Advanced topics in SAOM

Different types of data

Johan Koskinen

Department of Statistics
Stockholm University
University of Melbourne

February 24, 2023

Preamble

- All material is on the workshop repository https://github.com/johankoskinen/CHDH-SNA
  - Download the RMarkdown file CHDH-SNA-2.Rmd
- In order to run the Markdown you need
  - The R-package
  - The RStudio interface
- We will predominantly use the packages
  - `sna`
  - `network`
  - `RSiena`

Outline of workshops

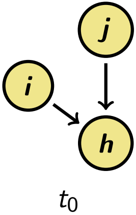
1. (Basic) Introduction to SAOM (Thursday AM)
   - SAOM as an agent-based model
   - How to estimate a SAOM
2. Analysing data with SAOM
   - Different model specifications
   - Different types of data
   - Trouble shooting and dealing with common issues
3. Extensions to SAOM
   - Even more types of data
   - Likelihood-based estimation
   - Settings and imperfect data

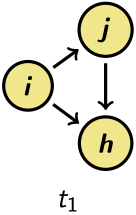
Outline of workshops

1. (Basic) Introduction to SAOM (Thursday AM)
   - SAOM as an agent-based model
   - How to estimate a SAOM
2. Analysing data with SAOM
   - Different model specifications
   - Different types of data
   - Trouble shooting and dealing with common issues
3. Extensions to SAOM
   - Even more types of data
   - Likelihood-based estimation
   - Settings and imperfect data

Continuous-time v discrete-time

Assume network $X(t_0)$ and $X(t_1)$
and that we are happy to fix $X(t_0)$
How do we model closure?





Network regression

$\sharp$ ties at t_1 close open triples at t_0

This approach does not care what happened to the mixed path $i \rightarrow h \leftarrow j$
ERGO: you may find closure even though closure has decreased!

Network regression

Basic problem ties at t_1 are modelled as independent conditional on t_0
Is there any way around this?

Network regression

Basic problem ties at t_1 are modelled as independent conditional on t_0
Is there any way around this?
Robins & Pattison (2001, Random graph models for temporal processes in social networks. J. Math. Sociol.) propose:

$$X(t_1)|X(t_0) \sim ERGM(\theta)$$

with $X(t_0)$ as covariate network.
This is permissible BUT model cannot be interpreted in terms of *change* as $X(t_1)$ is in equilibrium (Block et al., 2018)

Network regression

Dependencies in $X(t_1)$ cannot be accounted for through $X(t_0)$
Unless you assume *continuous-time* process.
What is the effect on prediction?

How does SAOM account for dependence?

SAOM induces marginal dependence in ties of $X(t_1)$ through assuming incremental changes in continuous time where a change only dependens on the past

What type of data do we want to explain

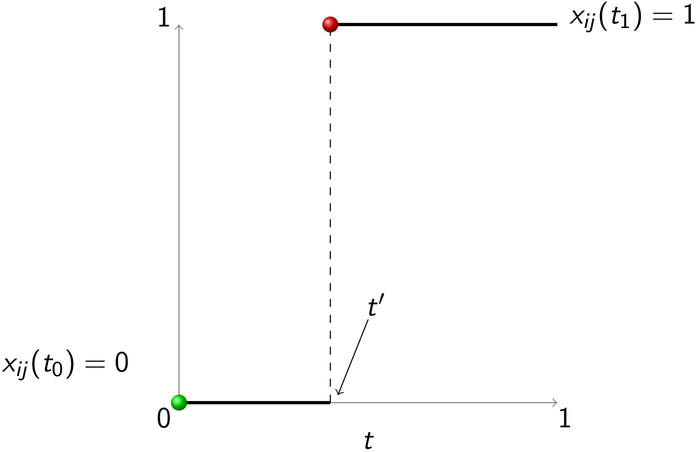
If an element x_{ij} has changed from

$$x_{ij}(t_0) = 0$$

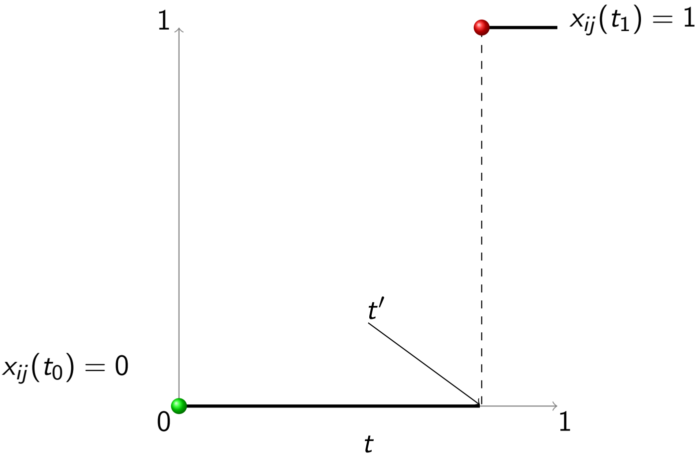
to

$$x_{ij}(t_1) = 1$$

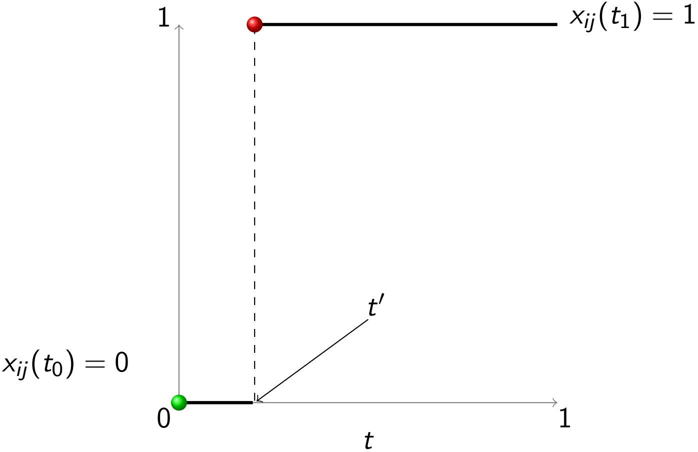
something has changed inbetween t_0 and t_1



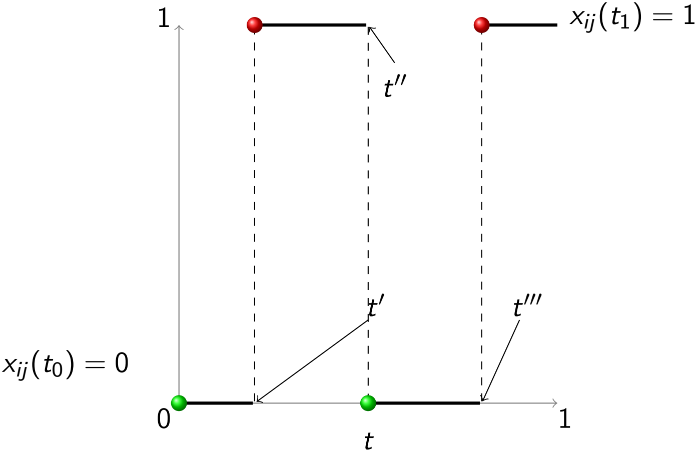
Change at t'



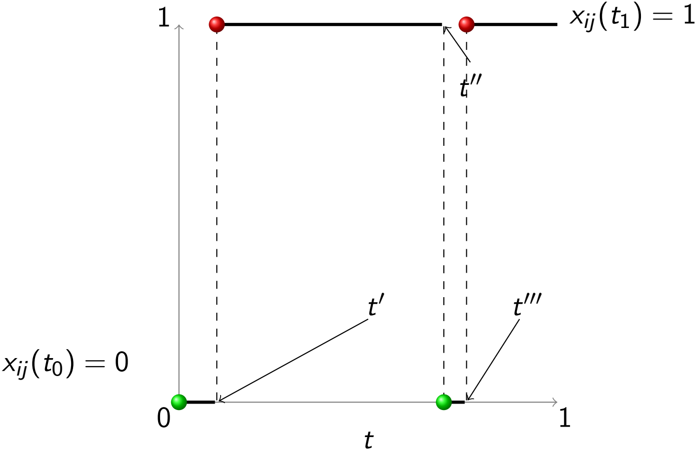
Change at t'



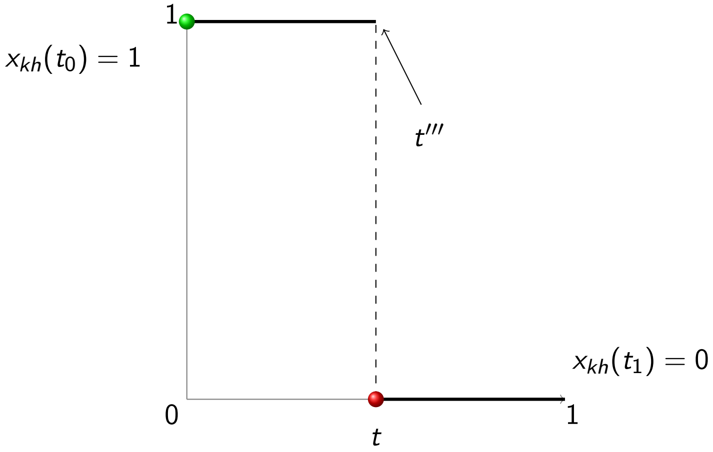
Change at t'



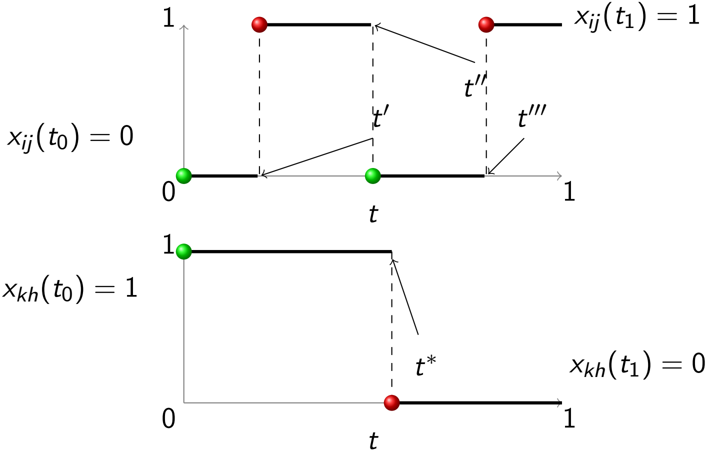
Change at t', t'', *and* t''



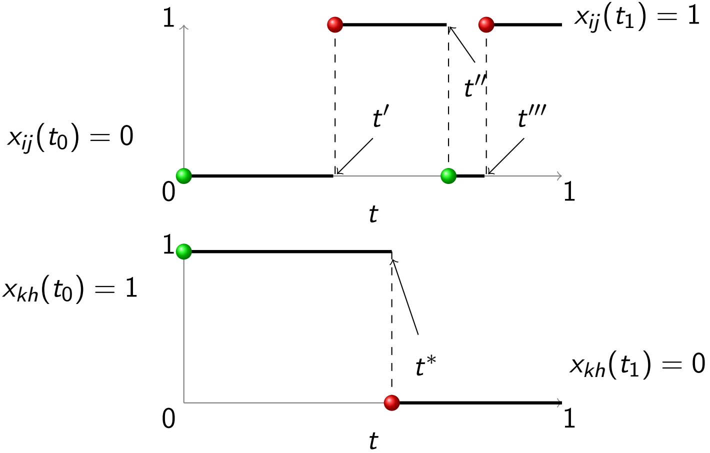
Change at t', t'', *and* t''

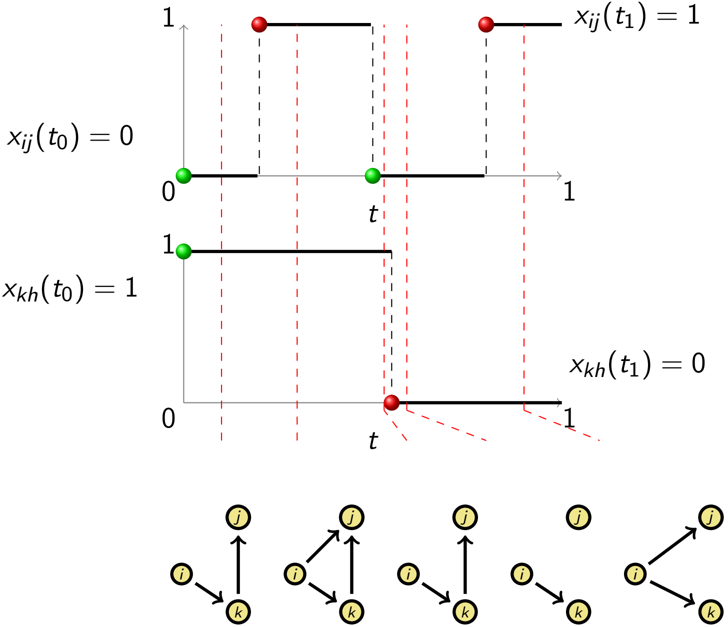


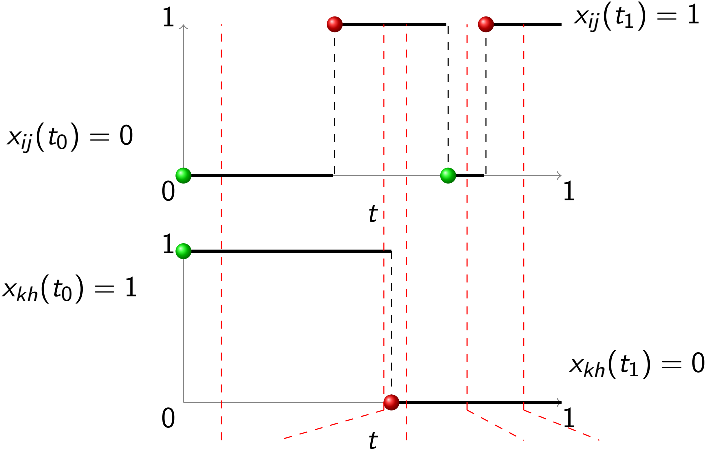
Change at t'''



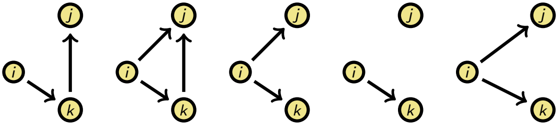
$x_{ij}(t_1) = 1$
$x_{ij}(t_0) = 0$
t'
t''
t'''
t
$x_{kh}(t_0) = 1$
t^*
$x_{kh}(t_1) = 0$
t
0
1


1
$x_{ij}(t_1) = 1$
t''
t'
t'''
$x_{ij}(t_0) = 0$
0
1
t
1
$x_{kh}(t_0) = 1$
t^*
$x_{kh}(t_1) = 0$
0
1
t


$x_{ij}(t_1) = 1$
$x_{ij}(t_0) = 0$
$x_{kh}(t_0) = 1$
$x_{kh}(t_1) = 0$
1
0
1
t
1
0
1
t
j
i
k
j
i
k
j
i
k
j
i
k
j
i
k



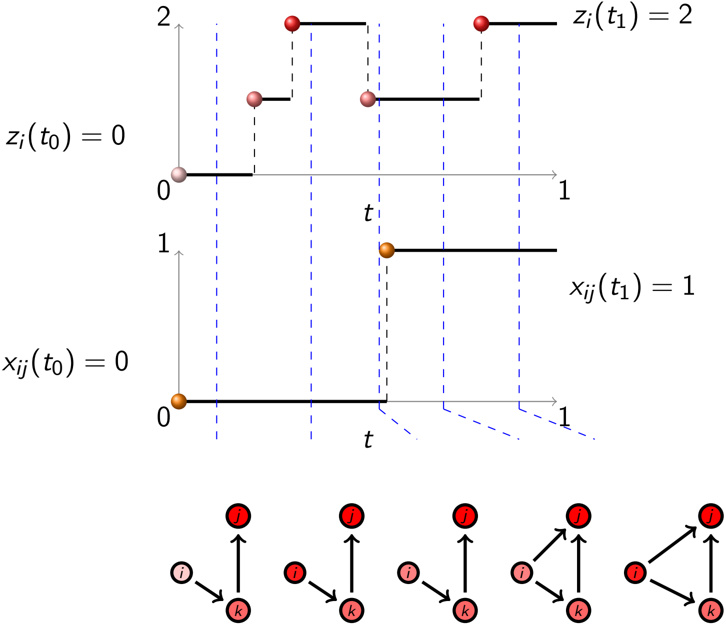
$x_{ij}(t_1) = 1$
$x_{ij}(t_0) = 0$
1
0
1
t
1
$x_{kh}(t_0) = 1$
$x_{kh}(t_1) = 0$
0
1
t



j
i
k
j
i
k
j
i
k
j
i
k
j
i
k

Actor-driven models

Network change process and behavior change process
run simultaneously, and influence each other
being each other's changing constraints.


2
$z_i(t_1) = 2$
$z_i(t_0) = 0$
0
1
t
1
$x_{ij}(t_1) = 1$
$x_{ij}(t_0) = 0$
0
1
t
i
j
k

Missing data in RSiena - MoM

For waves t_0, t_1, t_2
Default (MoM):

- if $X_{ij}(t_0) = NA$, $X_{ij}(t_0) := 0$, becuase networks are sparse
- if x, $X_{ij}(t_1)$ is simulated according to the model
  but $X_{ij}^{\mathrm{sim}}(t_1)$ is not used in calculating target statistics (Hipp et al. 2015, incorrect interpretation)
- if $X_{ij}(t_1) = NA$ and $X_{ij}(t_2) = NA$, $X_{ij}(t_2) := X_{ij}^{\mathrm{sim}}(t_1)$

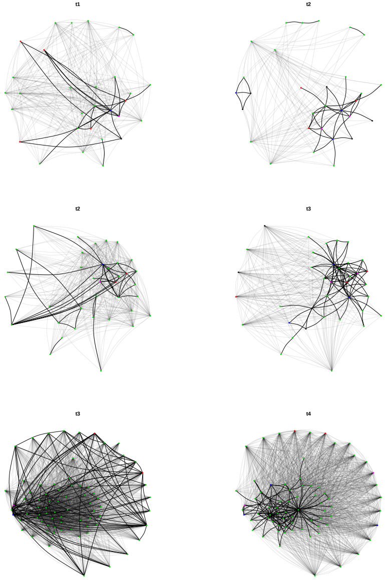
Covariates are imputed using mean.
Hipp et al. (2015) and Krause et al. (2018) impute $X_{ij}(t_0)$ using ERGM/stationary SAOM.

Missing data in RSiena - Likelihood based

For ML (Snijders, Koskinen, Schweinberger, 2010) and Bayes (Koskinen and SNijders, 2007):
Missing values are integrated out (by simulation from fully conditional posterior)
aver 1: no distribution for t_0 so imputed independently
aber 2: for reasons of parralelisation, if $X_{ij}(t_1) = NA$:
$X_{ij}^{\mathrm{sim}}(t_1)$ imputed for interval $t_0 \to t_1$, but
$X_{ij}(t_1) = NA$ is treated as a 'first' observation (i.e. imputed independently)

Missing data in RSiena - Likelihood based

Sampling paths for MoM are simulated forwards (unconstrained)
Sampling parths for ML/Bayes are simulated constrained, so that $X_{ij}^{\mathrm{sim}}(t_m) = X_{ij}^{\mathrm{obs}}(t_m)$, for $m > 0$.
Missing data for ML/Bayes hence reduces constrains and make estimation 'easier' (but less precise)



Waves t_0 through t_3 plotted pariwise $\{t_m, t_{m+1}\}$
Grey: missing dyad (Bright, Koskinen, Malm, 2019)

Changing composition and missing data

Changing composition of node-sets can be dealt with in three (and a half different ways):

- using `sienaCompositionChange` where actors actually enter and leave
- coding ties of leavers as structural zeros, code: `10`
- simlar to composition change, actor must have entered and left at some point, at which point their ties were NA

Using the Markov property, all can be combined with the multigroup option `sienaGroupCreate` by interval to reduce NA.

Using changing coposition

50 actors across 6 waves with 11, 20, and 33 entering and leaving

```
comp <- rep(list(c(1,6)), 50)
comp[[11]] <- c(3,6)
comp[[20]] <- c(1,4)
comp[[33]] <- c(1.5,3, 4.01,6)
changes <- sienaCompositionChange(comp)
```

33 enters halway between waves 1 and 3 with no ties; no one can have tie to 33 betwwen waves 3 and 4.

Structural zeros

For actor with tie-value `10`

they are in the network

if all outgoing ties are `10`, they can still be chosen

if exact times not known, equivalent to `sienaGroupCreate`

Changing composition through missings

If actor leaves inbetween t_m and t_{m+1}
setting ties at t_{m+1} to `10` loses information at t_m
using NA allows actor to choose and be chosen up until t_{m+1}

Co-evolution with continous outcomes

For the behaviours, the formula of the change probabilities is

$$p_{ihv}(\beta, z) = \frac{\exp(f(i,h,v))}{\sum_{k,u} \exp(f(i,k,u))}$$

where $f(i,h,v)$ is the objective function calculated for the potential new situation after a behaviour change,

$$f(i,h,v) = f_i^Z(\beta, z(i, h \rightsquigarrow v)) \ .$$

A multinomial logit form.

Co-evolution with continous outcomes

What if $Z_i \in \mathbb{R}$ or $Z_i \in \{0, \dots T\}$ for T LARGE?
to go from a value 0 to, say, 50, an actor has to make at least 50 changes w.p.

$$p_{ihv}(\beta, z) = \frac{\exp(f(i,h,v))}{\sum_{k,u} \exp(f(i,k,u))}$$

This attenuates the effects and lead to large rates

Co-evolution with continous outcomes

Solution, assume Brownian motion and apply stochastic differential equation:
Niezink, Snijders (2017). Co-evolution of Social Networks and Continuous Actor Attributes. The Annals of Applied Statistics
(NB: MoM)

Diffusion in stochastic oriented networks

Standard co-evolution of influence and selection in RSiena assumes
The behaviour switches on and off with multinomial probabilities
an unlimited number of times inbetween t_m and t_{m+1}
This does not afford:

- diffusion/contagion/adaption of something that monotonically increasing
- drawing on previous survival analysis techniques (e.g. Strang and Tuma, 1993)

Diffusion in stochastic oriented networks

Standard co-evolution of influence and selection in RSiena assumes
The behaviour switches on and off
an unlimited number of times inbetween t_m and t_{m+1}
This does not afford:

- diffusion/contagion/adaption of something that monotonically increasing
- drawing on previous survival analysis techniques (e.g. Strang and Tuma, 1993)

Diffusion in stochastic oriented networks

Diffusion of innovations in dynamic networks

Charlotte C. Greenan

University of Oxford, Oxford, UK.

Summary. The evolution of a dynamic social network and the diffusion of an innovation are jointly modelled, dependent on one another, using an extension of a stochastic actor oriented model developed by Snijders (2001), which is modified so that the adoption times follow a proportional hazards model. The asymptotic behaviour of the method of moments estimator is examined. The model is demonstrated on a dataset involving of the initiation of cannabis smoking amongst adolescents, and a simulation study is presented.

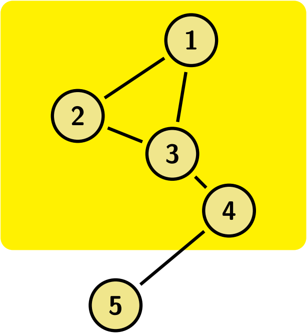
Keywords: Diffusion of innovations; Longitudinal analysis of network data; Method of moments; Proportional hazards; Stochastic actor oriented models.

Luckily: **Introduction**

Greenan, (2015). Diffusion of Innovations in Dynamic Networks. JRSS A

The settings model for thousands of nodes

Primary setting of i: everyone at distance less than or equal to 2

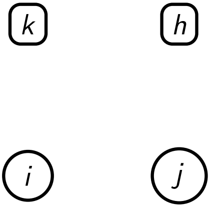


EGO: $\boldsymbol{i = 1}$

Ego ONLY evaluates primary setting
this changes dynamically

Two-mode networks

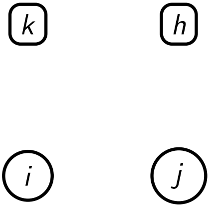
Set of agents $V = \{1, \ldots, n\}$ and,
targets $M = \{1, \ldots, m\}$.



Actor oriented:

Two-mode networks

Set of agents $V = \{1, \ldots, n\}$ and,
targets $M = \{1, \ldots, m\}$.

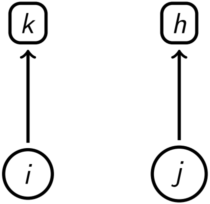


Actor oriented:

- rates only λ_i for one type of node $i \in V$

Two-mode networks

Set of agents $V = \{1, \dots, n\}$ and,
targets $M = \{1, \dots, m\}$.



Actor oriented:

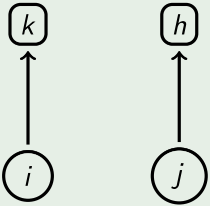
- rates only λ_i for one type of node $i \in V$
- $k \in M$ cannot create ties (hence no ties in $M \times M$)

Two-mode networks

Example (Koskinen and Edling, 2012)

Set of corporate boards $V = \{1, \ldots, n\}$ and,
Set of directors $M = \{1, \ldots, m\}$.

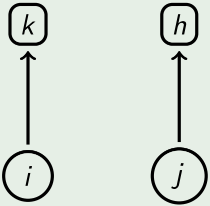
Actor oriented:



Two-mode networks

Example (Koskinen and Edling, 2012)

Set of corporate boards $V = \{1, \ldots, n\}$ and,
Set of directors $M = \{1, \ldots, m\}$.



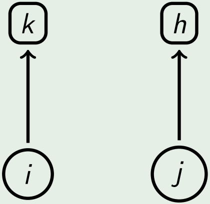
Actor oriented:

- What set of directors relevant?

Two-mode networks

Example (Koskinen and Edling, 2012)

Set of corporate boards $V = \{1, \ldots, n\}$ and,
Set of directors $M = \{1, \ldots, m\}$.



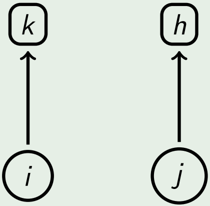
Actor oriented:

- What set of directors relevant?
- Board sizes $x_{i+} = \sum_j x_{ij}$ fixed

Two-mode networks

Example (Koskinen and Edling, 2012)

Set of corporate boards $V = \{1, \ldots, n\}$ and,
Set of directors $M = \{1, \ldots, m\}$.



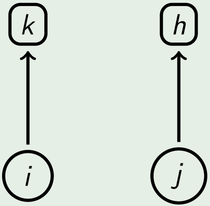
Actor oriented:

- What set of directors relevant?
- Board sizes $x_{i+} = \sum_j x_{ij}$ fixed
- Change: once a year at AGM

Two-mode networks

Example (Koskinen and Edling, 2012)

Set of corporate boards $V = \{1, \ldots, n\}$ and,
Set of directors $M = \{1, \ldots, m\}$.



Actor oriented:

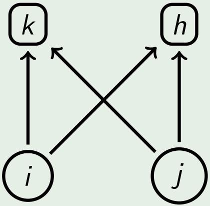
- What set of directors relevant?
- Board sizes $x_{i+} = \sum_j x_{ij}$ fixed
- Change: once a year at AGM

No triangles (SIC) $\Rightarrow$ 4-cycle is bipartite clustering:

Two-mode networks

Example (Koskinen and Edling, 2012)

Set of corporate boards $V = \{1, \ldots, n\}$ and,
Set of directors $M = \{1, \ldots, m\}$.



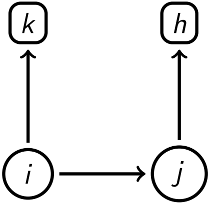
Actor oriented:

- What set of directors relevant?
- Board sizes $x_{i+} = \sum_j x_{ij}$ fixed
- Change: once a year at AGM

No triangles (SIC) $\Rightarrow$ 4-cycle is bipartite clustering: *peer-referal*

Analysing one-mode + two-mode networks

Set of people $V = \{1, \ldots, n\}$ and,
corporate boards/concepts/activities $M = \{1, \ldots, m\}$.



As actor oriented (rates only λ_i for one type of node $i \in V$)
$k \in M$ cannot create ties (hence no ties in $M \times M$)

Data structure - two distinct node-sets

Define two different node sets

```
people <- sienaNodeSet(nrppl, nodeSetName="people")
affiliations <- sienaNodeSet(nraffiliations,
nodeSetName="affiliations")
```

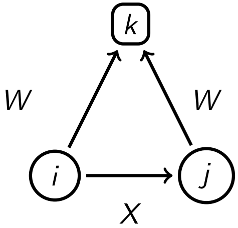
dependent variables

```
friendship <- sienaDependent(friendshipData)
aff <- sienaDependent(array(c(affiliations1, affiliations2),
dim=c(nrppl, nraffiliations,2)),
"bipartite", nodeSet=c("people", "affiliations"))
```

and the suite of effects is given by `getEffects`

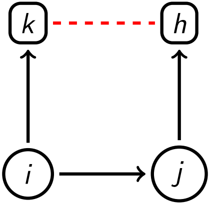
What type of effects?

As for multiplex networks,
some effects with multiple types of ties defined (others not) (*from W agreement*)



But as M not actors no dependent attribute on top-level

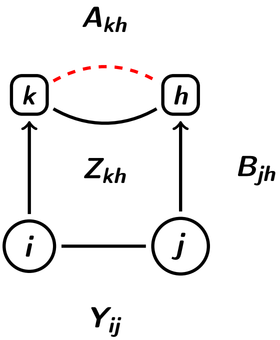
Set of people $V = \{1, \ldots, n\}$ and,
what if $M = \{1, \ldots, m\}$ have ties in $M \times M$?



. . . or have dependent attributes?

Define relevant networks

Example of people-people ties (Y_{ij}), people-concepts (B_{ik}), and concept-concept (Z_{kh}), with dyadic covariate (A_{kh})



The James Hollway trick

https://www.stats.ox.ac.uk/~snijders/siena/TwoModeAsSymmetricOneMode_Siena.R

Define three blocked matrices

$$\underset{(m+n)\times(m+n)}{\mathbf{D}} = \begin{pmatrix} \mathbf{Z} & \mathbf{0}_{m\times n} \\ \mathbf{0}_{n\times m} & \mathbf{0}_{n\times n} \end{pmatrix}$$

$$\underset{(m+n)\times(m+n)}{\mathbf{U}} = \begin{pmatrix} \mathbf{0}_{m\times m} & \mathbf{0}_{m\times n} \\ \mathbf{0}_{n\times m} & \mathbf{Y} \end{pmatrix}$$

$$\underset{(m+n)\times(m+n)}{\mathbf{V}} = \begin{pmatrix} \mathbf{0}_{m\times m} & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{0}_{n\times n} \end{pmatrix}$$

where $\mathbf{0}$ are blocks of structural zeros.

The James Hollway trick

The dyadic covariate for the concept-concept ties can be defined similarly

$$\underset{(|\mathcal{N}|+n)\times(|\mathcal{N}|+n)}{\mathbf{C}} = \begin{pmatrix} \mathbf{A} & \mathbf{0}_{|\mathcal{N}|\times n} \\ \mathbf{0}_{n\times|\mathcal{N}|} & \mathbf{0}_{n\times n} \end{pmatrix}$$

and the same for other dyadic covariates

Blocked SAOM for Basov's sociosemantic network

Reading in the networks

```
MatC <- as.matrix( read.table("blocked_c.txt") )
MatD <- as.matrix( read.table("blocked_d.txt") )
MatV <- as.matrix( read.table("blocked_v.txt") )
MatU <- as.matrix( read.table("blocked_u.txt") )
N <- dim(MatC)[1]
```

Defining the dependent variables and expert network covariate

```
LocalNet <- sienaDependent(
            array( c( MatD,MatD ),
            dim = c(N,N , 2 ) ) ,
            allowOnly=FALSE  )
SocialNet <- sienaDependent(
            array( c( MatU,MatU ),
            dim = c(N,N , 2 ) ) ,
            allowOnly=FALSE  )
UsageNet <- sienaDependent(
            array( c( MatV,MatV ),
            dim = c(N,N , 2 ) ) ,
            allowOnly=FALSE  )
ExpertNet <- coDyadCovar(MatC, centered=FALSE)
```

Defining the siena data structure

```
Blockdata <- sienaDataCreate( LocalNet,
                    SocialNet ,
                    UsageNet ,
                    ExpertNet )
```

Parameter	Point estimate	S.E.	Convergence
rate basic rate parameter LocalNet	0.5		
LocalNet: degree (density)	1.046	17.455	0.0234
LocalNet: transitive triads	0.096	0.0763	0.0578
LocalNet: ExpertNet	0.923	0.2183	0.0204
LocalNet: WW on X closure of ExpertNet	−0.003	0.0116	0.0284
LocalNet: shared incoming UsageNet	0.2	0.0274	0.0131
rate basic rate parameter SocialNet	50		
SocialNet: degree (density)	−1.259	0.265	0.125
SocialNet: transitive triads	0.475	0.2401	0.1154
basic rate parameter UsageNet	50		
UsageNet: outdegree (density)	−0.175	0.0423	0.0014
UsageNet: degree sqrt LocalNet pop.	0.563	0.0615	0.0429

Any effect for a dyadic covariate W on network X
(e.g. *from W agreement*)

can be defined in terms of network $X^{(s)}$ on network $X^{(u)}$, for $s, u \in \mathcal{R} = \{1, \ldots, R\}$.

Two dependent networks - no problem

Let $r = 1$: friendship
and $r = 2$: romantic

```
friendship <- sienaDependent(friendshipData)
```

```
romantic <- sienaDependent(romanticData)
```

from W agreement

Let $r = 1$: friendship

and $r = 2$: romantic

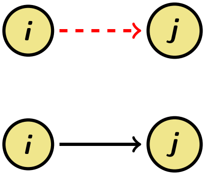
Closure of friendship by romantic:

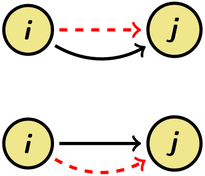
```
myeff <- includeEffects(myeff, name = 'romantic' , from,
  interaction1 = 'friendship')
```

Closure of romantic by friendship:

```
myeff <- includeEffects(myeff, name = 'friendship' ,from,
     interaction1 =  'romantic' )
```

Note that, for alignment, to investigate:

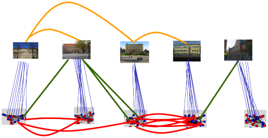




You need to use likelihood-based inference

Multilevel Analysis of Networks

Multilevel Network Analysis

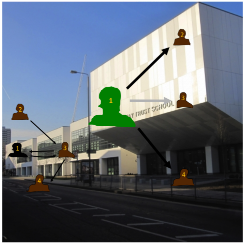


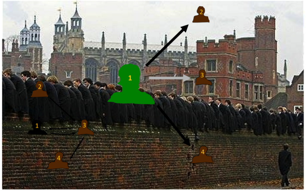
What if we have the same type of network observed in multiple, independent contexts?

How do dynamics play out in different contexts

should we have different models $p_{ij}^{[g]}(\beta^{[g]}, x^{[g]})$ for different groups g ?

How do dynamics play out in different contexts





should we have different models $p_{ij}^{[g]}(\beta^{[g]}, x^{[g]})$
for different groups g ?

Independently for groups $g = 1, 2, \ldots, G$ we have observations



$$x_{t_1}^{[1]}, \ldots, x_{t_M}^{[1]}$$
$$x_{t_1}^{[2]}, \ldots, x_{t_M}^{[2]}$$
$$x_{t_1}^{[2]}, \ldots, x_{t_M}^{[2]}$$
$$\vdots$$

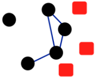



$$x_{t_1}^{[G-1]}, \ldots, x_{t_M}^{[G-1]}$$
$$x_{t_1}^{[G]}, \ldots, x_{t_M}^{[G]}$$


Multilevel Data structure

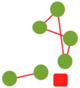
Independently for egos $g = 1, 2, \ldots, G$ we have observations

$$x_{t_1}^{[1]}, \ldots, x_{t_M}^{[1]}$$



$$x_{t_1}^{[2]}, \ldots, x_{t_M}^{[2]}$$

$$\vdots$$

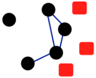


$$x_{t_1}^{[G]}, \ldots, x_{t_M}^{[G]}$$

Data structure

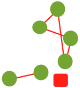
Independently for egos $g = 1, 2, \dots, G$ we have observations

$$x_{t_1}^{[1]}, \dots, x_{t_M}^{[1]}$$



$$x_{t_1}^{[2]}, \dots, x_{t_M}^{[2]}$$

$$\vdots$$

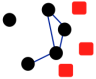


$$x_{t_1}^{[G]}, \dots, x_{t_M}^{[G]}$$

Approach one: separate analysis

Independently for egos $g = 1, 2, \ldots, G$ we have observations

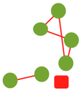
$x_{t_1}^{[1]}, \ldots, x_{t_M}^{[1]}$



$x_{t_1}^{[2]}, \ldots, x_{t_M}^{[2]}$

$\vdots$

$\vdots$



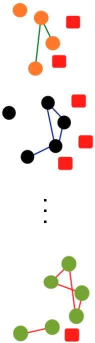
$x_{t_1}^{[G]}, \ldots, x_{t_M}^{[G]}$

Fit separate (unrestricted) models

- G unique explanations
- confounds systematic processes and context
- tedious
- egonets with little info.

Approach two: multigroup

Independently for egos $g = 1, 2, \ldots, G$ we have observations



$x_{t_1}^{[1]}, \ldots, x_{t_M}^{[1]}$

$x_{t_1}^{[2]}, \ldots, x_{t_M}^{[2]}$

$\vdots$

$x_{t_1}^{[G]}, \ldots, x_{t_M}^{[G]}$

Fit identical models assuming $\theta^{[g]} = \theta$

- strong assumption
- does not account for different contexts
- dominated by ‘large’ egos

Could use structural zeros but more efficient

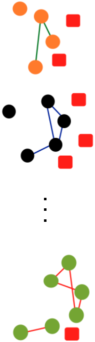
```
Group1 <- sienaDependent(array(c(N3401, HN3401),
  dim=c(45, 45, 2)))
Group3 <- sienaDependent(array(c(N3403, HN3403),
     dim=c(37, 37, 2)))
Group4 <- sienaDependent(array(c(N3404, HN3404),
     dim=c(33, 33, 2)))
Group6 <- sienaDependent(array(c(N3406, HN3406),
   dim=c(36, 36, 2)))
  dataset.1 <- sienaDataCreate(Friends = Group1)
  dataset.3 <- sienaDataCreate(Friends = Group3)
  dataset.4 <- sienaDataCreate(Friends = Group4)
  dataset.6 <- sienaDataCreate(Friends = Group6)
```

Then model can be defined as usula

```
FourGroups <- sienaGroupCreate(
       list(dataset.1, dataset.3, dataset.4, dataset.6))
 FourEffects <- getEffects(FourGroups)
 FourEffects <- includeEffects(FourEffects, transTrip)
```

Approach three: meta-analysis

Independently for egos $g = 1, 2, \ldots, G$ we have observations



$x^{[1]}_{t_1}, \ldots, x^{[1]}_{t_M}$

$x^{[2]}_{t_1}, \ldots, x^{[2]}_{t_M}$

$\vdots$

$x^{[G]}_{t_1}, \ldots, x^{[G]}_{t_M}$

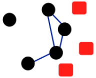
Fit separate identical models but parameters differ. POOL estimates

- ✓ assumes $(x^{[g]}_{t_m})$ from distribution of networks
- ✓ estimate the effects $\bar{\theta}$ "net of context"
- ✓ test differences across g
- ◎ $\hat{\theta}^{[g]}$ not estimable for small g
- ◎ pools $\hat{\theta}^{[g]}_r$ independently (for effects $r = 1, \ldots, p$)
- ◎ no ego covariates

Proposed approach: Hierarchical SAOM

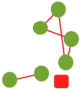
Independently for egos $g = 1, 2, \ldots, G$ we have observations

$x_{t_1}^{[1]}, \ldots, x_{t_M}^{[1]}$



$x_{t_1}^{[2]}, \ldots, x_{t_M}^{[2]}$

$\vdots$



$x_{t_1}^{[G]}, \ldots, x_{t_M}^{[G]}$

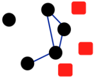
Assume a parametric model $\theta^{[g]} \sim N(\mu, \Sigma)$, and conditionally on this identical models

- ✓ estimate μ and Σ rather than $\theta^{[g]}$
- ✓ assumes $(x_{t_m}^{[g]})$ from distribution of network models

Proposed approach: Hierarchical SAOM

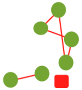
Independently for egos $g = 1, 2, \dots, G$ we have observations

$x_{t_1}^{[1]}, \dots, x_{t_M}^{[1]}$



$x_{t_1}^{[2]}, \dots, x_{t_M}^{[2]}$

$\vdots$



$x_{t_1}^{[G]}, \dots, x_{t_M}^{[G]}$

Assume a parametric model $\theta^{[g]} \sim N(\mu, \Sigma)$, and conditionally on this identical models

- ✓ estimate μ and Σ rather than $\theta^{[g]}$
- ✓ assumes $(x_{t_m}^{[g]})$ from distribution of network models
- ✓ $\hat{\theta}^{[g]}$ **does not have to** be estimable for all g
- ✓ pools $\hat{\theta}_r^{[g]}$ **consistently** (for effects $r = 1, \dots, p$)
- ✓ includes ego **covariates**

Use the multigroup option

```
FourEffects <- includeEffects(FourEffects, transTrip)
FourEffects <- setEffect(FourEffects, density, random=TRUE)
FourEffects <- setEffect(FourEffects, recip, random=TRUE)
print(FourEffects, includeRandoms=TRUE)
```

Bu

now you can chose to set some effecfs to vary across groups

Hierarchical SAOM: model and prior

Assuming the conjugate prior,

- $\Sigma^{-1} \sim \mathrm{wishart}_p(\Lambda_0^{-1}, \nu_0)$, and conditionally on Σ
- $\mu \mid \Sigma \sim N_p(\mu_0, \Sigma/\kappa_0)$.

The joint p.d.f., for data $x^{[1]}, \ldots, x^{[G]}$, is

Hierarchical SAOM: model and prior

Assuming the conjugate prior,

- $\Sigma^{-1} \sim \mathrm{wishart}_p(\Lambda_0^{-1}, \nu_0)$, and conditionally on Σ
- $\mu \mid \Sigma \sim N_p(\mu_0, \Sigma/\kappa_0)$.

The joint p.d.f., for data $x^{[1]}, \ldots, x^{[G]}$, is

$$f_{\mathrm{InvWish}}\left(\Sigma \mid \Lambda_0^{-1}, \nu_0\right) \phi_p\left(\mu \mid \mu_0, \Sigma/\kappa_0\right) \quad \text{prior}$$

Hierarchical SAOM: model and prior

Assuming the conjugate prior,

- $\Sigma^{-1} \sim \mathrm{wishart}_p(\Lambda_0^{-1}, \nu_0)$, and conditionally on Σ
- $\mu \mid \Sigma \sim N_p(\mu_0, \Sigma/\kappa_0)$.

The joint p.d.f., for data $x^{[1]}, \ldots, x^{[G]}$, is

$$f_{\mathrm{InvWish}}\left(\Sigma \mid \Lambda_0^{-1}, \nu_0\right) \phi_p\left(\mu \mid \mu_0, \Sigma/\kappa_0\right) \quad \text{prior}$$

$$\times \prod_{g=1}^{G} \phi_p(\theta^{[g]} \mid \mu, \Sigma) \quad \text{hierarchical model}$$

Hierarchical SAOM: model and prior

Assuming the conjugate prior,

- $\Sigma^{-1} \sim \mathrm{wishart}_p(\Lambda_0^{-1}, \nu_0)$, and conditionally on Σ
- $\mu \mid \Sigma \sim N_p(\mu_0, \Sigma/\kappa_0)$.

The joint p.d.f., for data $x^{[1]}, \dots, x^{[G]}$, is

$f_{\mathrm{InvWish}}\left(\Sigma \mid \Lambda_0^{-1}, \nu_0\right) \phi_p\left(\mu \mid \mu_0, \Sigma/\kappa_0\right)$	prior
$\times \prod_{g=1}^{G} \phi_p(\theta^{[g]} \mid \mu, \Sigma)$	hierarchical model
$\times \prod_{g=1}^{G} p_{\mathrm{SAOM}}(x^{[g]} \mid \theta^{[g]})$	network model

EXAMPLE OF MULTIPLE GROUPS

traditional nested data-structure

Example: data Andrea Knecht

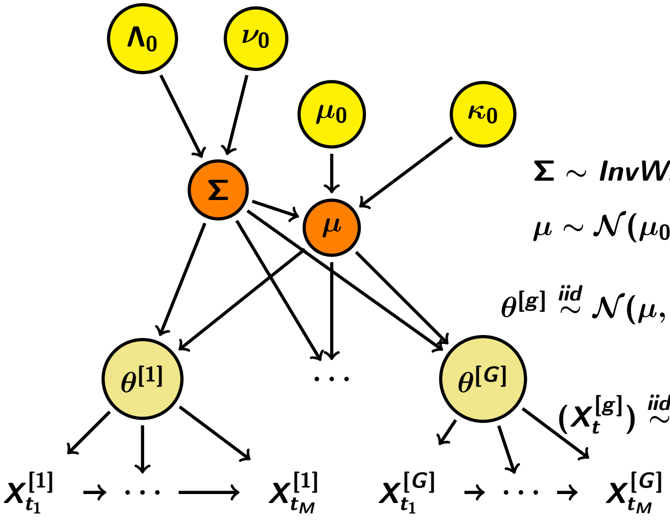
$G = 21$ school classes (Andrea Knecht, PhD thesis Utrecht, 2008; see Knecht, Snijders, Baerveldt, Steglich, & Raub, 2010)
We consider a model for a longitudinal study with 2 waves,
and with 9 parameters:
rate of change; outdegree; reciprocity; transitive triplets; 3-cycles;
delinquency ego, alter, ego $\times$ alter; sex similarity.

Use the multigroup option

```
groupModel.e <- sienaBayes(GroupsAlgo, data = TwentyOne_Groups,
initgainGlobal=0.1, initgainGroupwise = 0.001,
                effects = FourEffects, priorMu = Mu, priorSigma
priorKappa = 0.01,
prevAns = ans,
nwarm=200, nmain=1000, nrunMHBatches=40,
                nbrNodes=7, silentstart=FALSE)
```

now you can chose to set some effecfs to vary across groups

HSAOM (DAG)



$$\Sigma \sim InvWish_{p_1}(\Lambda_0^{-1}, \nu_0)$$

$$\mu \sim \mathcal{N}(\mu_0, \Sigma/\kappa_0)$$

$$\theta^{[g]} \overset{iid}{\sim} \mathcal{N}(\mu, \Sigma)$$

$$(X_t^{[g]}) \overset{iid}{\sim} SAOM(\theta^{[g]})$$

Hierarchical SAOM: estimation

We use a Bayesian MCMC inference scheme:
Draw posterior variates

$$(v^{[1]}, \dots, v^{[G]}, \theta^{[1]}, \dots, \theta^{[G]}, \mu, \Sigma)$$

by iteratively drawing from the full conditional posteriors

Hierarchical SAOM: estimation

We use a Bayesian MCMC inference scheme:
Draw posterior variates

$$(v^{[1]}, \dots, v^{[G]}, \theta^{[1]}, \dots, \theta^{[G]}, \mu, \Sigma)$$

by iteratively drawing from the full conditional posteriors

- $(v^{[1]}, \dots, v^{[G]}) \sim [v^{[1]}, \dots, v^{[G]} | \theta^{[1]}, \dots, \theta^{[G]}, x^{[1]}, \dots, x^{[G]}]$ (unobserved sample paths)

Hierarchical SAOM: estimation

We use a Bayesian MCMC inference scheme:
Draw posterior variates

$$(v^{[1]},\dots,v^{[G]},\theta^{[1]},\dots,\theta^{[G]},\mu,\Sigma)$$

by iteratively drawing from the full conditional posteriors

- $(v^{[1]},\dots,v^{[G]}) \sim [v^{[1]},\dots,v^{[G]}|\theta^{[1]},\dots,\theta^{[G]},x^{[1]},\dots,x^{[G]}]$ (unobserved sample paths)
- $(\theta^{[1]},\dots,\theta^{[G]}) \sim [\theta^{[1]},\dots,\theta^{[G]}|v^{[1]},\dots,v^{[G]},x^{[1]},\dots,x^{[G]},\mu,\Sigma]$ (group-level parameters)

Hierarchical SAOM: estimation

We use a Bayesian MCMC inference scheme:
Draw posterior variates

$$(v^{[1]}, \dots, v^{[G]}, \theta^{[1]}, \dots, \theta^{[G]}, \mu, \Sigma)$$

by iteratively drawing from the full conditional posteriors

- $(v^{[1]}, \dots, v^{[G]}) \sim [v^{[1]}, \dots, v^{[G]} | \theta^{[1]}, \dots, \theta^{[G]}, x^{[1]}, \dots, x^{[G]}]$ (unobserved sample paths)
- $(\theta^{[1]}, \dots, \theta^{[G]}) \sim [\theta^{[1]}, \dots, \theta^{[G]} | v^{[1]}, \dots, v^{[G]}, x^{[1]}, \dots, x^{[G]}, \mu, \Sigma]$ (group-level parameters)
- $(\mu, \Sigma) \sim [\mu, \Sigma | \theta^{[1]}, \dots, \theta^{[G]}]$ (global parameters)

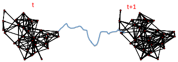
Hierarchical SAOM: estimation

We use a Bayesian MCMC inference scheme:
Draw posterior variates

$$(v^{[1]}, \ldots, v^{[G]}, \theta^{[1]}, \ldots, \theta^{[G]}, \mu, \Sigma)$$

by iteratively drawing from the full conditional posteriors

- $(v^{[1]}, \ldots, v^{[G]}) \sim [v^{[1]}, \ldots, v^{[G]} | \theta^{[1]}, \ldots, \theta^{[G]}, x^{[1]}, \ldots, x^{[G]}]$



  (unobserved sample paths)
- $(\theta^{[1]}, \ldots, \theta^{[G]}) \sim [\theta^{[1]}, \ldots, \theta^{[G]} | v^{[1]}, \ldots, v^{[G]}, x^{[1]}, \ldots, x^{[G]}, \mu, \Sigma]$ (group-level parameters)
- $(\mu, \Sigma) \sim [\mu, \Sigma | \theta^{[1]}, \ldots, \theta^{[G]}]$ (global parameters)

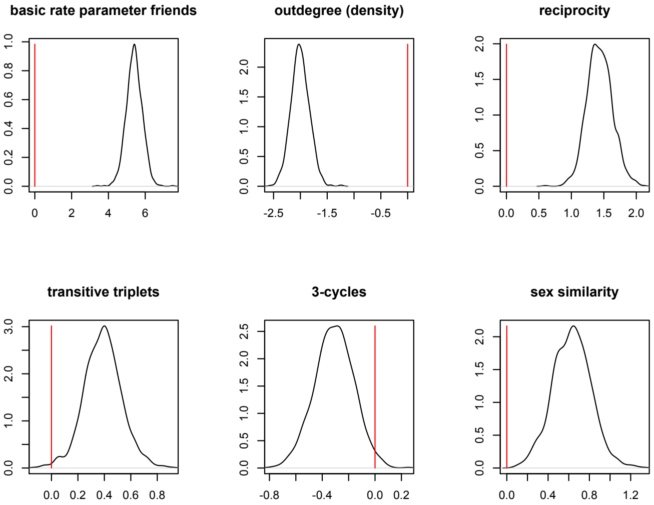


Figure: Posterior distributions μ

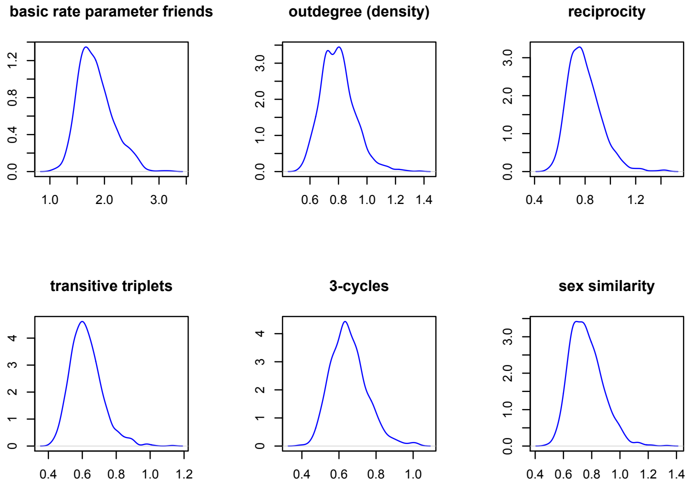


Figure: Posterior distributions SDs, square roots of diag elements Σ

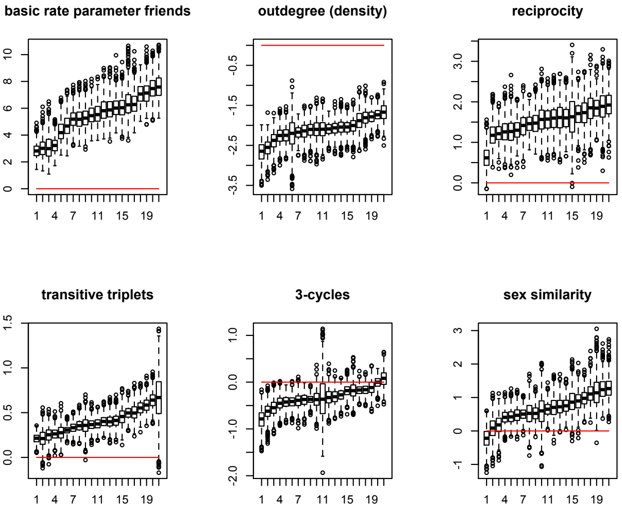


Figure: Posterior predictive distributions for $\theta^{[g]}$ (ordered)

Bayesian analysis of SAOM - final remarks

- estimation in `sienaBayes` that is in `RSienaTest` (download from RForge)
- implementation is robust but slow (should be improved in future implementations)
- A paper (started in 2010) is just published; preprint available at http://arxiv.org/abs/2201.12713