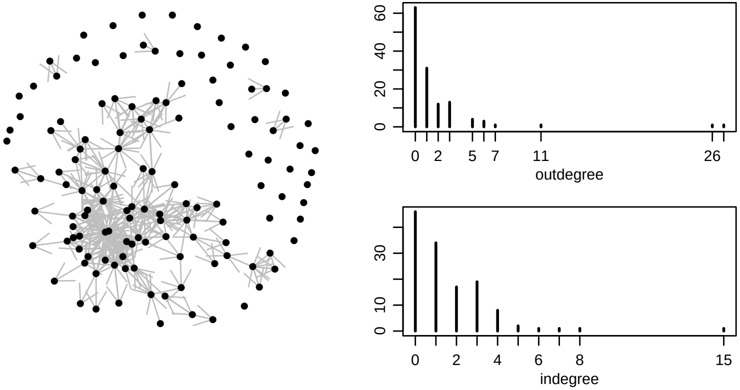
Testing reciprocity in RCE models

567 Statistical analysis of social networks

Peter Hoff

Statistics, University of Washington

Conflict in the 90s




```
sd(rsum(Y))

## [1] 3.589398

sd(csum(Y))

## [1] 1.984451
```

Model selection

```
fit.0<-glm( y ~ 1, family=binomial)
fit.r<-glm( y ~ C(factor(ridx),sum) , family=binomial)
fit.c<-glm( y ~ C(factor(cidx),sum) , family=binomial)
fit.rc<-glm( y ~ C(factor(ridx),sum)+C(factor(cidx),sum), family=binomial)

AIC(fit.0)

## [1] 2197.674

AIC(fit.r)

## [1] 1947.604

AIC(fit.c)

## [1] 2176.021

AIC(fit.rc)

## [1] 1897.398
```

For these data, the full RCE model is best among these four.

Evaluating the RCE model

$$H : \log \text{odds}(Y_{i,j} = 1) = \mu + a_i + b_j, \quad Y_{i,j}\text{'s independent}$$

Let's evaluate H with the following test statistics: $\mathbf{s}(\mathbf{Y}) = \{s_1(\mathbf{Y}), s_2(\mathbf{Y}), s_3(\mathbf{Y})\}$

- $s_1(\mathbf{Y}) = \text{sd}(\text{outdegree})$;
- $s_2(\mathbf{Y}) = \text{sd}(\text{indegree})$;
- $s_3(\mathbf{Y}) = \text{reciprocated dyads}$

$$s_3(\mathbf{Y}) = \sum_{i<j} y_{i,j} y_{j,i}$$

If H is true, then

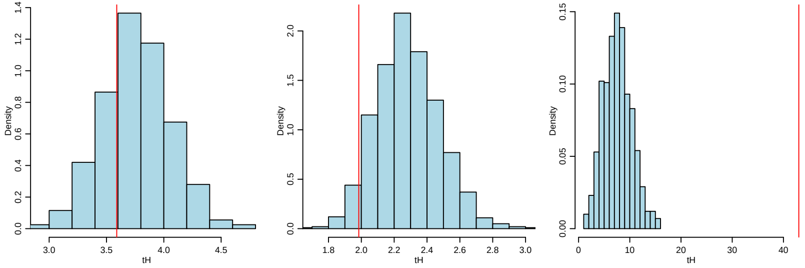
- $\mathbf{Y}$ should look like $\tilde{\mathbf{Y}} \sim RCE(\mu, \mathbf{a}, \mathbf{b})$ for some $(\mu, \mathbf{a}, \mathbf{b})$, but
- we can't simulate from this distribution as we don't know $(\mu, \mathbf{a}, \mathbf{b})$.

We will first use the ad-hoc "best-case" approach.

Best case comparison

```
mu.hat<-fit.rc$coef[1]
a.hat<- fit.rc$coef[1+1:(nrow(Y)-1)]   ; a.hat<-c(a.hat,-sum(a.hat) )
b.hat<- fit.rc$coef[nrow(Y)+1:(nrow(Y)-1)]   ; b.hat<-c(b.hat,-sum(b.hat) )

theta.mle<- mu.hat+ outer(a.hat,b.hat,"+")
p.mle<-exp(theta.mle)/(1+exp(theta.mle))

s.H<-NULL
for(s in 1:S)
{
  Ys<-matrix(rbinom(nrow(Y)^2,1,p.mle),nrow(Y),nrow(Y)) ; diag(Ys)<-NA
  s.H<-rbind(s.H,c(sd(rsum(Ys)),sd(csum(Ys)),sum(Ys*t(Ys)/2,na.rm=TRUE)))
}
```

Best case comparison



Conditional testing

Recall the "best case scenario" evaluation is somewhat ad-hoc.

Compare to the conditional evaluation:

Suppose $\mathbf{Y} \sim SRG(n, \theta)$:

- $\{\mathbf{Y}|y_{\cdot\cdot}\} \not\sim SRG(n, \theta)$;
- $\{\mathbf{Y}|y_{\cdot\cdot}\} \sim SRG(n, y_{\cdot\cdot})$.

Similarly, suppose $\mathbf{Y} \sim RCE(\mu, \mathbf{a}, \mathbf{b})$:

- $\{\mathbf{Y}|y_{\cdot\cdot}, \{y_{i\cdot}\}, \{y_{\cdot i}\}\} \not\sim RCE(\hat{\mu}, \hat{\mathbf{a}}, \hat{\mathbf{b}})$.
- $\{\mathbf{Y}|y_{\cdot\cdot}\{y_{i\cdot}\}, \{y_{\cdot i}\}\} \sim ?$

Return to the SRG:

$$\Pr(\mathbf{Y}_{[i,j]} = 1|\theta) = \theta = \frac{e^{\mu}}{1+e^{\mu}}$$
$$\Pr(\mathbf{Y}_{[i,j]} = y_{i,j}|\theta) = \frac{e^{\mu y_{i,j}}}{1+e^{\mu}}$$
$$\Pr(\mathbf{Y}|\theta) = e^{\mu y_{\cdot\cdot}} g(\mu)$$

This is a very simple **exponential family model**, or **exponentially parameterized random graph model (ERGM)**.

More generally, an ERGM is of the form

$$Pr(\mathbf{Y}|\theta) = e^{t(\mathbf{Y})\cdot\boldsymbol{\theta}} g(\boldsymbol{\theta}),$$

where

- $t(\mathbf{Y}) = (t_1(\mathbf{Y}), \ldots, t_p(\mathbf{Y}))$ is a vector of **statistics**;
- $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$ is a vector of **parameters**;
- $t(\mathbf{Y}) \cdot \boldsymbol{\theta} = \sum t_j(\mathbf{Y})\theta_j$.

RCE as ERGM

Can the RCE model be expressed as an ERGM?

$$\begin{aligned}
\Pr(\mathbf{Y}|\mu, \mathbf{a}, \mathbf{b}) &= \prod_{i \neq j} \frac{e^{(\mu + a_i + b_j) y_{i,j}}}{1 + e^{\mu + a_i + b_j}} \\
&= \exp(\mu y_{\cdot\cdot} + \sum a_i y_{i\cdot} + \sum b_j y_{\cdot j}) \prod_{i \neq j} (1 + e^{\mu + a_i + b_j})^{-1} \\
&= \exp(t(\mathbf{Y}) \cdot \boldsymbol{\theta}) g(\boldsymbol{\theta})
\end{aligned}$$

where

$$\begin{aligned}
t(\mathbf{Y}) &= (y_{\cdot\cdot}, y_{1\cdot}, \ldots, y_{n\cdot}, y_{\cdot 1}, \ldots, y_{\cdot n}) \\
\boldsymbol{\theta} &= (\mu, a_1, \ldots, a_n, b_1, \ldots, b_n)
\end{aligned}$$

So yes, the RCE model is an ERGM. The **sufficient statistics** that generate the model are the out and indegrees.

- the sum $y_{\cdot\cdot}$ can be computed from the degrees;
- the term "sufficient" means sufficient for inferring the parameters, assuming the model is correct.

Conditional tests for ERGMs

Suppose you want to evaluate the adequacy of an ERGM:

$$H : \Pr(\mathbf{Y}|\theta) = e^{t(\mathbf{Y})\cdot\boldsymbol{\theta}} g(\boldsymbol{\theta}) \ , \text{ for some } \theta \in \Theta.$$

Consider evaluation of H based on the statistics $s(\mathbf{Y})$
(where s is not a function of $\mathbf{t}$).

How can we reject or accept H based on s, without knowing $\boldsymbol{\theta}$?

Conditional tests for ERGMs

Recall our principle for testing:

If $\mathbf{Y} \sim ERGM(t, \theta)$ for some $\theta \in \Theta$, then

$\mathbf{Y}$ should "look like" another sample from $ERGM(t, \theta)$

- (but we can't generate these).

$\mathbf{Y}$ should "look like" samples $\tilde{\mathbf{Y}}$ from $ERGM(t, \theta)$ for which $t(\mathbf{Y}) = t(\tilde{\mathbf{Y}})$

- (can we generate these?)

Conditional tests for ERGMs

Let $t(\mathbf{Y}) = t_{obs}$.

$$\begin{aligned}
\Pr(\tilde{\mathbf{Y}}|\boldsymbol{\theta}, t(\tilde{\mathbf{Y}}) = t_{obs}) &= \frac{\Pr(\tilde{\mathbf{Y}} \cap t(\tilde{\mathbf{Y}}) = t_{obs}|\boldsymbol{\theta})}{\Pr(t(\tilde{\mathbf{Y}}) = t_{obs}|\boldsymbol{\theta})} \\
&= \frac{\exp(t(\tilde{\mathbf{Y}}) \cdot \boldsymbol{\theta}) g(\boldsymbol{\theta}) \times 1(t(\tilde{\mathbf{Y}}) = t_{obs})}{\sum_{\check{\mathbf{Y}}} \exp(t(\check{\mathbf{Y}}) \cdot \boldsymbol{\theta}) g(\boldsymbol{\theta}) \times 1(t(\check{\mathbf{Y}}) = t_{obs})} \\
&= \frac{1(t(\tilde{\mathbf{Y}}) = t_{obs})}{\sum_{\check{\mathbf{Y}}} 1(t(\check{\mathbf{Y}}) = t_{obs})}
\end{aligned}$$

This is the uniform distribution over graphs $\tilde{\mathbf{Y}}$ for which $t(\tilde{\mathbf{Y}}) = t_{obs}$ $(= t(\mathbf{Y}))$.

Conditional testing procedure

1. Compute $s_{\text{obs}} = s(\mathbf{Y})$;
2. For $k \in \{1, \ldots, K\}$:
   2.1 Simulate $\tilde{\mathbf{Y}}_k$ uniformly from graphs with $t(\tilde{\mathbf{Y}}) = t(\mathbf{Y})$;
   2.2 Compute $s_k = s(\tilde{\mathbf{Y}})$.
3. Compare s_{obs} to $s_1, \ldots, s_K$.

Conditionally uniform distributions

$$t(\mathbf{Y}) = \{y_{\cdot\cdot}, y_{1\cdot}, \dots, y_{n\cdot}, y_{\cdot 1}, \dots, y_{\cdot n}\} = t_{obs}$$

How can we simulate $\tilde{\mathbf{Y}}$ uniformly from the set of graphs with $t(\tilde{\mathbf{Y}}) = t_{obs}$?

Rejection sampling:. Given a current set of simulations $\{\tilde{\mathbf{Y}}^{(1)}, \dots, \tilde{\mathbf{Y}}^{(s)}\}$,

1. Simulate $\tilde{\mathbf{Y}} \sim SRG(n, y_{\cdot\cdot}^{obs})$
2. If $t(\tilde{\mathbf{Y}}) = t_{obs}$, then set $\tilde{\mathbf{Y}}^{(s+1)} = \tilde{\mathbf{Y}}$. Otherwise, return to step 1.

As you can imagine, this algorithm is not practical for large n.

MCMC sampling

MCMC sampling: Given a current set of simulations $\{\tilde{\mathbf{Y}}^{(1)}, \ldots, \tilde{\mathbf{Y}}^{(s)}\}$,

1. Make a random perturbation $\tilde{\mathbf{Y}}$ of $\tilde{\mathbf{Y}}^{(s)}$ so that $t(\tilde{\mathbf{Y}}) = t(\tilde{\mathbf{Y}}^{(s)}) = t_{obs}$;
2. Set $\tilde{\mathbf{Y}}^{(s+1)} = \tilde{\mathbf{Y}}$.

This generates a random *dependent* sequence from $\Pr(\tilde{\mathbf{Y}} | t(\tilde{\mathbf{Y}}) = t_{obs})$.

- dependent, because $\tilde{\mathbf{Y}}^{(s+1)}$ depends on $\tilde{\mathbf{Y}}^{(s)}$.
- contrast this with sampling $\tilde{\mathbf{Y}}^{(1)}, \ldots, \tilde{\mathbf{Y}}^{(s)}$ i.i.d. from $SRG(n, y_{\cdot\cdot})$.

Random perturbations

Given $\tilde{\mathbf{Y}}^{(s)}$, how can we construct a perturbation $\tilde{\mathbf{Y}}^{(s+1)}$ so $t(\tilde{\mathbf{Y}}^{(s+1)}) = t(\tilde{\mathbf{Y}}^{(s)})$?

NA	0	1	0	1	0	1	3
1	NA	0	1	1	0	0	3
0	1	NA	0	0	0	0	1
0	1	1	NA	0	1	1	4
1	0	0	0	NA	0	0	1
0	0	1	1	1	NA	1	4
1	0	1	0	0	0	NA	2
3	2	4	2	3	1	3	18

Suppose we randomly switch a tie with a non-tie, within a row:

Suppose we randomly switch a tie with a non-tie, within a row:

NA	0	1	0	1	0	1	3
1	NA	0	1	$1 \to 0$	$0 \to 1$	0	$3 \to 3$
0	1	NA	0	0	0	0	1
0	1	1	NA	0	1	1	4
1	0	0	0	NA	0	0	1
0	0	1	1	1	NA	1	4
1	0	1	0	0	0	NA	2
3	2	4	2	$3 \to 2$	$1 \to 2$	3	18

The outdegrees are maintained, but the indegrees change.

Random perturbations

Suppose we randomly switch a tie with a non-tie, within a column:

NA	0	1	0	1	0	1	3
1	NA	0	1	$1 \to 0$	0	0	$3 \to 2$
0	1	NA	0	0	0	0	1
0	1	1	NA	0	1	1	4
1	0	0	0	NA	0	0	1
0	0	1	1	1	NA	1	4
1	0	1	0	$0 \to 1$	0	NA	$2 \to 3$
3	2	4	2	$3 \to 3$	1	3	18

The indegrees are maintained, but the outdegrees change.

Random perturbations

To perturb while maintaining both in and outdegrees, we must update at least four cells at once:

NA	0	1	0	1	0	1	3
1	NA	$0 \to 1$	1	$1 \to 0$	0	0	$3 \to 3$
0	1	NA	0	0	0	0	1
0	1	1	NA	0	1	1	4
1	0	0	0	NA	0	0	1
0	0	1	1	1	NA	1	4
1	0	$1 \to 0$	0	$0 \to 1$	0	NA	$3 \to 3$
3	2	$3 \to 3$	2	$3 \to 3$	1	3	18

The in and outdegrees are maintained.

Random perturbations

To perturb while maintaining both in and outdegrees, we must update at least four cells at once:

Algorithm:

1. Set $\tilde{\mathbf{Y}}^{(s+1)} = \tilde{\mathbf{Y}}^{(s)}$.
2. Randomly select two rows $\mathbf{i} = (i_1, i_2)$ and two columns $\mathbf{j} = (j_1, j_2)$
3. Obtain the submatrix $\tilde{\mathbf{Y}}_{ij}^{(s+1)} = \tilde{\mathbf{Y}}_{[(i_1,i_2),(j_1,j_2)]}^{(s+1)}$.
4. Perturb $\tilde{\mathbf{Y}}_{ij}^{(s+1)}$ as follows:
   - If $\tilde{\mathbf{Y}}_{ij}^{(s+1)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, set $\tilde{\mathbf{Y}}_{ij}^{(s+1)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
   - If $\tilde{\mathbf{Y}}_{ij}^{(s+1)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, set $\tilde{\mathbf{Y}}_{ij}^{(s+1)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Iteration of this algorithm generates a random sequence $\tilde{\mathbf{Y}}^{(1)}, \ldots, \tilde{\mathbf{Y}}^{(S)}$

- the value of $t(\tilde{\mathbf{Y}}^{(s)})$ is constant throughout the sequence;
- the sequence visits a subset of $\tilde{\mathbf{Y}}$-values for which $t(\tilde{\mathbf{Y}}) = t(\tilde{\mathbf{Y}}^{(1)}) = t_{obs}$.

This means the algorithm samples uniformly from a subset of

$$\{\tilde{\mathbf{Y}} : t(\tilde{\mathbf{Y}}) = t_{obs}\}.$$

An MCMC sampler

To sample uniformly from *all* of $\{\tilde{\mathbf{Y}} : t(\tilde{\mathbf{Y}}) = t_{obs}\}$ we need to be able to make larger perturbations to $\tilde{\mathbf{Y}}$.

Here is an outline of an algorithm that works properly:

Given a sequence $\{\tilde{\mathbf{Y}}^{(1)}, \ldots, \tilde{\mathbf{Y}}^{(s)}\}$, generate $\tilde{\mathbf{Y}}^{(s+1)}$ as follows:

1. Construct $\tilde{\mathbf{Y}}_1$ by perturbing a random subsquare of $\tilde{\mathbf{Y}}^{(s)}$ as before;
2. Construct $\tilde{\mathbf{Y}}_2$ by perturbing a random triad of $\tilde{\mathbf{Y}}_1$;
3. Set $\tilde{\mathbf{Y}}^{(s+1)} = \tilde{\mathbf{Y}}_2$.

How is the perturbation of a triad done?

1. Randomly select a triad (i, j, k);
2. Perturb the triad:
    - If $i \to j \to k \to i$, then set $i \leftarrow j \leftarrow k \leftarrow i$;
    - If $i \leftarrow j \leftarrow k \leftarrow i$, then set $i \to j \to k \to i$.

Exercise: Show that such perturbations leave in and outdegrees unchanged.

A single iteration

```
rY.Yrc<-function(Y)
{
  n<-nrow(Y)

  ###
  i<-sample(1:n,4)
  Yi<-Y[ i[1:2], i[3:4] ]
  if( abs(Yi[1,1]+Yi[2,2]-Yi[1,2]-Yi[2,1])==2)
  { Y[ i[1:2], i[3:4] ] <-  1-Yi }
  ###


  ###
  i<-sample(1:n,3)
  idx<- rbind( c(i[1],i[2]) , c(i[1],i[3]) , c(i[2],i[3]) ,
               c(i[2],i[1]) , c(i[3],i[1]) , c(i[3],i[1])  )
  y<-Y[idx]
  if( all( y[2*(1:3)-1]== 1 - y[2*(1:3) ] )) { Y[idx]<- 1-y }
  ###

  Y
}
```

A more efficient sampler

```
rY.Yrc<-function(Y)
{

  ###
  n<-nrow(Y)
  i1<-resample( (1:n)[apply(Y,1,sum,na.rm=TRUE)>0 ]  ,1)
  j1<-resample(which(Y[i1,]==1),1) ; j2<- resample(which(Y[i1,]==0),1)
  yj1j2<-Y[,c(j1,j2) ]
  if(length(c(j1,j2))==2)
  {
    nnodes<- which( yj1j2[,1]==0 & yj1j2[,2]==1 )
    if(length(nnodes)>0)
    {
      i2<-resample(nnodes,1)
      if(length(i2)==1){  Y[c(i1,i2),c(j1,j2)] <- 1 - Y[c(i1,i2),c(j1,j2)] }
    }
  }
  ###



  ###
  Y1<-Y ; diag(Y1)<- 0
  Y2<-Y1%*%Y1
  ikt<-which( Y2*t(Y1)*(1-Y1)  > 0,arr.ind=TRUE )
  ik<-ikt[ resample(1:nrow(ikt),1)  ,]
  j<- resample(which(Y1[ik[1],]==1 & Y1[,ik[1]]==0 &
                     Y1[,ik[2] ]==1 & Y1[ik[2], ]==0 ), 1 )
  if(length(j)>0)
  {
    ijk<-c(ik[1],j,ik[2] )
    Y[ijk,ijk]<-1-Y[ijk,ijk]
  }
  ###
  Y
```

Evaluating the RCE model

$$H : \log \text{odds}(Y_{i,j} = 1) = \mu + a_i + b_j, \quad Y_{i,j}\text{'s independent}$$

If H is true, then

- $\mathbf{Y}$ should look like $\tilde{\mathbf{Y}} \sim RCE(\mu, \mathbf{a}, \mathbf{b})$ for some $(\mu, \mathbf{a}, \mathbf{b})$.
  - We can't simulate from this distribution as we don't know $(\mu, \mathbf{a}, \mathbf{b})$.
- $\mathbf{Y}$ should look like $\tilde{\mathbf{Y}} \sim$ uniform on $t(\tilde{\mathbf{Y}}) = t(\mathbf{Y})$.
  - We can sample from this distribution using MCMC.

For the conflict data, let's evaluate H with some test statistics:

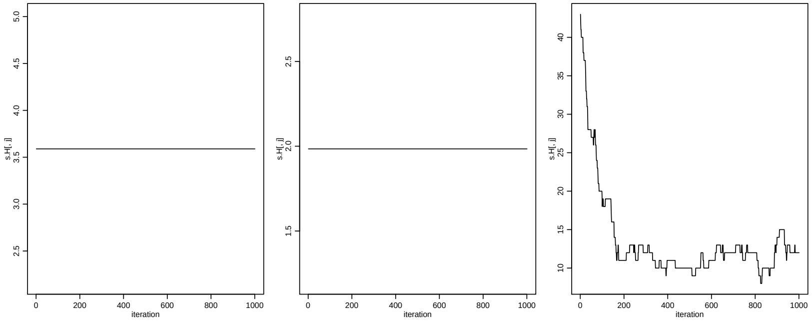
$\mathbf{s}(\mathbf{Y}) = \{s_1(\mathbf{Y}), s_2(\mathbf{Y}), s_3(\mathbf{Y})\}$

- $s_1(\mathbf{Y}) = \text{sd(outdegree)}$;
- $s_2(\mathbf{Y}) = \text{sd(indegree)}$;
- $s_3(\mathbf{Y}) =$ reciprocated dyads

$$s_3(\mathbf{Y}) = \sum_{i<j} y_{i,j} y_{j,i}$$

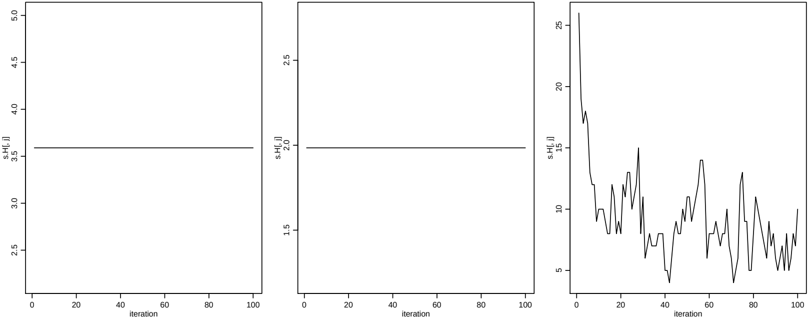
MCMC approximation

```
s.H<- c(sd(rsum(Y)),sd(csum(Y)),sum(Y*t(Y),na.rm=TRUE)/2)

Ys<-Y
for(s in 1:S)
{
  Ys<-rY.Yrc(Ys)
  s.H<-rbind(s.H,c(sd(rsum(Ys)),sd(csum(Ys)),sum(Ys*t(Ys)/2,na.rm=TRUE)))
}
```

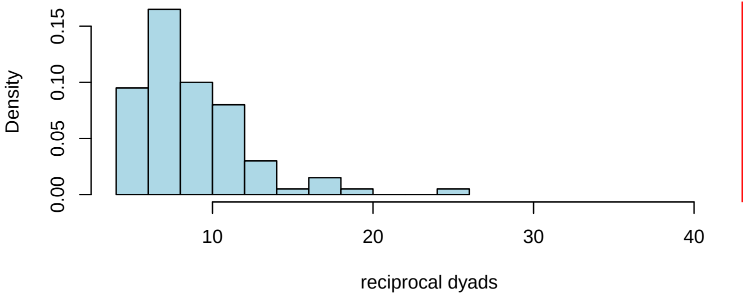


MCMC approximation

```
s.H<-matrix(0,nrow=Sbig/Sout,ncol=3)

Ys<-Y
for(s in 1:Sbig)
{
  Ys<-rY.Yrc(Ys)
  if(s%%Sout==0)
  {
    s.H[s/Sout,]<-c(sd(rsum(Ys)),sd(csum(Ys)),sum(Ys*t(Ys)/2,na.rm=TRUE))
  }
}
```



Reciprocity



The RCE model fails to capture the number of reciprocal dyads:

- There are more reciprocal dyads than expected under any RCE model;
- This makes sense - conflict is likely to be reciprocated;
- Statistically, this suggests that $Y_{i,j}$ and $Y_{j,i}$ *are not* independent.

References

- Wasserman and Faust, sections 13.5-13.8
- Rao, Jana, Bandyopadhyay 1996. A Markov chain Monte Carlo method for generating random (0,1)-matrices with given marginals.
- McDonald, Smith and Forster 2007. Markov chain Monte Carlo exact inference for social networks.

Dyads

A **dyad** is an unordered pair of nodes

- $\{i, j\} = \{j, i\}$ is the pair of nodes i and j.

For an undirected binary relation, the dyad can be in one of four states:

- $i \not\leftarrow j$ (null)
- $i \rightarrow j$ (asymmetric)
- $i \leftarrow j$ (asymmetric)
- $i \leftrightarrow j$ (mutual)

These correspond to the following values of $(y_{i,j}, y_{j,i})$

- $i \not\leftarrow j$ (0,0)
- $i \rightarrow j$ (1,0)
- $i \leftarrow j$ (0,1)
- $i \leftrightarrow j$ (1,1)

Reciprocity

Reciprocity or **mutuality** describes the tendency for ties to be mutual.

A quantification of reciprocity is a standard part of SNA

- Statistical analysis requires we account for dyadic dependence, if present.
- Reciprocity by itself may be of interest in terms of evaluating social theory.

The dyad census

Let

- $M=$ the number of mutual dyads;
- $A=$ the number of asymmetric dyads;
- $N=$ the number of null dyads.

Then $M + A + N =$ the number of dyads $= \binom{n}{2}$.

Computing M: Recall, $y_{i,j}y_{j,i} = 1$ only if both $y_{i,j}$ and $y_{j,i}$ are 1.

$$M = \sum_{i<j} y_{i,j}y_{j,i}$$

Computing A: A equals the number of links minus the number of reciprocated links.

$$A = y_{\cdot\cdot} - 2M$$

Computing N: Recall $M + A + N =$ the number of dyads $= \binom{n}{2}$.

$$N = \binom{n}{2} - M - A$$

Other formula are available.

Dyad census in R

```
M<-sum(Y*t(Y),na.rm=TRUE)/2

A<-sum(Y,na.rm=TRUE) - 2*M

N<- choose(nrow(Y),2) - M - A

M

## [1] 43

A

## [1] 117

N

## [1] 8225

### check
sum( (1-Y)*t((1-Y)),na.rm=TRUE)/2

## [1] 8225
```

Evaluating M

For what types of networks will

- M be large?
- M be small?

How do we evaluate M? What should it be compared to?

- Its distribution under some null model (RCE via MCMC);
- Its expected value under some simple conditions (direct calculation).

In the latter case, we

1. posit some simple conditions H on tie selection;
2. calculate $E[M|H]$;
3. compare M to $E[M|H]$.

This is less informative than comparing to a conditional distribution, but can be done without simulation in some cases.

Mutuality under fixed choice

Fixed nomination scheme:
A network survey instrument in which each individual is required to make exactly d nominations, where d is fixed in advance.

This is a common type of network survey instrument used in institutions:

- Each member given a roster of all members;
- Each member checks off their "top d" friends;
- Often ranks of the top d friends are included (fixed rank nomination)

Such an approach is useful when the goal is to

- to distinguish between the strong and weak ties;
- to control for outdegree heterogeneity.

Mutuality under fixed choice

H : Individuals make d nominations uniformly at random.

To say "this network has more mutuality than expected under randomness", we need to calculate $\mathrm{E}[M|H]$

$$\begin{aligned}
\mathrm{E}[M|H] &= \mathrm{E}[\sum_{i<j} y_{i,j} y_{j,i} | H] \\
&= \sum_{i<j} \mathrm{E}[y_{i,j} y_{j,i} | H] \\
&= \binom{n}{2} \mathrm{E}[y_{i,j} y_{j,i} | H] \\
&= \binom{n}{2} \Pr(y_{i,j} = 1 \text{ and } y_{j,i} = 1 | H)
\end{aligned}$$

$$\begin{aligned}
\Pr(y_{i,j} = 1 \text{ and } y_{j,i} = 1 | H) &= \Pr(y_{i,j} = 1 | H) \times \Pr(y_{j,i} = 1 | H) \\
&= \frac{d}{n-1} \times \frac{d}{n-1} = \frac{d^2}{(n-1)^2}
\end{aligned}$$

Mutuality under fixed choice

$$E[M|H] = \binom{n}{2} \frac{d^2}{(n-1)^2}$$
$$= \frac{n(n-1)}{2} \frac{d^2}{(n-1)^2} = \frac{nd^2}{2(n-1)}$$

Mutuality under free choice

H : Individuals make $y_{i\cdot}$ nominations uniformly at random.

It can be shown that

$$\mathrm{E}[M|H] = \frac{y_{\cdot\cdot}^2 - (\sum y_{i\cdot}^2)}{2(n-1)^2}$$

This allows us to evaluate mutuality, controlling for heterogeneity in outdegree.

```
yod<-rsum(Y)
( sum(yod)^2 - sum(yod^2) ) / ( 2*(nrow(Y)-1)^2)

## [1] 1.178715

mean(s.H[,3] )

## [1] 9.21

sum(Y*t(Y),na.rm=TRUE)/2

## [1] 43
```

For these data, there is more reciprocity than expected, controlling for either outdegrees or both in and outdegrees.

A reciprocity parameter

How should we measure reciprocity?

- reciprocity reflects dependence between $Y_{i,j}$ and $Y_{j,i}$;
- a reciprocity metric should measure "the effect" of $Y_{i,j}$ on $Y_{j,i}$

$$\Pr(Y_{j,i} = 1 | Y_{i,j} = 1) = \begin{cases} \Pr(Y_{j,i} = 1) & \text{under independence} \\ 0 & \text{under complete antireciprocity} \\ 1 & \text{under complete reciprocity} \end{cases}$$

Katz and Powell (1955) propose a reciprocity measure ρ:

$$\Pr(Y_{j,i} = 1 | Y_{i,j} = 1) = \Pr(Y_{j,i} = 1) + \rho \Pr(Y_{j,i} = 0)$$

- $\rho = 0$ implies independence;
- $\rho < 0$ implies antireciprocity;
- $\rho > 0$ implies reciprocity.

WF goes through a few ways to estimate this assuming various models.

Log odds ratio

An arguably more natural way to measure reciprocity is with a log-odds ratio:

$$\text{odds}(Y_{j,i} = 1 : Y_{i,j} = 1) = \frac{\Pr(Y_{j,i} = 1 | Y_{i,j} = 1)}{\Pr(Y_{j,i} = 0 | Y_{i,j} = 1)}$$

$$\text{odds}(Y_{j,i} = 1 : Y_{i,j} = 0) = \frac{\Pr(Y_{j,i} = 1 | Y_{i,j} = 0)}{\Pr(Y_{j,i} = 0 | Y_{i,j} = 0)}$$

$$\text{odds ratio}(Y_{j,i} = 1 : Y_{i,j} = 1, Y_{i,j} = 0) = \frac{p_{1|1}}{(1 - p_{1|1})} \frac{(1 - p_{1|0})}{p_{1|0}}$$

Where

$$p_{w|x} = \Pr(Y_{j,i} = w | Y_{i,j} = x)$$

Log odds ratio

Empirical estimates of these probabilities can be obtained from (M, A, N).

$$
\begin{aligned}
\Pr(Y_{j,i} = 1 | Y_{i,j} = 1) &= \frac{\Pr(Y_{i,j} = 1, Y_{j,i} = 1)}{\Pr(Y_{i,j} = 1)} \\
&= \frac{\Pr(Y_{i,j} = 1, Y_{j,i} = 1)}{\Pr(Y_{i,j} = 1, Y_{j,i} = 1) + \Pr(Y_{i,j} = 1, Y_{j,i} = 0)} \\
&\approx \frac{M/T}{M/T + A/(2T)} = \frac{2M}{2M + A}
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
\Pr(Y_{j,i} = 1 | Y_{i,j} = 0) &= \frac{\Pr(Y_{i,j} = 1, Y_{j,i} = 0)}{\Pr(Y_{i,j} = 0)} \\
&= \frac{\Pr(Y_{i,j} = 1, Y_{j,i} = 0)}{\Pr(Y_{i,j} = 0, Y_{j,i} = 0) + \Pr(Y_{i,j} = 1, Y_{j,i} = 0)} \\
&\approx \frac{A/(2T)}{A/(2T) + N/T} = \frac{A}{A + 2N}
\end{aligned}
$$

Reciprocity measure for 90s conflict data

```
M<-sum(Y*t(Y),na.rm=TRUE)/2

A<-sum(Y,na.rm=TRUE) - 2*M

N<- choose(nrow(Y),2) - M - A

M

## [1] 43

A

## [1] 117

N

## [1] 8225

p11<-2*M/(2*M+A)
p10<-A/(A+2*N)

p11

## [1] 0.4236453

p10

## [1] 0.007062232

log( p11 * (1-p10) /( (1-p11) * p10) )

## [1] 4.63808
```

Reciprocity measure in 90s conflict data

```
y<-c(Y) ; x<-c(t(Y))
mean(y[x==1],na.rm=TRUE)

## [1] 0.4236453

mean(y[x==0],na.rm=TRUE)

## [1] 0.007062232

fit<-glm(y~x,family=binomial)
fit

##
## Call:  glm(formula = y ~ x, family = binomial)
##
## Coefficients:
## (Intercept)            x
##      -4.946        4.638
##
## Degrees of Freedom: 16769 Total (i.e. Null);  16768 Residual
##   (130 observations deleted due to missingness)
## Null Deviance:     2196
## Residual Deviance: 1669  AIC: 1673

table(fit$fitted)

##
## 0.00706223230891567   0.423645320196599
##               16567                 203
```

Reciprocity via logistic regression

Exercise: Show that the log-odds ratio is the logistic regression coefficient.

Note: This use of `glm` is not really fitting a model - the outcome is on both sides of the regression equation.