Higher order patterns via factor models

567 Statistical analysis of social networks

Peter Hoff

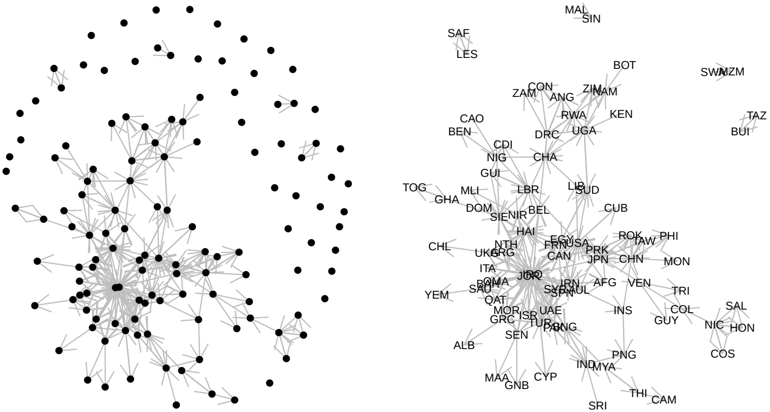
Statistics, University of Washington

Conflict data

```
Y<-conflict90s$conflicts

Xn<-conflict90s$nodevars

colnames(Xn)

## [1] "pop"    "gdp"    "polity"

Xn[,1:2]<-log(Xn[,1:2])
```

```
table(Y)

## Y
##     0     1     2     3     4     5     6     7     8
## 16567   154    22    14     7     2     2     1     1
```

Dichotomized conflict data



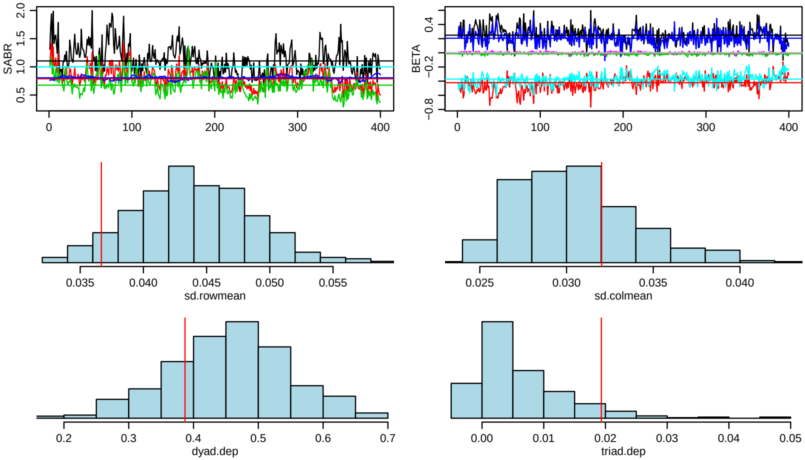
SRM fit to ordinal data

```
#### nodal covariates only
fit_n<-ame(Y,Xrow=Xn,Xcol=Xn,model="ord",nscan=10000)
```

```
summary(fit_n)

##
## beta:
##              pmean   psd z-stat p-val
## pop.row      0.256 0.121  2.114 0.034
## gdp.row     -0.430 0.099 -4.334 0.000
## polity.row -0.014 0.017 -0.797 0.425
## pop.col      0.207 0.096  2.151 0.031
## gdp.col     -0.371 0.078 -4.774 0.000
## polity.col -0.001 0.014 -0.062 0.950
##
## Sigma_ab pmean:
##       a     b
## a 1.134 0.801
## b 0.801 0.687
##
## rho pmean:
##  0.804
```

SRM fit to ordinal data

```
plot(fit_n)
```



Triad dependence measure

```
gofstats

## function (Y)
## {
##     sd.rowmean <- sd(rowMeans(Y, na.rm = TRUE), na.rm = TRUE)
##     sd.colmean <- sd(colMeans(Y, na.rm = TRUE), na.rm = TRUE)
##     dyad.dep <- cor(c(Y), c(t(Y)), use = "complete.obs")
##     E <- Y - mean(Y, na.rm = TRUE)
##     D <- 1 * (!is.na(E))
##     E[is.na(E)] <- 0
##     triad.dep <- sum(diag(E %*% E %*% E))/(sum(diag(D %*% D %*%
##         D)) * sd(c(Y), na.rm = TRUE)^3)
##     gof <- c(sd.rowmean, sd.colmean, dyad.dep, triad.dep)
##     names(gof) <- c("sd.rowmean", "sd.colmean", "dyad.dep", "triad.dep")
##     gof
## }
## <environment: namespace:amen>
```

Triad dependence measure

Let $\mathbf{E} = (\mathbf{Y} - \mathbf{1}\mathbf{1}^T \bar{y}_{\cdot\cdot})/s_y$, that is, $e_{i,j} = (y_{i,j} - \bar{y}_{\cdot\cdot})/s_y$.

- $\bar{y}_{\cdot\cdot}$ is the grand mean;
- s_y is the sample standard deviation of the $y_{i,j}$'s
- $e_{i,j}$ is like a "scaled residual" from the simple mean model $y_{i,j} = \mu + \epsilon_{i,j}$.

The sum of the diagonal of E^3 gives the scaled third-order moment:

$$\text{trace}(\mathbf{E}^3) = \sum_i \sum_j \sum_k e_{i,j} e_{j,k} e_{k,i}$$

The GOF statistic used in `amen` is the average of the $e_{i,j} e_{j,k} e_{k,i}$'s:

$$t(\mathbf{Y}) = \frac{\sum_i \sum_j \sum_k e_{i,j} e_{j,k} e_{i,j}}{\#\text{ of ordered triads}}$$

Triad dependence measure

Let $\mathbf{E} = (\mathbf{Y} - \mathbf{1}\mathbf{1}^T \bar{y}_{\cdot\cdot})/s_y$, that is, $e_{i,j} = (y_{i,j} - \bar{y}_{\cdot\cdot})/s_y$.

- $\bar{y}_{\cdot\cdot}$ is the grand mean;
- s_y is the sample standard deviation of the $y_{i,j}$'s
- $e_{i,j}$ is like a "scaled residual" from the simple mean model $y_{i,j} = \mu + \epsilon_{i,j}$.

The sum of the diagonal of E^3 gives the scaled third-order moment:

$$\text{trace}(\mathbf{E}^3) = \sum_i \sum_j \sum_k e_{i,j} e_{j,k} e_{k,i}$$

The GOF statistic used in `amen` is the average of the $e_{i,j} e_{j,k} e_{k,i}$'s:

$$t(\mathbf{Y}) = \frac{\sum_i \sum_j \sum_k e_{i,j} e_{j,k} e_{i,j}}{\# \text{ of ordered triads}}$$

Triad dependence measure

Let $\mathbf{E} = (\mathbf{Y} - \mathbf{1}\mathbf{1}^T \bar{y}_{\cdot\cdot})/s_y$, that is, $e_{i,j} = (y_{i,j} - \bar{y}_{\cdot\cdot})/s_y$.

- $\bar{y}_{\cdot\cdot}$ is the grand mean;
- s_y is the sample standard deviation of the $y_{i,j}$'s
- $e_{i,j}$ is like a "scaled residual" from the simple mean model $y_{i,j} = \mu + \epsilon_{i,j}$.

The sum of the diagonal of E^3 gives the scaled third-order moment:

$$\text{trace}(\mathbf{E}^3) = \sum_i \sum_j \sum_k e_{i,j} e_{j,k} e_{k,i}$$

The GOF statistic used in `amen` is the average of the $e_{i,j} e_{j,k} e_{k,i}$'s:

$$t(\mathbf{Y}) = \frac{\sum_i \sum_j \sum_k e_{i,j} e_{j,k} e_{i,j}}{\# \text{ of ordered triads}}$$

Triad dependence measure

Let $\mathbf{E} = (\mathbf{Y} - \mathbf{1}\mathbf{1}^T \bar{y}_{\cdot\cdot})/s_y$, that is, $e_{i,j} = (y_{i,j} - \bar{y}_{\cdot\cdot})/s_y$.

- $\bar{y}_{\cdot\cdot}$ is the grand mean;
- s_y is the sample standard deviation of the $y_{i,j}$'s
- $e_{i,j}$ is like a "scaled residual" from the simple mean model $y_{i,j} = \mu + \epsilon_{i,j}$.

The sum of the diagonal of E^3 gives the scaled third-order moment:

$$\text{trace}(\mathbf{E}^3) = \sum_i \sum_j \sum_k e_{i,j} e_{j,k} e_{k,i}$$

The GOF statistic used in `amen` is the average of the $e_{i,j} e_{j,k} e_{k,i}$'s:

$$t(\mathbf{Y}) = \frac{\sum_i \sum_j \sum_k e_{i,j} e_{j,k} e_{i,j}}{\#\text{ of ordered triads}}$$

Triad dependence and transitivity

This triadic dependence measure is related to **transitivity** for binary data:

Transitive triple: An ordered triple (i, j, k) is transitive if $i \rightarrow j \rightarrow k \rightarrow i$ $(y_{i,j} = y_{j,k} = y_{k,i})$.

$$\text{trans}(\mathbf{Y}) = \sum_i \sum_j \sum_k y_{i,j} y_{j,k} y_{k,i}$$

Triad dependence and transitivity

This triadic dependence measure is related to **transitivity** for binary data:

Transitive triple: An ordered triple (i, j, k) is transitive if $i \rightarrow j \rightarrow k \rightarrow i$ $(y_{i,j} = y_{j,k} = y_{k,i})$.

$$\text{trans}(\mathbf{Y}) = \sum_i \sum_j \sum_k y_{i,j} y_{j,k} y_{k,i}$$

Triad dependence and transitivity

This triadic dependence measure is related to **transitivity** for binary data:

Transitive triple: An ordered triple (i, j, k) is transitive if $i \rightarrow j \rightarrow k \rightarrow i$ $(y_{i,j} = y_{j,k} = y_{k,i})$.

$$\text{trans}(\mathbf{Y}) = \sum_i \sum_j \sum_k y_{i,j} y_{j,k} y_{k,i}$$

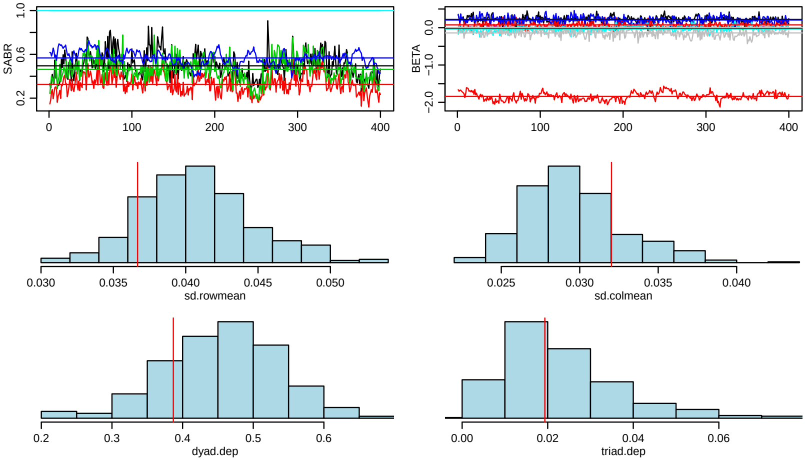
SRM fit to ordinal data with dyadic covariates

```
#### dyadic and nodal covariates
fit_dn<-ame(Y,Xdyad=Xd,Xrow=Xn,Xcol=Xn,model="ord",nscan=10000)
```

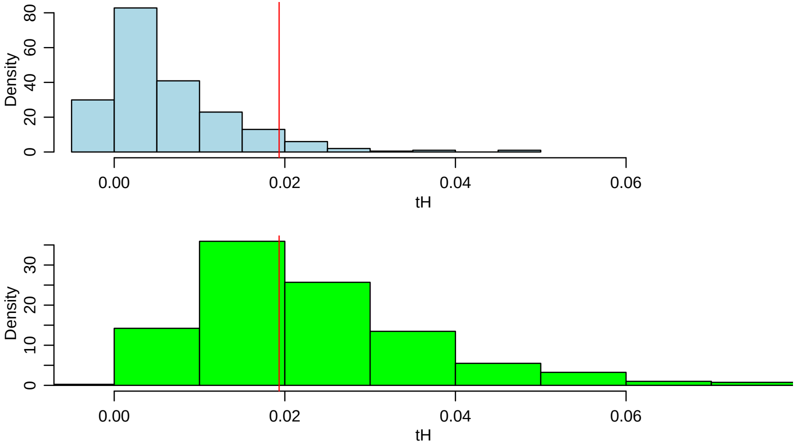
```
summary(fit_dn)

##
## beta:
##                      pmean   psd  z-stat p-val
## pop.row              0.212 0.095   2.230 0.026
## gdp.row              0.077 0.078   0.992 0.321
## polity.row          -0.028 0.013  -2.045 0.041
## pop.col              0.205 0.092   2.232 0.026
## gdp.col             -0.029 0.078  -0.376 0.707
## polity.col           0.002 0.012   0.130 0.896
## polity_int.dyad     -0.003 0.001  -2.419 0.016
## imports.dyad        -0.142 0.100  -1.415 0.157
## shared_igos.dyad    -0.017 0.005  -3.383 0.001
## distance.dyad       -1.837 0.101 -18.244 0.000
##
## Sigma_ab pmean:
##       a     b
## a 0.500 0.332
## b 0.332 0.462
##
## rho pmean:
##  0.566
```

SRM fit to ordinal data with dyadic covariates

```r
plot(fit_dn)
```



GOF comparison



Dyadic functions of nodal characteristics

Adding these dyadic covariates improved the fit with regard to $t(\mathbf{Y})$.

Note that each of these dyadic covariates was a function of nodal covariates:

- $x_{i,j,1} = f_1(\text{polity}_i, \text{polity}_j)$
- $x_{i,j,2} = f_2(\text{igo}_i, \text{igo}_j)$
- $x_{i,j,3} = f_3(\text{location}_i, \text{location}_j)$

This suggests models of the form

$$y_{i,j} \sim \beta_0 + \beta_1 x_{i,j}$$
$$x_{i,j} = s(x_i, x_j)$$

where $s(\cdot,\cdot)$ is some (non-additive) function of the nodal characteristics x_i, x_j.

Dyadic functions of nodal characteristics

Adding these dyadic covariates improved the fit with regard to $t(\mathbf{Y})$.

Note that each of these dyadic covariates was a function of nodal covariates:

- $x_{i,j,1} = f_1(\text{polity}_i, \text{polity}_j)$
- $x_{i,j,2} = f_2(\text{igo}_i, \text{igo}_j)$
- $x_{i,j,3} = f_3(\text{location}_i, \text{location}_j)$

This suggests models of the form

$$\begin{aligned} y_{i,j} &\sim \beta_0 + \beta_1 x_{i,j} \\ x_{i,j} &= s(x_i, x_j) \end{aligned}$$

where $s(\cdot,\cdot)$ is some (non-additive) function of the nodal characteristics x_i, x_j.

Dyadic functions of nodal characteristics

Adding these dyadic covariates improved the fit with regard to $t(\mathbf{Y})$.

Note that each of these dyadic covariates was a function of nodal covariates:

- $x_{i,j,1} = f_1(\text{polity}_i, \text{polity}_j)$
- $x_{i,j,2} = f_2(\text{igo}_i, \text{igo}_j)$
- $x_{i,j,3} = f_3(\text{location}_i, \text{location}_j)$

This suggests models of the form

$$y_{i,j} \sim \beta_0 + \beta_1 x_{i,j}$$
$$x_{i,j} = s(x_i, x_j)$$

where $s(\cdot,\cdot)$ is some (non-additive) function of the nodal characteristics x_i, x_j.

Dyadic functions of nodal characteristics

Adding these dyadic covariates improved the fit with regard to $t(\mathbf{Y})$.

Note that each of these dyadic covariates was a function of nodal covariates:

- $x_{i,j,1} = f_1(\text{polity}_i, \text{polity}_j)$
- $x_{i,j,2} = f_2(\text{igo}_i, \text{igo}_j)$
- $x_{i,j,3} = f_3(\text{location}_i, \text{location}_j)$

This suggests models of the form

$$\begin{aligned} y_{i,j} &\sim \beta_0 + \beta_1 x_{i,j} \\ x_{i,j} &= s(x_i, x_j) \end{aligned}$$

where $s(\cdot,\cdot)$ is some (non-additive) function of the nodal characteristics x_i, x_j.

Dyadic functions of nodal characteristics

Adding these dyadic covariates improved the fit with regard to $t(\mathbf{Y})$.

Note that each of these dyadic covariates was a function of nodal covariates:

- $x_{i,j,1} = f_1(\text{polity}_i, \text{polity}_j)$
- $x_{i,j,2} = f_2(\text{igo}_i, \text{igo}_j)$
- $x_{i,j,3} = f_3(\text{location}_i, \text{location}_j)$

This suggests models of the form

$$\begin{aligned} y_{i,j} &\sim \beta_0 + \beta_1 x_{i,j} \\ x_{i,j} &= s(x_i, x_j) \end{aligned}$$

where $s(\cdot,\cdot)$ is some (non-additive) function of the nodal characteristics x_i, x_j.

Dyadic functions of nodal characteristics

Adding these dyadic covariates improved the fit with regard to $t(\mathbf{Y})$.

Note that each of these dyadic covariates was a function of nodal covariates:

- $x_{i,j,1} = f_1(\text{polity}_i, \text{polity}_j)$
- $x_{i,j,2} = f_2(\text{igo}_i, \text{igo}_j)$
- $x_{i,j,3} = f_3(\text{location}_i, \text{location}_j)$

This suggests models of the form

$$y_{i,j} \sim \beta_0 + \beta_1 x_{i,j}$$
$$x_{i,j} = s(x_i, x_j)$$

where $s(\cdot,\cdot)$ is some (non-additive) function of the nodal characteristics x_i, x_j.

Homophily and stochastic equivalence

More generally, let

- u_i be a covariate of i as a sender of ties;
- v_j be covariate of j as a receiver of ties.

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$$

Such a model can describe various types of higher-order dependence, including

transitivity

stochastic equivalence

Homophily and stochastic equivalence

More generally, let

u_i be a covariate of i as a sender of ties;

v_j be covariate of j as a receiver of ties.

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$$

Such a model can describe various types of higher-order dependence, including

transitivity

stochastic equivalence

Homophily and stochastic equivalence

More generally, let

- u_i be a covariate of i as a sender of ties;
- v_j be covariate of j as a receiver of ties.

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$$

Such a model can describe various types of higher-order dependence, including

transitivity

stochastic equivalence

Transitivity via homophily

Homophily on covariates can explain transitivity:

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(x_i, x_j)$$

If $\beta_1 > 0$ and the $y_{i,j}$'s are binary, then

- $y_{i,j} = 1 \Rightarrow x_i \approx x_j$;
- $y_{i,k} = 1 \Rightarrow x_i \approx x_k$;
- $x_i \approx x_j, \; x_i \approx x_k \Rightarrow x_j \approx x_k$.
- $x_j \approx x_k \Rightarrow y_{j,k} = 1$.

Similarly, homophily can explain triadic dependence for ordinal $y_{i,j}$'s.

Transitivity via homophily

Homophily on covariates can explain transitivity:

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(x_i, x_j)$$

If $\beta_1 > 0$ and the $y_{i,j}$'s are binary, then

- $y_{i,j} = 1 \Rightarrow x_i \approx x_j$;
- $y_{i,k} = 1 \Rightarrow x_i \approx x_k$;
- $x_i \approx x_j,\ x_i \approx x_k \Rightarrow x_j \approx x_k$.
- $x_j \approx x_k \Rightarrow y_{j,k} = 1$.

Similarly, homophily can explain triadic dependence for ordinal $y_{i,j}$'s.

Transitivity via homophily

Homophily on covariates can explain transitivity:

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(x_i, x_j)$$

If $\beta_1 > 0$ and the $y_{i,j}$'s are binary, then

- $y_{i,j} = 1 \Rightarrow x_i \approx x_j$;
- $y_{i,k} = 1 \Rightarrow x_i \approx x_k$;
- $x_i \approx x_j, \; x_i \approx x_k \Rightarrow x_j \approx x_k$.
- $x_j \approx x_k \Rightarrow y_{j,k} = 1$.

Similarly, homophily can explain triadic dependence for ordinal $y_{i,j}$'s.

Transitivity via homophily

Homophily on covariates can explain transitivity:

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(x_i, x_j)$$

If $\beta_1 > 0$ and the $y_{i,j}$'s are binary, then

- $y_{i,j} = 1 \Rightarrow x_i \approx x_j$;
- $y_{i,k} = 1 \Rightarrow x_i \approx x_k$;
- $x_i \approx x_j,\ x_i \approx x_k \Rightarrow x_j \approx x_k$.
- $x_j \approx x_k \Rightarrow y_{j,k} = 1$.

Similarly, homophily can explain triadic dependence for ordinal $y_{i,j}$'s.

Transitivity via homophily

Homophily on covariates can explain transitivity:

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(x_i, x_j)$$

If $\beta_1 > 0$ and the $y_{i,j}$'s are binary, then

- $y_{i,j} = 1 \Rightarrow x_i \approx x_j$;
- $y_{i,k} = 1 \Rightarrow x_i \approx x_k$;
- $x_i \approx x_j, \; x_i \approx x_k \Rightarrow x_j \approx x_k$.
- $x_j \approx x_k \Rightarrow y_{j,k} = 1$.

Similarly, homophily can explain triadic dependence for ordinal $y_{i,j}$'s.

Transitivity via homophily

Homophily on covariates can explain transitivity:

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(x_i, x_j)$$

If $\beta_1 > 0$ and the $y_{i,j}$'s are binary, then

- $y_{i,j} = 1 \Rightarrow x_i \approx x_j$;
- $y_{i,k} = 1 \Rightarrow x_i \approx x_k$;
- $x_i \approx x_j, \; x_i \approx x_k \Rightarrow x_j \approx x_k$.
- $x_j \approx x_k \Rightarrow y_{j,k} = 1$.

Similarly, homophily can explain triadic dependence for ordinal $y_{i,j}$'s.

Stochastic equivalence

Returning to the more general model:

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$$

If $u_i = u_k$ then i and j are **equivalent as senders** in terms of the model.

Example: Logistic regression

$$\Pr(Y_{i,j} = 1) = \frac{e^{\beta_0 + \beta_1 s(u_i, v_j)}}{1 + e^{\beta_0 + \beta_1 s(u_i, v_j)}}$$

If $u_i = u_k = u$, then

$$\Pr(Y_{i,j} = 1) = \frac{e^{\beta_0 + \beta_1 s(u_i, v_j)}}{1 + e^{\beta_0 + \beta_1 s(u_i, v_j)}} = \Pr(Y_{k,j} = 1),$$

and nodes i and k are **stochastically equivalent** (as senders).

Stochastic equivalence

Returning to the more general model:

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$$

If $u_i = u_k$ then i and j are **equivalent as senders** in terms of the model.

Example: Logistic regression

$$\Pr(Y_{i,j} = 1) = \frac{e^{\beta_0 + \beta_1 s(u_i, v_j)}}{1 + e^{\beta_0 + \beta_1 s(u_i, v_j)}}$$

If $u_i = u_k = u$, then

$$\Pr(Y_{i,j} = 1) = \frac{e^{\beta_0 + \beta_1 s(u_i, v_j)}}{1 + e^{\beta_0 + \beta_1 s(u_i, v_j)}} = \Pr(Y_{k,j} = 1),$$

and nodes i and k are **stochastically equivalent** (as senders).

Stochastic equivalence

Returning to the more general model:

$$y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$$

If $u_i = u_k$ then i and j are **equivalent as senders** in terms of the model.

Example: Logistic regression

$$\Pr(Y_{i,j} = 1) = \frac{e^{\beta_0 + \beta_1 s(u_i, v_j)}}{1 + e^{\beta_0 + \beta_1 s(u_i, v_j)}}$$

If $u_i = u_k = u$, then

$$\Pr(Y_{i,j} = 1) = \frac{e^{\beta_0 + \beta_1 s(u_i, v_j)}}{1 + e^{\beta_0 + \beta_1 s(u_i, v_j)}} = \Pr(Y_{k,j} = 1),$$

and nodes i and k are **stochastically equivalent** (as senders).

Homophily and stochastic equivalence

In our hypothetical model of social relations, we've seen how nodal characteristics relate to homophily and stochastic equivalence.

Homophily: Similar nodes link to each other

- "similar" in terms of characteristics (potentially unobserved)
- homophily leads to transitive or clustered social networks
- observed transitivity may be due to exogenous or endogenous factors

(See Shalizi and Thomas 2010 for a more careful discussion)

Stochastic equivalence: Similar nodes have similar relational patterns

- similar nodes may or may not link to each other
- equivalent nodes can be thought of as having the same "role"

Homophily and stochastic equivalence

In our hypothetical model of social relations, we've seen how nodal characteristics relate to homophily and stochastic equivalence.

Homophily: Similar nodes link to each other

- "similar" in terms of characteristics (potentially unobserved)
- homophily leads to transitive or clustered social networks
- observed transitivity may be due to exogenous or endogenous factors

(See Shalizi and Thomas 2010 for a more careful discussion)

Stochastic equivalence: Similar nodes have similar relational patterns

- similar nodes may or may not link to each other
- equivalent nodes can be thought of as having the same "role"

Homophily and stochastic equivalence

In our hypothetical model of social relations, we've seen how nodal characteristics relate to homophily and stochastic equivalence.

Homophily: Similar nodes link to each other

- "similar" in terms of characteristics (potentially unobserved)
- homophily leads to transitive or clustered social networks
- observed transitivity may be due to exogenous or endogenous factors

(See Shalizi and Thomas 2010 for a more careful discussion)

Stochastic equivalence: Similar nodes have similar relational patterns

- similar nodes may or may not link to each other
- equivalent nodes can be thought of as having the same "role"

Homophily and stochastic equivalence

In our hypothetical model of social relations, we've seen how nodal characteristics relate to homophily and stochastic equivalence.

Homophily: Similar nodes link to each other

- "similar" in terms of characteristics (potentially unobserved)
- homophily leads to transitive or clustered social networks
- observed transitivity may be due to exogenous or endogenous factors

(See Shalizi and Thomas 2010 for a more careful discussion)

Stochastic equivalence: Similar nodes have similar relational patterns

- similar nodes may or may not link to each other
- equivalent nodes can be thought of as having the same "role"

Homophily and stochastic equivalence

In our hypothetical model of social relations, we've seen how nodal characteristics relate to homophily and stochastic equivalence.

Homophily: Similar nodes link to each other

- "similar" in terms of characteristics (potentially unobserved)
- homophily leads to transitive or clustered social networks
- observed transitivity may be due to exogenous or endogenous factors

(See Shalizi and Thomas 2010 for a more careful discussion)

Stochastic equivalence: Similar nodes have similar relational patterns

- similar nodes may or may not link to each other
- equivalent nodes can be thought of as having the same "role"

Homophily and stochastic equivalence

In our hypothetical model of social relations, we've seen how nodal characteristics relate to homophily and stochastic equivalence.

Homophily: Similar nodes link to each other

- "similar" in terms of characteristics (potentially unobserved)
- homophily leads to transitive or clustered social networks
- observed transitivity may be due to exogenous or endogenous factors

(See Shalizi and Thomas 2010 for a more careful discussion)

Stochastic equivalence: Similar nodes have similar relational patterns

- similar nodes may or may not link to each other
- equivalent nodes can be thought of as having the same "role"

Homophily and stochastic equivalence

In our hypothetical model of social relations, we've seen how nodal characteristics relate to homophily and stochastic equivalence.

Homophily: Similar nodes link to each other

- "similar" in terms of characteristics (potentially unobserved)
- homophily leads to transitive or clustered social networks
- observed transitivity may be due to exogenous or endogenous factors

(See Shalizi and Thomas 2010 for a more careful discussion)

Stochastic equivalence: Similar nodes have similar relational patterns

- similar nodes may or may not link to each other
- equivalent nodes can be thought of as having the same "role"

Homophily and stochastic equivalence

In our hypothetical model of social relations, we've seen how nodal characteristics relate to homophily and stochastic equivalence.

Homophily: Similar nodes link to each other

- "similar" in terms of characteristics (potentially unobserved)
- homophily leads to transitive or clustered social networks
- observed transitivity may be due to exogenous or endogenous factors

(See Shalizi and Thomas 2010 for a more careful discussion)

Stochastic equivalence: Similar nodes have similar relational patterns

- similar nodes may or may not link to each other
- equivalent nodes can be thought of as having the same "role"

Homophily and stochastic equivalence

In our hypothetical model of social relations, we've seen how nodal characteristics relate to homophily and stochastic equivalence.

Homophily: Similar nodes link to each other

- "similar" in terms of characteristics (potentially unobserved)
- homophily leads to transitive or clustered social networks
- observed transitivity may be due to exogenous or endogenous factors

(See Shalizi and Thomas 2010 for a more careful discussion)

Stochastic equivalence: Similar nodes have similar relational patterns

- similar nodes may or may not link to each other
- equivalent nodes can be thought of as having the same "role"

Homophily and stochastic equivalence

In our hypothetical model of social relations, we've seen how nodal characteristics relate to homophily and stochastic equivalence.

Homophily: Similar nodes link to each other

- "similar" in terms of characteristics (potentially unobserved)
- homophily leads to transitive or clustered social networks
- observed transitivity may be due to exogenous or endogenous factors

(See Shalizi and Thomas 2010 for a more careful discussion)

Stochastic equivalence: Similar nodes have similar relational patterns

- similar nodes may or may not link to each other
- equivalent nodes can be thought of as having the same "role"

Homophily and stochastic equivalence

In our hypothetical model of social relations, we've seen how nodal characteristics relate to homophily and stochastic equivalence.

Homophily: Similar nodes link to each other

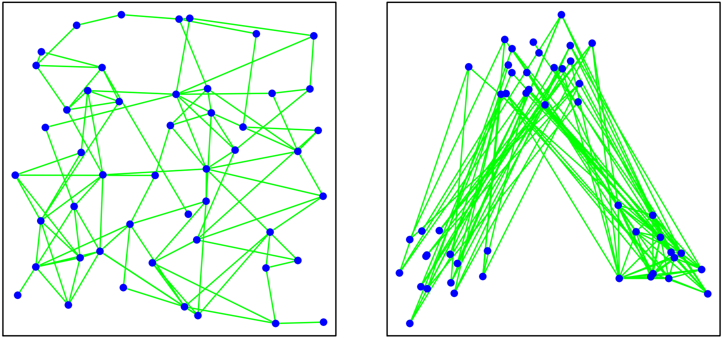
- "similar" in terms of characteristics (potentially unobserved)
- homophily leads to transitive or clustered social networks
- observed transitivity may be due to exogenous or endogenous factors

(See Shalizi and Thomas 2010 for a more careful discussion)

Stochastic equivalence: Similar nodes have similar relational patterns

- similar nodes may or may not link to each other
- equivalent nodes can be thought of as having the same "role"

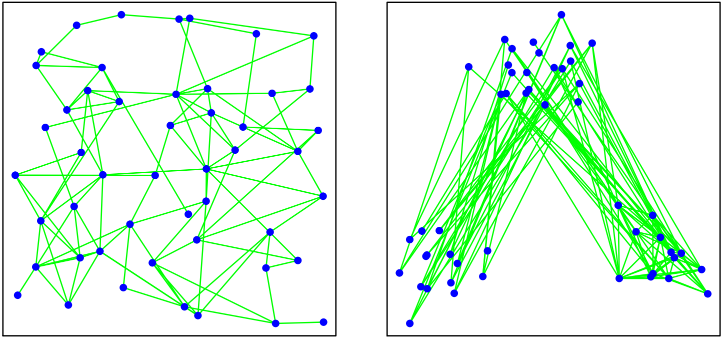
Visualizing stochastic equivalence



For which network is homophily a plausible explanation?

Which network exhibits a large degree of stochastic equivalence?

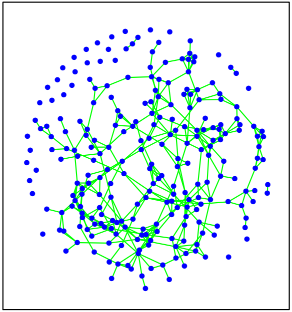
Visualizing stochastic equivalence

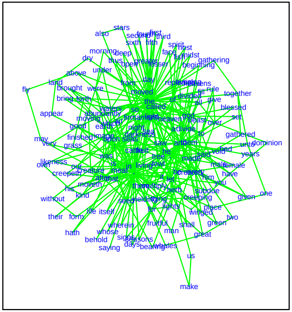


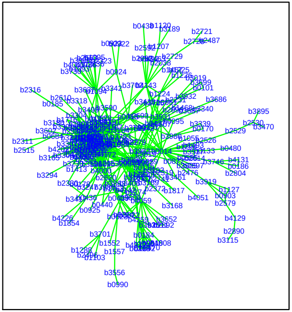
For which network is homophily a plausible explanation?

Which network exhibits a large degree of stochastic equivalence?

Homophily and stochastic equivalence in real networks

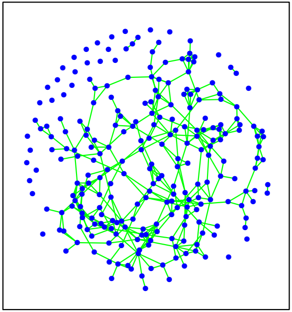


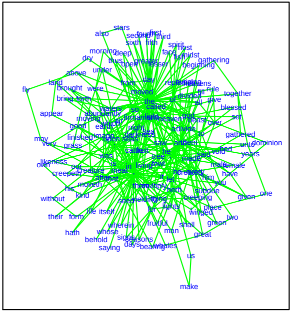


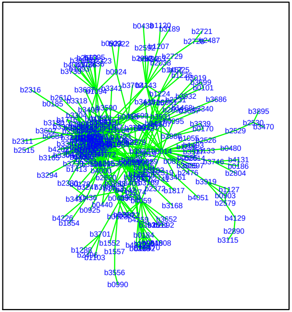


- AddHealth friendships: friendships among 247 12th-graders
- Word neighbors in Genesis: neighboring occurrences among 158 words
- Protein binding interactions: binding patterns among 230 proteins

Homophily and stochastic equivalence in real networks

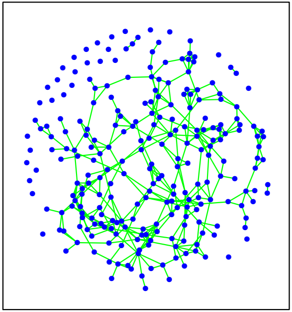


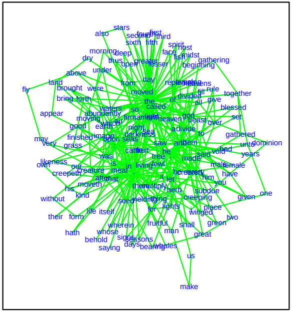


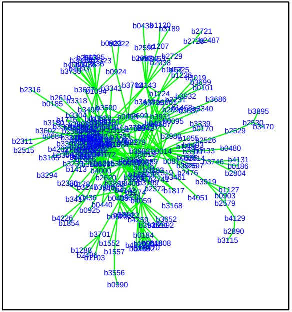


- AddHealth friendships: friendships among 247 12th-graders
- Word neighbors in Genesis: neighboring occurrences among 158 words
- Protein binding interactions: binding patterns among 230 proteins

Homophily and stochastic equivalence in real networks

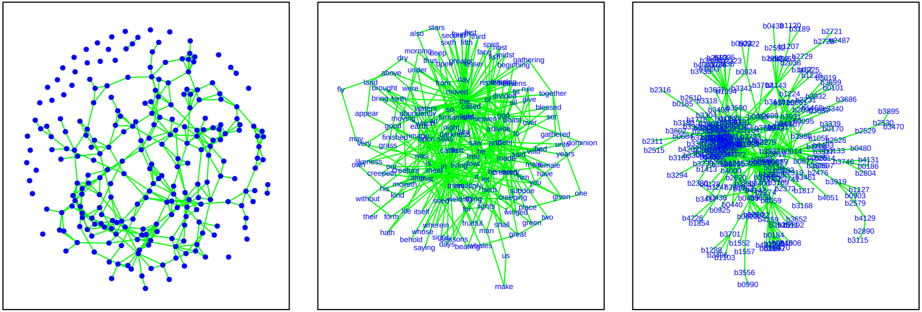






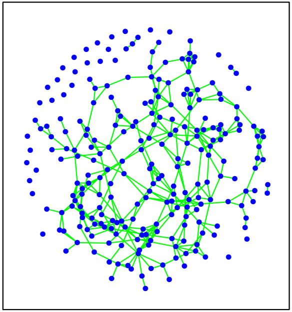
- AddHealth friendships: friendships among 247 12th-graders
- Word neighbors in Genesis: neighboring occurrences among 158 words
- Protein binding interactions: binding patterns among 230 proteins

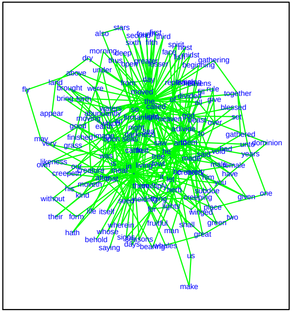
Homophily and stochastic equivalence in real networks

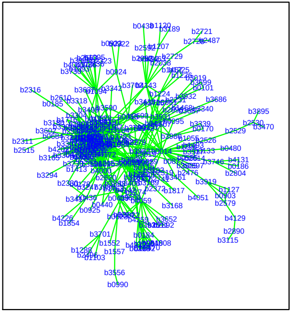


- AddHealth friendships: friendships among 247 12th-graders
- Word neighbors in Genesis: neighboring occurrences among 158 words
- Protein binding interactions: binding patterns among 230 proteins

Homophily and stochastic equivalence in real networks







- AddHealth friendships: friendships among 247 12th-graders
- Word neighbors in Genesis: neighboring occurrences among 158 words
- Protein binding interactions: binding patterns among 230 proteins

Latent factor models

Homophily and stochastic equivalence from unobserved variables can be represented with a **latent factor model**:

$$y_{i,j} \sim \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j,$$

- $\boldsymbol{u}_i$ is a vector of latent factors describing i as a sender of ties;
- $\boldsymbol{v}_j$ is a vector of latent factors describing j as a receiver of ties;
- $\mathbf{D}$ is a diagonal matrix of factor weights.

Normal, binomial, ordinal data can be represented with this structure as follows:

$$z_{i,j} = \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

where g is some increasing function:

- $g(z) = z$ for normal data;
- $g(z) = 1(z > 0)$ for binomial data;
- $g(z) =$ some increasing step function for ordinal data.

Latent factor models

Homophily and stochastic equivalence from unobserved variables can be represented with a **latent factor model**:

$$y_{i,j} \sim \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j,$$

- $\boldsymbol{u}_i$ is a vector of latent factors describing i as a sender of ties;
- $\boldsymbol{v}_j$ is a vector of latent factors describing j as a receiver of ties;
- $\mathbf{D}$ is a diagonal matrix of factor weights.

Normal, binomial, ordinal data can be represented with this structure as follows:

$$z_{i,j} = \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

where g is some increasing function:

- $g(z) = z$ for normal data;
- $g(z) = 1(z > 0)$ for binomial data;
- $g(z) =$ some increasing step function for ordinal data.

Latent factor models

Homophily and stochastic equivalence from unobserved variables can be represented with a **latent factor model**:

$$y_{i,j} \sim \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j,$$

- $\boldsymbol{u}_i$ is a vector of latent factors describing i as a sender of ties;
- $\boldsymbol{v}_j$ is a vector of latent factors describing j as a receiver of ties;
- $\mathbf{D}$ is a diagonal matrix of factor weights.

Normal, binomial, ordinal data can be represented with this structure as follows:

$$z_{i,j} = \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

where g is some increasing function:

- $g(z) = z$ for normal data;
- $g(z) = 1(z > 0)$ for binomial data;
- $g(z) =$ some increasing step function for ordinal data.

Latent factor models

Homophily and stochastic equivalence from unobserved variables can be represented with a **latent factor model**:

$$y_{i,j} \sim \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j,$$

- $\boldsymbol{u}_i$ is a vector of latent factors describing i as a sender of ties;
- $\boldsymbol{v}_j$ is a vector of latent factors describing j as a receiver of ties;
- $\mathbf{D}$ is a diagonal matrix of factor weights.

Normal, binomial, ordinal data can be represented with this structure as follows:

$$\begin{aligned} z_{i,j} &= \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j} \\ y_{i,j} &= g(z_{i,j}) \end{aligned}$$

where g is some increasing function:

- $g(z) = z$ for normal data;
- $g(z) = 1(z > 0)$ for binomial data;
- $g(z) =$ some increasing step function for ordinal data.

Latent factor models

Homophily and stochastic equivalence from unobserved variables can be represented with a **latent factor model**:

$$y_{i,j} \sim \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j,$$

- $\boldsymbol{u}_i$ is a vector of latent factors describing i as a sender of ties;
- $\boldsymbol{v}_j$ is a vector of latent factors describing j as a receiver of ties;
- $\mathbf{D}$ is a diagonal matrix of factor weights.

Normal, binomial, ordinal data can be represented with this structure as follows:

$$z_{i,j} = \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

where g is some increasing function:

- $g(z) = z$ for normal data;
- $g(z) = 1(z > 0)$ for binomial data;
- $g(z) =$ some increasing step function for ordinal data.

Latent factor models

Homophily and stochastic equivalence from unobserved variables can be represented with a **latent factor model**:

$$y_{i,j} \sim \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j,$$

- $\boldsymbol{u}_i$ is a vector of latent factors describing i as a sender of ties;
- $\boldsymbol{v}_j$ is a vector of latent factors describing j as a receiver of ties;
- $\mathbf{D}$ is a diagonal matrix of factor weights.

Normal, binomial, ordinal data can be represented with this structure as follows:

$$z_{i,j} = \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

where g is some increasing function:

- $g(z) = z$ for normal data;
- $g(z) = 1(z > 0)$ for binomial data;
- $g(z) =$ some increasing step function for ordinal data.

Latent factor models

Homophily and stochastic equivalence from unobserved variables can be represented with a **latent factor model**:

$$y_{i,j} \sim \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j,$$

- $\boldsymbol{u}_i$ is a vector of latent factors describing i as a sender of ties;
- $\boldsymbol{v}_j$ is a vector of latent factors describing j as a receiver of ties;
- $\mathbf{D}$ is a diagonal matrix of factor weights.

Normal, binomial, ordinal data can be represented with this structure as follows:

$$z_{i,j} = \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

where g is some increasing function:

- $g(z) = z$ for normal data;
- $g(z) = 1(z > 0)$ for binomial data;
- $g(z) =$ some increasing step function for ordinal data.

Latent factor models

Homophily and stochastic equivalence from unobserved variables can be represented with a **latent factor model**:

$$y_{i,j} \sim \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j,$$

- $\boldsymbol{u}_i$ is a vector of latent factors describing i as a sender of ties;
- $\boldsymbol{v}_j$ is a vector of latent factors describing j as a receiver of ties;
- $\mathbf{D}$ is a diagonal matrix of factor weights.

Normal, binomial, ordinal data can be represented with this structure as follows:

$$\begin{aligned} z_{i,j} &= \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j} \\ y_{i,j} &= g(z_{i,j}) \end{aligned}$$

where g is some increasing function:

- $g(z) = z$ for normal data;
- $g(z) = 1(z > 0)$ for binomial data;
- $g(z) =$ some increasing step function for ordinal data.

Latent factor models

Homophily and stochastic equivalence from unobserved variables can be represented with a **latent factor model**:

$$y_{i,j} \sim \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j,$$

- $\boldsymbol{u}_i$ is a vector of latent factors describing i as a sender of ties;
- $\boldsymbol{v}_j$ is a vector of latent factors describing j as a receiver of ties;
- $\mathbf{D}$ is a diagonal matrix of factor weights.

Normal, binomial, ordinal data can be represented with this structure as follows:

$$\begin{aligned} z_{i,j} &= \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j} \\ y_{i,j} &= g(z_{i,j}) \end{aligned}$$

where g is some increasing function:

- $g(z) = z$ for normal data;
- $g(z) = 1(z > 0)$ for binomial data;
- $g(z) =$ some increasing step function for ordinal data.

Understanding latent factors

$$\mathbf{Z} = \mathbf{U}^T \mathbf{D} \mathbf{V} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{D} \mathbf{v}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} d_r u_{i,r} v_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a 2 factor model, we have

$$z_{i,j} = d_1 (u_{i,1} \times v_{j,1}) + d_2 (u_{i,2} \times v_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_i \approx u_j$: similarity of latent factors implies approximate stoch equivalence;
- $u_i \approx v_j$: similarity of latent factors implies high probability of a tie.

Understanding latent factors

$$\mathbf{Z} = \mathbf{U}^T\mathbf{D}\mathbf{V} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T\mathbf{D}\mathbf{v}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} d_r u_{i,r} v_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a 2 factor model, we have

$$z_{i,j} = d_1(u_{i,1} \times v_{j,1}) + d_2(u_{i,2} \times v_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_i \approx u_j$: similarity of latent factors implies approximate stoch equivalence;
- $u_i \approx v_j$: similarity of latent factors implies high probability of a tie.

Understanding latent factors

$$\mathbf{Z} = \mathbf{U}^T \mathbf{D} \mathbf{V} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{D} \mathbf{v}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} d_r u_{i,r} v_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a 2 factor model, we have

$$z_{i,j} = d_1(u_{i,1} \times v_{j,1}) + d_2(u_{i,2} \times v_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_i \approx u_j$: similarity of latent factors implies approximate stoch equivalence;
- $u_i \approx v_j$: similarity of latent factors implies high probability of a tie.

Understanding latent factors

$$\mathbf{Z} = \mathbf{U}^T \mathbf{D} \mathbf{V} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{D} \mathbf{v}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} d_r u_{i,r} v_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a 2 factor model, we have

$$z_{i,j} = d_1 (u_{i,1} \times v_{j,1}) + d_2 (u_{i,2} \times v_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $\boldsymbol{u}_i \approx \boldsymbol{u}_j$: similarity of latent factors implies approximate stoch equivalence;
- $\boldsymbol{u}_i \approx \boldsymbol{v}_j$: similarity of latent factors implies high probability of a tie.

Understanding latent factors

$$\mathbf{Z} = \mathbf{U}^T\mathbf{D}\mathbf{V} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T\mathbf{D}\mathbf{v}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} d_r u_{i,r} v_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a 2 factor model, we have

$$z_{i,j} = d_1(u_{i,1} \times v_{j,1}) + d_2(u_{i,2} \times v_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $\boldsymbol{u}_i \approx \boldsymbol{u}_j$: similarity of latent factors implies approximate stoch equivalence;
- $\boldsymbol{u}_i \approx \boldsymbol{v}_j$: similarity of latent factors implies high probability of a tie.

Understanding latent factors

$$\mathbf{Z} = \mathbf{U}^T \mathbf{D} \mathbf{V} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{D} \mathbf{v}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} d_r u_{i,r} v_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a 2 factor model, we have

$$z_{i,j} = d_1(u_{i,1} \times v_{j,1}) + d_2(u_{i,2} \times v_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $\boldsymbol{u}_i \approx \boldsymbol{u}_j$: similarity of latent factors implies approximate stoch equivalence;
- $\boldsymbol{u}_i \approx \boldsymbol{v}_j$: similarity of latent factors implies high probability of a tie.

Understanding latent factors

$$\mathbf{Z} = \mathbf{U}^T \mathbf{D} \mathbf{V} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{D} \mathbf{v}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} d_r u_{i,r} v_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a 2 factor model, we have

$$z_{i,j} = d_1 (u_{i,1} \times v_{j,1}) + d_2 (u_{i,2} \times v_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $\boldsymbol{u}_i \approx \boldsymbol{u}_j$: similarity of latent factors implies approximate stoch equivalence;
- $\boldsymbol{u}_i \approx \boldsymbol{v}_j$: similarity of latent factors implies high probability of a tie.

Matrix decomposition interpretation

Recall from linear algebra:

- Every $m \times n$ matrix $\mathbf{Z}$ can be written

$$\mathbf{Z} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

  where $\mathbf{D} = \text{diag}(d_1, \ldots, d_n)$, $\mathbf{U}$ and $\mathbf{V}$ are orthonormal.

- If $\mathbf{U}\mathbf{D}\mathbf{V}^T$ is the svd of $\mathbf{Z}$, then

$$\hat{\mathbf{Z}}_k = \mathbf{U}_{[,1:k]}\mathbf{D}_{[1:k,1:k]}\mathbf{V}^T_{[,1:k]}$$

  is the least-squares rank-k approximation to $\mathbf{Z}$.

Matrix decomposition interpretation

Recall from linear algebra:

- Every $m \times n$ matrix $\mathbf{Z}$ can be written

  $$\mathbf{Z} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

  where $\mathbf{D} = \text{diag}(d_1, \ldots, d_n)$, $\mathbf{U}$ and $\mathbf{V}$ are orthonormal.

- If $\mathbf{U}\mathbf{D}\mathbf{V}^T$ is the svd of $\mathbf{Z}$, then

  $$\hat{\mathbf{Z}}_k = \mathbf{U}_{[,1:k]}\mathbf{D}_{[1:k,1:k]}\mathbf{V}^T_{[,1:k]}$$

  is the least-squares rank-k approximation to $\mathbf{Z}$.

Matrix decomposition interpretation

Recall from linear algebra:

- Every $m \times n$ matrix $\mathbf{Z}$ can be written

$$\mathbf{Z} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

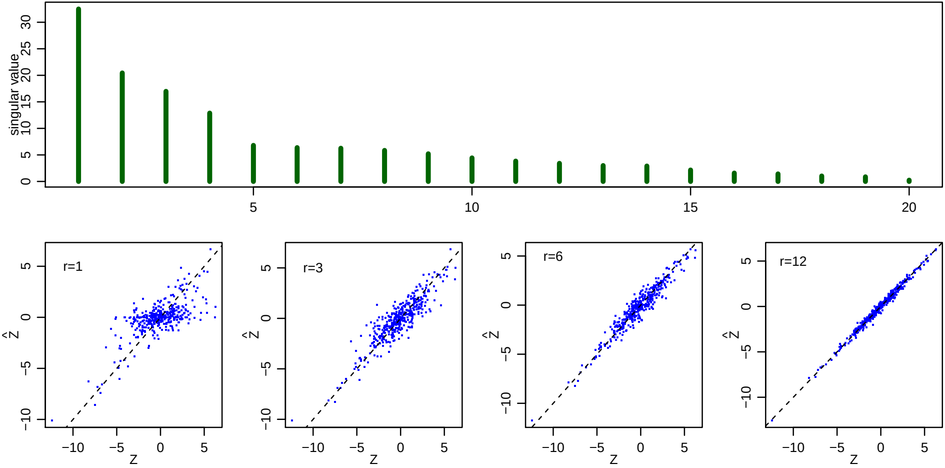
  where $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n)$, $\mathbf{U}$ and $\mathbf{V}$ are orthonormal.

- If $\mathbf{U}\mathbf{D}\mathbf{V}^T$ is the svd of $\mathbf{Z}$, then

$$\hat{\mathbf{Z}}_k \equiv \mathbf{U}_{[,1:k]}\mathbf{D}_{[1:k,1:k]}\mathbf{V}^T_{[,1:k]}$$

  is the least-squares rank-k approximation to $\mathbf{Z}$.

Least squares matrix approximations



LFM for symmetric data

Probit version of the symmetric latent factor model:

$$\begin{aligned} y_{i,j} &= g(z_{i,j}) \ , && \text{where } g \text{ is a nondecreasing function} \\ z_{i,j} &= \mathbf{u}_i^T \Lambda \mathbf{u}_j + \epsilon_{i,j} \ , && \text{where } \mathbf{u}_i \in \mathbb{R}^K \ , \ \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_K) \\ \{\epsilon_{i,j}\} &\overset{iid}{\sim} \text{normal}(0,1) && \end{aligned}$$

Writing $\{z_{i,j}\}$ as a matrix ,

$$\mathbf{Z} = \mathbf{U} \Lambda \mathbf{U}^T + \mathbf{E}$$

Recall from linear algebra:

- Every $n \times n$ symmetric matrix $\mathbf{Z}$ can be written

  $$\mathbf{Z} = \mathbf{U} \Lambda \mathbf{U}^T$$

  where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and $\mathbf{U}$ is orthonormal.
- If $\mathbf{U} \Lambda \mathbf{U}^T$ is the eigendecomposition of $\mathbf{Z}$, then

  $$\hat{\mathbf{Z}}_k = \mathbf{U}_{[,1:k]} \Lambda_{[1:k,1:k]} \mathbf{U}^T_{[,1:k]}$$

  is the least-squares rank-k approximation to $\mathbf{Z}$.

LFM for symmetric data

Probit version of the symmetric latent factor model:

$$\begin{aligned} y_{i,j} &= g(z_{i,j}) \ , && \text{where } g \text{ is a nondecreasing function} \\ z_{i,j} &= \mathbf{u}_i^T \Lambda \mathbf{u}_j + \epsilon_{i,j} \ , && \text{where } \mathbf{u}_i \in \mathbb{R}^K \ , \ \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_K) \\ \{\epsilon_{i,j}\} &\overset{iid}{\sim} \text{normal}(0,1) && \end{aligned}$$

Writing $\{z_{i,j}\}$ as a matrix ,

$$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T + \mathbf{E}$$

Recall from linear algebra:

- Every $n \times n$ symmetric matrix $\mathbf{Z}$ can be written

$$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T$$

where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and $\mathbf{U}$ is orthonormal.

- If $\mathbf{U}\Lambda\mathbf{U}^T$ is the eigendecomposition of $\mathbf{Z}$, then

$$\hat{\mathbf{Z}}_k \equiv \mathbf{U}_{[,1:k]} \Lambda_{[1:k,1:k]} \mathbf{U}^T_{[,1:k]}$$

is the least-squares rank-k approximation to $\mathbf{Z}$.

LFM for symmetric data

Probit version of the symmetric latent factor model:

$$
\begin{aligned}
y_{i,j} &= g(z_{i,j}) \;, & &\text{where } g \text{ is a nondecreasing function} \\
z_{i,j} &= \mathbf{u}_i^T \Lambda \mathbf{u}_j + \epsilon_{i,j} \;, & &\text{where } \mathbf{u}_i \in \mathbb{R}^K \;, \; \Lambda{=}\text{diag}(\lambda_1, \ldots, \lambda_K) \\
\{\epsilon_{i,j}\} &\overset{iid}{\sim} \text{normal}(0,1)
\end{aligned}
$$

Writing $\{z_{i,j}\}$ as a matrix ,

$$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T + \mathbf{E}$$

Recall from linear algebra:

- Every $n \times n$ symmetric matrix $\mathbf{Z}$ can be written

  $$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T$$

  where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and $\mathbf{U}$ is orthonormal.
- If $\mathbf{U}\Lambda\mathbf{U}^T$ is the eigendecomposition of $\mathbf{Z}$, then

  $$\hat{\mathbf{Z}}_k \equiv \mathbf{U}_{[,1:k]}\Lambda_{[1:k,1:k]}\mathbf{U}_{[,1:k]}^T$$

  is the least-squares rank-k approximation to $\mathbf{Z}$.

LFM for symmetric data

Probit version of the symmetric latent factor model:

$$\begin{aligned} y_{i,j} &= g(z_{i,j}) \ , && \text{where } g \text{ is a nondecreasing function} \\ z_{i,j} &= \mathbf{u}_i^T \Lambda \mathbf{u}_j + \epsilon_{i,j} \ , && \text{where } \mathbf{u}_i \in \mathbb{R}^K \ , \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_K) \\ \{\epsilon_{i,j}\} &\overset{iid}{\sim} \text{normal}(0,1) \end{aligned}$$

Writing $\{z_{i,j}\}$ as a matrix ,

$$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T + \mathbf{E}$$

Recall from linear algebra:

- Every $n \times n$ symmetric matrix $\mathbf{Z}$ can be written

  $$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T$$

  where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and $\mathbf{U}$ is orthonormal.
- If $\mathbf{U}\Lambda\mathbf{U}^T$ is the eigendecomposition of $\mathbf{Z}$, then

  $$\hat{\mathbf{Z}}_k = \mathbf{U}_{[,1:k]}\Lambda_{[1:k,1:k]}\mathbf{U}_{[,1:k]}^T$$

  is the least-squares rank-k approximation to $\mathbf{Z}$.

LFM for symmetric data

Probit version of the symmetric latent factor model:

$$\begin{aligned} y_{i,j} &= g(z_{i,j}) \ , && \text{where } g \text{ is a nondecreasing function} \\ z_{i,j} &= \mathbf{u}_i^T \Lambda \mathbf{u}_j + \epsilon_{i,j} \ , && \text{where } \mathbf{u}_i \in \mathbb{R}^K \ , \ \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_K) \\ \{\epsilon_{i,j}\} &\overset{iid}{\sim} \text{normal}(0,1) \end{aligned}$$

Writing $\{z_{i,j}\}$ as a matrix ,

$$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T + \mathbf{E}$$

Recall from linear algebra:

- Every $n \times n$ symmetric matrix $\mathbf{Z}$ can be written

  $$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T$$

  where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and $\mathbf{U}$ is orthonormal.
- If $\mathbf{U}\Lambda\mathbf{U}^T$ is the eigendecomposition of $\mathbf{Z}$, then

  $$\hat{\mathbf{Z}}_k \equiv \mathbf{U}_{[,1:k]}\Lambda_{[1:k,1:k]}\mathbf{U}_{[,1:k]}^T$$

  is the least-squares rank-k approximation to $\mathbf{Z}$.

LFM for symmetric data

Probit version of the symmetric latent factor model:

$$
\begin{aligned}
y_{i,j} &= g(z_{i,j}) \ , && \text{where } g \text{ is a nondecreasing function} \\
z_{i,j} &= \mathbf{u}_i^T \Lambda \mathbf{u}_j + \epsilon_{i,j} \ , && \text{where } \mathbf{u}_i \in \mathbb{R}^K \ , \ \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_K) \\
\{\epsilon_{i,j}\} &\overset{iid}{\sim} \text{normal}(0,1)
\end{aligned}
$$

Writing $\{z_{i,j}\}$ as a matrix ,

$$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T + \mathbf{E}$$

Recall from linear algebra:

- Every $n \times n$ symmetric matrix $\mathbf{Z}$ can be written

  $$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T$$

  where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and $\mathbf{U}$ is orthonormal.
- If $\mathbf{U}\Lambda\mathbf{U}^T$ is the eigendecomposition of $\mathbf{Z}$, then

  $$\hat{\mathbf{Z}}_k \equiv \mathbf{U}_{[,1:k]}\Lambda_{[1:k,1:k]}\mathbf{U}_{[,1:k]}^T$$

  is the least-squares rank-k approximation to $\mathbf{Z}$.

LFM for symmetric data

Probit version of the symmetric latent factor model:

$$\begin{aligned} y_{i,j} &= g(z_{i,j}) \;, && \text{where } g \text{ is a nondecreasing function} \\ z_{i,j} &= \mathbf{u}_i^T \Lambda \mathbf{u}_j + \epsilon_{i,j} \;, && \text{where } \mathbf{u}_i \in \mathbb{R}^K \;, \; \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_K) \\ \{\epsilon_{i,j}\} &\overset{iid}{\sim} \text{normal}(0,1) \end{aligned}$$

Writing $\{z_{i,j}\}$ as a matrix ,

$$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T + \mathbf{E}$$

Recall from linear algebra:

- Every $n \times n$ symmetric matrix $\mathbf{Z}$ can be written

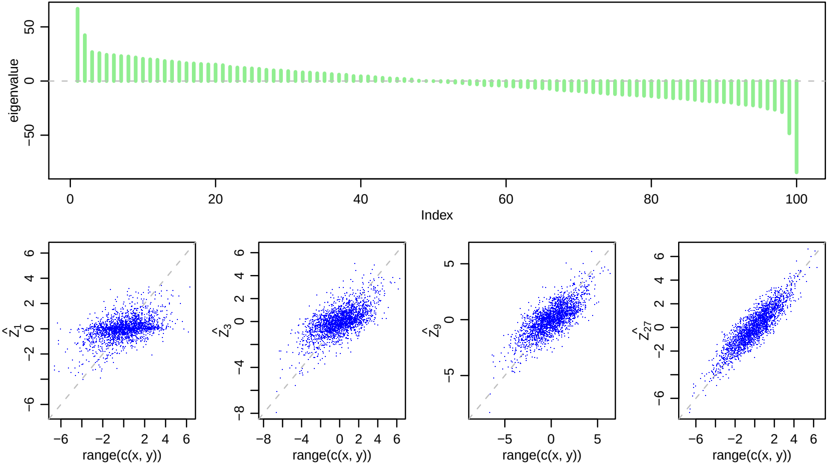
  $$\mathbf{Z} = \mathbf{U}\Lambda\mathbf{U}^T$$

  where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and $\mathbf{U}$ is orthonormal.
- If $\mathbf{U}\Lambda\mathbf{U}^T$ is the eigendecomposition of $\mathbf{Z}$, then

  $$\hat{\mathbf{Z}}_k \equiv \mathbf{U}_{[,1:k]} \Lambda_{[1:k,1:k]} \mathbf{U}_{[,1:k]}^T$$

  is the least-squares rank-k approximation to $\mathbf{Z}$.

Least squares approximations of increasing rank



Understanding eigenvectors and eigenvalues

$$\mathbf{Z} = \mathbf{U}^T \boldsymbol{\Lambda} \mathbf{U} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \boldsymbol{\Lambda} \mathbf{u}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} \lambda_r u_{i,r} u_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a rank-2 model, we have

$$z_{i,j} = \lambda_1 (u_{i,1} \times u_{j,1}) + \lambda_2 (u_{i,2} \times u_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_{i,r} \approx u_{j,r}$: equality of latent factors; represents stochastic equivalence;
- $\lambda_r > 0$: positive eigenvalues represent homophily;
- $\lambda_r < 0$: negative eigenvalues represent antihomophily.

Understanding eigenvectors and eigenvalues

$$\mathbf{Z} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{\Lambda} \mathbf{u}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} \lambda_r u_{i,r} u_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a rank-2 model, we have

$$z_{i,j} = \lambda_1 (u_{i,1} \times u_{j,1}) + \lambda_2 (u_{i,2} \times u_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_{i,r} \approx u_{j,r}$: equality of latent factors; represents stochastic equivalence;
- $\lambda_r > 0$: positive eigenvalues represent homophily;
- $\lambda_r < 0$: negative eigenvalues represent antihomophily.

Understanding eigenvectors and eigenvalues

$$\mathbf{Z} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{\Lambda} \mathbf{u}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} \lambda_r u_{i,r} u_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a rank-2 model, we have

$$z_{i,j} = \lambda_1 (u_{i,1} \times u_{j,1}) + \lambda_2 (u_{i,2} \times u_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_{i,r} \approx u_{j,r}$: equality of latent factors; represents stochastic equivalence;
- $\lambda_r > 0$: positive eigenvalues represent homophily;
- $\lambda_r < 0$: negative eigenvalues represent antihomophily.

Understanding eigenvectors and eigenvalues

$$\mathbf{Z} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{\Lambda} \mathbf{u}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} \lambda_r u_{i,r} u_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a rank-2 model, we have

$$z_{i,j} = \lambda_1 (u_{i,1} \times u_{j,1}) + \lambda_2 (u_{i,2} \times u_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_{i,r} \approx u_{j,r}$: equality of latent factors; represents stochastic equivalence;
- $\lambda_r > 0$: positive eigenvalues represent homophily;
- $\lambda_r < 0$: negative eigenvalues represent antihomophily.

Understanding eigenvectors and eigenvalues

$$\mathbf{Z} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{\Lambda} \mathbf{u}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} \lambda_r u_{i,r} u_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a rank-2 model, we have

$$z_{i,j} = \lambda_1 (u_{i,1} \times u_{j,1}) + \lambda_2 (u_{i,2} \times u_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_{i,r} \approx u_{j,r}$: equality of latent factors; represents stochastic equivalence;
- $\lambda_r > 0$: positive eigenvalues represent homophily;
- $\lambda_r < 0$: negative eigenvalues represent antihomophily.

Understanding eigenvectors and eigenvalues

$$\mathbf{Z} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{\Lambda} \mathbf{u}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} \lambda_r u_{i,r} u_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a rank-2 model, we have

$$z_{i,j} = \lambda_1 (u_{i,1} \times u_{j,1}) + \lambda_2 (u_{i,2} \times u_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_{i,r} \approx u_{j,r}$: equality of latent factors; represents stochastic equivalence;
- $\lambda_r > 0$: positive eigenvalues represent homophily;
- $\lambda_r < 0$: negative eigenvalues represent antihomophily.

Understanding eigenvectors and eigenvalues

$$\mathbf{Z} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{\Lambda} \mathbf{u}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} \lambda_r u_{i,r} u_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a rank-2 model, we have

$$z_{i,j} = \lambda_1 (u_{i,1} \times u_{j,1}) + \lambda_2 (u_{i,2} \times u_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_{i,r} \approx u_{j,r}$: equality of latent factors; represents stochastic equivalence;
- $\lambda_r > 0$: positive eigenvalues represent homophily;
- $\lambda_r < 0$: negative eigenvalues represent antihomophily.

Understanding eigenvectors and eigenvalues

$$\mathbf{Z} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{\Lambda} \mathbf{u}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} \lambda_r u_{i,r} u_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a rank-2 model, we have

$$z_{i,j} = \lambda_1 (u_{i,1} \times u_{j,1}) + \lambda_2 (u_{i,2} \times u_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_{i,r} \approx u_{j,r}$: equality of latent factors; represents stochastic equivalence;
- $\lambda_r > 0$: positive eigenvalues represent homophily;
- $\lambda_r < 0$: negative eigenvalues represent antihomophily.

Understanding eigenvectors and eigenvalues

$$\mathbf{Z} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{\Lambda} \mathbf{u}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} \lambda_r u_{i,r} u_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a rank-2 model, we have

$$z_{i,j} = \lambda_1 (u_{i,1} \times u_{j,1}) + \lambda_2 (u_{i,2} \times u_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_{i,r} \approx u_{j,r}$: equality of latent factors; represents stochastic equivalence;
- $\lambda_r > 0$: positive eigenvalues represent homophily;
- $\lambda_r < 0$: negative eigenvalues represent antihomophily.

Understanding eigenvectors and eigenvalues

$$\mathbf{Z} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} + \mathbf{E}$$

$$\begin{aligned} z_{i,j} &= \mathbf{u}_i^T \mathbf{\Lambda} \mathbf{u}_j + \epsilon_{i,j} \\ &= \sum_{r=1}^{R} \lambda_r u_{i,r} u_{j,r} + \epsilon_{i,j} \end{aligned}$$

For example, in a rank-2 model, we have

$$z_{i,j} = \lambda_1 (u_{i,1} \times u_{j,1}) + \lambda_2 (u_{i,2} \times u_{j,2}) + \epsilon_{i,j}$$

Interpretation

- $u_{i,r} \approx u_{j,r}$: equality of latent factors; represents stochastic equivalence;
- $\lambda_r > 0$: positive eigenvalues represent homophily;
- $\lambda_r < 0$: negative eigenvalues represent antihomophily.

AME models

Returning to directed relations:

SRM: We have motivated the SRM in order to represent 2nd order dependence:

- within row dependence, within column dependence;
- within dyad dependence.

$$z_{i,j} = \beta^T x_{i,j} + a_i + b_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

This model is made up of additive random effects.

LFM: We have motivated the LFM to model more complex structures:

- third order dependence and transitivity;
- stochastic equivalence.

$$z_{i,j} = \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

This model is made up of multiplicative random effects.

AME models

Returning to directed relations:

SRM: We have motivated the SRM in order to represent 2nd order dependence:

- within row dependence, within column dependence;
- within dyad dependence.

$$z_{i,j} = \beta^T x_{i,j} + a_i + b_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

This model is made up of additive random effects.

LFM: We have motivated the LFM to model more complex structures:

- third order dependence and transitivity;
- stochastic equivalence.

$$z_{i,j} = \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

This model is made up of multiplicative random effects.

AME models

Returning to directed relations:

SRM: We have motivated the SRM in order to represent 2nd order dependence:

- within row dependence, within column dependence;
- within dyad dependence.

$$z_{i,j} = \beta^T x_{i,j} + a_i + b_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

This model is made up of additive random effects.

LFM: We have motivated the LFM to model more complex structures:

- third order dependence and transitivity;
- stochastic equivalence.

$$z_{i,j} = \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

This model is made up of multiplicative random effects.

AME models

Returning to directed relations:

SRM: We have motivated the SRM in order to represent 2nd order dependence:

- within row dependence, within column dependence;
- within dyad dependence.

$$z_{i,j} = \beta^T x_{i,j} + a_i + b_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

This model is made up of additive random effects.

LFM: We have motivated the LFM to model more complex structures:

- third order dependence and transitivity;
- stochastic equivalence.

$$z_{i,j} = \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

This model is made up of multiplicative random effects.

AME models

Returning to directed relations:

SRM: We have motivated the SRM in order to represent 2nd order dependence:

- within row dependence, within column dependence;
- within dyad dependence.

$$
\begin{aligned}
z_{i,j} &= \beta^T x_{i,j} + a_i + b_j + \epsilon_{i,j} \\
y_{i,j} &= g(z_{i,j})
\end{aligned}
$$

This model is made up of additive random effects.

LFM: We have motivated the LFM to model more complex structures:

- third order dependence and transitivity;
- stochastic equivalence.

$$
\begin{aligned}
z_{i,j} &= \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j} \\
y_{i,j} &= g(z_{i,j})
\end{aligned}
$$

This model is made up of multiplicative random effects.

AME models

Returning to directed relations:

SRM: We have motivated the SRM in order to represent 2nd order dependence:

- within row dependence, within column dependence;
- within dyad dependence.

$$\begin{aligned} z_{i,j} &= \beta^T x_{i,j} + a_i + b_j + \epsilon_{i,j} \\ y_{i,j} &= g(z_{i,j}) \end{aligned}$$

This model is made up of additive random effects.

LFM: We have motivated the LFM to model more complex structures:

- third order dependence and transitivity;
- stochastic equivalence.

$$\begin{aligned} z_{i,j} &= \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j} \\ y_{i,j} &= g(z_{i,j}) \end{aligned}$$

This model is made up of multiplicative random effects.

AME models

Returning to directed relations:

SRM: We have motivated the SRM in order to represent 2nd order dependence:

- within row dependence, within column dependence;
- within dyad dependence.

$$
\begin{aligned}
z_{i,j} &= \beta^T x_{i,j} + a_i + b_j + \epsilon_{i,j} \\
y_{i,j} &= g(z_{i,j})
\end{aligned}
$$

This model is made up of additive random effects.

LFM: We have motivated the LFM to model more complex structures:

- third order dependence and transitivity;
- stochastic equivalence.

$$
\begin{aligned}
z_{i,j} &= \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j} \\
y_{i,j} &= g(z_{i,j})
\end{aligned}
$$

This model is made up of multiplicative random effects.

AME models

Returning to directed relations:

SRM: We have motivated the SRM in order to represent 2nd order dependence:

- within row dependence, within column dependence;
- within dyad dependence.

$$z_{i,j} = \beta^T x_{i,j} + a_i + b_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

This model is made up of additive random effects.

LFM: We have motivated the LFM to model more complex structures:

- third order dependence and transitivity;
- stochastic equivalence.

$$z_{i,j} = \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

This model is made up of multiplicative random effects.

AME models

Returning to directed relations:

SRM: We have motivated the SRM in order to represent 2nd order dependence:

- within row dependence, within column dependence;
- within dyad dependence.

$$\begin{aligned} z_{i,j} &= \beta^T x_{i,j} + a_i + b_j + \epsilon_{i,j} \\ y_{i,j} &= g(z_{i,j}) \end{aligned}$$

This model is made up of additive random effects.

LFM: We have motivated the LFM to model more complex structures:

- third order dependence and transitivity;
- stochastic equivalence.

$$\begin{aligned} z_{i,j} &= \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j} \\ y_{i,j} &= g(z_{i,j}) \end{aligned}$$

This model is made up of multiplicative random effects.

AME models

Combining them gives an **additive and multiplicative effects** model

$$z_{i,j} = \beta^T x_{i,j} + a_i + b_j + \boldsymbol{u}_i^T \mathbf{D} \boldsymbol{v}_j + \epsilon_{i,j}$$
$$y_{i,j} = g(z_{i,j})$$

```
     ame(Y, Xdyad=NULL, Xrow=NULL, Xcol=NULL, rvar = !(model=="rrl") ,
     cvar = TRUE, dcor = TRUE, R = 0, model="nrm",...)
```

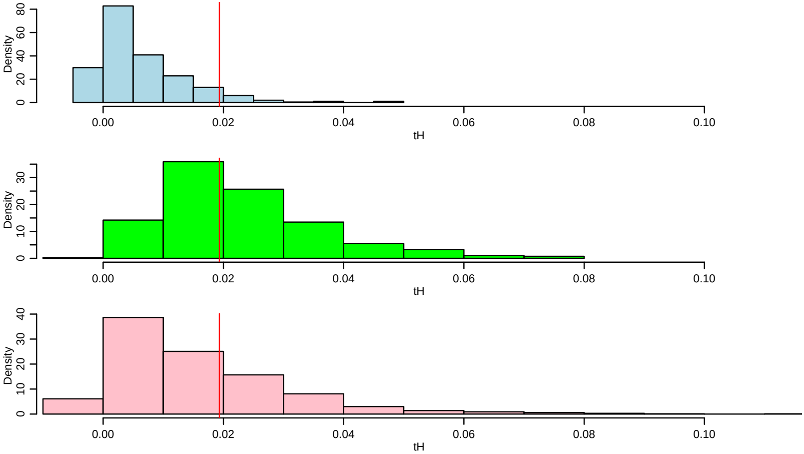
```
Arguments:

       Y: an n x n square relational matrix of relations.  See model
          below for various data types.

   Xdyad: an n x n x pd array of covariates

    Xrow: an n x pr matrix of nodal row covariates

    Xcol: an n x pc matrix of nodal column covariates

    rvar: logical: fit row random effects?

    cvar: logical: fit column random effects?

    dcor: logical: fit a dyadic correlation?

       R: integer: dimension of the multiplicative effects (can be
          zero)

   model: character: one of "nrm","bin","ord","cbin","frn","rrl" - see
```

AME fit to ordinal data with nodal covariates

```
#### nodal covariates and latent factors
fit_nr2<-ame(Y,Xrow=Xn,Xcol=Xn,R=2,model="ord",nscan=10000)
```

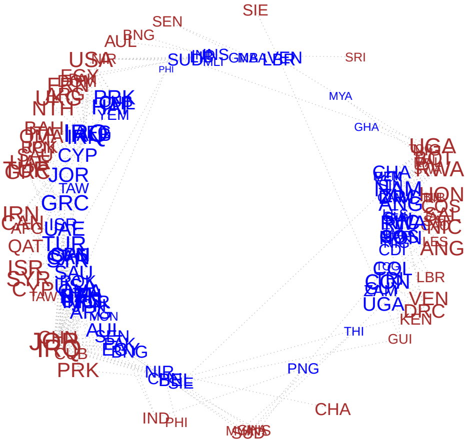
```
summary(fit_nr2)

##
## beta:
##               pmean   psd z-stat p-val
## pop.row      0.236 0.101  2.329 0.020
## gdp.row     -0.092 0.179 -0.516 0.606
## polity.row -0.030 0.015 -2.057 0.040
## pop.col      0.243 0.087  2.783 0.005
## gdp.col     -0.173 0.136 -1.267 0.205
## polity.col -0.009 0.013 -0.652 0.515
##
## Sigma_ab pmean:
##       a     b
## a 0.307 0.167
## b 0.167 0.206
##
## rho pmean:
##  0.838
```

GOF comparison



Plots of latent factors

```
circplot(Y,fit_nr2$U,fit_nr2$V)
```



Blockmodels

A **blockmodel** is

- a partition of the nodes into classes;
- an estimation of the rate of ties between and within classes.

Such a model includes

- a classification function $c : \{1, \ldots, n\} \to \{1, \ldots, K\}$,
  i.e. $c_i = k$ means i is in block/group k.
- a between-group tie density matrix:

$$\Theta = \begin{pmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \vdots & \vdots \\ \theta_{K1} & \cdots & \theta_{KK} \end{pmatrix}$$

Under this model,

$$\Pr(Y_{i,j} = 1) = \theta_{c_i, c_j}$$

Note: When the classes/blocks are defined by known binary covariates, the blockmodel is essentially the same as binary regression with indicators for block memberships.

Blockmodels

A **blockmodel** is

- a partition of the nodes into classes;
- an estimation of the rate of ties between and within classes.

Such a model includes

- a classification function $c : \{1, \ldots, n\} \rightarrow \{1, \ldots, K\}$,
  i.e. $c_i = k$ means i is in block/group k.
- a between-group tie density matrix:

$$\Theta = \begin{pmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \vdots & \vdots \\ \theta_{K1} & \cdots & \theta_{KK} \end{pmatrix}$$

Under this model,

$$\Pr(Y_{i,j} = 1) = \theta_{c_i, c_j}$$

Note: When the classes/blocks are defined by known binary covariates, the blockmodel is essentially the same as binary regression with indicators for block memberships.

Blockmodels

A **blockmodel** is

- a partition of the nodes into classes;
- an estimation of the rate of ties between and within classes.

Such a model includes

- a classification function $c : \{1, \ldots, n\} \to \{1, \ldots, K\}$,
  i.e. $c_i = k$ means i is in block/group k.
- a between-group tie density matrix:

$$\Theta = \begin{pmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \vdots & \vdots \\ \theta_{K1} & \cdots & \theta_{KK} \end{pmatrix}$$

Under this model,

$$\Pr(Y_{i,j} = 1) = \theta_{c_i, c_j}$$

Note: When the classes/blocks are defined by known binary covariates, the blockmodel is essentially the same as binary regression with indicators for block memberships.

Blockmodels

A **blockmodel** is

- a partition of the nodes into classes;
- an estimation of the rate of ties between and within classes.

Such a model includes

- a classification function $c : \{1, \ldots, n\} \to \{1, \ldots, K\}$,
  i.e. $c_i = k$ means i is in block/group k.
- a between-group tie density matrix:

$$\Theta = \begin{pmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \vdots & \vdots \\ \theta_{K1} & \cdots & \theta_{KK} \end{pmatrix}$$

Under this model,

$$\Pr(Y_{i,j} = 1) = \theta_{c_i, c_j}$$

Note: When the classes/blocks are defined by known binary covariates, the blockmodel is essentially the same as binary regression with indicators for block memberships.

Blockmodels

A **blockmodel** is

- a partition of the nodes into classes;
- an estimation of the rate of ties between and within classes.

Such a model includes

- a classification function $c : \{1, \ldots, n\} \rightarrow \{1, \ldots, K\}$,

  i.e. $c_i = k$ means i is in block/group k.
- a between-group tie density matrix:

$$\Theta = \begin{pmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \vdots & \vdots \\ \theta_{K1} & \cdots & \theta_{KK} \end{pmatrix}$$

Under this model,

$$\Pr(Y_{i,j} = 1) = \theta_{c_i, c_j}$$

Note: When the classes/blocks are defined by known binary covariates, the blockmodel is essentially the same as binary regression with indicators for block memberships.

Blockmodels

A **blockmodel** is

- a partition of the nodes into classes;
- an estimation of the rate of ties between and within classes.

Such a model includes

- a classification function $c : \{1, \ldots, n\} \to \{1, \ldots, K\}$,

  i.e. $c_i = k$ means i is in block/group k.
- a between-group tie density matrix:

$$\Theta = \begin{pmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \vdots & \vdots \\ \theta_{K1} & \cdots & \theta_{KK} \end{pmatrix}$$

Under this model,

$$\Pr(Y_{i,j} = 1) = \theta_{c_i, c_j}$$

Note: When the classes/blocks are defined by known binary covariates, the blockmodel is essentially the same as binary regression with indicators for block memberships.

Blockmodels

A **blockmodel** is

- a partition of the nodes into classes;
- an estimation of the rate of ties between and within classes.

Such a model includes

- a classification function $c : \{1, \ldots, n\} \to \{1, \ldots, K\}$,

  i.e. $c_i = k$ means i is in block/group k.
- a between-group tie density matrix:

$$\Theta = \begin{pmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \vdots & \vdots \\ \theta_{K1} & \cdots & \theta_{KK} \end{pmatrix}$$

Under this model,

$$\Pr(Y_{i,j} = 1) = \theta_{c_i, c_j}$$

Note: When the classes/blocks are defined by known binary covariates, the blockmodel is essentially the same as binary regression with indicators for block memberships.

Blockmodels

A **blockmodel** is

- a partition of the nodes into classes;
- an estimation of the rate of ties between and within classes.

Such a model includes

- a classification function $c : \{1, \ldots, n\} \to \{1, \ldots, K\}$,

  i.e. $c_i = k$ means i is in block/group k.
- a between-group tie density matrix:

$$\Theta = \begin{pmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \vdots & \vdots \\ \theta_{K1} & \cdots & \theta_{KK} \end{pmatrix}$$

Under this model,

$$\Pr(Y_{i,j} = 1) = \theta_{c_i, c_j}$$

Note: When the classes/blocks are defined by known binary covariates, the blockmodel is essentially the same as binary regression with indicators for block memberships.

Blockmodels

A **blockmodel** is

- a partition of the nodes into classes;
- an estimation of the rate of ties between and within classes.

Such a model includes

- a classification function $c : \{1, \ldots, n\} \to \{1, \ldots, K\}$,
  i.e. $c_i = k$ means i is in block/group k.
- a between-group tie density matrix:

$$\Theta = \begin{pmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \vdots & \vdots \\ \theta_{K1} & \cdots & \theta_{KK} \end{pmatrix}$$

Under this model,

$$\Pr(Y_{i,j} = 1) = \theta_{c_i,c_j}$$

Note: When the classes/blocks are defined by known binary covariates, the blockmodel is essentially the same as binary regression with indicators for block memberships.

Blockmodels

A **blockmodel** is

- a partition of the nodes into classes;
- an estimation of the rate of ties between and within classes.

Such a model includes

- a classification function $c : \{1, \ldots, n\} \to \{1, \ldots, K\}$,

  i.e. $c_i = k$ means i is in block/group k.
- a between-group tie density matrix:

$$\Theta = \begin{pmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \vdots & \vdots \\ \theta_{K1} & \cdots & \theta_{KK} \end{pmatrix}$$

Under this model,

$$\Pr(Y_{i,j} = 1) = \theta_{c_i, c_j}$$

Note: When the classes/blocks are defined by known binary covariates, the blockmodel is essentially the same as binary regression with indicators for block memberships.

Stochastic equivalence and blockmodels

The blockmodel is a model of the form we've been discussing:

$$y_{i,j} \sim \beta_0 + \beta_1 s(x_i, x_j)$$

Let

- $\beta_0 = 0,\ \beta_1 = 1;$
- $x_i = c_i,\ x_j = c_j;$
- $s(x_i, x_j) = \theta_{c_i, c_j}$

Stochastic equivalence:
All nodes within the same block are stochastically equivalent, under this model.

If $c_i = c_k = c$, then

$$\Pr(Y_{i,j} = 1) = \Pr(Y_{k,j} = 1) = \theta_{c, c_j}.$$

Stochastic equivalence and blockmodels

The blockmodel is a model of the form we've been discussing:

$$y_{i,j} \sim \beta_0 + \beta_1 s(x_i, x_j)$$

Let

- $\beta_0 = 0,\ \beta_1 = 1;$
- $x_i = c_i,\ x_j = c_j;$
- $s(x_i, x_j) = \theta_{c_i, c_j}$

Stochastic equivalence:
All nodes within the same block are stochastically equivalent, under this model.

If $c_i = c_k = c$, then

$$\Pr(Y_{i,j} = 1) = \Pr(Y_{k,j} = 1) = \theta_{c, c_j}.$$

Stochastic equivalence and blockmodels

The blockmodel is a model of the form we've been discussing:

$$y_{i,j} \sim \beta_0 + \beta_1 s(x_i, x_j)$$

Let

- $\beta_0 = 0$, $\beta_1 = 1$;
- $x_i = c_i$, $x_j = c_j$;
- $s(x_i, x_j) = \theta_{c_i,c_j}$

Stochastic equivalence:
All nodes within the same block are stochastically equivalent, under this model.

If $c_i = c_k = c$, then

$$\Pr(Y_{i,j} = 1) = \Pr(Y_{k,j} = 1) = \theta_{c,c_j}.$$

Stochastic equivalence and blockmodels

The blockmodel is a model of the form we've been discussing:

$$y_{i,j} \sim \beta_0 + \beta_1 s(x_i, x_j)$$

Let

- $\beta_0 = 0$, $\beta_1 = 1$;
- $x_i = c_i$, $x_j = c_j$;
- $s(x_i, x_j) = \theta_{c_i,c_j}$

Stochastic equivalence:
All nodes within the same block are stochastically equivalent, under this model.

If $c_i = c_k = c$, then

$$\Pr(Y_{i,j} = 1) = \Pr(Y_{k,j} = 1) = \theta_{c,c_j}.$$

Stochastic equivalence and blockmodels

The blockmodel is a model of the form we've been discussing:

$$y_{i,j} \sim \beta_0 + \beta_1 s(x_i, x_j)$$

Let

- $\beta_0 = 0$, $\beta_1 = 1$;
- $x_i = c_i$, $x_j = c_j$;
- $s(x_i, x_j) = \theta_{c_i, c_j}$

Stochastic equivalence:
All nodes within the same block are stochastically equivalent, under this model.

If $c_i = c_k = c$, then

$$\Pr(Y_{i,j} = 1) = \Pr(Y_{k,j} = 1) = \theta_{c, c_j}.$$

Stochastic equivalence and blockmodels

The blockmodel is a model of the form we've been discussing:

$$y_{i,j} \sim \beta_0 + \beta_1 s(x_i, x_j)$$

Let

- $\beta_0 = 0$, $\beta_1 = 1$;
- $x_i = c_i$, $x_j = c_j$;
- $s(x_i, x_j) = \theta_{c_i, c_j}$

Stochastic equivalence:
All nodes within the same block are stochastically equivalent, under this model.

If $c_i = c_k = c$, then

$$\Pr(Y_{i,j} = 1) = \Pr(Y_{k,j} = 1) = \theta_{c, c_j}.$$

Stochastic blockmodels

Consider the task of **identifying** stochastically equivalent classes from the data.

In the simplest case of an undirected binary relation, we want to find

latent classes $c_1, \ldots, c_n \in \{1, \ldots, K\}$;

between-class rates $\Theta = \{\theta_{k,l} : 1 \leq k, l \leq K\}$

that make the probability of our data large:

$$\Pr(\mathbf{Y} = \mathbf{y} | \boldsymbol{c}, \Theta) = \prod_{i \neq j} \theta_{c_i, c_j}^{y_{i,j}} (1 - \theta_{c_i, c_j})^{1 - y_{i,j}}$$

$$= \prod_{k=1}^{K} \prod_{l=1}^{K} \theta_{k,l}^{s_{k,l}} (1 - \theta_{k,l})^{n_{k,l} - s_{k,l}},$$

where

- $n_{k,l}$= number of pairs (i,j) for which $c_i = k$ and $c_j = l$
- $s_{k,l}$= number of pairs (i,j) for which $c_i = k$ and $c_j = l$ and $y_{i,j} = 1$.

This model is sometimes called the **stochastic blockmodel**.
It is a blockmodel where the classes are estimated from the data.

Stochastic blockmodels

Consider the task of **identifying** stochastically equivalent classes from the data.

In the simplest case of an undirected binary relation, we want to find

latent classes $c_1, \ldots, c_n \in \{1, \ldots, K\}$;

between-class rates $\Theta = \{\theta_{k,l} : 1 \leq k, l \leq K\}$

that make the probability of our data large:

$$\begin{aligned} \Pr(\mathbf{Y} = \mathbf{y} | \boldsymbol{c}, \Theta) &= \prod_{i \neq j} \theta_{c_i, c_j}^{y_{i,j}} (1 - \theta_{c_i, c_j})^{1 - y_{i,j}} \\ &= \prod_{k=1}^{K} \prod_{l=1}^{K} \theta_{k,l}^{s_{k,l}} (1 - \theta_{k,l})^{n_{k,l} - s_{k,l}}, \end{aligned}$$

where

- $n_{k,l}$= number of pairs (i, j) for which $c_i = k$ and $c_j = l$
- $s_{k,l}$= number of pairs (i, j) for which $c_i = k$ and $c_j = l$ and $y_{i,j} = 1$.

This model is sometimes called the **stochastic blockmodel**.
It is a blockmodel where the classes are estimated from the data.

Stochastic blockmodels

Consider the task of **identifying** stochastically equivalent classes from the data.

In the simplest case of an undirected binary relation, we want to find

latent classes $c_1, \ldots, c_n \in \{1, \ldots, K\}$;

between-class rates $\Theta = \{\theta_{k,l} : 1 \leq k, l \leq K\}$

that make the probability of our data large:

$$\Pr(\mathbf{Y} = \mathbf{y} | \boldsymbol{c}, \Theta) = \prod_{i \neq j} \theta_{c_i, c_j}^{y_{i,j}} (1 - \theta_{c_i, c_j})^{1 - y_{i,j}}$$

$$= \prod_{k=1}^{K} \prod_{l=1}^{K} \theta_{k,l}^{s_{k,l}} (1 - \theta_{k,l})^{n_{k,l} - s_{k,l}},$$

where

- $n_{k,l}$= number of pairs (i, j) for which $c_i = k$ and $c_j = l$
- $s_{k,l}$= number of pairs (i, j) for which $c_i = k$ and $c_j = l$ and $y_{i,j} = 1$.

This model is sometimes called the **stochastic blockmodel**.
It is a blockmodel where the classes are estimated from the data.

Stochastic blockmodels

Consider the task of **identifying** stochastically equivalent classes from the data.

In the simplest case of an undirected binary relation, we want to find

latent classes $c_1, \ldots, c_n \in \{1, \ldots, K\}$;

between-class rates $\Theta = \{\theta_{k,l} : 1 \le k, l \le K\}$

that make the probability of our data large:

$$\begin{aligned}\Pr(\mathbf{Y} = \mathbf{y} | \boldsymbol{c}, \Theta) &= \prod_{i \neq j} \theta_{c_i, c_j}^{y_{i,j}} (1 - \theta_{c_i, c_j})^{1 - y_{i,j}} \\ &= \prod_{k=1}^{K} \prod_{l=1}^{K} \theta_{k,l}^{s_{k,l}} (1 - \theta_{k,l})^{n_{k,l} - s_{k,l}},\end{aligned}$$

where

- $n_{k,l}=$ number of pairs (i, j) for which $c_i = k$ and $c_j = l$
- $s_{k,l}=$ number of pairs (i, j) for which $c_i = k$ and $c_j = l$ and $y_{i,j} = 1$.

This model is sometimes called the **stochastic blockmodel**.
It is a blockmodel where the classes are estimated from the data.

Stochastic blockmodels

Consider the task of **identifying** stochastically equivalent classes from the data.

In the simplest case of an undirected binary relation, we want to find

latent classes $c_1, \dots, c_n \in \{1, \dots, K\}$;

between-class rates $\Theta = \{\theta_{k,l} : 1 \leq k, l \leq K\}$

that make the probability of our data large:

$$\begin{aligned}
\Pr(\mathbf{Y} = \mathbf{y} | \boldsymbol{c}, \Theta) &= \prod_{i \neq j} \theta_{c_i, c_j}^{y_{i,j}} (1 - \theta_{c_i, c_j})^{1 - y_{i,j}} \\
&= \prod_{k=1}^{K} \prod_{l=1}^{K} \theta_{k,l}^{s_{k,l}} (1 - \theta_{k,l})^{n_{k,l} - s_{k,l}},
\end{aligned}$$

where

- $n_{k,l}$= number of pairs (i, j) for which $c_i = k$ and $c_j = l$
- $s_{k,l}$= number of pairs (i, j) for which $c_i = k$ and $c_j = l$ and $y_{i,j} = 1$.

This model is sometimes called the **stochastic blockmodel**.
It is a blockmodel where the classes are estimated from the data.

Stochastic blockmodel

You can't take derivative to find the MLEs of $c_1, \ldots, c_n$. Instead, use

- EM algorithm
- Gibbs sampling/MCMC

The basic model can be extended in various ways:

- covariates/regressors;
- directed data
  - dyadic correlation;
  - separate sender and receiver classes.

Much of this was done in Nowicki and Snijders (2001).

A variety of more complex variants have been recently developed.

Stochastic blockmodel

You can't take derivative to find the MLEs of $c_1, \dots, c_n$. Instead, use

- EM algorithm
- Gibbs sampling/MCMC

The basic model can be extended in various ways:

- covariates/regressors;
- directed data
    - dyadic correlation;
    - separate sender and receiver classes.

Much of this was done in Nowicki and Snijders (2001).

A variety of more complex variants have been recently developed.

Stochastic blockmodel

You can't take derivative to find the MLEs of $c_1, \ldots, c_n$. Instead, use

- EM algorithm
- Gibbs sampling/MCMC

The basic model can be extended in various ways:

- covariates/regressors;
- directed data
    - dyadic correlation;
    - separate sender and receiver classes.

Much of this was done in Nowicki and Snijders (2001).

A variety of more complex variants have been recently developed.

Stochastic blockmodel

You can't take derivative to find the MLEs of $c_1, \ldots, c_n$. Instead, use

- EM algorithm
- Gibbs sampling/MCMC

The basic model can be extended in various ways:

- covariates/regressors;
- directed data
  - dyadic correlation;
  - separate sender and receiver classes.

Much of this was done in Nowicki and Snijders (2001).

A variety of more complex variants have been recently developed.

Stochastic blockmodel

You can't take derivative to find the MLEs of $c_1, \ldots, c_n$. Instead, use

- EM algorithm
- Gibbs sampling/MCMC

The basic model can be extended in various ways:

- covariates/regressors;
- directed data
  - dyadic correlation;
  - separate sender and receiver classes.

Much of this was done in Nowicki and Snijders (2001).

A variety of more complex variants have been recently developed.

Stochastic blockmodel

You can't take derivative to find the MLEs of $c_1, \ldots, c_n$. Instead, use

- EM algorithm
- Gibbs sampling/MCMC

The basic model can be extended in various ways:

- covariates/regressors;
- directed data
  - dyadic correlation;
  - separate sender and receiver classes.

Much of this was done in Nowicki and Snijders (2001).

A variety of more complex variants have been recently developed.

Stochastic blockmodel

You can't take derivative to find the MLEs of $c_1, \ldots, c_n$. Instead, use

- EM algorithm
- Gibbs sampling/MCMC

The basic model can be extended in various ways:

- covariates/regressors;
- directed data
  - dyadic correlation;
  - separate sender and receiver classes.

Much of this was done in Nowicki and Snijders (2001).

A variety of more complex variants have been recently developed.

Stochastic blockmodel

You can't take derivative to find the MLEs of $c_1, \ldots, c_n$. Instead, use

- EM algorithm
- Gibbs sampling/MCMC

The basic model can be extended in various ways:

- covariates/regressors;
- directed data
  - dyadic correlation;
  - separate sender and receiver classes.

Much of this was done in Nowicki and Snijders (2001).

A variety of more complex variants have been recently developed.

Stochastic blockmodel

You can't take derivative to find the MLEs of $c_1, \ldots, c_n$. Instead, use

- EM algorithm
- Gibbs sampling/MCMC

The basic model can be extended in various ways:

- covariates/regressors;
- directed data
    - dyadic correlation;
    - separate sender and receiver classes.

Much of this was done in Nowicki and Snijders (2001).

A variety of more complex variants have been recently developed.

Stochastic blockmodel

You can't take derivative to find the MLEs of $c_1, \ldots, c_n$. Instead, use

- EM algorithm
- Gibbs sampling/MCMC

The basic model can be extended in various ways:

- covariates/regressors;
- directed data
    - dyadic correlation;
    - separate sender and receiver classes.

Much of this was done in Nowicki and Snijders (2001).

A variety of more complex variants have been recently developed.

Stochastic blockmodel

You can't take derivative to find the MLEs of $c_1, \ldots, c_n$. Instead, use

- EM algorithm
- Gibbs sampling/MCMC

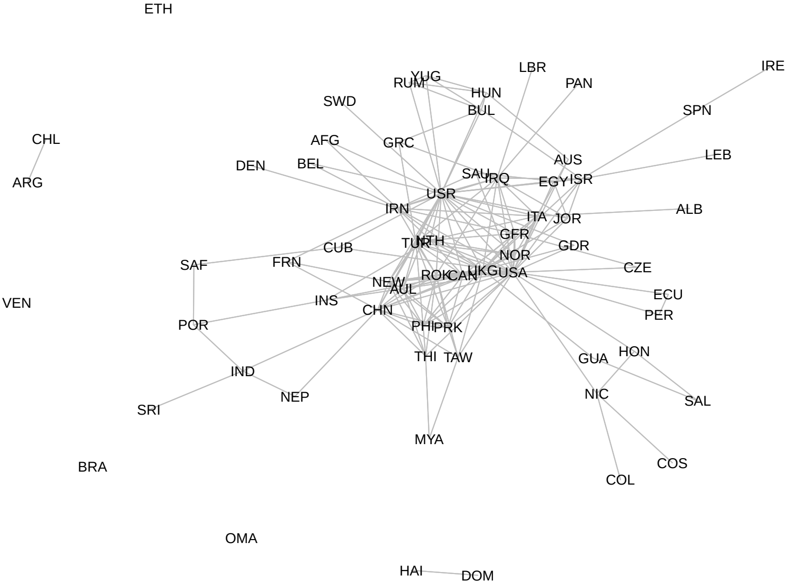
The basic model can be extended in various ways:

- covariates/regressors;
- directed data
  - dyadic correlation;
  - separate sender and receiver classes.

Much of this was done in Nowicki and Snijders (2001).

A variety of more complex variants have been recently developed.

Stochastic blockmodel for Cold War data



Stochastic blockmodel for Cold War data

A three-class model gives the following inferred classes:

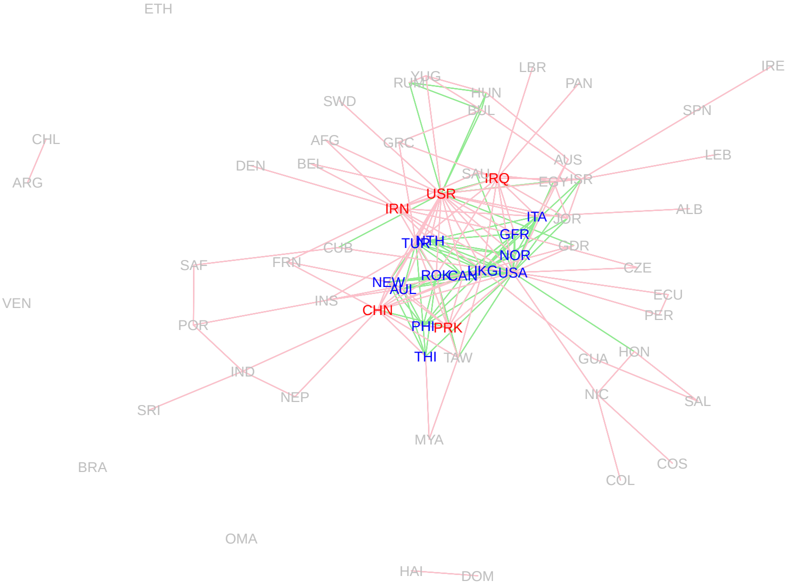
```
rownames(Y)[latentclass==1]

## [1] "CHN" "IRN" "IRQ" "PRK" "USR"

rownames(Y)[latentclass==2]

##  [1] "AUL" "CAN" "GFR" "ITA" "NEW" "NOR" "NTH" "PHI" "ROK" "THI" "TUR"
## [12] "UKG" "USA"
```

and everyone else in class 3, with the following between-class means:

```
round(M,2)

##       [,1]  [,2] [,3]
## [1,]  0.02 -0.89 0.50
## [2,] -0.89  4.47 1.07
## [3,]  0.50  1.07 0.86
```

These rates are on the probit scale (recall the data are ordinal).

Latent class for Cold War data



Matrix form of the blockmodel

The essential feature of the blockmodel is the representation

$$y_{i,j} \sim \theta_{c_i,c_j}$$

where

- c_i, c_j are unobserved latent class variables;
- Θ is a matrix of between-class intensities.

This model structure can be expressed in matrix form as follows:

$$\theta_{c_i,c_j} = \boldsymbol{u}_i^T \Theta \boldsymbol{u}_j,$$

where

- $\boldsymbol{u}_i$ is a $K \times 1$ vector of all 0s except $\boldsymbol{u}_i[c_i] = 1$;
- Θ is the $K \times K$ matrix of between class intensities.

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \theta_{31}$$

This model is a *special case* of the latent factor model.
Community detection methods are (more or less) special cases of blockmodels.

Matrix form of the blockmodel

The essential feature of the blockmodel is the representation

$$y_{i,j} \sim \theta_{c_i,c_j}$$

where

- c_i, c_j are unobserved latent class variables;
- Θ is a matrix of between-class intensities.

This model structure can be expressed in matrix form as follows:

$$\theta_{c_i,c_j} = \boldsymbol{u}_i^T \Theta \boldsymbol{u}_j,$$

where

- $\boldsymbol{u}_i$ is a $K \times 1$ vector of all 0s except $\boldsymbol{u}_i[c_i] = 1$;
- Θ is the $K \times K$ matrix of between class intensities.

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \theta_{31}$$

This model is a *special case* of the latent factor model.
Community detection methods are (more or less) special cases of blockmodels.

Matrix form of the blockmodel

The essential feature of the blockmodel is the representation

$$y_{i,j} \sim \theta_{c_i,c_j}$$

where

- c_i, c_j are unobserved latent class variables;
- Θ is a matrix of between-class intensities.

This model structure can be expressed in matrix form as follows:

$$\theta_{c_i,c_j} = \boldsymbol{u}_i^T \Theta \boldsymbol{u}_j,$$

where

- $\boldsymbol{u}_i$ is a $K \times 1$ vector of all 0s except $\boldsymbol{u}_i[c_i] = 1$;
- Θ is the $K \times K$ matrix of between class intensities.

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \theta_{31}$$

This model is a *special case* of the latent factor model.
Community detection methods are (more or less) special cases of blockmodels.

Matrix form of the blockmodel

The essential feature of the blockmodel is the representation

$$y_{i,j} \sim \theta_{c_i,c_j}$$

where

- c_i, c_j are unobserved latent class variables;
- Θ is a matrix of between-class intensities.

This model structure can be expressed in matrix form as follows:

$$\theta_{c_i,c_j} = \boldsymbol{u}_i^T \Theta \boldsymbol{u}_j,$$

where

- $\boldsymbol{u}_i$ is a $K \times 1$ vector of all 0s except $\boldsymbol{u}_i[c_i] = 1$;
- Θ is the $K \times K$ matrix of between class intensities.

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \theta_{31}$$

This model is a *special case* of the latent factor model.
Community detection methods are (more or less) special cases of blockmodels.

Matrix form of the blockmodel

The essential feature of the blockmodel is the representation

$$y_{i,j} \sim \theta_{c_i,c_j}$$

where

- c_i, c_j are unobserved latent class variables;
- Θ is a matrix of between-class intensities.

This model structure can be expressed in matrix form as follows:

$$\theta_{c_i,c_j} = \boldsymbol{u}_i^T \Theta \boldsymbol{u}_j,$$

where

- $\boldsymbol{u}_i$ is a $K \times 1$ vector of all 0s except $\boldsymbol{u}_i[c_i] = 1$;
- Θ is the $K \times K$ matrix of between class intensities.

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \theta_{31}$$

This model is a *special case* of the latent factor model.
Community detection methods are (more or less) special cases of blockmodels.

Matrix form of the blockmodel

The essential feature of the blockmodel is the representation

$$y_{i,j} \sim \theta_{c_i,c_j}$$

where

- c_i, c_j are unobserved latent class variables;
- Θ is a matrix of between-class intensities.

This model structure can be expressed in matrix form as follows:

$$\theta_{c_i,c_j} = \boldsymbol{u}_i^T \Theta \boldsymbol{u}_j,$$

where

- $\boldsymbol{u}_i$ is a $K \times 1$ vector of all 0s except $\boldsymbol{u}_i[c_i] = 1$;
- Θ is the $K \times K$ matrix of between class intensities.

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \theta_{31}$$

This model is a *special case* of the latent factor model.
Community detection methods are (more or less) special cases of blockmodels.

Matrix form of the blockmodel

The essential feature of the blockmodel is the representation

$$y_{i,j} \sim \theta_{c_i, c_j}$$

where

- c_i, c_j are unobserved latent class variables;
- Θ is a matrix of between-class intensities.

This model structure can be expressed in matrix form as follows:

$$\theta_{c_i, c_j} = \boldsymbol{u}_i^T \Theta \boldsymbol{u}_j,$$

where

- $\boldsymbol{u}_i$ is a $K \times 1$ vector of all 0s except $\boldsymbol{u}_i[c_i] = 1$;
- Θ is the $K \times K$ matrix of between class intensities.

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \theta_{31}$$

This model is a *special case* of the latent factor model.
Community detection methods are (more or less) special cases of blockmodels.

Summary of LFM

Features of real datasets:

- third-order triadic dependence;
- groups of nodes with similar roles.

Explanations of features:

- homophily on covariates can generate triadic dependence;
- stochastic equivalence can generate groups of similar nodes;
- both features can be represented by the "model" $y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$

Statistical modeling:

- Generate derived dyadic variables $s(u_i, v_j)$ from nodal covariates u_i, v_j.
  - indicators for categorical variables: $s(u_i, v_j) = 1(u_i = v_j)$;
  - interactions for continuous variables: $s(u_i, v_j) = u_i v_j$.
- If insufficient for model fit, estimate additional latent factors from data:
  - stochastic blockmodel: latent categorical variables;
  - latent factor model: latent continuous variables.

Summary of LFM

Features of real datasets:

- third-order triadic dependence;
- groups of nodes with similar roles.

Explanations of features:

- homophily on covariates can generate triadic dependence;
- stochastic equivalence can generate groups of similar nodes;
- both features can be represented by the “model” $y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$

Statistical modeling:

- Generate derived dyadic variables $s(u_i, v_j)$ from nodal covariates u_i, v_j.
  - indicators for categorical variables: $s(u_i, v_j) = 1(u_i = v_j)$;
  - interactions for continuous variables: $s(u_i, v_j) = u_i v_j$.
- If insufficient for model fit, estimate additional latent factors from data:
  - stochastic blockmodel: latent categorical variables;
  - latent factor model: latent continuous variables.

Summary of LFM

Features of real datasets:

- third-order triadic dependence;
- groups of nodes with similar roles.

Explanations of features:

- homophily on covariates can generate triadic dependence;
- stochastic equivalence can generate groups of similar nodes;
- both features can be represented by the "model" $y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$

Statistical modeling:

- Generate derived dyadic variables $s(u_i, v_j)$ from nodal covariates u_i, v_j.
  - indicators for categorical variables: $s(u_i, v_j) = 1(u_i = v_j)$;
  - interactions for continuous variables: $s(u_i, v_j) = u_i v_j$.
- If insufficient for model fit, estimate additional latent factors from data:
  - stochastic blockmodel: latent categorical variables;
  - latent factor model: latent continuous variables.

Summary of LFM

Features of real datasets:

- third-order triadic dependence;
- groups of nodes with similar roles.

Explanations of features:

- homophily on covariates can generate triadic dependence;
- stochastic equivalence can generate groups of similar nodes;
- both features can be represented by the "model" $y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$

Statistical modeling:

- Generate derived dyadic variables $s(u_i, v_j)$ from nodal covariates u_i, v_j.
  - indicators for categorical variables: $s(u_i, v_j) = 1(u_i = v_j)$;
  - interactions for continuous variables: $s(u_i, v_j) = u_i v_j$.
- If insufficient for model fit, estimate additional latent factors from data:
  - stochastic blockmodel: latent categorical variables;
  - latent factor model: latent continuous variables.

Summary of LFM

Features of real datasets:

- third-order triadic dependence;
- groups of nodes with similar roles.

Explanations of features:

- homophily on covariates can generate triadic dependence;
- stochastic equivalence can generate groups of similar nodes;
- both features can be represented by the “model” $y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$

Statistical modeling:

- Generate derived dyadic variables $s(u_i, v_j)$ from nodal covariates u_i, v_j.
    - indicators for categorical variables: $s(u_i, v_j) = 1(u_i = v_j)$;
    - interactions for continuous variables: $s(u_i, v_j) = u_i v_j$.
- If insufficient for model fit, estimate additional latent factors from data:
    - stochastic blockmodel: latent categorical variables;
    - latent factor model: latent continuous variables.

Summary of LFM

Features of real datasets:

- third-order triadic dependence;
- groups of nodes with similar roles.

Explanations of features:

- homophily on covariates can generate triadic dependence;
- stochastic equivalence can generate groups of similar nodes;
- both features can be represented by the “model” $y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$

Statistical modeling:

- Generate derived dyadic variables $s(u_i, v_j)$ from nodal covariates u_i, v_j.
  - indicators for categorical variables: $s(u_i, v_j) = 1(u_i = v_j)$;
  - interactions for continuous variables: $s(u_i, v_j) = u_i v_j$.
- If insufficient for model fit, estimate additional latent factors from data:
  - stochastic blockmodel: latent categorical variables;
  - latent factor model: latent continuous variables.

Summary of LFM

Features of real datasets:

- third-order triadic dependence;
- groups of nodes with similar roles.

Explanations of features:

- homophily on covariates can generate triadic dependence;
- stochastic equivalence can generate groups of similar nodes;
- both features can be represented by the “model” $y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$

Statistical modeling:

- Generate derived dyadic variables $s(u_i, v_j)$ from nodal covariates u_i, v_j.
  - indicators for categorical variables: $s(u_i, v_j) = 1(u_i = v_j)$;
  - interactions for continuous variables: $s(u_i, v_j) = u_i v_j$.
- If insufficient for model fit, estimate additional latent factors from data:
  - stochastic blockmodel: latent categorical variables;
  - latent factor model: latent continuous variables.

Summary of LFM

Features of real datasets:

- third-order triadic dependence;
- groups of nodes with similar roles.

Explanations of features:

- homophily on covariates can generate triadic dependence;
- stochastic equivalence can generate groups of similar nodes;
- both features can be represented by the "model" $y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$

Statistical modeling:

- Generate derived dyadic variables $s(u_i, v_j)$ from nodal covariates u_i, v_j.
  - indicators for categorical variables: $s(u_i, v_j) = 1(u_i = v_j)$;
  - interactions for continuous variables: $s(u_i, v_j) = u_i v_j$.
- If insufficient for model fit, estimate additional latent factors from data:
  - stochastic blockmodel: latent categorical variables;
  - latent factor model: latent continuous variables.

Summary of LFM

Features of real datasets:

- third-order triadic dependence;
- groups of nodes with similar roles.

Explanations of features:

- homophily on covariates can generate triadic dependence;
- stochastic equivalence can generate groups of similar nodes;
- both features can be represented by the "model" $y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$

Statistical modeling:

- Generate derived dyadic variables $s(u_i, v_j)$ from nodal covariates u_i, v_j.
  - indicators for categorical variables: $s(u_i, v_j) = 1(u_i = v_j)$;
  - interactions for continuous variables: $s(u_i, v_j) = u_i v_j$.
- If insufficient for model fit, estimate additional latent factors from data:
  - stochastic blockmodel: latent categorical variables;
  - latent factor model: latent continuous variables.

Summary of LFM

Features of real datasets:

- third-order triadic dependence;
- groups of nodes with similar roles.

Explanations of features:

- homophily on covariates can generate triadic dependence;
- stochastic equivalence can generate groups of similar nodes;
- both features can be represented by the "model" $y_{i,j} \sim \beta_0 + \beta_1 \times s(u_i, v_j)$

Statistical modeling:

- Generate derived dyadic variables $s(u_i, v_j)$ from nodal covariates u_i, v_j.
    - indicators for categorical variables: $s(u_i, v_j) = 1(u_i = v_j)$;
    - interactions for continuous variables: $s(u_i, v_j) = u_i v_j$.
- If insufficient for model fit, estimate additional latent factors from data:
    - stochastic blockmodel: latent categorical variables;
    - latent factor model: latent continuous variables.