.game(nv, ne, type = 'gnm', directed = TRUE)
  cl.rg[i] <- clust.coef.dir(g.rg)
  apl.rg[i] <- average.path.length(g.rg)
}
```

```r
obs.cl <- clust.coef.dir(macaque); obs.apl <- average.path.length(macaque)
p.cl <- sum(cl.rg <= obs.cl)/ntrials; p.apl <- sum(apl.rg <= obs.apl)/ntrials
data.frame(Attributes = c('Clustering coefficient', 'Average Path Length'),
           p.values = c(p.cl, p.apl))
```

```
##                Attributes p.values
## 1 Clustering coefficient        1
## 2    Average Path Length        1
```

Interpretation of result: `macaque` dataset has substantially more clustering, and longer shortest paths between vertex pairs than a random network, indicating that the small-world behavior is not clear in this dataset.