Network Dependencies in Social Space, Geographical Space, and Temporal Space. Part I [1]

Johan Koskinen

Social Networks Lab
Melbourne School of Psychological Sciences
University of Melbourne

NetGlow June 2022

[1] More material at http://www.stats.ox.ac.uk/siena/; Material from Snijders greatly acknowledged

The two basic types of data

NETWORK

nodes: Andras, Per, Zsofia

have **ties:** Andras $\rightarrow$ Per

BEHAVIOUR

attributes of nodes: Andras, Per, Zsofia drink

Zsofia does not smoke

The two basic types of data

NETWORK

nodes: Andras, Per, Zsofia

have **ties:** Andras $\rightarrow$ Per

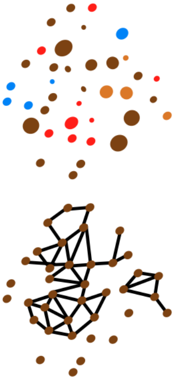
BEHAVIOUR

attributes of nodes: Andras, Per, Zsofia drink

Zsofia does not smoke

SAOM: longitudinal modelling

We have **observations** on NETWORK and BEHAVIOUR



At some fixed points in time

starting at t_0

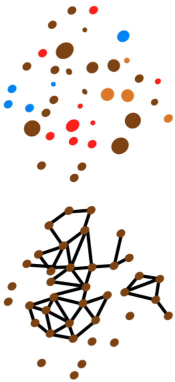
followed by t_1

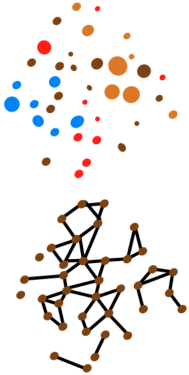
$t_0 < t_1$

inferential task: explain how t_0 change into t_1

SAOM: longitudinal modelling

We have **observations** on NETWORK and BEHAVIOUR





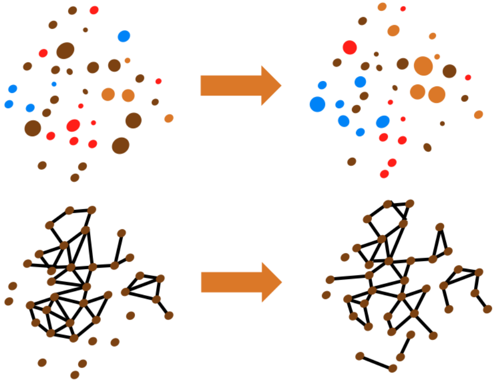
At some fixed points in time

starting at t_0
followed by t_1
$t_0 < t_1$

inferential task: explain how t_0 change into t_1

SAOM: longitudinal modelling

We have **observations** on NETWORK and BEHAVIOUR



At some fixed points in time

starting at t_0

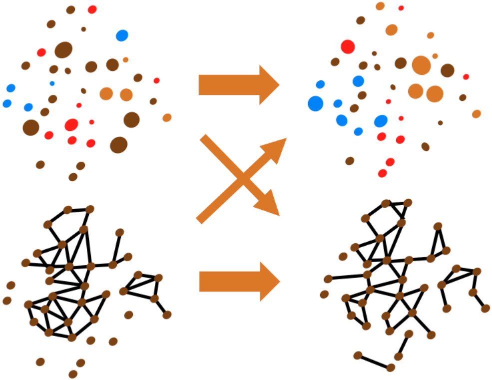
followed by t_1

$t_0 < t_1$

inferential task: explain how t_0 change into t_1

SAOM: longitudinal modelling

We have **observations** on NETWORK and BEHAVIOUR



Especially the **co-evolution**:

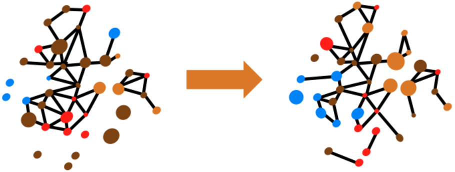
selection

influence

inferential task: explain how t_0 change into t_1

SAOM: longitudinal modelling

We have **observations** on NETWORK and BEHAVIOUR



Especially the **co-evolution**:

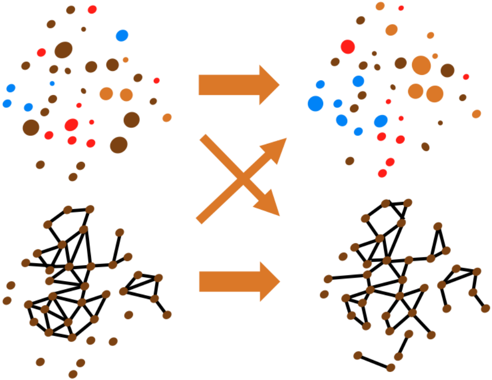
selection

influence

inferential task: explain how t_0 change into t_1

SAOM: longitudinal modelling

We have **observations** on NETWORK and BEHAVIOUR



Especially the **co-evolution**:

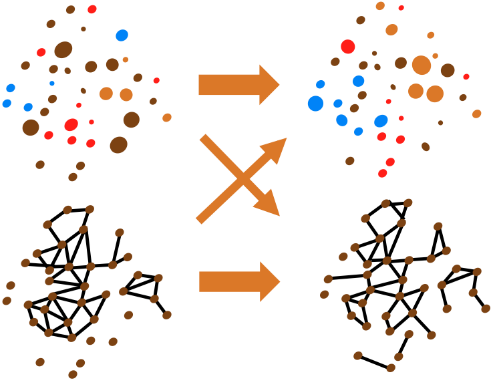
selection

influence

inferential task: explain how t_0 change into t_1

The SAO Model

We have **observations** on NETWORK and BEHAVIOUR



Especially the
co-evolution:

selection

influence

HOW?

inferential task: explain how t_0 change into t_1

The SAO Model

Assume **PARTIAL observations** on a process

observations:

at t_0

and t_1

the rest:

missing

the process explains how t_0 change into t_1

The SAO Model

Assume **PARTIAL observations** on a process

observations:

at t_0

and t_1

the rest:

missing

the process explains how t_0 change into t_1

What type of data do we want to explain: adjacency matrix

Data represented as adjacency matrices

$$\mathbf{x} = \begin{pmatrix} . & 0 & 0 & 0 & 1 \\ 1 & . & 0 & 0 & 0 \\ 1 & 1 & . & 0 & 0 \\ 0 & 0 & 0 & . & 0 \\ 0 & 0 & 1 & 1 & . \end{pmatrix}$$

where $x_{ij} = 1$ or 0 according to wether $i \to j$ or not.

What type of data do we want to explain: longitudinal

Data represented as adjacency matrices
where elements change

$$x(t_0) = \begin{pmatrix} . & 0 & 0 & 0 & 1 \\ 1 & . & 0 & 0 & 0 \\ 1 & 1 & . & 0 & 0 \\ 0 & 0 & 0 & . & 0 \\ 0 & 0 & 1 & 1 & . \end{pmatrix}$$

What type of data do we want to explain

Data represented as adjacency matrices
where elements change

$$x(t_1) = \begin{pmatrix} . & 1 & 0 & 0 & 1 \\ 1 & . & 0 & 0 & 0 \\ 1 & 0 & . & 0 & 0 \\ 0 & 0 & 0 & . & 0 \\ 1 & 0 & 1 & 1 & . \end{pmatrix}$$

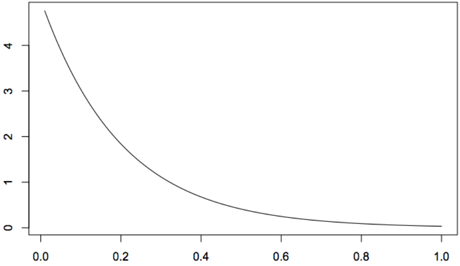
What type of data do we want to explain

Data represented as adjacency matrices
where elements change

$$x(t_2) = \begin{pmatrix} . & 1 & 0 & 1 & 1 \\ 1 & . & 0 & 0 & 1 \\ 1 & 1 & . & 0 & 0 \\ 0 & 0 & 0 & . & 0 \\ 1 & 0 & 0 & 1 & . \end{pmatrix}$$

SAOM: the rate of change

At random points in time, at rates λ_i



nodes/individuals/actors are given opportunities to change

SAOM: the *direction* of change

Conditional on an actor having an opportunity for change
the probability for each outcome

SAOM: the *direction* of change

Conditional on an actor having an opportunity for change the probability for each outcome

- is modelled like multinomial logistic regression

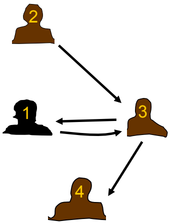
SAOM: the *direction* of change

Conditional on an actor having an opportunity for change
the probability for each outcome

- is modelled like multinomial logistic regression
- reflects the attractiveness of the outcome to the actor

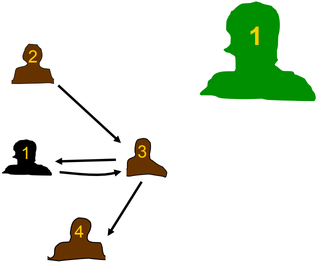
Example: i has oppotunity to change/toggle x_{ij} to $1 - x_{ij}$
We call the new network $X(i \rightsquigarrow j)$

$$x = \begin{array}{|c|c|c|c|} \hline - & 0 & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$



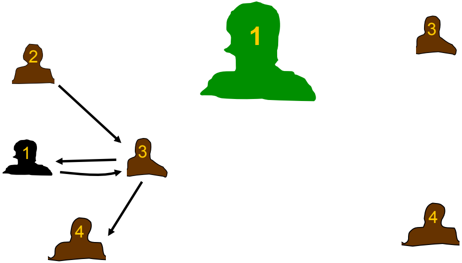
$x =$

-	0	1	0
0	-	1	0
1	0	-	1
0	0	0	-



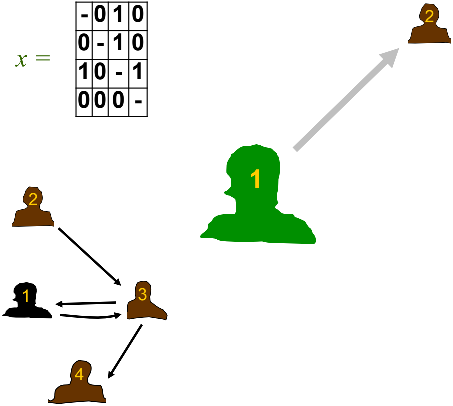
$x =$

-	0	1	0
0	-	1	0
1	0	-	1
0	0	0	-



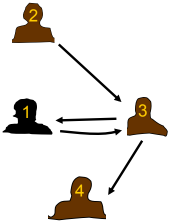
$x =$

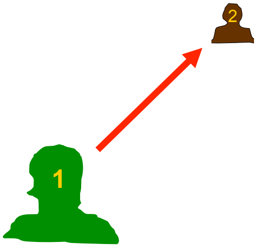
-	0	1	0
0	-	1	0
1	0	-	1
0	0	0	-

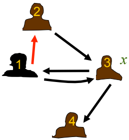


$x =$

-	0	1	0
0	-	1	0
1	0	-	1
0	0	0	-



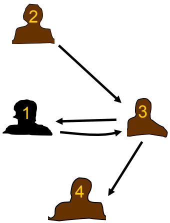


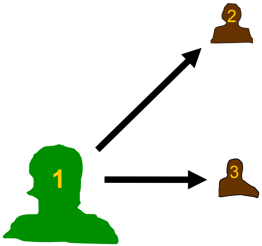


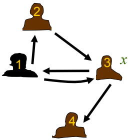
$x(1 \to 2) =$

-	1	1	0
0	-	1	0
1	0	-	1
0	0	0	-

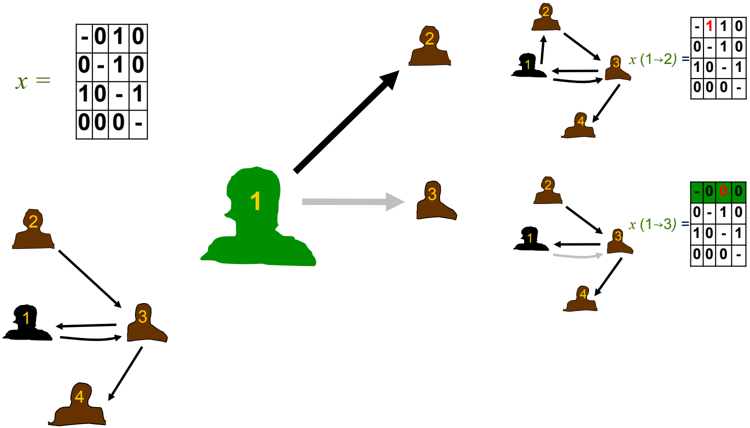
$$x = \begin{array}{|c|c|c|c|} \hline - & 0 & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$





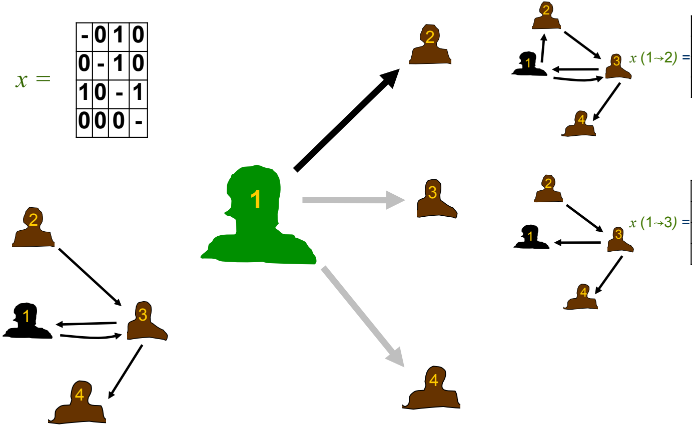


$$x(1 \to 2) = \begin{array}{|c|c|c|c|} \hline - & \color{red}{1} & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$


x =
- 0 1 0
0 - 1 0
1 0 - 1
0 0 0 -
1
2
3
4
x (1→2) =
- 1 1 0
0 - 1 0
1 0 - 1
0 0 0 -
x (1→3) =
- 0 0 0
0 - 1 0
1 0 - 1
0 0 0 -

$x =$

-	0	1	0
0	-	1	0
1	0	-	1
0	0	0	-



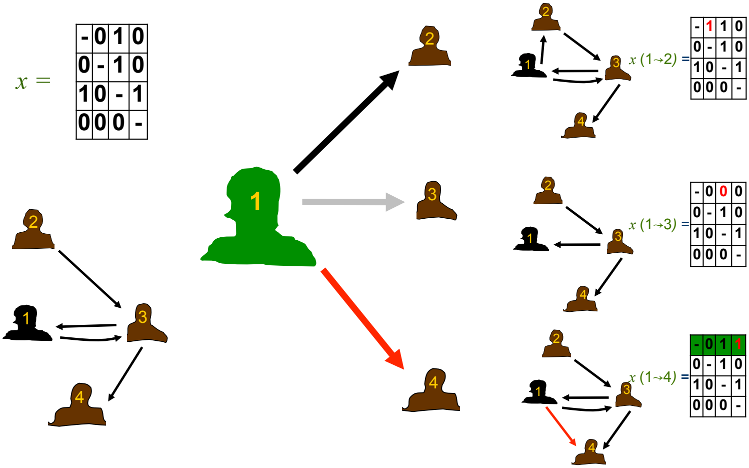
$x(1 \to 2) =$

-	1	1	0
0	-	1	0
1	0	-	1
0	0	0	-

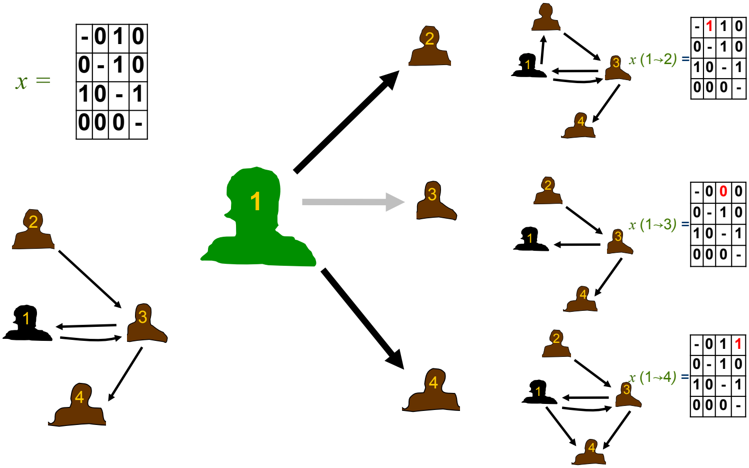
$x(1 \to 3) =$

-	0	0	0
0	-	1	0
1	0	-	1
0	0	0	-

$$x = \begin{array}{|c|c|c|c|} \hline - & 0 & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$



$$x = \begin{array}{|c|c|c|c|} \hline - & 0 & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$



Of the three changes (for $j = 2, 3, 4$) available to i (here 1)

Of the three changes (for $j = 2, 3, 4$) available to i (here 1) the probability that i toggles the tie $i \to j$ is given by

Of the three changes (for $j = 2, 3, 4$) available to i (here 1) the probability that i toggles the tie $i \to j$ is given by

One-step jump probability

$$p_{ij}(\beta, G) = \frac{\exp\left(f_i(\beta, G(i \rightsquigarrow j))\right)}{\sum_{h=1}^{n} \exp\left(f_i(\beta, G(i \rightsquigarrow h))\right)},$$

where

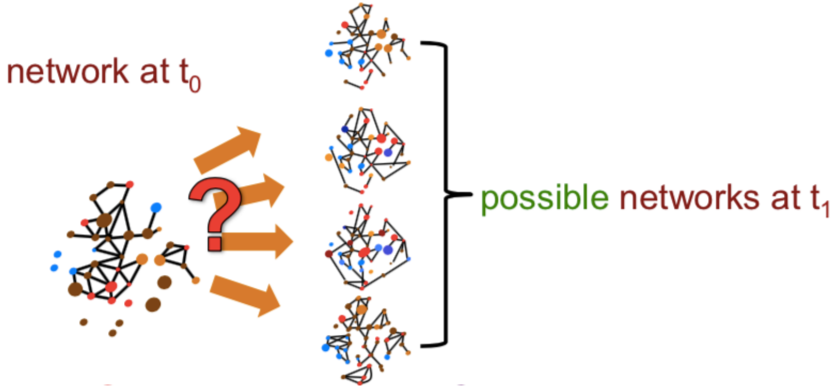
Of the three changes (for $j = 2, 3, 4$) available to i (here 1) the probability that i toggles the tie $i \to j$ is given by

One-step jump probability

$$p_{ij}(\beta, G) = \frac{\exp\left(f_i(\beta, G(i \rightsquigarrow j))\right)}{\sum_{h=1}^{n} \exp\left(f_i(\beta, G(i \rightsquigarrow h))\right)} ,$$

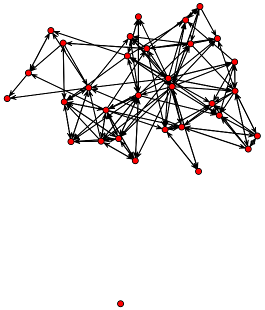
where

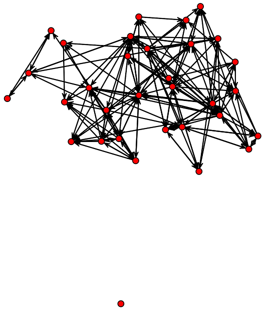
- $G(i \rightsquigarrow j)$ is the network resulting from the change
- β are statistical parameters
- f_i describes the attractiveness of $G(i \rightsquigarrow j)$ to i


network at t_0
?
possible networks at t_1

van de Bunt data set

```
library('RSiena')
library('network')
library('sna')
tmp4[is.na(tmp4)] <- 0 # remove missing
par(mfrow = c(1,2))
coordin <- plot(as.network(tmp3))
plot(as.network(tmp4),coord=coordin)
```





Let us assume that i ONLY cares about not having too many or two few ties:

$$f_i(\beta, X) = \exp\left\{ \beta \sum_j x_{ij} \right\}$$

meaning that

$$p_{ij}(\beta, X) = \frac{\exp\left\{\beta(1 - 2x_{ij})\right\}}{\sum_{h=1}^{n} \exp\left\{\beta(1 - 2x_{ih})\right\}} ,$$

because if currently $x_{ij} = 1$, then the number of ties for i in $G(i \rightsquigarrow j)$ will be one less (-1), and if currently $x_{ij} = 0$ then there will be one more $(+1)$

Simulation settings: actors only care about degree

Let the rate be equal for all $\lambda_i = \lambda = 3.8311$

- ✓ is each iteration, actor with shortest waiting time 'wins' (and gets to change)
- ✓ on average every actor gets 3.8 opportunities to change

and set $\beta = -1.1059$

- ✓ if $\beta = 0$ actor would not care if tie was added or deleted
- ✓ here $\beta < 0$ meaning that actor wants less than half of the possible ties

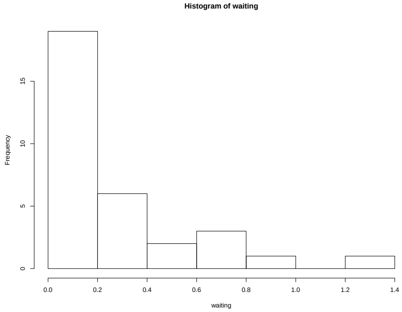
van de Bunt data set

```
mynet1 <- sienaDependent(array(c(tmp3, tmp4),
                              dim=c(32, 32,2)))
mydata <- sienaDataCreate(mynet1)
myeff <- getEffects(mydata)
myeff <- includeEffects(myeff, recip,include=FALSE)
myeff$initialValue[
             myeff$shortName == 'Rate'] <- 3.8311
myeff$initialValue[
        myeff$shortName=='density'][1] <- -1.1059
```

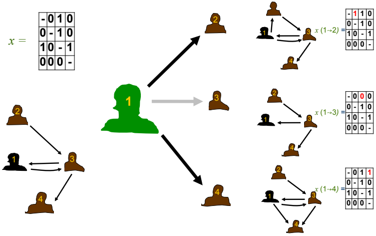
Model: rate

```
waiting <- rexp(32,3.8311)
hist(waiting)
which( waiting == min(waiting))
```

⟸ the winner

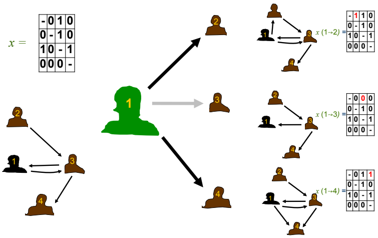


Model: conditional one-step change probability



$$\Pr(1 \rightsquigarrow 2) = \frac{e^{-1.1059}}{e^{-1.1059} + e^{1.1059} + e^{-1.1059} + 1} = 0.07$$

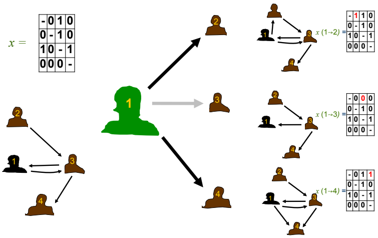
Model: conditional one-step change probability



$$\Pr(1 \rightsquigarrow 2) = \frac{e^{-1.1059}}{e^{-1.1059}+e^{1.1059}+e^{-1.1059}+1} = 0.07$$

$$\Pr(1 \rightsquigarrow 3) = \frac{e^{1.1059}}{e^{-1.1059}+e^{1.1059}+e^{-1.1059}+1} = 0.65$$

Model: conditional one-step change probability



$\Pr(1 \rightsquigarrow 2) =$

$\frac{e^{-1.1059}}{e^{-1.1059}+e^{1.1059}+e^{-1.1059}+1} =$

0.07

$\Pr(1 \rightsquigarrow 3) =$

$\frac{e^{1.1059}}{e^{-1.1059}+e^{1.1059}+e^{-1.1059}+1} =$

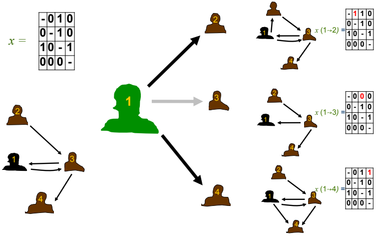
0.65

$\Pr(1 \rightsquigarrow 4) =$

$\frac{e^{-1.1059}}{e^{-1.1059}+e^{1.1059}+e^{-1.1059}+1} =$

0.07

Model: conditional one-step change probability



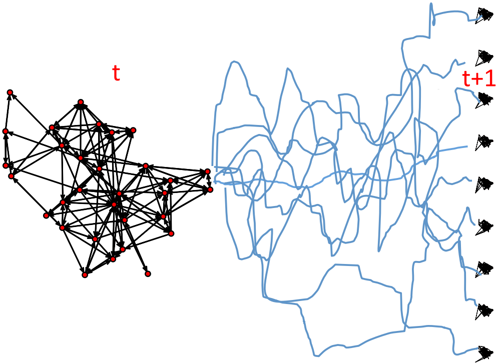
$\Pr(1 \rightsquigarrow 2) = \frac{e^{-1.1059}}{e^{-1.1059}+e^{1.1059}+e^{-1.1059}+1} =$
0.07

$\Pr(1 \rightsquigarrow 3) = \frac{e^{1.1059}}{e^{-1.1059}+e^{1.1059}+e^{-1.1059}+1} =$
0.65

$\Pr(1 \rightsquigarrow 4) = \frac{e^{-1.1059}}{e^{-1.1059}+e^{1.1059}+e^{-1.1059}+1} =$
0.07

Of course, in van de Bunt every actor has 31+1 choices for change

Now let us simulate



Simulate from t_0 to t_1 (`simOnly = TRUE`)

```
sim_model  <-  sienaAlgorithmCreate(
                         projname = 'sim_model',
                              cond = FALSE,
                              useStdInits = FALSE, nsub = 0 ,
                              simOnly = TRUE)
sim_ans <- siena07( sim_model, data = mydata,
                              effects = myeff,
                              returnDeps = TRUE, batch=TRUE )
```

The object `sim_ans` will now contain 1000 simulated networks

Extract networks

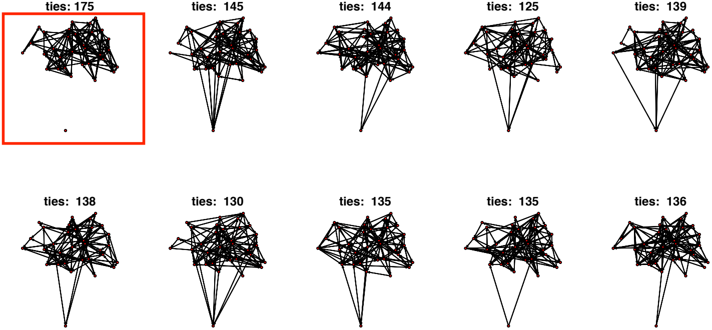
```
n <- dim(tmp4)[1]
mySimNets <- reshapeRSienaDeps( sim_ans , n )
```

The object `mySimNets` is a 1000 by n by n array of adjacency matrices

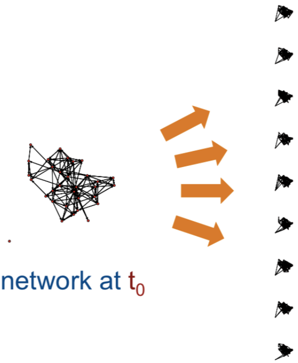
Plot observed network and 9 simulated

```
pdf(file='simnets1.pdf', width = 9,height =4.5)
par(mfrow=c(2,5),  oma = c(0,4,0,0) + 0.1,
    mar = c(5,0,1,1) + 0.1)
plot(as.network(tmp4),coord=coordin,
              main=paste('ties:',sum(tmp4) ) )
apply(mySimNets[1:9,,],1,function(x)
                  plot(as.network(x),
                       coord=coordin,
                       main=paste('ties: ',sum(x))) )
dev.off()
```

The observed at t_1 and possible netowrks at t_1



Simulated networks v t_1 obs



network at t_0

Dyad Census

```
> dyad.census(tmp4)
     Mut Asym Null
[1,]  46   83  367
> dyad.census(mySimNets[1:9,,])
      Mut Asym Null
 [1,]  27   91  378
 [2,]  20  104  372
 [3,]  16   93  387
 [4,]  19  101  376
 [5,]  21   96  379
 [6,]  16   98  382
 [7,]  11  113  372
 [8,]  20   95  381
 [9,]  15  106  375
```

Conclusions

A process where i ONLY cares about not having too many or two few ties does to replicate the reciprochity at t_1

Assume that i ALSO cares about

Conclusions

A process where i ONLY cares about not having too many or two few ties does to replicate the reciprochity at t_1

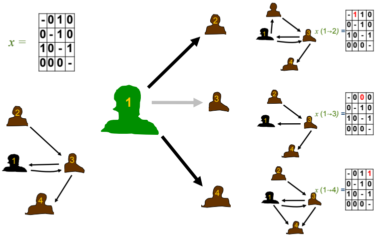
Assume that i ALSO cares about having ties $i \rightarrow j$ reciprocated $j \rightarrow i$

$$f_i(\beta, X) = \exp\left\{\beta_d \sum_j x_{ij} + \beta_r \sum_j x_{ij} x_{ji}\right\}$$

meaning that probability that i toggles relationship to j

$$p_{ij}(\beta, X) = \frac{\exp\left\{\beta_d(1-2x_{ij}) + \beta_r(1-2x_{ij})x_{ji}\right\}}{\sum_{h=1}^{n} \exp\left\{\beta_d(1-2x_{ih}) + \beta_r(1-2x_{ih})x_{hi}\right\}},$$

Model



Objective function:

$\beta_d \sum_j x_{ij} + \beta_r \sum_j x_{ij} x_{ji}$

- adding 1 → 2: β_d
- deleting 1 → 3: $\beta_d - \beta_r$
- adding 1 → 4: β_d

Our simulated networks had too few reciprocated dyads so we need to set $\beta_r \ldots$

Simulation settings: actors care aobut degree and reciprocity

Let the rate be equal for all $\lambda_i = \lambda = 4.2525$

- ✓ on average every actor gets 4.3 opportunities to change

and set $\beta_d = -1.4163$

- ✓ here $\beta_d < 0$ - actors do not want too many ties

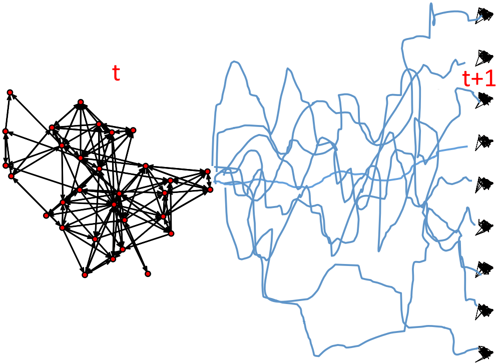
and set $\beta_r = 1.1383$

- ✓ here $\beta_r > 0$ - actors prefer reciprocated to assymetric ties

van de Bunt data set

```
myeff <- includeEffects(myeff, recip,include=TRUE)
myeff$initialValue[
             myeff$shortName == 'Rate'] <- 4.2525
myeff$initialValue[
        myeff$shortName =='density'][1] <- -1.4163
myeff$initialValue[
          myeff$shortName =='recip'][1] <- 1.1383
```

Now let us simulate



Simulate from t_0 to t_1 now with reciprocity (`simOnly = TRUE`)

```
sim_model  <-  sienaAlgorithmCreate(
                     projname = 'sim_model',
                          cond = FALSE,
                          useStdInits = FALSE, nsub = 0 ,
                          simOnly = TRUE)
sim_ans <- siena07( sim_model, data = mydata,
                          effects = myeff,
                          returnDeps = TRUE, batch=TRUE )
```

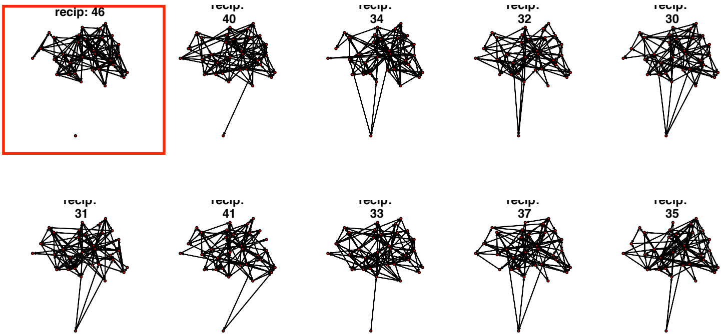
The object `sim_ans` will now contain 1000 simulated networks

NOTE: this piece of code is unchanged

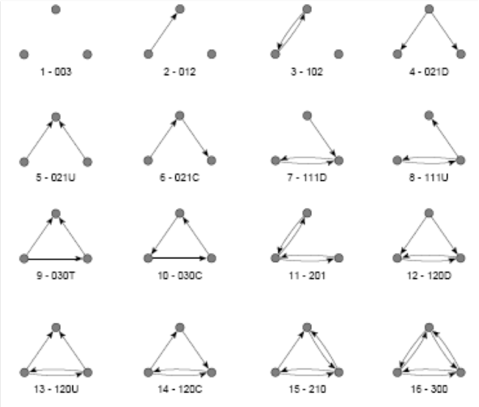
Plot observed network and 9 simulated

```
mySimNets <- reshapeRSienaDeps(sim_ans,n)
plot(as.network(tmp4),coord=coordin,
              main=paste('recip:',dyad.census(tmp4)[1] )  )
apply(mySimNets[1:9,,],1,function(x)
                  plot(as.network(x),
                       coord=coordin,
                       main=paste('recip:
   ',dyad.census(x)[1] ) ) )
```

The observed at t_1 and possible netowrks at t_1



Simulated networks v t_1 obs: triad census



Simulated networks v t_1 obs: triad census

```
> triad.census(tmp4)
      003  012 102 021D 021U 021C 111D 111U 030T 030C 201 120D 120U 120C 210 300
[1,] 2078 1329 745  146   80   52   65  217   37    0  68   16   65   10  30  22
> triad.census(mySimNets[1:9,,])
       003  012 102 021D 021U 021C 111D 111U 030T 030C 201 120D 120U 120C 210 300
 [1,] 2317 1296 681   78   51  111   97  174   15    2  74    5   17   11  26   5
 [2,] 2408 1280 607  130   44  123   88  158   23    3  39   13    8   11  18   7
 [3,] 2603 1257 604   73   36   84   69  135    8    1  42   11   11    8  16   2
 [4,] 2493 1356 575   92   53  105   69  124   16    2  36    6    8    9  11   5
 [5,] 2470 1342 533   98   66  106   75  154   16    3  41    4   18    9  20   5
 [6,] 2492 1119 762   81   40   83   97  163   12    2  58    8   11    6  11  15
 [7,] 2394 1358 574   86   55  133   89  144   19    3  57    7   10   11  19   1
 [8,] 2466 1265 671   76   37   76   83  163   15    3  55    6    7   12  17   8
 [9,] 2450 1235 650  100   61  116  105  128   11    2  40   10   10   18  23   1
```

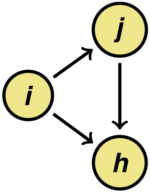
Reciprocity is clearly not enough to explain the incidence of *transitive triangles* and *simmelian ties* (3 Mutual 0, Assymetric, 0 Null)

Assume that *i* ALSO cares about *closure*

$$f_i(\beta, X) = \exp\left\{\beta_d \sum_j x_{ij} + \beta_r \sum_j x_{ij}x_{ji} + \beta_t s_{i,t}(x)\right\}$$

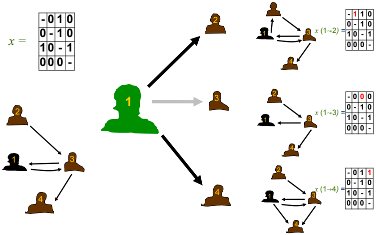
Modelled through, e.g.

transitive triplets effect,
number of transitive patterns in *i*'s ties
$(i \to j,\ j \to h,\ i \to h)$
$s_{i,t}(x) = \sum_{j,h} x_{ij}\, x_{jh}\, x_{ih}$



transitive triplet

Model



Objective function including $s_{i,t}(x)$:

- adding 1 $\rightarrow$ 2: $\ldots + \beta_t$
- deleting 1 $\rightarrow$ 3: no change in closure
- adding 1 $\rightarrow$ 4: $\ldots + \beta_t$

Our simulated networks had too few 030T and 300 so we need to set $\beta_t \ldots$

Assume actors care about degree, reciprocity, and closure

Let the rate be equal for all $\lambda_i = \lambda = 4.5017$

✓ on average every actor gets 4.5 opportunities to change

and set $\beta_d = -1.9024$

✓ here $\beta_d < 0$ - actors do not want too many ties

and set $\beta_r = 0.6794$

✓ here $\beta_r > 0$ - actors prefer reciprocated to assymetric ties

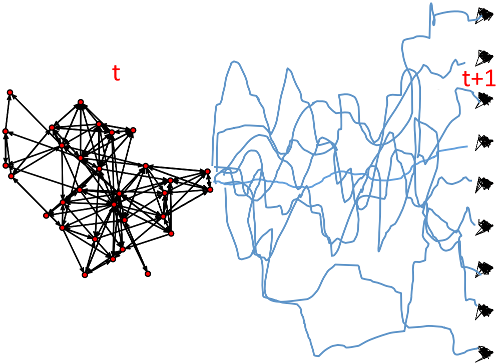
and set $\beta_t = 0.3183$

✓ here $\beta_r > 0$ - actors prefer ties that close open triads

van de Bunt data set

```
myeff <- includeEffects(myeff, recip,include=TRUE)
myeff <- includeEffects(myeff, transTrip,include=TRUE)
myeff$initialValue[
             myeff$shortName == 'Rate'] <- 4.5017
myeff$initialValue[
        myeff$shortName =='density'][1] <- -1.9024
myeff$initialValue[
          myeff$shortName =='recip'][1] <-  0.6794
myeff$initialValue[
          myeff$shortName =='transTrip'][1] <-  0.3183
```

Now let us simulate



Simulate from t_0 to t_1 now with transitivity (`simOnly = TRUE`)

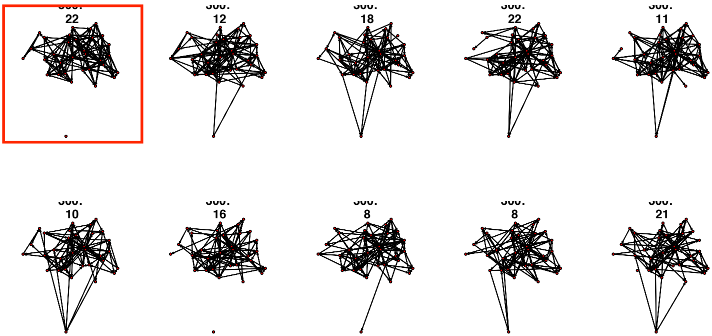
```
sim_model  <-  sienaAlgorithmCreate(
                         projname = 'sim_model',
                              cond = FALSE,
                              useStdInits = FALSE, nsub = 0 ,
                              simOnly = TRUE)
sim_ans <- siena07( sim_model, data = mydata,
                              effects = myeff,
                              returnDeps = TRUE, batch=TRUE )
```

NOTE: this piece of code is unchanged

Plot observed network and 9 simulated

```
mySimNets <- reshapeRSienaDeps(sim_ans,n)
plot(as.network(tmp4),coord=coordin,
              main=paste('recip:',triad.census(tmp4)[16] )  )
apply(mySimNets[1:9,,],1,function(x)
                  plot(as.network(x),
                       coord=coordin,
                       main=paste('300:
   ',triad.census(x)[16] ) ) )
```

The observed at t_1 and possible netowrks at t_1



Simulated networks v t_1 obs: triad census

```
> triad.census(tmp4)
      003  012 102 021D 021U 021C 111D 111U 030T 030C 201 120D 120U 120C 210 300
[1,] 2078 1329 745  146   80   52   65  217   37    0  68   16   65   10  30  22
> triad.census(mySimNets[1:9,,])
      003  012 102 021D 021U 021C 111D 111U 030T 030C 201 120D 120U 120C 210 300
 [1,] 2320 1475 456  128   48  141   73  158   31    2  30   11   26   18  31  12
 [2,] 2411 1352 580  117   54   96   62  152   25    2  32    9   29    2  19  18
 [3,] 2293 1279 663  143   47   92   70  186   20    4  39   18   35   14  35  22
 [4,] 2171 1384 626  160   65  114   76  180   32    4  32   25   35   19  26  11
 [5,] 2530 1395 446  136   51   99   54  129   25    4  19   10   25    6  21  10
 [6,] 2637 1179 535   97   34   81   74  173   19    3  52   10   16    8  26  16
 [7,] 2476 1346 521  105   44  104   82  141   27    4  32    9   14   19  28   8
 [8,] 2641 1262 588   47   40   74   77  122    9    5  47    7    6   12  15   8
 [9,] 2671 1272 510   60   46   99   71  126   15    5  21    5   14    9  15  21
```

Reciprocity togehter with transitivity seems enough to explain the incidence of *transitive triangles* and *simmelian ties* (3 Mutual 0, Assymetric, 0 Null)

Questions?

What effects are there?

- RSiena Manual `http://www.stats.ox.ac.uk/~snijders/siena/RSiena_Manual.pdf` - check for `shortName`
- scroll through the effects available to you for your data `myeff` - check for `shortName`
- also `effectsDocumentation(myeff)`

Where did I get these numbers?

Estimation by Method of Moments: data

Basics for data

- You need at least 2 observations on $X(t)$ for waves $t_0,\ t_1$
- First observations is fixed and contains no information about θ
- No assumption of a stationary network distribution

Estimation by Method of Moments: procedure

How to estimate $\theta = (\lambda, \beta)$*?*

- pick starting values for θ
- simulate from $X(t_0)$ until t_1 - call the simulated network (-s) X_{rep}
- if statistic $Z_k(X_{\text{rep}})$ for parameter k is different to $Z_k(X_{\text{obs}})$, adjust accordingly

Estimation by Method of Moments: aim

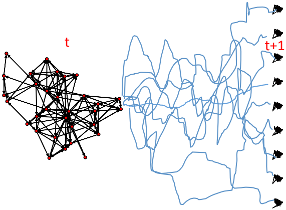
For suitable statistic $Z = (Z_1, \ldots, Z_K)$,
i.e., K variables which can be calculated from the network;
the statistic Z_k must be *sensitive* to the parameter θ_k
e.g. number of mutual dyads is sensitive to the reciprocity paramter (as we have seen)

The MoM estimate is a value: $\hat{\theta}$ of θ such that for

- observed stats $Z(X_{\mathrm{obs}})$
- and the the expected value $\mathrm{E}_{\theta}(Z(X_{\mathrm{rep}}))$

$$E_{\hat{\theta}}\left\{Z(X_{\mathrm{rep}})\right\} = Z(X_{\mathrm{obs}}) \,.$$

Method of Moments matches the moments



Do we have to do this for every update of the parameter θ?

Robbins-Monro algorithm

The moment equation $E_{\hat{\theta}}\{Z\} = z$ cannot be solved by analytical or the usual numerical procedures

Stochastic approximation (Robbins-Monro, 1951)

Iteration step:

$$\hat{\theta}_{N+1} = \hat{\theta}_N - a_N D^{-1}(z_N - z) \,, \tag{1}$$

where z_N is a simulation of Z with parameter $\hat{\theta}_N$,

D is a suitable matrix, and $a_N \to 0$.

Computer algorithm has 3 phases:

1. brief phase for preliminary estimation of $\partial \mathrm{E}_\theta \{Z\} / \partial \theta$ for defining D;

Computer algorithm has 3 phases:

1. brief phase for preliminary estimation of $\partial \mathrm{E}_\theta \{Z\} / \partial \theta$ for defining D;
2. estimation phase with Robbins-Monro updates, where a_N remains constant in *subphases* and decreases between subphases;

Computer algorithm has 3 phases:

1. brief phase for preliminary estimation of $\partial \mathrm{E}_\theta\{Z\}/\partial\theta$ for defining D;
2. estimation phase with Robbins-Monro updates, where a_N remains constant in *subphases* and decreases between subphases;
3. final phase where θ remains constant at estimated value; this phase is for checking that

$$\mathrm{E}_{\hat{\theta}}\{Z\} \approx z ,$$

and for estimating D_θ and Σ_θ to calculate standard errors.

Convergence

We say that $E_{\hat{\theta}}\{Z\} = z$ is approximately satidfied if, for each statistic $Z_k(X_{\rm obs})$ is within 0.1 standard deviation of $\mathrm{E}_{\theta}(Z(X_{\rm rep}))$. This is provided in the output as the *convergence t-ratio* (and the overall maximum convergence ratio is less than 0.25)

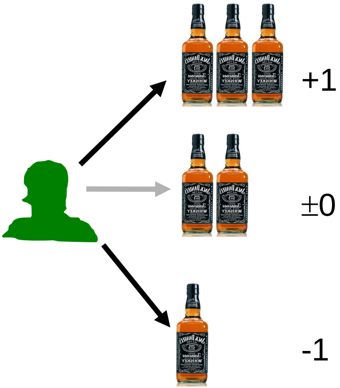
Change to BEHAVIOUR

Satisfaction with new state: $f_i +$ random component

Change to BEHAVIOUR

Satisfaction with new state: $f_i +$ random component

Change to BEHAVIOUR



Satisfaction with new state: f_i+ random component

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,

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

Again, multinomial logit form.

Things that go into the objective functions - selection

Homophily effects:

counts of the number of ties to people that are “like me”

Things that go into the objective functions - influence

Controls:

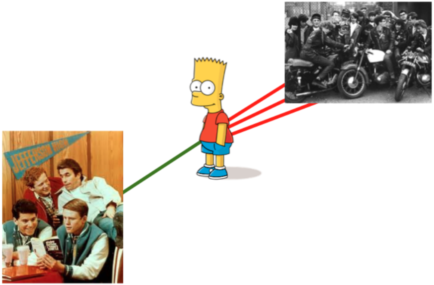
1. Gender
2. Age
3. Education

For influence effects:
immitation
persuation
etc

Things that go into the objective functions - influence

Controls:

1. Gender
2. Age
3. Education



For influence effects:

immitation

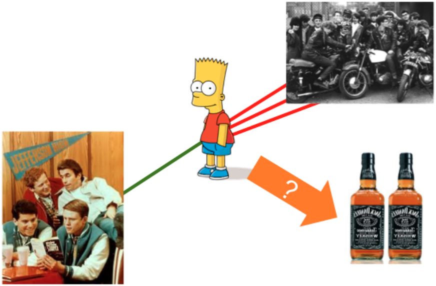
persuation

etc

Things that go into the objective functions - influence

Controls:

1. Gender
2. Age
3. Education



For influence effects:

immitation

persuation

etc

Summary

What is the purpose of having the embedded Markov Chain in continuous time?

Summary

What is the purpose of having the embedded Markov Chain in continuous time?

DYNAMICS

can model change of *tie* as dependent on current ties AND behaviour

can model change in *behaviour* as dependent on current behaviour AND the behavior of those you are tied to

Summary

What is the purpose of having the embedded Markov Chain in continuous time?

STATISTICAL

This is a statistical model that has estimable parameters for selection and influence

This is a generative model from which we can also generate replicate data AND assess GOF

Summary

What is the purpose of having the embedded Markov Chain in continuous time?

STATISTICAL

This is a statistical model that has estimable parameters for selection and influence

This is a generative model from which we can also generate replicate data AND assess GOF

Compare

- Generalized Estimation Equations
- Regressing behaviour wave 1 on wave 0

Example: 50 girls in a Scottish secondary school

Study of smoking initiation and friendship (starting age 12-13 years)
(following up on earlier work by P. West, M. Pearson & others).
with sociometric & behavior questionnaires at three moments, at appr. 1 year intervals.

Smoking: values 1–3;
drinking: values 1–5;

covariates:
gender, smoking of parents and siblings (binary),
money available (range 0–40 pounds/week).

Rename data that was automatically loaded

```
friend.data.w1 <- s501
friend.data.w2 <- s502
friend.data.w3 <- s503
drink <- s50a
smoke <- s50s
friendshipData <- array( c( friend.data.w1,
                            friend.data.w2,
                            friend.data.w3 ),
                         dim = c( 50, 50, 3 ) )
```

Define dependent/independent data

```
friendship <- sienaDependent(friendshipData)
drinking <- sienaDependent( drink, type = "behavior" )
smoke1 <- coCovar( smoke[ , 1 ] )
```

Join data and get effects

```
NBdata <- sienaDataCreate( friendship,
                           smoke1,
                           drinking )
NBeff <- getEffects( NBdata )
```

Define structural network effects

```
NBeff <- includeEffects( NBeff, transTrip, transRecTrip )
```

Define covariate effects on the network (selection)

```
NBeff <- includeEffects( NBeff,
                                   egoX, egoSqX, altX, altSqX,
diffSqX,
                                   interaction1 = "drinking" )
NBeff <- includeEffects( NBeff, egoX, altX, simX,
                                   interaction1 = "smoke1" )
```

Define effects on drinking (influence)

```
NBeff <- includeEffects( NBeff, avAlt, name="drinking",
                         interaction1 = "friendship" )
```

Define estimation settings and estimate

```
myalgorithm1 <- sienaAlgorithmCreate( projname = 's50_NB' )
NBans <- siena07( myalgorithm1,
                         data = NBdata, effects = NBeff,
 returnDeps = TRUE )
```

Result selection

Effect	par.	(s.e.)	t stat.
constant friendship rate (period 1)	6.21	(1.08)	-0.0037
constant friendship rate (period 2)	5.01	(0.87)	0.0042
outdegree (density)	–2.82	(0.27)	–0.0809
reciprocity	2.82	(0.35)	0.0559
transitive triplets	0.90	(0.16)	0.0741
transitive recipr. triplets	–0.52	(0.24)	0.0695
smoke1 alter	0.07	(0.17)	0.0343
smoke1 ego	–0.00	(0.15)	0.0747
smoke1 similarity	0.25	(0.24)	0.0158
drinking alter	–0.06	(0.15)	0.0158
drinking squared alter	–0.11	(0.14)	0.0704
drinking ego	0.04	(0.13)	0.0496
drinking squared ego	0.22	(0.12)	0.0874
drinking diff. squared	–0.10	(0.05)	0.0583

convergence t ratios all < 0.09.
Overall maximum convergence ratio 0.19.

Result Influence

Effect	par.	(s.e.)	t stat.
rate drinking (period 1)	1.31	(0.34)	–0.0692
rate drinking (period 2)	1.82	(0.54)	0.0337
drinking linear shape	0.42	(0.24)	0.0301
drinking quadratic shape	–0.56	(0.33)	0.0368
drinking average alter	1.24	(0.81)	0.0181

convergence t ratios all < 0.09.
Overall maximum convergence ratio 0.19.

Everything you need to know (including scipts for all kinds of data) is avaiable at

`http://www.stats.ox.ac.uk/~snijders/siena/`