Covariate effects in ERGMs
567 Statistical analysis of social networks

Peter Hoff

Statistics, University of Washington
Dutch college data

Friendship ties among 32 college students enrolled in a particular program.

- Relations on a 6 point scale, from "dislike" to "best friends";
- Relations measured at seven time points;
- Sex, smoking status and subprogram category also available.

Questions:

- What are the effects of sex, smoking status and subgroup on tie formation?
- Is there substantial in and outdegree heterogeneity, or reciprocity?
- How does the network evolve over time?
For now, we’ll analyze

- the indicator of a positive relation;
- the network at the final timepoint.
Preliminary analysis

```r
mean( Y7, na.rm=TRUE)
## [1] 0.1693548
mean( Y7[ X$male==1, X$male==1 ], na.rm=TRUE )
## [1] 0.3571429
mean( Y7[ X$male==0, X$male==0 ], na.rm=TRUE )
## [1] 0.1956522
mean( Y7[ X$smoke==1, X$smoke==1 ], na.rm=TRUE )
## [1] 0.2692308
mean( Y7[ X$smoke==0, X$smoke==0 ], na.rm=TRUE )
## [1] 0.2017544
SP<-outer(X$prog,X$prog,"==")
mean( Y7[SP], na.rm=TRUE)
## [1] 0.2861111
```
Preliminary analysis

```r
mean(outdegree[X$smoker==1]) - mean(outdegree[X$smoker==0])
## [1] 0.3562753

mean(indegree[X$smoker==1]) - mean(indegree[X$smoker==0])
## [1] 0.2267206
```
Degree heterogeneity and Reciprocity

### degree analysis

\[ \text{sd(outdegree)} \]

## [1] 4.662825

\[ \text{sd(indegree)} \]

## [1] 2.514474

\[ \text{cor(outdegree, indegree)} \]

## [1] 0.1705822

### dyad census

\[ \text{M<-sum(Y7}\text{t(Y7), na.rm=TRUE)}/2} \]

\[ \text{A<-sum(Y7, na.rm=TRUE) - 2*M} \]

\[ \text{N<- choose(nrow(Y7), 2) - M - A} \]

\[ \text{p11<-2*M/(2*M+A)} \]

\[ \text{p10<-A/(A+2*N)} \]

\[ \text{log( p11 * (1-p10) / (1-p11) * p10 )} \]

## [1] 2.250258
Preliminary analysis

Some preliminary findings:

- Covariate effects:
  - homophily by sex, smoking behavior and program;
  - smokers seem more outgoing and popular.

- Network patterns:
  - positive reciprocity;
  - outdegree variance is larger than indegree variance, and little correlation between the two.

To summarize covariate effects, some researchers employ “network regression:”

- convert sociomatrix and covariate matrices to vectors;
- perform ordinary regression (OLS, logistic regression, Poisson regression).

Such a procedure should **not** be called network regression:

- it is just **regression**;
- it ignores the network structure to the data.
Logistic regression

Nevertheless, regression with appropriate covariates might be adequate.

In particular, network patterns could be explained by covariates:

- degree heterogeneity could be explained by one or more nodal covariates;
- reciprocity could be explained by a group comembership variable.

Let’s do an ordinary logistic regression and evaluate the fit.

```R
XM<-array(dim=c(n,n,5))
XM[,1]<-matrix(X[,2],n,n)
XM[,2]<-t(XM[,1])
XM[,3]<-outer(X[,1],X[,1],"==")
XM[,4]<-outer(X[,2],X[,2],"==")
XM[,5]<-outer(X[,3],X[,3],"==")

y7<-c(Y7)
x<-apply(XM,3,"c")
colnames(x)<-c("rsmoke","csmoke","ssex","ssmoke","sprog")

fit.glm<-glm( y7 ~ x ,family=binomial)
```
Logistic regression fit

summary(fit(glm))

##
## Call:
glm(formula = y7 ~ x, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1144  -0.6909  -0.4446  -0.3119   2.4689
##
## Coefficients:
##            Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.2122   0.2671  -12.027  < 2e-16 ***
## xrsmoke      0.2548   0.1882   1.354   0.175716
## xcsmoke      0.2130   0.1881   1.132   0.257500
## xssex        0.6930   0.2079   3.334  0.000857 ***
## xssmoke      0.7983   0.1863   4.285  1.83e-05 ***
## xsprog       1.1030   0.1800   6.127  8.94e-10 ***
## ---
## Signif. codes:  0 '***'  0.001 '**'  0.01 '*'  0.05 '.'  0.1 ' ' 1
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 902.45  on 991  degrees of freedom
## Residual deviance: 819.39  on 986  degrees of freedom
## (32 observations deleted due to missingness)
## AIC: 831.39
##
## Number of Fisher Scoring iterations: 5
Fitting logistic regression in ergm

Logistic regression is an ERGM with independent relations.

Suppose our model is

$$\log \text{odds}(y_{i,j} = 1) = \beta_0 + \beta_r x_{r,i} + \beta_c x_{c,j} + \beta_d x_{d,i,j}$$

Then

$$\Pr(Y = y | X, \beta) = \prod_{i \neq j} \left( \frac{e^{(\beta_0 + \beta_r x_{r,i} + \beta_c x_{c,j} + \beta_d x_{d,i,j}) y_{i,j}}}{1 + e^{\beta_0 + \beta_r x_{r,i} + \beta_c x_{c,j} + \beta_d x_{d,i,j}}} \right)$$

$$= c(X, \beta) \times \exp \left( \beta_0 \sum_{i \neq j} y_{i,j} + \beta_r \sum_{i \neq j} x_{r,i} y_{i,j} + \beta_c \sum_{i \neq j} x_{c,j} y_{i,j} + \beta_d \sum_{i \neq j} x_{d,i,j} y_{i,j} \right)$$

The sufficient statistics simplify to the four-dimensional vector

$$t(y) = \left( y_{..}, \sum_{i=1}^{n} x_{r,i} y_{i..}, \sum_{j=1}^{n} x_{c,j} y_{..j}, \sum_{i \neq j} x_{d,i,j} y_{i,j} \right).$$
As logistic regression is an ERGM, we should be able to fit it with `ergm`.

We first need to convert the data to a `network` object:

```r
library(ergm)
netdat<-network(Y7, vertex.attr=X)
```

Sometimes you want to add vertex attributes one at a time:

```r
netdat<-network(Y7)
set.vertex.attribute(netdat,"male",X[,1])
set.vertex.attribute(netdat,"smoker",X[,2])
set.vertex.attribute(netdat,"program",X[,3])
```
Fitting logistic regression in ergm

The model is then fit, as before, by specifying sufficient statistics:

```r
fit.ergm <- ergm( netdat ~ edges + nodeocov("smoker") + nodeicov("smoker") + nodematch("male") + nodematch("smoker") + nodematch("program") )
```

The terms `nodeicov`, `nodeocov` and `nodematch` create sufficient statistics out of nodal covariates:

- `nodeocov` creates a row regression effect;
- `nodeicov` creates a column regression effect;
- `nodematch` creates a dyadic binary indicator.

See the `ergm` manual for more details.
Fitting logistic regression in ergm

```
summary(fit.ergm)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:  netdat ~ edges + nodeocov("smoker") + nodeicov("smoker") + nodematch("male")
##          nodematch("smoker") + nodematch("program")
##
## Iterations:  5 out of 20
##
## Monte Carlo MLE Results:
##
##       Estimate Std. Error MCMC %  p-value
## edges       -3.2122   0.2671 0     < 1e-04  ***
## nodeocov.smoker  0.2548   0.1882 0       0.176026
## nodeicov.smoker  0.2130   0.1881 0       0.257774
## nodematch.male    0.6930   0.2079 0       0.000889  ***
## nodematch.smoker  0.7983   0.1863 0       < 1e-04  ***
## nodematch.program 1.1030   0.1800 0       < 1e-04  ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Null Deviance: 1375 on 992 degrees of freedom
## Residual Deviance: 1292 on 986 degrees of freedom
##
## AIC: 1304    BIC: 1334  (Smaller is better.)
```
Goodness of fit

We fit this model without regard to its network structure:

- across sender heterogeneity/within sender correlation;
- across receiver heterogeneity/within receiver correlation;
- reciprocity/within dyad correlation.

It is possible that such patterns could be explained by covariates:

- heterogeneity in smoking leads to heterogeneity in degree;
- homophily for sex, smoking and group leads to reciprocity.

Let’s examine this with a goodness of fit evaluation.
Goodness of fit

\[ s.\text{obs}<-c(\text{sd(rsum(Y7))}, \text{sd(csum(Y7))}, \text{mdyad(Y7)}) \]

\[ \text{py.hat}<-\text{fit.glm}\$\text{fitted} \]

\[ s.\text{SIM}<-\text{NULL} \]

for (s in 1:S) {
  \{
    Ysim<-matrix(\text{NA}, \text{nrow(Y7)}, \text{nrow(Y7)})
    Ysim[\text{!is.na(Y7)}] <- \text{rbinom(length(py.hat), 1, py.hat)}
    s.\text{SIM}<-\text{rbind}(s.\text{SIM}, c(\text{sd(rsum(Ysim))}, \text{sd(csum(Ysim))}, \text{mdyad(Ysim)}))
  \}

Goodness of fit

\[
\text{mean}(s.SIM[,1]>=s.obs[1])
\]
## [1] 0

\[
\text{mean}(s.SIM[,2]>=s.obs[2])
\]
## [1] 0.149

\[
\text{mean}(s.SIM[,3]>=s.obs[3])
\]
## [1] 0

**Evaluation:** These results indicate

- more outdegree heterogeneity than expected under the MLE;
- more reciprocity than expected;
- indegree heterogeneity is as expected.
This lack of fit can be addressed by adding statistics to the model:

```
fit.p1cov.1 <- ergm( netdat ~ edges + sender + receiver + mutual +
                      nodeocov("smoker") + nodeicov("smoker") +
                      nodematch("male") + nodematch("smoker") + nodematch("program") )
```

```
summary(fit.p1cov.1)
```

### Summary of model fit

#### Formula: netdat ~ edges + sender + receiver + mutual + nodeocov("smoker") +
nodeicov("smoker") + nodematch("male") + nodematch("smoker") +
nodematch("program")

#### Iterations: 4 out of 20

#### Monte Carlo MLE Results:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>MCMC %</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>edges</td>
<td>-6.38755</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>sender2</td>
<td>0.18046</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>sender3</td>
<td>2.19305</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>sender4</td>
<td>-Inf</td>
<td>0.00000</td>
<td>0</td>
<td>&lt;1e-04  ***</td>
</tr>
<tr>
<td>sender5</td>
<td>-Inf</td>
<td>0.00000</td>
<td>0</td>
<td>&lt;1e-04  ***</td>
</tr>
<tr>
<td>sender6</td>
<td>0.13125</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>sender7</td>
<td>3.47546</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>sender8</td>
<td>0.03050</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>sender9</td>
<td>1.87725</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
Regression terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodeocov.smoker</td>
<td>0.1858202</td>
</tr>
<tr>
<td>nodeicov.smoker</td>
<td>0.2769255</td>
</tr>
<tr>
<td>nodematch.male</td>
<td>0.9771959</td>
</tr>
<tr>
<td>nodematch.smoker</td>
<td>0.8809058</td>
</tr>
<tr>
<td>nodematch.program</td>
<td>1.2139767</td>
</tr>
</tbody>
</table>
$p_1$ with alternative term order

```r
fit.p1cov.2 <- ergm(netdat ~
    nodeocov("smoker") + nodeicov("smoker") + 
    nodematch("male") + nodematch("smoker") + nodematch("program") +
    edges + sender + receiver + mutual )
```

```r
summary(fit.p1cov.2)

##
## ========================== 
## Summary of model fit 
## ==========================
##
## Formula: netdat ~ nodeocov("smoker") + nodeicov("smoker") + nodematch("male") + 
## nodematch("smoker") + nodematch("program") + edges + sender + receiver + mutual 
##
## Iterations: 4 out of 20 
##
## Monte Carlo MLE Results: 
##
##|                              | Estimate | Std. Error | MCMC % | p-value |
##|------------------------------|----------|------------|--------|---------|
##| nodeocov.smoker             | -1.676301| NA         | NA     | NA      |
##| nodeicov.smoker             | -1.451256| NA         | NA     | NA      |
##| nodematch.male              | 0.968922 | NA         | NA     | NA      |
##| nodematch.smoker            | 0.873663 | NA         | NA     | NA      |
##| nodematch.program           | 1.208628 | NA         | NA     | NA      |
##| edges                        | -2.720451| NA         | NA     | NA      |
##| sender2                      | -1.695372| NA         | NA     | NA      |
##| sender3                      | 0.301114 | NA         | NA     | NA      |
##| sender4                      | -Inf     | 0.000000   | 0      | <1e-04  *** |
```

```
Warning: The following terms have infinite coefficient estimates: sender4 sender5 sender13 sender18 sender20 sender27 sender30 sender31 receiver18
```
The problem is **confounding** between these effects and the sender and receiver effects.

To illustrate this issue, consider a simple model with just

- sender effects;
- one sender-specific covariate.

\[
\Pr(Y_{i,j} = y_{i,j}) = \frac{e^{(\mu + a_i + \beta x_i)y_{i,j}}}{1 + e^{\mu + a_i + \beta x_i}}
\]
Confounding

The sufficient statistics can be found by summing the exponent over pairs:

\[
\sum_{i \neq j} \mu y_{i,j} + a_i y_{i,j} + \beta x_i y_{i,j} = \mu y_\cdot + \sum_i a_i y_i + \beta \sum_i x_i y_i.
\]

Naively, the parameters and sufficient statistics are

\[
\theta = (\mu, a_1, \ldots, a_n, \beta)
\]

\[
t(y) = (y_\cdot, y_1, \ldots, y_n, \sum_i x_i y_i)
\]

Note that

1. \(y_\cdot\) is a function of \(y_1, \ldots, y_n\) (this leads to side conditions on the \(a_i\)'s);
2. \(\sum x_i y_i\) is a function of \(y_1, \ldots, y_n\) (the \(x_i\)'s are treated as “fixed”).

This latter phenomenon means that \(\beta\) and the \(a_i\)'s are not jointly estimable.
Confounding

Let’s examine this more explicitly:

\[
\mathbf{t}(\mathbf{y}) \cdot \theta(\mu, \mathbf{a}, \beta) = \mu y.. + \sum_i a_i y_i + \beta \sum_i x_i y_i.
\]

\[
\mathbf{t}(\mathbf{y}) \cdot \theta(\mu, \mathbf{a} - c\mathbf{x}, \beta + c) = \mu y.. + \sum_i (a_i - cx_i) y_i + (\beta + c) \sum_i x_i y_i.
\]

\[
= \mu y.. + \sum_i a_i y_i + \beta \sum_i x_i y_i.
\]

\[
= \mathbf{t}(\mathbf{y}) \cdot \theta(\mu, \mathbf{a}, \beta).
\]

**Nonidentifiability:**

This result implies that for any two values of \(\beta\), say \(\beta_1\) and \(\beta_2\), there are vectors \(\mathbf{a}_1\) and \(\mathbf{a}_2\) such that

\[
l(\mu, \mathbf{a}_1, \beta_1 : \mathbf{y}) = l(\mu, \mathbf{a}_2, \beta_2 : \mathbf{y}).
\]

The data information can’t distinguish between \((\mu, \mathbf{a}_1, \beta_1)\) and \((\mu, \mathbf{a}_2, \beta_2)\).
Modeling options

There are three commonly used methods of addressing this issue:

1. fit the model without sender and receiver effects;
2. fit the model without sender and receiver regressors;
3. use a random effects model.

We don’t want to do 1 if the logistic regression model has been rejected.

We will fit the model in item 2, but use a two-stage procedure for estimating nodal covariate effects: For example,

- obtain $\hat{\theta} = (\hat{\mu}, \hat{\alpha})$;
- fit the regression model $\hat{a}_i = \beta x_i + \epsilon_i$.

This is an ad-hoc approximation to the random effects approach:

- Model $y_{i,j}$ as a function of $a_i$;
- Model $a_i$ as a function of $x_i$

$$a_i = \beta x_i + \epsilon_i$$

$$\{\epsilon_1, \ldots, \epsilon_n\} \sim \text{i.i.d. normal}(0, \sigma_a^2)$$

We will cover such models shortly.
Dyadic covariates for \( p_1 \)

```
fit.p1cov.d <- ergm(netdat ~
    nodematch("male") + nodematch("smoker") + nodematch("program") +
    edges + mutual + sender + receiver )
```

```
summary(fit.p1cov.d)
```

```
## Summary of model fit

## Formula: netdat ~ nodematch("male") + nodematch("smoker") + nodematch("program") +
## edges + mutual + sender + receiver

## Iterations: 4 out of 20

## Monte Carlo MLE Results:

<table>
<thead>
<tr>
<th>Term</th>
<th>Estimate</th>
<th>Std. Error</th>
<th>MCMC % p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodematch.male</td>
<td>0.97589</td>
<td>0.24573</td>
<td>&lt; 1e-04 ***</td>
</tr>
<tr>
<td>nodematch.smoker</td>
<td>0.89583</td>
<td>0.21151</td>
<td>&lt; 1e-04 ***</td>
</tr>
<tr>
<td>nodematch.program</td>
<td>1.20418</td>
<td>0.21740</td>
<td>&lt; 1e-04 ***</td>
</tr>
<tr>
<td>edges</td>
<td>-5.84758</td>
<td>0.85072</td>
<td>&lt; 1e-04 ***</td>
</tr>
<tr>
<td>mutual</td>
<td>3.92387</td>
<td>0.56460</td>
<td>&lt; 1e-04 ***</td>
</tr>
<tr>
<td>sender2</td>
<td>-0.07422</td>
<td>0.97722</td>
<td>0.93948</td>
</tr>
<tr>
<td>sender3</td>
<td>1.92965</td>
<td>0.93595</td>
<td>0.03952 *</td>
</tr>
<tr>
<td>sender4</td>
<td>-Inf</td>
<td>0.00000</td>
<td>&lt; 1e-04 ***</td>
</tr>
<tr>
<td>sender5</td>
<td>-Inf</td>
<td>0.00000</td>
<td>&lt; 1e-04 ***</td>
</tr>
<tr>
<td>sender6</td>
<td>-0.15274</td>
<td>1.08341</td>
<td>0.88791</td>
</tr>
<tr>
<td>sender7</td>
<td>3.38719</td>
<td>0.90082</td>
<td>0.00018 ***</td>
</tr>
</tbody>
</table>
```

---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Null Deviance: 1375 on 992 degrees of freedom
Residual Deviance: NaN on 925 degrees of freedom
AIC: NaN BIC: NaN (Smaller is better.)

Warning: The following terms have infinite coefficient estimates:
sender4, sender5, sender13, sender18, sender20, sender27, sender30, sender31, receiver18.
Extracting row and column effects

\[ a.hat<-c(0, fit.p1cov.d$coef[4+(2:nrow(Y))] ) \]
\[ b.hat<-c(0, fit.p1cov.d$coef[4+ nrow(Y)-1 + (2:nrow(Y))] ) \]
Nodal covariate effects

How does a covariate $x = \{x_1, \ldots, x_n\}$ relate to

- outgoingness ($a_1, \ldots, a_n$)?
- popularity ($b_1, \ldots, b_n$)?

![Box plots showing the distribution of $a$ and $b$ for different smoking statuses](image)
lm(a.hat~xsmoke)

## Error in lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...):
NA.NaN.Inf in 'y'

The problem here is that \( \hat{a}_i \) is \(-\infty\) for nodes with zero outdegree. What can we do?

0. give up;
1. remove the problematic observations;
2. replace the problematic observations with some large negative value;
3. fit a random effects model.

Item 1 removes information and biases the results:

- Zero degree nodes are highly informative about covariate effects.
- Their removal could bias the estimated effects towards zero.

Item 2 requires we can pick the “right” replacement value.
The results suggest that smoking doesn’t have a large effect on sender or receiver effects, and hence on outgoingness or popularity.
Ad-hoc statistical evaluation

**However:** What if

- all $-\infty a_i$’s corresponded to smokers?
- all $-\infty b_j$’s corresponded to nonsmokers?

Either possibility would suggest estimating the parameter as further away from zero, making it “more significant.”

```r
xsmoke[is.na(a.hat)]
## [1] 0 0 1 0 0 0 0 1
xsmoke[is.na(b.hat)]
## [1] 0
mean(xsmoke)
## [1] 0.40625
mean(xsmoke[is.na(a.hat)])
## [1] 0.25
```

These results don’t give indications of strong relationships between smoking and the tendency to send or receive ties.
mdyad(Y)

## [1] 43

## expected value conditional on outdegree  WF p. 517

\[
\frac{(\text{sum}(Y,\text{na.rm=}\text{TRUE})^2 - \text{sum}(\text{rsum}(Y)^2))}{2 \times (\text{nrow}(Y)-1)^2}
\]

## [1] 1.178715
fit.0 <- ergm(Y ~ edges)
s.SIM0 <- NULL
for(s in 1:S)
{
  Ysim <- as.matrix(simulate(fit.0))
  diag(Ysim) <- NA
  s.SIM0 <- rbind(s.SIM0, c(sd(rsum(Ysim)), sd(csum(Ysim)), mdyad(Ysim)))
}
Additionally, we have the following covariates:

- **Nodal covariates:**
  - population
  - gdp
  - polity

- **Dyad covariates:**
  - exports
  - shared IGOs
  - geographic distance

Let’s see if these covariates account for any of the network patterns.
Coding covariates

It is common to log values of money, population and distance:

```r
colnames(Xn)
## [1] "pop" "gdp" "polity"
Xn[,1:2]<-log(Xn[,1:2])
colnames(Xn)<-c("lpop","lgdp","polity")
netdat<-network(Y,vertex.attr=as.data.frame(Xn))
```

Dyad covariates enter into `ergm` via the `edgecov` function:

```r
fit.cov.ergm<-ergm( netdat ~ edges +
    nodeocov("lpop") + nodeocov("lgdp") + nodeocov("polity") +
    nodeicov("lpop") + nodeicov("lgdp") + nodeicov("polity") +
    edgecov(Xpol) + edgecov(Xigo) + edgecov(Xldst) + edgecov(Xlexp) + edgecov(Xlimp))
```
Logistic regression fit

summary(fit.cov.ergm)

##
## ==========================  
## Summary of model fit  
## ==========================  
##
## Formula: netdat ~ edges + nodeocov("lpop") + nodeocov("lgdp") + nodeocov("polity") +  
## nodeicov("lpop") + nodeicov("lgdp") + nodeicov("polity") +  
## edgecov(Xpol) + edgecov(Xigo) + edgecov(Xldst) + edgecov(Xlexp) +  
## edgecov(Xlimp)  
##
## Iterations: 9 out of 20  
##
## Monte Carlo MLE Results:  
##
##|              | Estimate | Std. Error | MCMC | % | p-value     |
##|--------------|----------|------------|------|---|-------------|
##| edges        | -2.548601| 0.362832   | 0    | < 1e-04 ***|
##| nodeocov.lpop| 0.204650 | 0.083405   | 0    | 0.014150 *  |
##| nodeocov.lgdp| 0.277993 | 0.080569   | 0    | 0.000561 ***|
##| nodeocov.polity | -0.081600 | 0.012262 | 0    | < 1e-04 ***|
##| nodeicov.lpop | 0.193615 | 0.083862   | 0    | 0.020970 *  |
##| nodeicov.lgdp | 0.171160 | 0.079843   | 0    | 0.032071 *  |
##| nodeicov.polity | -0.037818 | 0.012390 | 0    | 0.002274 ** |
##| edgecov.Xpol  | -0.004510 | 0.001659   | 0    | 0.006551 ** |
##| edgecov.Xigo | -0.011437 | 0.005592   | 0    | 0.040844 *  |
##| edgecov.Xldst | -2.663417 | 0.142696   | 0    | < 1e-04 ***|
##| edgecov.Xlexp| 0.058343  | 0.426599   | 0    | 0.891219    |
##| edgecov.Xlimp | -0.035318 | 0.428694   | 0    | 0.934341    |
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
s.SIM<-NULL

for(s in 1:S)
{
  Ysim<-as.matrix(simulate(fit.cov.ergm))
  diag(Ysim)<-NA
  s.SIM<-rbind(s.SIM, c(sd(rsum(Ysim)),sd(csum(Ysim)),mdyad(Ysim)))
}
Improvement via covariates
Summary

• Covariates can be included in the ERGMs.
  • dyad level covariates: nodematch, edgecov and others;
  • node level covariates: nodeocov, nodeicov and others.

• Covariates can often partially explain degree heterogeneity and reciprocity.

• Node-level parameters are confounded with node-level covariate effects.
  • two stage approach: fit node-level parameters, and then relate to covariates;
  • random effects model: next lecture.