

Foundations of Data Science II: Statistical Network Analysis II

Isabella Gollini
`isabella.gollini@ucd.ie`

December 7th 2023



University College Dublin
Ireland's Global University

What will we do today?

- 1 Networks as Random Graphs - The Erdős-Rényi Model
- 2 Exponential Random Graph Models
- 3 Latent Space Models

Networks as Random Graphs - The Erdős-Rényi Model

Network Structures and Adjacency Matrix

Networks are equivalent to graphs consisting of:

- a set of n nodes (or vertices);
- a set of edges (or links or ties) connecting nodes.

An **adjacency matrix** is a way of representing a graph as a square $(0, 1)$ -matrix \mathbf{Y} encoding the presence or absence of an edge between two nodes.

- Non-diagonal elements of the adjacency matrix are defined as:

$$Y_{ij} = \begin{cases} 1, & \text{if } i \sim j \\ 0, & \text{if } i \not\sim j \end{cases}$$

- Diagonal elements $Y_{ii} = 0$ representing self-loops are not generally allowed.

Undirected Networks

In undirected networks edges are non-directional: $Y_{ij} = Y_{ji}$.

An undirected network can be represented by a symmetric adjacency matrix.

- The number of edge variables is $m = \binom{n}{2} = \frac{n(n-1)}{2}$
- The degree of a node i is $\deg(i) = \sum_j y_{ij}$

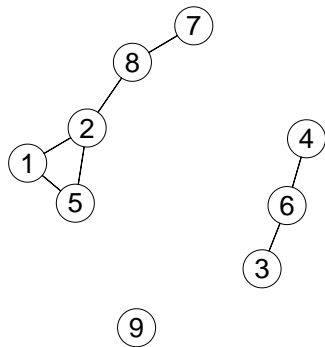
Example

```
library('statnet')  
n <- 9  
set.seed(1)  
y <- network(n, directed = FALSE)  
y[,]
```

```
##    1 2 3 4 5 6 7 8 9  
## 1 0 1 0 0 1 0 0 0  
## 2 1 0 0 0 1 0 0 1  
## 3 0 0 0 0 0 1 0 0  
## 4 0 0 0 0 0 1 0 0  
## 5 1 1 0 0 0 0 0 0  
## 6 0 0 1 1 0 0 0 0  
## 7 0 0 0 0 0 0 0 1  
## 8 0 1 0 0 0 0 1 0  
## 9 0 0 0 0 0 0 0 0
```

Example - An undirected network graph y

```
plot(y,  
  vertex.cex = 6,  
  vertex.col = 0,  
  label = 1:n,  
  label.pos = 5,  
  label.cex = 2)
```

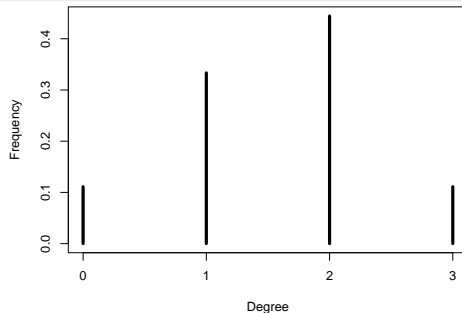


Degree distributio

Degree distribution

The degree distribution $\text{deg}(k)$ of a network is then defined to be the proportion of nodes in the network with degree k for $k = 0, 1, 2, \dots, n - 1$.

```
deg_dist <- table(degree(y, gmode = "graph"))  
plot(deg_dist / sum(deg_dist),  
      ylab = 'Frequency',  
      xlab = 'Degree',  
      lwd = 4)
```



Directed Networks

In directed networks edges are directional and can be represented by an asymmetric adjacency matrix.

- The number of edge variables is $m = n(n - 1)$
- The in-degree of a node i is $\text{in-deg}(i) = \sum_j y_{ij}$
- The out-degree of a node i is $\text{out-deg}(i) = \sum_j y_{ij}$

In-Degree distribution

The in-degree distribution $\text{in-deg}(k)$ of a network is then defined to be the proportion of nodes in the network with in-degree k for $k = 0, 1, 2, \dots, n - 1$.

Out-Degree distribution

The out-degree distribution $\text{out-deg}(k)$ of a network is then defined to be the proportion of nodes in the network with out-degree k for $k = 0, 1, 2, \dots, n - 1$.

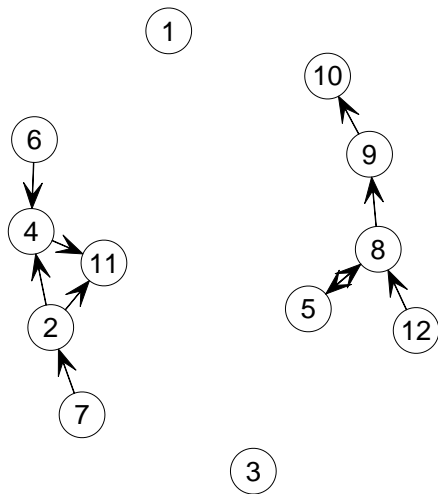
Example

```
n_d <- 12  
set.seed(1)  
y_d <- network(n_d)  
y_d[,]
```

##		1	2	3	4	5	6	7	8	9	10	11	12
##	1	0	0	0	0	0	0	0	0	0	0	0	0
##	2	0	0	0	1	0	0	0	0	0	0	1	0
##	3	0	0	0	0	0	0	0	0	0	0	0	0
##	4	0	0	0	0	0	0	0	0	0	0	1	0
##	5	0	0	0	0	0	0	0	1	0	0	0	0
##	6	0	0	0	1	0	0	0	0	0	0	0	0
##	7	0	1	0	0	0	0	0	0	0	0	0	0
##	8	0	0	0	0	1	0	0	0	1	0	0	0
##	9	0	0	0	0	0	0	0	0	0	1	0	0
##	10	0	0	0	0	0	0	0	0	0	0	0	0
##	11	0	0	0	0	0	0	0	0	0	0	0	0
##	12	0	0	0	0	0	0	0	1	0	0	0	0

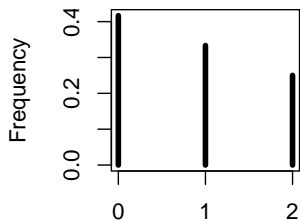
Example - A directed network graph y_d

```
plot(y_d,  
     vertex.cex = 5, vertex.col = 0,  
     arrowhead.cex = 3,  
     label = 1:n_d, label.pos = 5,  
     label.cex = 2)
```

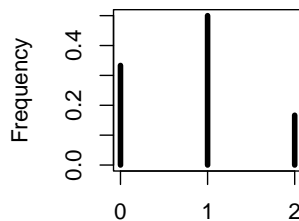


Example - In-degree and out-degree distr. of y_d

```
ideg_dist <- table(degree(y_d, cmode = "indegree"))
odeg_dist <- table(degree(y_d, cmode = "outdegree"))
par(mfrow = c(1, 2), mar = c(4, 4, 0, 4))
plot(ideg_dist/ sum(ideg_dist),
     xlab = 'In-degree', ylab = 'Frequency', lwd = 4)
plot(odeg_dist/ sum(odeg_dist),
     xlab = 'Out-degree', ylab = 'Frequency', lwd = 4)
```



In-degree



Out-degree

In statistical network analysis, network structures are described by probability distributions so we assume that the presence or absence of edges between nodes is the outcome of a random or stochastic process.

Erdős-Rényi

The Erdős-Rényi random graph model for undirected networks assumes:

- n is fixed (and therefore m is fixed);
- $Y_{ij} | \eta \stackrel{iid}{\sim} \text{Bernoulli}(\eta)$

The model likelihood is:

$$\begin{aligned} p(\mathbf{y} \mid \eta) &= \prod_{i < j} \eta^{y_{ij}} (1 - \eta)^{1 - y_{ij}} \\ &= \eta^{\sum_{i < j} y_{ij}} (1 - \eta)^{\sum_{i < j} (1 - y_{ij})}. \end{aligned}$$

It is easy to calculate the MLE for

$$\hat{\eta} = \frac{\sum_{i < j} y_{ij}}{m}$$

Example

```
e_y <- summary(y ~ edges)
m_y <- choose(n, 2)

eta_mle <- unname(e_y / m_y)
eta_mle

## [1] 0.1944444
```

Some properties

- The expected number of edges is $E(e(\mathbf{y})) = m\eta$.
- The distribution of the degree of any particular node follows a Binomial distribution:

$$\Pr(\deg(i) = k) = \binom{n-1}{k} \eta^k (1-\eta)^{n-1-k}.$$

- The random graph model belongs to the **exponential family** so by considering the parameter $\theta = \text{logit}(\eta)$:

$$\theta = \log\left(\frac{\eta}{1-\eta}\right) \Leftrightarrow \eta = \frac{\exp(\theta)}{1 + \exp(\theta)},$$

the model likelihood can be written as:

$$p(\mathbf{y} \mid \theta) = \prod_{i < j} \frac{\exp(\theta) y_{ij}}{1 + \exp(\theta)} = \frac{\exp\left(\theta \cdot \sum_{i < j} y_{ij}\right)}{\underbrace{z(\theta)}_{\text{normalising constant}}}.$$

Some properties

- The number of edges $e(\mathbf{y}) = \sum_{i < j} y_{ij}$ is called **edge statistic**.
- Interpretation of θ :
 - $\Pr(Y_{ij} = 1 \mid \theta = 0) = \eta = 0.5$.
 - $\Pr(Y_{ij} = 1 \mid \theta < 0) = \eta < 0.5$.
 - $\Pr(Y_{ij} = 1 \mid \theta > 0) = \eta > 0.5$.

```
theta_mle <- log(eta_mle / (1 - eta_mle))  
theta_mle  
  
## [1] -1.421386
```

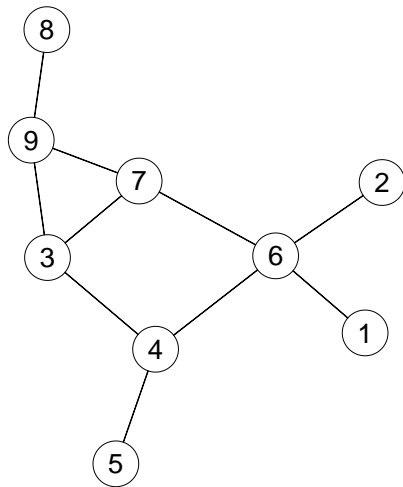
Simulation

To generate random graphs using the Erdős-Rényi model given an edge probability η we just need to simulate m iid realisations from a Bernoulli distribution with a fixed η value.

```
eta <- 0.3
theta <- log(eta / (1 - eta))
n <- 9
set.seed(1)
y_sim <- simulate(~ edges,
                  coef = theta,
                  basis = network(n, directed = FALSE))
```

Example - A randomly generated undirected network y_{sim} on 9 nodes with edge probability = 0.3

```
plot(y_sim,  
     vertex.cex = 5, vertex.col = 0,  
     label = 1:n, label.pos = 5,  
     label.cex = 2)
```



Pros and cons of Erdős-Rényi model

- It is a simple and scalable model.
- It is inappropriate for modeling many complex real-life networks as it has, for example, low clustering.

Dyadic-independence

Reciprocity model for directed networks

The Erdős-Rényi model for directed networks assumes that the Y_{ij} and Y_{ji} are iid Bernoulli random variables, but it does not allow for tendencies toward reciprocation.

- The reciprocity model assumes that the edge variables (Y_{ij}, Y_{ji}) of a dyad (i, j) depend individually on the edge parameter θ_e and mutually by the reciprocity parameter θ_r :

$$Y_{ij} \stackrel{iid}{\sim} \text{Bernoulli} \left(\frac{\exp(\theta_e + \theta_r y_{ji})}{1 + \exp(\theta_e + \theta_r y_{ji})} \right)$$

- The dyadic probabilities are:

$$p(y_{ij}, y_{ji} \mid \theta_e, \theta_r) = \frac{\exp(\theta_e(y_{ij} + y_{ji}) + \theta_r y_{ij} y_{ji})}{z_{ij}(\theta_e, \theta_r)},$$

where $z_{ij}(\theta_e, \theta_r) = 1 + 2 \exp(\theta_e) + \exp(2\theta_e + \theta_r)$ is the the dyadic normalising constant.

Reciprocity model for directed networks

- The model likelihood is:

$$p(\mathbf{y} \mid \theta_e, \theta_r) = \prod_{i < j} p(y_{ij}, y_{ji} \mid \theta_e, \theta_r) = \frac{\exp \{ \theta_e e(\mathbf{y}) + \theta_r r(\mathbf{y}) \}}{\underbrace{z(\theta_e, \theta_r)}_{\text{normalising constant}}},$$

where

- $e(\mathbf{y}) = \sum_{i < j} y_{ij} + y_{ji}$ is the edge parameter and
- $r(\mathbf{y}) = \sum_{i < j} y_{ij} y_{ji}$ is the reciprocity statistic, i.e., the number of reciprocated edges in the network.

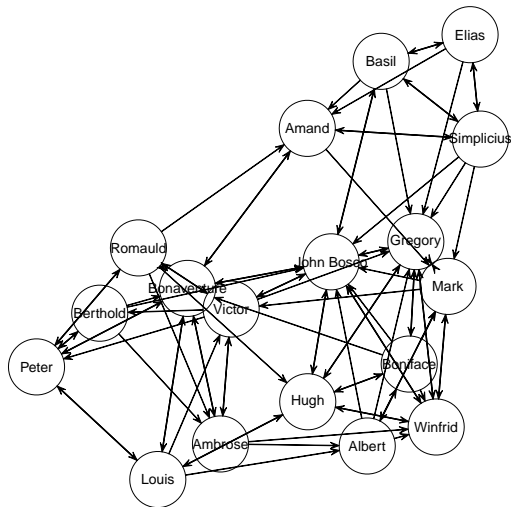
Example

Sampson (1969) recorded positive affect relations among a group of monks while he was a resident as an experimenter at the cloister

```
data(sampson)  
y_d <- samplike
```


Example

```
set.seed(1)
plot(y_d,
     vertex.cex = 6,
     vertex.col = 0,
     label = y_d %v%
       'vertex.names',
     label.pos = 5)
```



Example

```
r_model <- ergm(y_d ~ edges + mutual)
summary(r_model)$coefficients
```

##		Estimate	Std. Error	MCMC %	z value	Pr(> z)
## edges		-1.775154	0.1996407	0	-8.891744	6.015709e-19
## mutual		2.319160	0.4309846	0	5.381074	7.404277e-08

Example

The edge parameter $\hat{\theta}_e < 0$ meaning that the probability of observing an edge in the network is less than 0.5:

$$\begin{aligned}\Pr(Y_{ji} = 1 \mid \hat{\theta}_e, \hat{\theta}_r) &= \Pr(Y_{ji} = 1, Y_{ij} = 0 \mid \hat{\theta}_e, \hat{\theta}_r) \\ &\quad + \Pr(Y_{ji} = 1, Y_{ij} = 1 \mid \hat{\theta}_e, \hat{\theta}_r) \\ &= \frac{\exp(\hat{\theta}_e)}{z_{ij}} + \frac{\exp(2\hat{\theta}_e + \hat{\theta}_r)}{z_{ij}}\end{aligned}$$

```
theta_e <- unname(r_model$coefficients["edges"])
theta_r <- unname(r_model$coefficients["mutual"])
z_ij <- 1 + 2 * exp(theta_e) + exp(theta_e + theta_r)
(exp(theta_e) + exp(2 * theta_e + theta_r)) / z_ij

## [1] 0.1506999
```

Example

The reciprocity parameter $\hat{\theta}_r > 0$ meaning that the probability of observing an edge $Y_{ji} = 1$ reciprocating an existing edge $Y_{ij} = 1$ is higher than 0.5:

$$\Pr(Y_{ji} = 1 \mid Y_{ij} = 1, \hat{\theta}_e, \hat{\theta}_r) = \frac{\exp(\hat{\theta}_e + \hat{\theta}_r)}{1 + \exp(\hat{\theta}_e + \hat{\theta}_r)}.$$

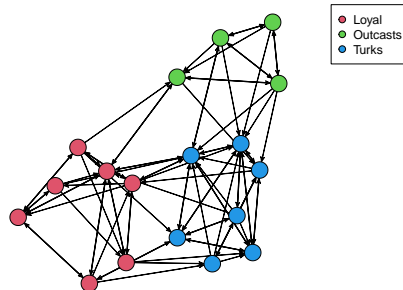
```
exp(theta_e + theta_r) / (1 + exp(theta_e + theta_r))
```

```
## [1] 0.6327438
```

Nodal and dyadic covariates

- Nodal and dyadic covariates can also be introduced as exogenous predictors of network edges.
- For example, we can introduce in the reciprocity model the homophily statistic represented by the number of edges connecting nodes with the same nodal covariate.

```
set.seed(1)
plot(y_d, vertex.cex = 3,
     vertex.col =
       unclass(factor(y_d %v% 'group')) + 1)
legend('topright', pt.bg = 2:4,
      pt.cex = 1.2, pch = 21,
      legend = sort(unique(y_d %v% 'group')))
```



- We can therefore include this categorical covariate \mathbf{x} and let it interact with the reciprocity effect in order to understand whether reciprocity is concentrated within groups.
- Our model can be written as:

$$p(\mathbf{y} \mid \mathbf{x}, \theta_e, \theta_r, \theta_h) = \frac{\exp \{ \theta_e e(\mathbf{y}) + \theta_r r(\mathbf{y}) + \theta_h h(\mathbf{y}, \mathbf{x}) \}}{\underbrace{z(\theta_e, \theta_r, \theta_h)}_{\text{normalising constant}}},$$

where the mutual homophily statistic $h(\mathbf{y}, \mathbf{x}) = \sum_{i < j} y_{ij} y_{ji} x_{ij}$, with $x_{ij} = 1$, if $x_i = x_j$ and $x_{ij} = 0$, otherwise.

Nodal and dyadic covariates

```
r_model_2 <- ergm(y_d ~ edges + mutual + mutual('group'))  
summary(r_model_2)$coefficients
```

##	Estimate	Std. Error	MCMC %	z value	Pr(> z)
## edges	-1.7517224	0.2077578	0	-8.431558	3.410921e-17
## mutual	0.8082342	0.5745339	0	1.406765	1.594971e-01
## mutual.group	2.9098390	0.5351458	0	5.437470	5.404256e-08

Conditional edge probabilities

Let's interpret the results by estimating the conditional edge probabilities.

```
edge_prob_y_ij_1 <- function(y_ji, x_ij, theta){  
  
  num <- unname(exp(theta["edges"] +  
                    theta["mutual"] * y_ji +  
                    theta["mutual.group"] * y_ji * x_ij))  
  
  den <- 1 + num  
  
  return(num / den)  
  
}
```


Conditional edge probabilities

The edge parameter $\hat{\theta}_e < 0$ meaning that we expect:

$$\Pr(Y_{ij} = 1 \mid Y_{ji} = 0, X_{ij} = 0, \hat{\theta}_e, \hat{\theta}_r, \hat{\theta}_h) = \frac{\exp(\hat{\theta}_e)}{1 + \exp(\hat{\theta}_e)} < 0.5.$$

```
edge_prob_y_ij_1(y_ji = 0, x_ij = 0,  
                 theta = r_model_2$coefficients)
```

```
## [1] 0.1478301
```

Conditional edge probabilities

The reciprocity parameter $\hat{\theta}_r$ is not significant meaning that we expect:

$$\Pr(Y_{ij} = 1 \mid Y_{ji} = 1, X_{ij} = 0, \hat{\theta}_e, \hat{\theta}_r, \hat{\theta}_h) = \frac{\exp(\hat{\theta}_e + \hat{\theta}_r)}{1 + \exp(\hat{\theta}_e + \hat{\theta}_r)}$$

not significantly different from $\Pr(Y_{ij} = 1 \mid Y_{ji} = 0, X_{ij} = 0, \hat{\theta}_e, \hat{\theta}_r, \hat{\theta}_h)$

```
edge_prob_y_ij_1(y_ji = 1, x_ij = 0,  
                 theta = r_model_2$coefficients)
```

```
## [1] 0.2801963
```

```
edge_prob_y_ij_1(y_ji = 0, x_ij = 0,  
                 theta = r_model_2$coefficients)
```

```
## [1] 0.1478301
```

Conditional edge probabilities

The mutual homophily parameter $\hat{\theta}_h > |\hat{\theta}_e| > 0$ meaning that we expect:

$$\Pr(Y_{ij} = 1 \mid Y_{ji} = 1, X_{ij} = 1, \hat{\theta}_e, \hat{\theta}_r, \hat{\theta}_h) = \frac{\exp(\hat{\theta}_e + \hat{\theta}_r + \hat{\theta}_h)}{1 + \exp(\hat{\theta}_e + \hat{\theta}_r + \hat{\theta}_h)} > 0.5$$

```
edge_prob_y_ij_1(y_ji = 1, x_ij = 1,  
                 theta = r_model_2$coefficients)
```

```
## [1] 0.8772186
```

Conditional dyadic probabilities

Now let's observe the impact of the edge and reciprocity parameter on the estimated dyadic probability distributions.

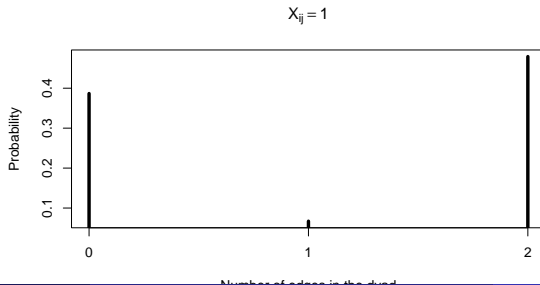
```
dyadic_prob <- function(x_ij, theta){  
  theta_e <- unname(theta["edges"])  
  theta_r <- unname(theta["mutual"])  
  theta_h <- ifelse(x_ij == 1,  
                    theta["mutual.group"], 0)  
  z_ij <- 1 + 2 * exp(theta_e) +  
    exp(2 * theta_e + theta_r + theta_h)  
  
  p_00 <- 1 / z_ij  
  p_10 <- exp(theta_e) / z_ij # = p_01  
  p_11 <- exp(2 * theta_e + theta_r + theta_h) / z_ij  
  
  return(c(p_00 = p_00, p_10 = p_10, p_11 = p_11))  
}
```

Conditional dyadic probabilities

Dyadic probability distribution of nodes who belong to the same group

$p(y_{ij}, y_{ji} \mid X_{ij} = 1, \hat{\theta}_e, \hat{\theta}_r, \hat{\theta}_h)$:

```
plot(dyadic_prob(x_ij = 1, theta = r_model_2$coefficients),  
     type = 'h', xlab = 'Number of edges in the dyad',  
     ylab = 'Probability', main = expression(X[ij] == 1),  
     lwd = 4, xaxt="n")  
axis(1, at = 1:3, labels = 0:2)
```

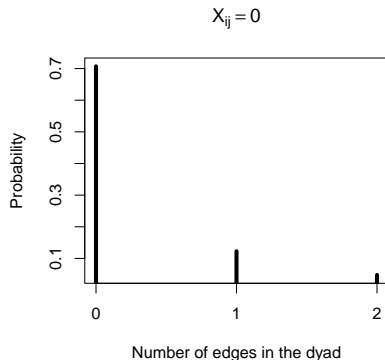


Conditional dyadic probabilities

Dyadic probability for nodes who do not belong to the same group

$$p(y_{ij}, y_{ji} \mid X_{ij} = 0, \hat{\theta}_e, \hat{\theta}_r, \hat{\theta}_h) :$$

```
plot(dyadic_prob(x_ij = 0,  
               theta = r_model_2$coefficients),  
     type = 'h',  
     xlab = 'Number of edges in the dyad',  
     ylab = 'Probability',  
     main = expression(X[ij] == 0),  
     lwd = 4, xaxt = "n")  
axis(1, at = 1:3, labels = 0:2)
```



Dyadic probability distribution for two nodes that do not belong to the same group. The mode of the distribution corresponds to the empty dyad.

Conditional odds ratios of edges

It is useful to calculate the conditional odds ratios of single edge variable Y_{ij} given the rest of the network structure \mathbf{Y}_{-ij} and potential covariate information \mathbf{x} :

$$\frac{\Pr(Y_{ij} = 1 \mid \mathbf{Y}_{-ij}, \mathbf{x}, \boldsymbol{\theta})}{\Pr(Y_{ij} = 0 \mid \mathbf{Y}_{-ij}, \mathbf{x}, \boldsymbol{\theta})} = \exp \left\{ \boldsymbol{\theta}^\top \delta_{ij}^+(\mathbf{y}) \right\},$$

where $\delta_{ij}^+(\mathbf{y})$ is the change statistic representing the difference between the number of statistics when $Y_{ij} = 1$ and the number of statistics when $Y_{ij} = 0$.

Example

Let's calculate the conditional odds ratio for the edge variable $Y_{2,7}$ involving the Gregory (node 2) and Mark (node 7).

The change statistic $\delta_{2,7}^+(\mathbf{y}) = s(\mathbf{y}; Y_{2,7} = 1) - s(\mathbf{y}; Y_{2,7} = 0)$ is:

```
y_d[2, 7] <- 1
s_y_ij_1 <- summary(y_d ~ edges + mutual + mutual('group'))

y_d[2, 7] <- 0
s_y_ij_0 <- summary(y_d ~ edges + mutual + mutual('group'))

delta_y_ij <- s_y_ij_1 - s_y_ij_0
delta_y_ij

##           edges           mutual mutual.group
##             1             1             1
```


Example

The conditional odds ratio $\frac{\Pr(Y_{2,7} = 1 \mid \mathbf{Y}_{-ij}, x_{ij}, \hat{\theta}_e, \hat{\theta}_r, \hat{\theta}_h)}{\Pr(Y_{2,7} = 0 \mid \mathbf{Y}_{-ij}, x_{ij}, \hat{\theta}_e, \hat{\theta}_r, \hat{\theta}_h)}$ is:

```
theta <- unname(r_model_2$coefficients)
OR_y_ij <- c(exp(t(theta) %*% delta_y_ij))
OR_y_ij

## [1] 7.144557
```

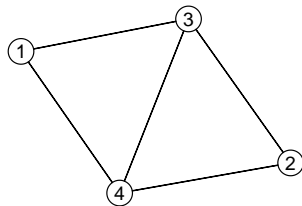
Markov models

Markov dependence assumption

Markov dependence assumption

Two edge variables Y_{ij} and Y_{kl} are conditionally dependent given the rest of \mathbf{Y} , if they share a node.

```
n <- 4
set.seed(123)
y <- network(n, directed = FALSE)
plot(y, vertex.cex = 4,
     vertex.col = 0,
     label = 1:n, label.pos = 5)
```



The Hammersley-Clifford theorem

The **Hammersley-Clifford theorem**¹ informs us that Markov random graphs are completely characterised by the following set of network statistics:

- the edge statistic $e(\mathbf{y})$;
- the star statistics $s(\mathbf{y}) = \sum_{i=1}^n \binom{\deg(i)}{k}$;
- the triangle statistic $t(\mathbf{y}) = \sum_{i < j < k} y_{ij} y_{jk} y_{ki}$.

¹Hammersley, J. M. and Clifford, P. (1971), Markov fields on finite graphs and lattices.

Phase transition

- A fundamental property of Markov random graphs is that parameters regulate how many of their corresponding configurations are expected on average.
- The statistics are highly interdependent: we cannot increase the expected number of triangles without also increasing the expected number of edges and stars.
- A dramatic transition from low-density graphs to high-density graphs associated to some parameter values illustrates one of the difficulties with Markov models. This phenomenon is known as phase transition and it is characterized by high uncertainty about the number of statistics corresponding to certain parameter values.

Schweinberger, Michael. (2011). Instability, sensitivity, and degeneracy of discrete exponential families. *Journal of the American Statistical Association*, 106(496), 1361–1370.

Example

A basic Markov model is defined by the edge statistic, 2-star and 3-stars statistics (capturing the mean, standard deviation, and skew of the degree distribution, respectively); and triangle (capturing clustering).

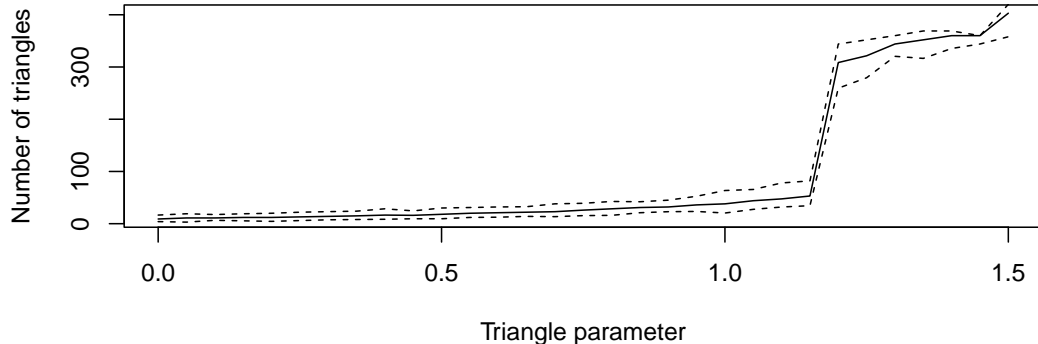
```
sim <- function(theta_triangle){  
  simulate(~ edges + kstar(2:3) + triangle,  
    basis = network(30, directed = FALSE),  
    coef = c(-3, 0.5, -0.2, theta_triangle),  
    output = 'stats',  
    control = control.simulate.formula(MCMC.burnin = 50000,  
                                         MCMC.interval = 1000),  
    nsim = 100)[,4] |>  
    quantile(probs = c(0.025, 0.50, 0.975))  
}  
set.seed(123)  
theta_triangle_values <- seq(0, 1.5, by = 0.05)  
simt <- sapply(theta_triangle_values, sim)
```

Example

```
plot(x = theta_triangle_values, y = simt[2, ], main = '', type = 'l',  
     xlab = 'Triangle parameter',  
     ylab = 'Number of triangles')  
lines(x = theta_triangle_values,  
      y = simt[1, ],  
      lty = 2)  
lines(x = theta_triangle_values,  
      y = simt[3, ],  
      lty = 2)
```

Example

Median and 95% interval for number of edges and triangles as function of triangle parameter. Roughly after $\theta_{triangle} = 1.11$ there is a dramatic rise in the number of triangles (and all the other statistics).



Example

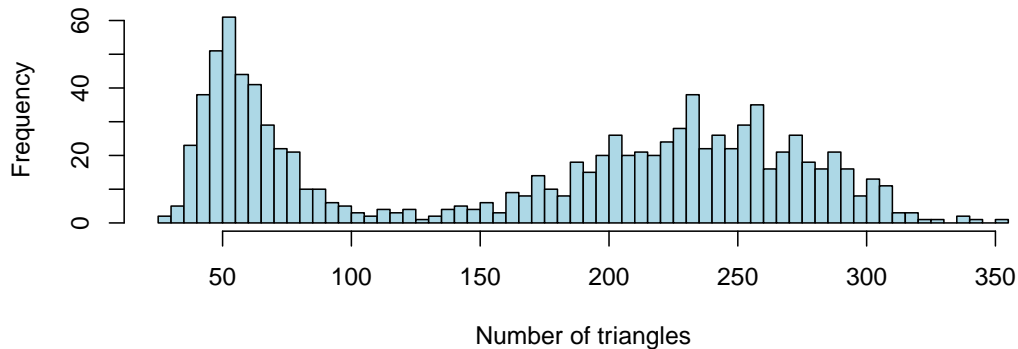
```
triangle.coef <- 1.1502
n.sim <- 1000

set.seed(1)
y_sims <- simulate(~ edges + kstar(2:3) + triangle,
                  basis = network(30, directed = FALSE),
                  coef = c(-3, 0.5, -0.2, triangle.coef),
                  output = 'stats',
                  control = control.simulate.formula(MCMC.burnin = 50000,
                                                    MCMC.interval = 1000),
                  nsim = n.sim)

hist(y_sims[, 4],
     breaks = 50,
     col = 'lightblue',
     main = "",
     xlab = 'Number of triangles')
```

Example

When, for example, $\theta_{triangle} = 1.1502$ the 95% interval extends from about 30 to 340 triangles and the distribution is bimodal.



Example

The high variability occurs because the model places most of the probability on low-density graphs and high-density graphs, and very little probability on graphs in-between. Bimodality is problematic for both inference and interpretation of parameters. Similar phase transitions can be observed for the star parameters.

Geometrically-weighted statistics

A natural solution to the phase transition is to use a statistic that places decreasing weights on the higher degrees and therefore constraining on the natural parameter space. An elegant way is to use degree counts with geometrically decreasing weights. The **geometrically weighted degree** statistic (for undirected networks) is defined as:

$$\text{gwdeg}_\phi(\mathbf{y}) = e^\phi \sum_{k=1}^{n-2} \left\{ 1 - \left(1 - e^{-\phi} \right)^k \right\} \text{deg}_k(\mathbf{y}),$$

where $\text{deg}_k(\mathbf{y})$ is the number of nodes with degree k and $\phi > 0$ is a decay parameter controlling the geometric rate of decrease in the weights. For large values of ϕ , the contribution of the higher degree nodes is greatly decreased. As $\phi \rightarrow 0$ the statistic places increasing weight on the high degree graphs.

Geometrically-weighted statistics

An equivalent statistic for triangles is the **geometrically weighted edgewise shared partner** statistic

$$\text{gwesp}_\phi(\mathbf{y}) = e^\phi \sum_{k=1}^{n-2} \left\{ 1 - \left(1 - e^{-\phi} \right)^k \right\} \text{esp}_k(\mathbf{y}),$$

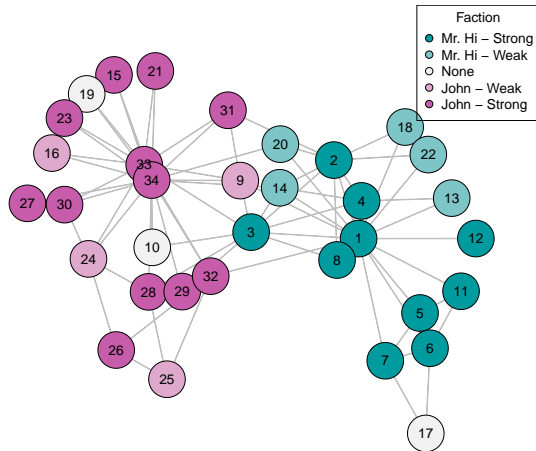
where the edgewise shared partner term $\text{esp}_k(\mathbf{y})$ is the number of connected nodes who have exactly k shared partners.

Snijders, T. A., Pattison, P. E., Robins, G. L., and Handcock, M. S. (2006). New specifications for exponential random graph models. *Sociological methodology*, 36(1), 99-153.

Example

Zachary (1977) reported observations of social relations between 34 individuals in a university karate club: 32 ordinary club members and officers, the club president ("John A." - node 34), and the part-time instructor ("Mr. Hi" - node 1).

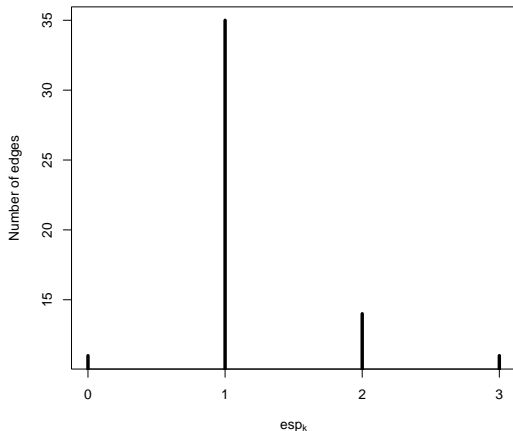
```
data(zach); y <- zach
cp <- hcl.colors(5, 'Tropic') # palette
f_id <- y %v% 'faction.id' + 3
ids <- unique(f_id)
factions <- unique(y %v% 'faction')
plot(y, label = 1:(dim(y[, ])[1]),
     label.cex = 1, label.pos = 5,
     vertex.cex = 4, edge.col = 'grey',
     vertex.col = cp[f_id])
legend('topright', pt.cex = 1.2, pch = 21,
     pt.bg = cp[sort(ids)],
     legend = factions[order(ids)],
     title = 'Faction')
```



Example

Edgewise shared partner distribution of the Zachary karate club network.

```
par(mar = rep(4, 4))
plot(summary(y ~ esp(0:(n - 1))),
     type = 'h',
     ylab = 'Number of edges',
     xlab = expression(esp[k]),
     main = '',
     lwd = 4,
     xaxt = 'n')
axis(1, at = 1:n, labels = 0:(n - 1))
```



Example

Let's fit a Markov model to the karate club network using the edge statistic; edge homophily based on the factions; the GW degree and edgewise shared partner statistics. A generally convenient default value for the decay parameter ϕ of the GW statistics is $\log 2 \approx 0.7$.

```
formula <- y ~ edges + nodematch('faction') + gwesp(decay = 0.7, fixed = TRUE)
set.seed(1)
gwm_model <- ergm(formula)
summary(gwm_model)$coefficients
```

##	Estimate	Std. Error	MCMC %	z value	Pr(> z)
## edges	-3.2417493	0.2514523	0	-12.892105	4.986410e-38
## nodematch.faction	1.0696267	0.2067894	0	5.172543	2.309299e-07
## gwesp.fixed.0.7	0.6180764	0.1289106	0	4.794611	1.629907e-06

Example - Curved exponential family

When the decay parameter is not fixed, it can be estimated. In this case the geometrically weighted statistics are non linear, *curved*, functions of the decay parameter ϕ .

Hunter, D. R., and Handcock, M. S. (2006). Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*, 15(3), 565-583.

Example

Let's calculate the conditional odds ratio for the edge variable $Y_{33,34}$.

```
theta <- unname(gwm_model$coefficients)

y[33, 34] <- 1; s_y_ij_1 <- summary(formula)
y[33, 34] <- 0; s_y_ij_0 <- summary(formula)

delta_y_ij <- s_y_ij_1 - s_y_ij_0
delta_y_ij

##              edges nodematch.faction  gwesp.fixed.0.7
##              1.00000              1.00000           16.79241

OR_y_ij <- c(exp(t(theta) %*% delta_y_ij))
OR_y_ij

## [1] 3666.062
```

Example

Let's calculate the conditional odds ratio for the edge variable $Y_{10,17}$.

```
y[10, 17] <- 1
s_y_ij_1 <- summary(formula)

y[10, 17] <- 0
s_y_ij_0 <- summary(formula)

delta_y_ij <- s_y_ij_1 - s_y_ij_0
delta_y_ij

##              edges nodematch.faction   gwesp.fixed.0.7
##      1.000000e+00      1.000000e+00      -1.421085e-14

OR_y_ij <- c(exp(t(theta) %*% delta_y_ij))
OR_y_ij

## [1] 0.1139355
```

Example

Let's calculate the conditional odds ratio for the edge variable $Y_{1,34}$.

```
y[1, 34] <- 1
s_y_ij_1 <- summary(formula)

y[1, 34] <- 0
s_y_ij_0 <- summary(formula)

delta_y_ij <- s_y_ij_1 - s_y_ij_0
delta_y_ij

##              edges nodematch.faction  gwesp.fixed.0.7
##              1.000000              0.000000              7.025657

OR_y_ij <- c(exp(t(theta) %*% delta_y_ij))
OR_y_ij

## [1] 3.0061
```

Estimation and Goodness of fit

To sample a graph \mathbf{y} from an estimated (intractable) Markov model with parameter $\hat{\theta}$, we rely on Markov chain Monte Carlo (MCMC) procedures which consists of generating a sequence of M graphs that are successively updated through small changes.

Metropolis-Hastings algorithm

Start at any network graph \mathbf{y} (e.g. empty graph). Iterate M times:

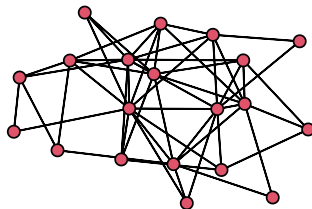
- 1 Propose \mathbf{y}' by 'toggling' a randomly selected tie-variable of \mathbf{y} ;
- 2 Accept $\mathbf{y} = \mathbf{y}'$ with probability

$$\min \left(1, \frac{\exp \left\{ \hat{\boldsymbol{\theta}}^T s(\mathbf{y}') \right\}}{\exp \left\{ \hat{\boldsymbol{\theta}}^T s(\mathbf{y}) \right\}} \right)$$

(reject otherwise)

Simulation

```
n <- 20
set.seed(1)
y_sim <- simulate(~ edges + gwesp(0.7, fixed = TRUE),
                  coef = c(-2, 0.5),
                  basis = network(n, directed = FALSE),
                  control = control.simulate(MCMC.interval = 5000))
plot(y_sim, vertex.cex = 2)
```



- Dyadic-independence models can be formulated as a generalised linear models (GLM) and they can therefore be estimated using iterative generalized (or weighted) least squares (IGLS), yielding maximum likelihood estimators.
- One of the most popular estimation approaches for Markov models (and implemented in statnet) is the **Geyer-Thompson** approach.
- This is an MCMC algorithm that takes one large sample of M network graphs for a provisional value of the parameter vector to approximate the likelihood ratio:

$$\log \left\{ \frac{p(\mathbf{y} | \boldsymbol{\theta})}{p(\mathbf{y} | \boldsymbol{\theta}')} \right\} \approx (\boldsymbol{\theta}' - \boldsymbol{\theta})^t s(\mathbf{y}) - \log \left\{ \frac{1}{M} \sum_{i=1}^M \exp \left[(\boldsymbol{\theta}' - \boldsymbol{\theta})^\top s(\mathbf{y}_i) \right] \right\}$$

Geyer, C. J., and Thompson, E. A. (1992). Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society: Series B*, 54(3), 657-683.

Goodness of fit (GOF)

The GOF procedure is carried out as follows:

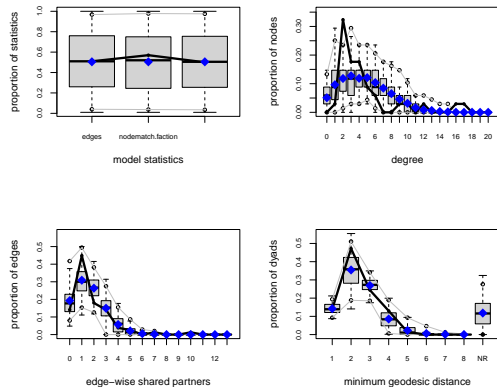
- ① We simulate from the model to generate a distribution of graphs.
- ② We assess whether a summary measure $s_k(\mathbf{y}_{obs})$ for the observed graph is far from what we would predict under the fitted model, the standardized difference $[s_k(x_{obs}) - \bar{s}_k] / SD(s_k(x))$ is calculated, where \bar{s}_k and $SD(s_k(x))$ are the mean and standard deviation, respectively, calculated over the generated sample of graphs.
- ③ If this ratio is large in absolute value, then the observed measure is far from what is expected under the model and the corresponding p-value will be close to 0.

Hunter, D. R., Goodreau, S. M., and Handcock, M. S. (2008). Goodness of fit of social network models. *Journal of the American Statistical Association*, 103(481), 248-258.

Example

```
data(zach)
y <- zach
formula <- y ~ edges +
  nodematch('faction') +
  gwesp(decay = 0.7, fixed = TRUE)
set.seed(1)
gwm_model <- ergm(formula)
gwm_gof <- gof(gwm_model)
par(mfrow = c(2, 2))
plot(gwm_gof)
```

Goodness-of-fit diagnostics



Wald tests, information criteria (e.g. AIC) and likelihood ratio tests can be performed between models using approximate likelihoods.

Latent Space Models

Latent Space Model

- Hoff *et al.* (2002) introduced a model that assumes that each node n has an unknown position z_n in a D -dim *Euclidean latent space*.

$$p(Y|Z, \alpha) = \prod_{i \neq j}^N p(y_{ij}|z_i, z_j, \alpha) = \prod_{i \neq j}^N \frac{\exp(\alpha - |z_i - z_j|)^{y_{ij}}}{1 + \exp(\alpha - |z_i - z_j|)}$$

with $p(\alpha) = \mathcal{N}(\xi, \psi^2)$, $p(z_n) \stackrel{iid}{=} \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_D)$ and σ^2, ξ, ψ^2 are fixed parameters.

The posterior distribution *cannot* be calculated analytically.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460), 1090-1098.

You can fit the model using a *Variational inference approach* that is considerably quicker but less accurate than MCMC. (Gollini Murphy 2013)

- The posterior probability of the unknown (Z, α) is:

$$p(Z, \alpha | Y) = p(Y | Z, \alpha) p(\alpha) \prod_{n=1}^N p(z_n) \times C$$

where C is the unknown normalising constant

- You can use the variational posterior $q(Z, \alpha | Y)$:

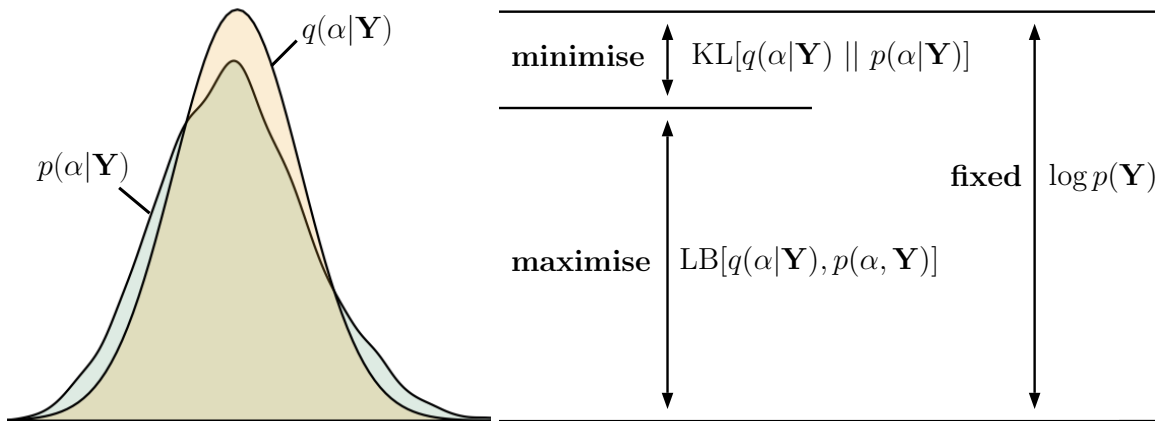
$$q(Z, \alpha | Y) = q(\alpha) \prod_{n=1}^N q(z_n)$$

where $q(\alpha) = \mathcal{N}(\tilde{\xi}, \tilde{\psi}^2)$ and $q(z_n) = \mathcal{N}(\tilde{z}_n, \tilde{\Sigma})$.

Gollini, I., and Murphy, T. B. (2016), Joint Modelling of Multiple Network Views, *Journal of Computational and Graphical Statistics*, 25:1, 246-265.

Variational Approach

- The basic idea behind the variational approach is to find a lower bound of the log marginal likelihood $\log p(\mathbf{Y})$ by introducing the variational posterior distribution $q(\mathbf{Z}, \alpha | \mathbf{Y})$.



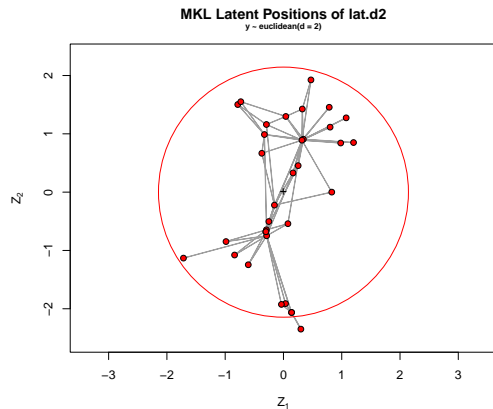
- We compare the results obtained by using two R packages:
 - **latentnet** (Krivitsky and Handcock, 2008, 2022): Euclidean Distance, MCMC
 - **lvm4net** (Gollini, 2014): Squared Euclidean Distance, Variational
- Krivitsky PN, Handcock MS (2008) Fitting position latent cluster models for social networks with latentnet. *Journal of Statistical Software*, 24(5).
- Krivitsky P, Handcock M (2022). latentnet: Latent Position and Cluster Models for Statistical Networks. The Statnet Project (<https://statnet.org>). R package version 2.10.6.
- Gollini I, (2019) lvm4net: Latent Variable Models for Networks, R package version 0.3. R package version Available on GitHub: <https://github.com/igollini/lvm4net>

Example - latentnet

```
plot(lat.d2)
```

```
data(zach)
y <- zach
library(latentnet)
system.time(lat.d2 <-
  ergmm(y ~ euclidean(d = 2)))

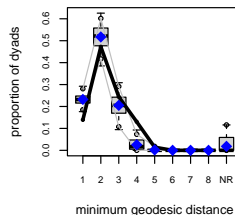
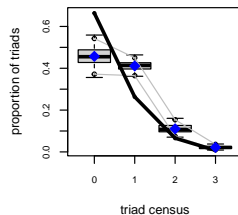
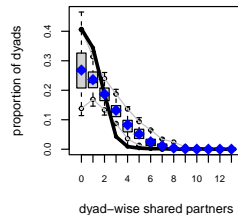
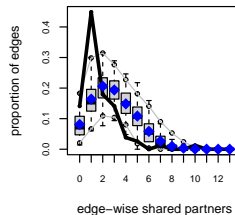
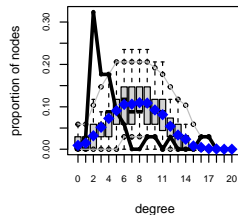
##      user  system elapsed
##    3.116    0.026    3.152
```



Example - latentnet

Goodness-of-fit diagnostics

```
gf.lat.d2 <-  
  gof(lat.d2,  
    GOF = ~ degree +  
      esp + dsp +  
      triadcensus +  
      distance)  
par(mfrow = c(2,3))  
plot(gf.lat.d2)
```

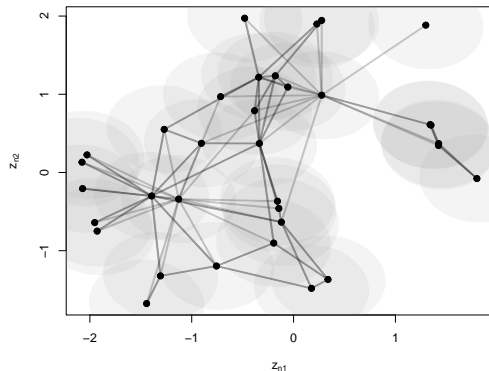


Example - lvm4net

```
plot(lvm.d2, y[, ],  
     drawCB = TRUE)
```

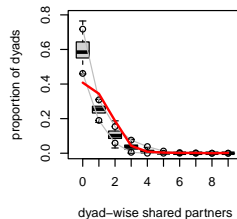
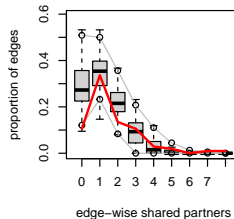
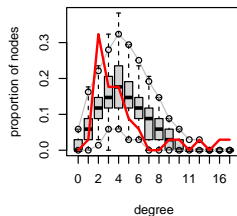
```
library(lvm4net)  
system.time(  
  lvm.d2 <- lsm(y[, ], D = 2,  
               nstart = 5))
```

```
##      user  system elapsed  
##    0.330    0.004    0.336
```

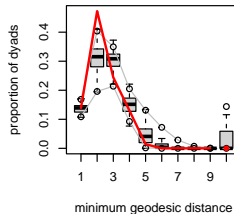
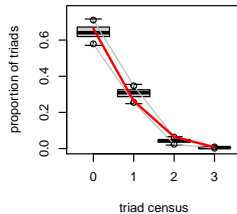


Example - lvm4net

Goodness-of-fit diagnostics



```
lsmGof <-  
  goflsm(lvm.d2, Y = y[,])
```



Variational Inference VS MCMC

- Closed form posteriors
- Far faster than MCMC based methods
- In the absence of posterior dependence, the lower bound would match the log likelihood.
- Underestimates variances
- Difficult to assess how tight the lower bound is.
- Sensitive to starting values (local minima)
- As long as the posterior dependence is weak, the VA may be useful:
 - Larger networks
 - For starting point of MCMC algorithms
 - To explore the model space.

- LSM with covariates and sender and receiver:

$$p(Y|Z, \alpha, X, \beta) = \prod_{i \neq j}^N \frac{\exp(\alpha + \beta^T \mathbf{x}_{ij} - |z_i - z_j| + \delta_i + \gamma_j)^{y_{ij}}}{1 + \exp(\alpha + \beta^T \mathbf{x}_{ij} - |z_i - z_j| + \delta_i + \gamma_j)}$$

where β and \mathbf{x}_{ij} are vectors of length M . In directed graph we don't have sender and receiver, but sociality factor.

- The Projection model for strongly asymmetric graphs:

$$p(Y|Z, \alpha, X, \beta) = \prod_{i \neq j}^N \frac{\exp\left(\alpha + \beta^T \mathbf{x}_{ij} - \frac{|z_i^T z_j|}{|z_j|}\right)^{y_{ij}}}{1 + \exp\left(\alpha + \beta^T \mathbf{x}_{ij} - \frac{|z_i^T z_j|}{|z_j|}\right)}$$

- Latent Position Cluster Models (Handcock et al. 2007) extends the LSM assuming that the latent positions come from a mixture model:

$$z_i \sim \sum_{g=1}^G \lambda_g \text{MVN}_d(\mu_g, \sigma^2 \mathbf{I})$$

where λ_g is the probability that a node belongs to group g and $\sum_{g=1}^G \lambda_g = 1$.

Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2), 301-354.

Some useful references

- Caimo, A., and Gollini, I. (2023). Recent Advances In Exponential Random Graph Modelling. In *Mathematical Proceedings of the Royal Irish Academy* (Vol. 123, No. 1, pp. 1-12). Royal Irish Academy.
- Salter-Townshend, M., White, A., Gollini, I., and Murphy, T. B. (2012). Review of statistical network analysis: models, algorithms, and software. *Statistical Analysis and Data Mining*, 5(4), 243.
- Schweinberger, M., Krivitsky, P. N., Butts, C. T., and Stewart, J. R. (2020) Exponential-Family Models of Random Graphs: Inference in Finite, Super and Infinite Population Scenarios. *Statistical Science* 35 (4), 627-662