Introduction to Stochastic Actor-Oriented Models

Fundamentals of SAOMs

Johan Koskinen

Department of Statistics
Stockholm University
University of Melbourne

February 27, 2025

Stockholm University

Preamble

- All material is on the workshop repository https://github.com/johankoskinen/CHDH-SNA
  - Download the RMarkdown file CHDH-SNA-3.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 (Friday AM part 1)
   - SAOM as an agent-based model
   - How to estimate a SAOM
2. (Social Influence) Analysing social influence with SAOM (Friday PM part 2)
   - Accounting for nodal attributes
   - Modelling change of nodal attributes
   - Trouble shooting and dealing with common issues
3. Advanced topics in SAOM (Friday PM)
   - Even more types of data
   - Likelihood-based estimation
   - Settings and imperfect data
   - Modelling multiple parallel networks

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

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

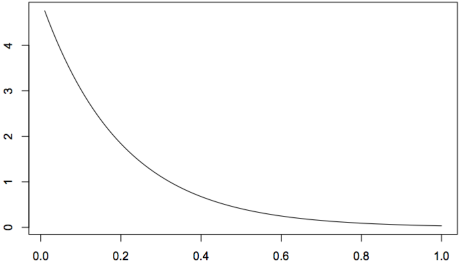
What type of data do we want to explain

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

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

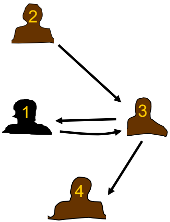
Micro-step

When actor i has opportunity to change
They may toggle x_{ij} to $1 - x_{ij}$
We call the new network

$$X(i \rightsquigarrow j)$$

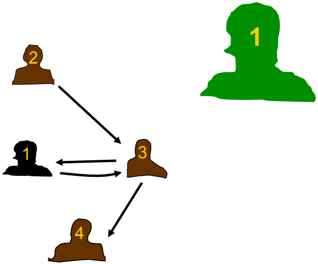
the network X that *differs* in exactly **one** tie-variable x_{ij}

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

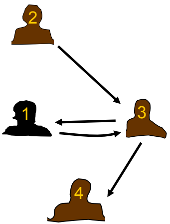


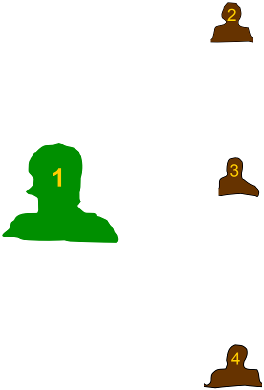
$x =$

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

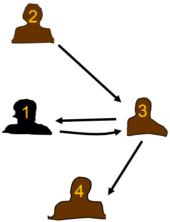


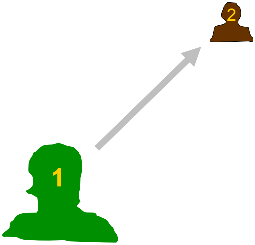
$$x = \begin{array}{|c|c|c|c|} \hline - & 0 & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$



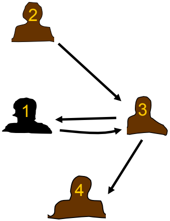


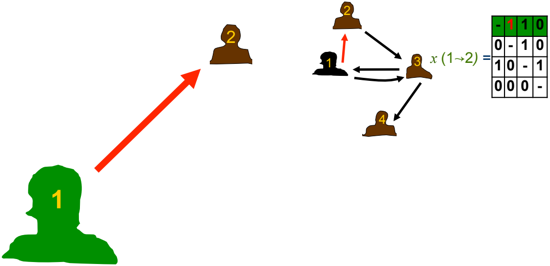
$$x = \begin{array}{|c|c|c|c|} \hline - & 0 & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$



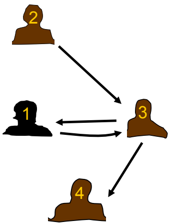


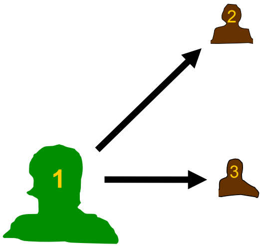
$$x = \begin{array}{|c|c|c|c|} \hline - & 0 & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$

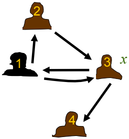




$$x = \begin{array}{|c|c|c|c|}\hline - & 0 & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$

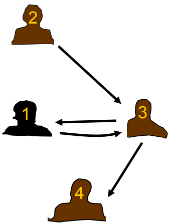


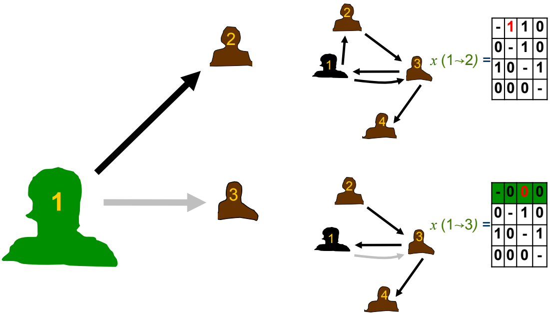




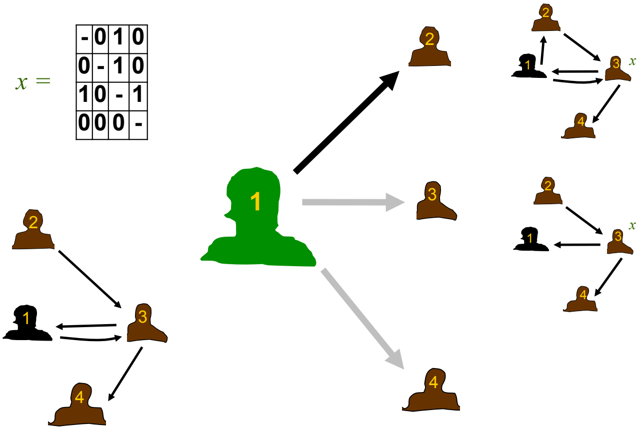
$$x\,(1\to 2) = \begin{array}{|c|c|c|c|}\hline - & \color{red}{1} & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$

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





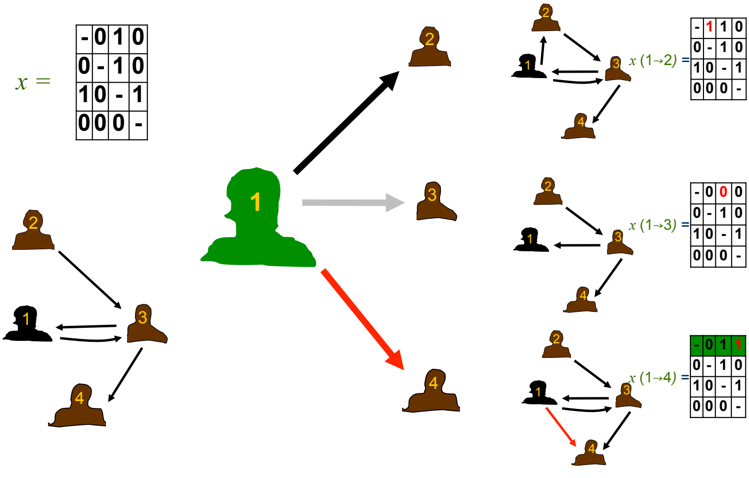
$$x = \begin{array}{|c|c|c|c|}\hline - & 0 & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$



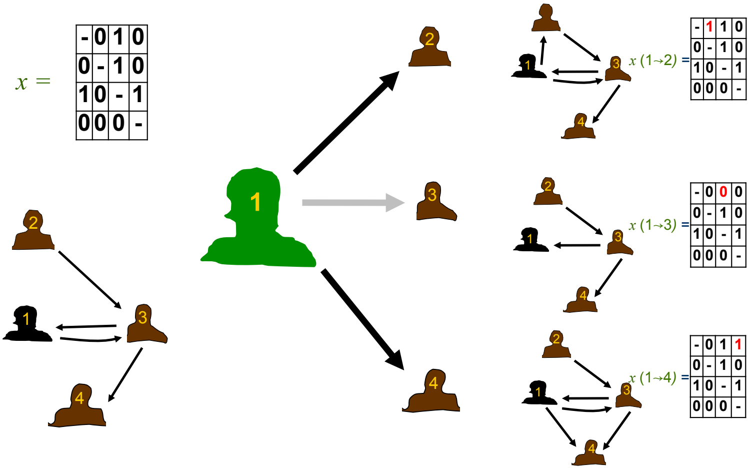
$$x\,(1 \rightarrow 2) = \begin{array}{|c|c|c|c|}\hline - & \color{red}{1} & 1 & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$

$$x\,(1 \rightarrow 3) = \begin{array}{|c|c|c|c|}\hline - & 0 & \color{red}{0} & 0 \\ \hline 0 & - & 1 & 0 \\ \hline 1 & 0 & - & 1 \\ \hline 0 & 0 & 0 & - \\ \hline \end{array}$$

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



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



Probability of a change

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

$$p_{ij}(\beta, \underbrace{X}_{\text{current network}}) = \underbrace{\frac{\exp\left(f_i(\beta, \underbrace{X(i \rightsquigarrow j)}_{\text{new network}})\right)}{\sum_{h=1}^{n} \exp\left(f_i(\beta, X(i \rightsquigarrow h))\right)}}_{\text{all possible changes}},$$

where

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

Probability of a change: utility

One-step jump probability: *can* be derived as:

- Each network X has **untility** $U_i(X, t)$ for i
- Actor i chooses network X that maximises $U_i(X, t)$

If (random) utility has form

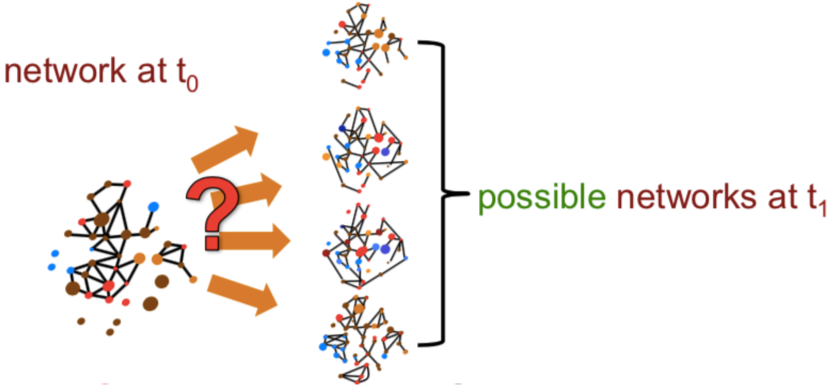
$$U_i(X,t) = \underbrace{f_i(\beta, X)}_{\text{objective function}} \quad + \quad \epsilon_{it} \, .$$

$$\Uparrow$$

random component

Actors are (*myopically*) maximising the utility of their network ties

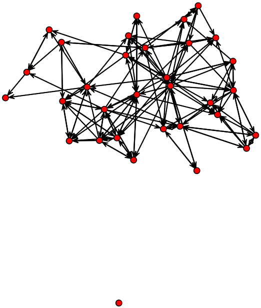
Agent-based: Change driven by incremental updates

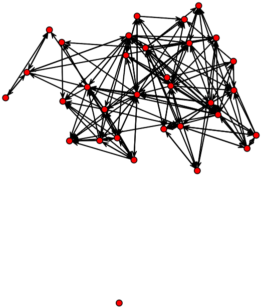


van de Bunt data set: R

Markdown: https://raw.githubusercontent.com/johankoskinen/CHDH-SNA/main/Markdowns/CHDH-SNA-3.Rmd

```
library('RSiena')
library('network')
library('sna')
tmp3[is.na(tmp3)] <- 0 # remove missing
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:

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

meaning that

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

because if

- currently $x_{ij} = 1$, then
- the number of ties for i in $X(i \rightsquigarrow j)$ will be one less (-1),
- and if currenlty $x_{ij} = 0 \ldots$

Simulation settings: actors only care about degree

Let the rate be equal for all $\lambda_i = \lambda = 5.7288$

- ✓ is each iteration, actor with shortest waiting time 'wins' (and gets to change)
- ✓ on average every actor gets 5.7 opportunities to change

and set $\beta = -0.7349$

- ✓ 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

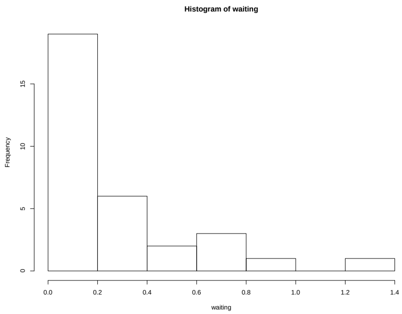
van de Bunt data set

```
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'] <- 5.7288
myeff$initialValue[
        myeff$shortName=='density'][1] <- -0.7349
```

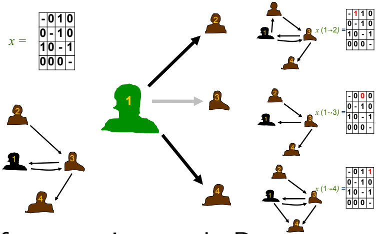
Model: rate

```
waiting <- rexp(32,5.7288)
hist(waiting)
which( waiting == min(waiting))
```

⟸ the winner



Model: conditional one-step change probability



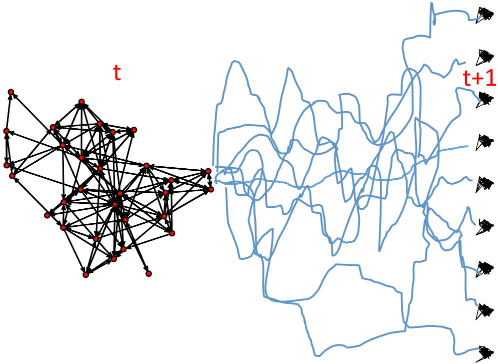
$$\Pr(1 \rightsquigarrow 2) = \frac{e^{-0.7349}}{e^{-0.7349}+e^{1.1059}+e^{-0.7349}+1} = 0.1185$$

$$\Pr(1 \rightsquigarrow 3) = \frac{e^{1.1059}}{e^{-0.7349}+e^{1.1059}+e^{-0.7349}+1} = 0.5156$$

$$\Pr(1 \rightsquigarrow 4) = \frac{e^{-0.7349}}{e^{-0.7349}+e^{1.1059}+e^{-0.7349}+1} = 0.1185$$

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

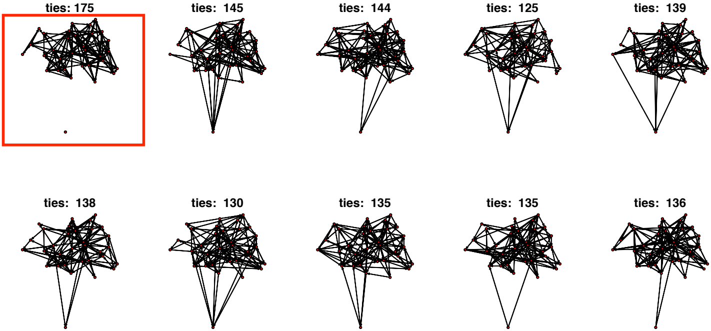
```
n <- dim(tmp4)[1]
mySimNets <- reshapeRSienaDeps( sim_ans , n )
```

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

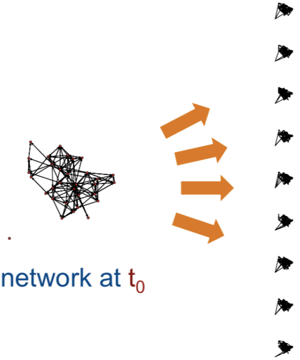
Plot observed network and 9 simulated

```
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



Dyad Census

```
> dyad.census(tmp4)
     Mut Asym Null
[1,]  46   83  367
> dyad.census(mySimNets[1:9,,])
      Mut Asym Null
 [1,]  23  128  345
 [2,]  25  133  338
 [3,]  17  136  343
 [4,]  20  134  342
 [5,]  16  143  337
 [6,]  21  136  339
 [7,]  26  118  352
 [8,]  23  128  345
 [9,]  30  122  344
```

Conclusions

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

Assume that i ALSO cares about

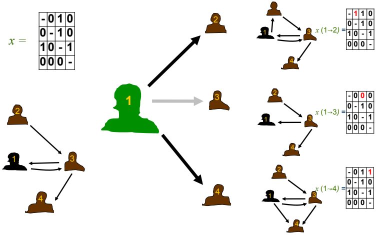
having ties $i \to j$ reciprocated $j \to i$

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

meaning that probability that i toggles relationship to j

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

Model



Objective function:

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

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

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

Simulation settings: actors care about degree and reciprocity

Let the rate be equal for all $\lambda_i = \lambda = 6.3477$

- ✓ on average every actor gets 6.3 opportunities to change

and set $\beta_d = -1.1046$

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

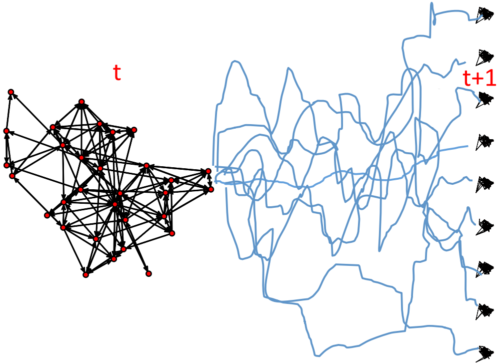
and set $\beta_r = 1.2608$

- ✓ 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'] <- 6.3477
myeff$initialValue[
        myeff$shortName =='density'][1] <- -1.1046
myeff$initialValue[
          myeff$shortName =='recip'][1] <- 1.2608
```

Now let us simulate



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

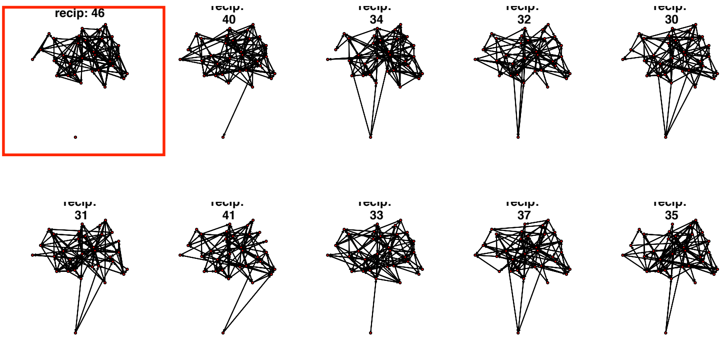
```
sim_model  <-  sienaAlgorithmCreate(
                     projname = 'sim_model',
                          cond = FALSE,
                          useStdInits = FALSE, nsub = 0 ,
                          simOnly = TRUE)
sim_ans <- siena07( sim_model, data = mydata,
                          effects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

The object sim_ans will now contain 1000 simulated networks
NOTE: this piece of code is unchanged

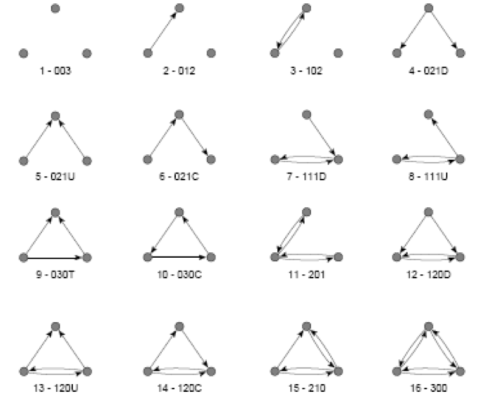
Plot observed network and 9 simulated

```
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,] 1968 1381 718   95   84  160  148  224   16    4  77   13   13   17  33   9
 [2,] 2067 1348 703   85   82  154  150  191   16    8  75    9   11   25  27   9
 [3,] 2073 1397 687  102   75  158  129  181   18    2  68   14   13   23  13   7
 [4,] 2185 1313 733   78   60  132  102  172   20    7  89    7   10   20  28   4
 [5,] 2040 1340 766   89   64  155  129  189   18    7  82   12   11   19  27  12
 [6,] 2206 1403 669   76   68  135  122  143   17    6  64    8   12    9  17   5
 [7,] 1788 1357 760  113   89  169  195  238   30   11  98   14   16   34  37  11
 [8,] 2164 1301 681   70   65  136  168  174   12    5  91   15    8   30  28  12
 [9,] 1988 1383 729  111   67  151  148  173   26   10  82   13   22   19  32   6
```

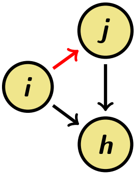
Reciprocity is clearly not enough to explain the incidence of *transitive triangles* and *simmelian ties* (3 Mutual 0, Assymetric, 0 Null)

Assume that i ALSO cares about *closure*

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

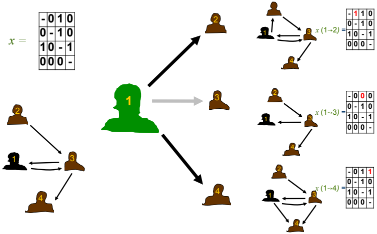
Modelled through, e.g.

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 $\lambda_i = \lambda = 7.0959$

- ✓ on average every actor gets 7 opportunities to change

and set $\beta_d = -1.6468$

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

and set $\beta_r = 0.8932$

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

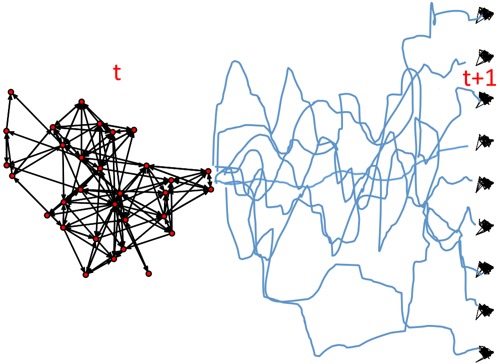
and set $\beta_t = 0.2772$

- ✓ here $\beta_t > 0$ - 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'] <- 7.0959
myeff$initialValue[
        myeff$shortName =='density'][1] <- 1.6468
myeff$initialValue[
          myeff$shortName =='recip'][1] <-  0.8932
myeff$initialValue[
          myeff$shortName =='transTrip'][1] <-  0.2772
```

Now let us simulate



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

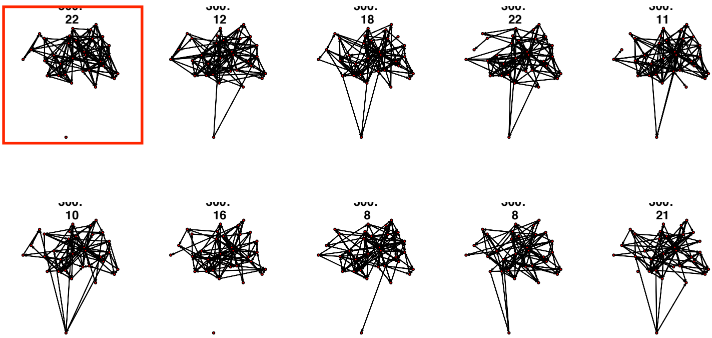
```
sim_model  <-  sienaAlgorithmCreate(
                     projname = 'sim_model',
                          cond = FALSE,
                          useStdInits = FALSE, nsub = 0 ,
                          simOnly = TRUE)
sim_ans <- siena07( sim_model, data = mydata,
                          effects = 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,] 2029 1223 662  108   79  159  138  257   36    6  82   19   35   29  68  30
 [2,] 2163 1292 727   89   62  125  136  160   15    4  61    9   23   25  29  40
 [3,] 2551 1191 657   44   34   84   88  143    5    0  78    6   16   14  27  22
 [4,] 1990 1430 576  100   84  173  128  200   23   10 103   12   38   22  45  26
 [5,] 2219 1390 631  103   83  121   84  175   25    5  45    8   21    9  31  10
 [6,] 2031 1218 799   75   56  109  136  213   15    9 114   15   25   25  61  59
 [7,] 2079 1443 597  104   72  160   97  206   24    4  54   17   24   24  38  17
 [8,] 2105 1256 764   77   55  114   99  212   12    5  98   11   33   16  50  53
 [9,] 2405 1260 569  103   55  126   78  184   22    6  41   12   25    8  46  20
```

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

Computer simulation algorithm for arbitrary rate function $\lambda_i(\alpha, \rho, x)$

1. Set $t = 0$ and $x = X(0)$.
2. Generate S according to the exponential distribution with mean $1/\lambda_+(\alpha, \rho, x)$ where $\lambda_+(\alpha, \rho, x) = \sum_i \lambda_i(\alpha, \rho, x)$.
3. Select $i \in \{1, ..., n\}$ using probabilities $\frac{\lambda_i(\alpha,\rho,x)}{\lambda_+(\alpha,\rho,x)}$.

4. Select $j \in \{1, ..., n\}$, $j \neq i$ using probabilities $p_{ij}(\beta, x)$.
5. Set $t = t + S$ and $x = x(i \rightsquigarrow j)$.
6. Go to step 2
   (unless stopping criterion is satisfied).

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 $Z_k(X_{\mathrm{rep}})$ for parameter k is different to $Z_k(X_{\mathrm{obs}})$, adjust accordingly

Estimation by Method of Moments: aim

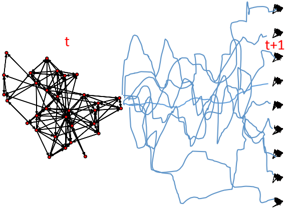
For suitable statistic $Z = (Z_1, \ldots, Z_K)$,
i.e., K variables which can be calculated from the network;
the statistic Z_k must be *sensitive* to the parameter θ_k
e.g. number of mutual dyads is sensitive to the reciprocity parameter (as we have seen)

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:

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

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;
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 satisfied if, for each statistic $Z_k(X_{\mathrm{obs}})$ is within 0.1 standard deviation of $\mathrm{E}_{\theta}(Z(X_{\mathrm{rep}}))$.
This is provided in the output as the *convergence t-ratio* (and the overall maximum convergence ratio is less than 0.25)

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

Halfway 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

Testing assumptions: Goodness-of-fit (GOF)

We can (almost) always get estimates
but model is very complex
so how do we know that it is realistic?

Two routines for goodness-of-fit

- `sienaTimeTest()`
  for testing time heterogeneity
- `sienaGOF()`
  for checking that the model reproduces the features of the observed networks (that were not modelled).

Time-test

Standard assumptions M waves, the $M-1$ periods follow the same model with the same parameters.
Use

- `sienaTimeTest()`
  to test if some parameters differ across any of the periods
- if test 'positive'
  include interactions with time using
  `includeTimeDummy()`

see `RscriptSienaTimeTest.r`

Extension 2: Is model homogenous over time

Example time test (Lospinoso et al., 2010)
vdb_tt.R:

```
vdb.ans1 <- siena07(vdb.model, data=vdb.data,
                       effects=vdb.eff,
                      useCluster=FALSE, initC=TRUE)
        timetest.1 <- sienaTimeTest(vdb.ans1)
        summary(timetest.1)
        plot(timetest.1, effects=c(1,3))
```

Goodness of fit

Principle: simulate replicate data
and check how simulations compare to observed data
This is exactly what we did in 'Simulating SAOM'
What are we looking for?
does model capture features that we have not modelled?

built in GOF-function

Siena has function `sienaGOF()`
This operates on your siena-object
generated from `siena07()` with option `returnDeps = TRUE`

choosing features for GOF

Some preprogrammed ‘auxiliary’ functions
that can be passed to `sienaGOF` are:
`OutdegreeDistribution()`
`IndegreeDistribution()`
`BehaviorDistribution()`
you can also create custom functions

More help on GOF

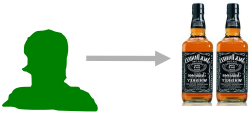
Use ? function and sienaGOF_new.R

```
results1 <- siena07(myalg, data=mydata,
                 effects=myeff, returnDeps=TRUE)
gof1.od <- sienaGOF(results1, verbose=TRUE,
              varName="friendship",
              OutdegreeDistribution,
              cumulative=TRUE, levls=0:10)
gof1.od
plot(gof1.od)
```

See example script
https://www.stats.ox.ac.uk/~snijders/siena/sienaGOF_vdB.R

Modelling behaviour change - social influence

Change to BEHAVIOUR



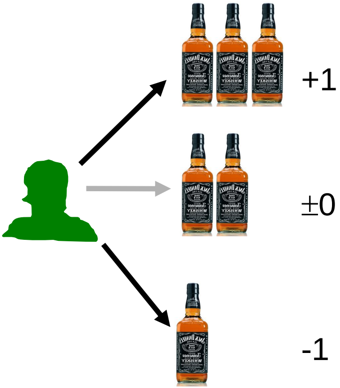
± 0

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:

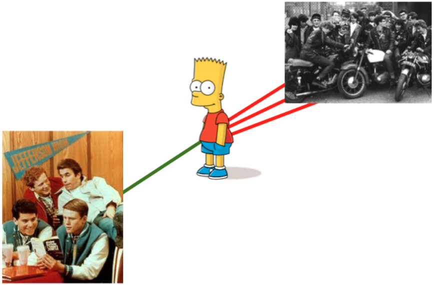
1. Gender
2. Age
3. Education

For influence effects:
immitation
persuation
etc

Things that go into the objective functions - influence

Controls:

1. Gender
2. Age
3. Education

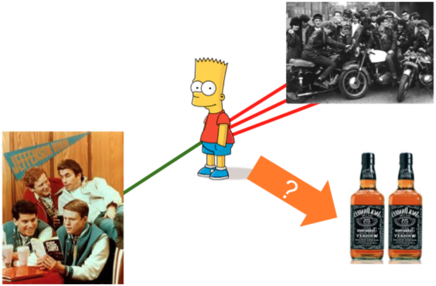


For influence effects:
immitation
persuation
etc

Things that go into the objective functions - influence

Controls:

1. Gender
2. Age
3. Education



For influence effects:
immitation
persuation
etc

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.

More structural effects

Default effects

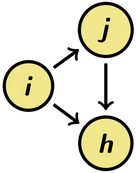
Choose possible network effects for actor i, e.g.:
(others to whom actor i is tied are called here i's 'friends')

1. *out-degree effect*, controlling the density / average degree,
   $s_{i1}(x) = x_{i+} = \sum_j x_{ij}$
2. *reciprocity effect*, number of reciprocated ties
   $s_{i2}(x) = \sum_j x_{ij} x_{ji}$

Four ways of closure (1)

Four potential effects representing network closure:

3. *transitive triplets effect*,
   number of transitive patterns in i's ties
   $(i \to j,\ j \to h,\ i \to h)$
   $s_{i3}(x) = \sum_{j,h} x_{ij}\, x_{jh}\, x_{ih}$



transitive triplet

4. *transitive ties effect*,
   number of actors j to whom i is tied indirectly
   (through at least one intermediary: $x_{ih} = x_{hj} = 1$)
   and also directly $x_{ij} = 1$),
   $s_{i4}(x) = \#\{j \mid x_{ij} = 1,\ \max_h(x_{ih}\, x_{hj}) > 0\}$

Four ways of closure (2)

5. *indirect ties effect*,
   number of actors j to whom i is tied indirectly
   (through at least one intermediary: $x_{ih} = x_{hj} = 1$)
   but not directly $x_{ij} = 0$),
   $=$ number of geodesic distances equal to 2,
   $s_{i5}(x) = \#\{j \mid x_{ij} = 0, \max_h(x_{ih}\, x_{hj}) > 0\}$

Four ways of closure (3)

6. *balance* or structural equivalence,
similarity between outgoing ties of i
with outgoing ties of his friends,

$$s_{i6}(x) = \sum_{j=1}^{n} x_{ij} \sum_{\substack{h=1 \\ h \neq i,j}}^{g} \left(1 - |x_{ih} - x_{jh}|\right) ,$$

[note that $(1 - |x_{ih} - x_{jh}|) = 1$ if $x_{ih} = x_{jh}$,
and 0 if $x_{ih} \neq x_{jh}$, so that

$$\sum_{\substack{h=1 \\ h \neq i,j}}^{g} \left(1 - |x_{ih} - x_{jh}|\right)$$

measures agreement between i and j .]

Four ways of closure (4)

Differences between these three network closure effects:

- transitive triplets effect: i more attracted to j
  if there are *more* indirect ties $i \rightarrow h \rightarrow j$;
- transitive ties effect: i more attracted to j
  if there is *at least one* such indirect connection ;
- balance effect:
  i prefers others j who make same choices as i.

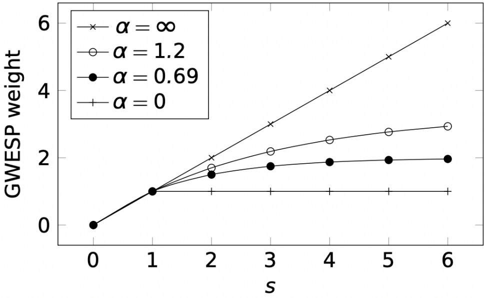
One way of closure: GWESP

Nowadays, we often use GWESP (geometrically weighted edgewise shared partners) - combines `transTrip` and `transTies`:

$$GWESP(i,\alpha) = \sum_j x_{ij} e^{\alpha} \left[1 - (1 - e^{-\alpha})^{\overbrace{\sum_h x_{ih} x_{jh}}^{\sharp\text{com. partn.}}} \right]$$

- for $\alpha \geq 0$ (effect parameter $= 100 \times \alpha$).
- Default $\alpha = \log(2)$, `parameter = 69`

One way of closure: GWESP



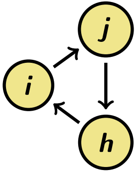
Weight tie $i \to j$ for $s = \sum_h x_{ih} x_{jh}$

Degree-based effects

7. *in-degree related popularity effect*, sum friends' in-degrees
   $s_{i7}(x) = \sum_j x_{ij} \sqrt{x_{+j}} = \sum_j x_{ij} \sqrt{\sum_h x_{hj}}$
   related to dispersion of in-degrees
   (can also be defined without the $\sqrt{}$ sign);
8. *out-degree related popularity effect*,
   sum friends' out-degrees
   $s_{i8}(x) = \sum_j x_{ij} \sqrt{x_{j+}} = \sum_j x_{ij} \sqrt{\sum_h x_{jh}}$
   related to association in-degrees — out-degrees;
9. *Outdegree-related activity effect* ,
   $s_{i9}(x) = \sum_j x_{ij} \sqrt{x_{i+}} = x_{i+}^{1.5}$
   related to dispersion of out-degrees;
10. *Indegree-related activity effect* ,
    $s_{i10}(x) = \sum_j x_{ij} \sqrt{x_{+i}} = x_{i+} \sqrt{x_{+i}}$
    related to association in-degrees — out-degrees;

Four ways of closure (5)

11. *three-cycle effect*,
    number of three-cycles in i's ties
    $(i \to j,\ j \to h,\ h \to i)$
    $s_{i11}(x) = \sum_{j,h}\ x_{ij}\, x_{jh}\, x_{hi}$



three-cycle

    This represents a kind of generalized reciprocity, and absence of hierarchy.

12. ... and potentially many others ...

More on selection effects

Selection effects

Preferences of actors dependent on their degrees:

- out ego - out alter degrees
- out ego - in alter degrees
- in ego - out alter degrees
- in ego - in alter degrees

All these are product interactions between the two degrees (or square roots).

Selection effects: types of evaluations

Four kinds of evaluation function effect associated with actor covariate v_i.

This applies also to behavior variables Z_h.

13. *covariate-related popularity*, 'alter'
    sum of covariate over all of i 's friends
    $s_{i13}(x) = \sum_j x_{ij} \, v_j$;
14. *covariate-related activity*, 'ego'
    i's out-degree weighted by covariate
    $s_{i14}(x) = v_i \, x_{i+}$;

Selection effects: similarity

15. *covariate-related similarity*,
    sum of measure of covariate similarity
    between i and his friends,
    $s_{i15}(x) = \sum_j x_{ij} \, \mathrm{sim}(v_i, v_j)$
    where $\mathrm{sim}(v_i, v_j)$ is the similarity between v_i and v_j,
    $$\mathrm{sim}(v_i, v_j) = 1 - \frac{|v_i - v_j|}{R_V} \; ,$$
    R_V being the range of V;
16. *covariate-related interaction*, ‘ego $\times$ alter’
    $s_{i16}(x) = v_i \sum_j x_{ij} \, v_j$;

Selection effects: similarity

Snijders and Lomi (2019) *Beyond homophily: Incorporating actor variables in statistical network models*:

- for (non-binary) variables v_i
  - combination of tendencies of
    - homophily,
    - aspiration, and
    - social norm
  - yields 5 effects:
    1. ego $x_{ij} v_i$
    2. alter $x_{ij} v_j$
    3. ego-squared $x_{ij} v_i^2$
    4. ego-alter difference squared $x_{ij}(v_i - v_j)^2$ and
    5. alter squared $x_{ij} v_j^2$

Do we really *have* to use this?

Example van de Bunt (1)

Example (Gerhard van de Bunt)

Data

- 32 university freshmen (24 fem and 8 male)
- (here) 3 obs. $(t_1,\ t_2,\ t_3)$ at 6, 9, and 12 weeks
- The relation: ‘friendly relation’.

Missing entries $x_{ij}(t_m)$ set to 0 and not used in calculations of statistics.
Densities increase from 0.15 at t_1 via 0.18 to 0.22 at t_3 .

Example van de Bunt (2)

Example (Gerhard van de Bunt (cont.))

Very simple model: only out-degree and reciprocity effects

	Model 1	
Effect	par.	(s.e.)
Rate $t_1 - t_2$	3.51	(0.54)
Rate $t_2 - t_3$	3.09	(0.49)
Out-degree	-1.10	(0.15)
Reciprocity	1.79	(0.27)

rate parameters:
per actor about 3 opportunities for change between observations;
out-degree parameter negative:
on average, cost of friendship ties higher than their benefits;
reciprocity effect strong and highly significant ($t = 1.79/0.27 = 6.6$).

Example van de Bunt (3)

Example (Gerhard van de Bunt (cont.))

Evaluation function is

$$f_i(x) = \sum_j \Big(-1.10\, x_{ij} + 1.79\, x_{ij}\, x_{ji} \Big).$$

This expresses 'how much actor i likes the network'.
Adding a reciprocated tie (i.e., for which $x_{ji} = 1$) gives

$$-1.10 + 1.79 = 0.69.$$

Adding a non-reciprocated tie (i.e., for which $x_{ji} = 0$) gives

$$-1.10,$$

i.e., this has negative benefits.
Gumbel distributed disturbances are added:
these have variance $\pi^2/6 = 1.645$ and s.d. 1.28.

Example van de Bunt (4)

Example (Gerhard van de Bunt: with simple closure)

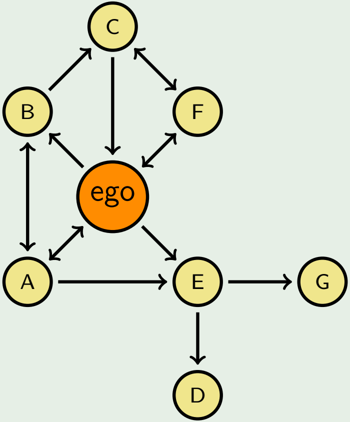
The estimates with only transitive ties:

Structural model with one network closure effect

	Model 3	
Effect	par.	(s.e.)
Rate $t_1 - t_2$	3.89	(0.60)
Rate $t_2 - t_3$	3.06	(0.47)
Out-degree	−2.14	(0.38)
Reciprocity	1.55	(0.28)
Transitive ties	1.30	(0.41)

Example van de Bunt (5)

Example (Gerhard van de Bunt: with simple closure (cont.))



for ego:

out-degree $x_{i+} = 4$
$\#\{\text{recipr. ties}\} = 2,$
$\#\{\text{trans. ties }\} = 3.$

Example van de Bunt (6)

Example (Gerhard van de Bunt: with simple closure (cont.))

The evaluation function is

$$f_i(x) = \sum_j \Big(-2.14\, x_{ij} + 1.55\, x_{ij}\, x_{ji} + 1.30\, x_{ij} \max_h \big(x_{ih}\, x_{hj}\big)\Big)$$

$\Big($ note: $\sum_j x_{ij} \max_h \big(x_{ih}\, x_{hj}\big)$ is $\#\{$trans. ties $\}$ $\Big)$
so its current value for this actor is

$$f_i(x) = -2.14 \times 4 + 1.55 \times 2 + 1.30 \times 3 = -1.56.$$

Example van de Bunt (7)

Example (Gerhard van de Bunt: with simple closure (cont.))

Options when 'ego' has opportunity for change:

	out-degr.	recipr.	trans. ties	gain	prob.
current	4	2	3	0.00	0.061
new tie to C	5	3	5	+2.01	0.455
new tie to D	5	2	4	+0.46	0.096
new tie to G	5	2	4	+0.46	0.096
drop tie to A	3	1	0	–3.31	0.002
drop tie to B	3	2	1	–0.46	0.038
drop tie to E	3	2	2	+0.84	0.141
drop tie to F	3	1	3	+0.59	0.110

The actor adds random influences to the gain (with s.d. 1.28), and chooses the change with the highest total 'value'.

Example van de Bunt (8)

Example (Gerhard van de Bunt: with more closure)

	Model 3	
Effect	par.	(s.e.)
Rate $t_1 - t_2$	4.64	(0.80)
Rate $t_2 - t_3$	3.53	(0.57)
Out-degree	–0.90	(0.58)
Reciprocity	2.27	(0.41)
Transitive triplets	0.35	(0.06)
Transitive ties	0.75	(0.45)
Three-cycles	–0.72	(0.21)
In-degree popularity ($\surd$)	–0.71	(0.27)

Conclusions:
Reciprocity, transitivity; negative 3-cycle effect; negative popularity effect.

Example van de Bunt (9)

Example (Gerhard van de Bunt: Add effects of gender & program, smoking similarity)

	Model 4	
Effect	par.	(s.e.)
Rate $t_1 - t_2$	4.71	(0.80)
Rate $t_2 - t_3$	3.54	(0.59)
Out-degree	–0.81	(0.61)
Reciprocity	2.14	(0.45)
Transitive triplets	0.33	(0.06)
Transitive ties	0.67	(0.46)
Three-cycles	–0.64	(0.22)
In-degree popularity ($\surd$)	–0.72	(0.28)
Sex (M) alter	0.52	(0.27)
Sex (M) ego	–0.15	(0.27)
Sex similarity	0.21	(0.22)
Program similarity	0.65	(0.26)
Smoking similarity	0.25	(0.18)

Conclusions:
Trans. ties now not needed any more to represent transitivity;
men more popular;
program similarity.

Example van de Bunt (10)

Example (Gerhard van de Bunt: selection table)

We may do the calculations with $F = 0, \; M = 1$ (even if centered) The joint effect of the gender-related effects
for the tie variable x_{ij} from i to j is

$$-0.15\, z_i \; + \; 0.52\, z_j \; + \; 0.21\, I\{z_i = z_j\} \; .$$

$i \;\; \backslash \;\; j$	F	M
F	0.21	0.52
M	-0.15	0.58

Conclusion:
men seem not to like female friends...?

More on influence effects

Influence effects

Many different reasons why networks
are important for behavior:

1. *imitation* :
   individuals imitate others
   (basic drive; uncertainty reduction).
2. *social capital* :
   individuals may use resources of others;
3. *coordination* :
   individuals can achieve some goals
   only by concerted behavior;

In this presentation, only imitation is considered,
but the other two reasons are also of eminent importance.

Influence effects

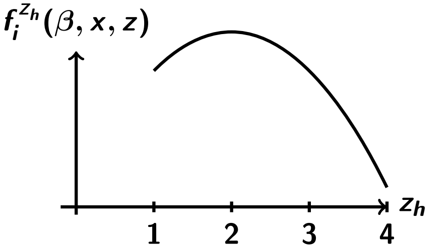
Basic effects for dynamics of behavior f_i^Z:

$$f_i^Z(\beta, x, z) = \sum_{k=1}^{L} \beta_k \, s_{ik}(x, z) \, ,$$

1. *tendency* ,
   $s_{i1}^Z(x, z) = z_{ih}$
2. *quadratic tendency*, ‘effect behavior on itself’,
   $s_{i2}^Z(x, z) = z_{ih}^2$
   Quadratic tendency effect important for model fit.

Influence effects

For a negative quadratic tendency parameter,
the model for behavior is a unimodal preference model.



For positive quadratic tendency parameters ,
the behavior objective function can be bimodal
('positive feedback').

Influence effects

3. *behavior-related average similarity*,
   average of behavior similarities between i and friends
   $s_{i3}(x) = \frac{1}{x_{i+}} \sum_j x_{ij} \, \mathrm{sim}(z_{ih}, z_{jh})$
   where $\mathrm{sim}(z_{ih}, z_{jh})$ is the similarity between v_i and v_j,
   $$\mathrm{sim}(z_{ih}, z_{jh}) = 1 - \frac{|z_{ih} - z_{jh}|}{R_{Z^h}} \ ,$$
   R_{Z^h} being the range of Z^h;
4. *average behavior alter* — an alternative to similarity:
   $s_{i4}(x, z) = z_{ih} \frac{1}{x_{i+}} \sum_j x_{ij} z_{jh}$

Effects 3 and 4 are alternatives for each other:
they express the same theoretical idea of influence
in mathematically different ways.
The data will have to differentiate between them.

Influence effects

Network position can also have influence on behaviour dynamics
e.g. through degrees rather than through behaviour
of those to whom one is tied:

5. *popularity-related tendency*, (in-degree)
   $s_{i5}(x, z) = z_{ih}\, x_{+i}$
6. *activity-related tendency*, (out-degree)
   $s_{i6}(x, z) = z_{ih}\, x_{i+}$

Influence effects

7. *dependence on other behaviours* $(h \neq \ell)$,
   $s_{i7}(x, z) = z_{ih}\, z_{i\ell}$

For both the network and the behaviour dynamics,
extensions are possible depending on the network position.

Influence effects

The *similarity effect* in evaluation function :

sum of absolute behaviour differences between i and his friends
$s_{i2}(x,z) = \sum_j x_{ij}\,\mathrm{sim}(z_{ih}, z_{jh})$.

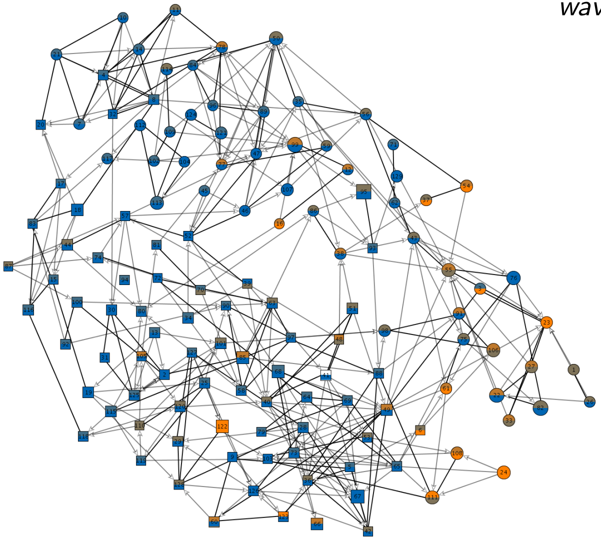
This is fundamental both
to network selection based on behaviour,
and to behavior change based on network position.

Influence effects: Example

Example (Smoke rings)

Study of smoking initiation and friendship
(following up on earlier work by P. West, M. Pearson & others).
One school year group from a Scottish secondary school
starting at age 12-13 years, was monitored over 3 years;
total of 160 pupils, of which 129 pupils present at all 3 observations; 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).

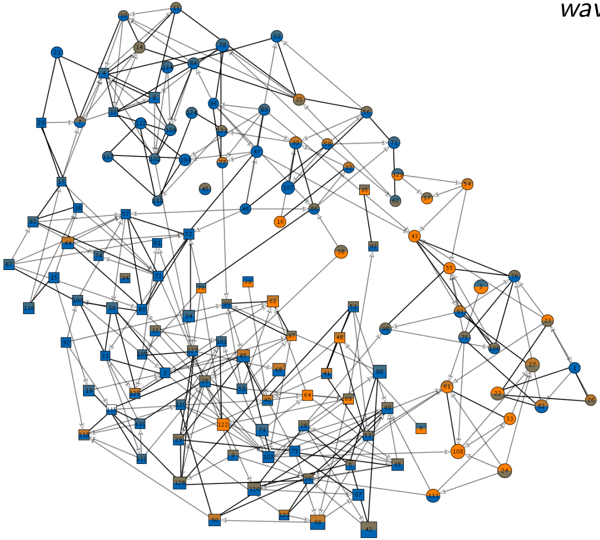


wave 1

girls: circles
boys: squares

node size: pocket money

color: top = drinking
bottom = smoking
(orange = high)

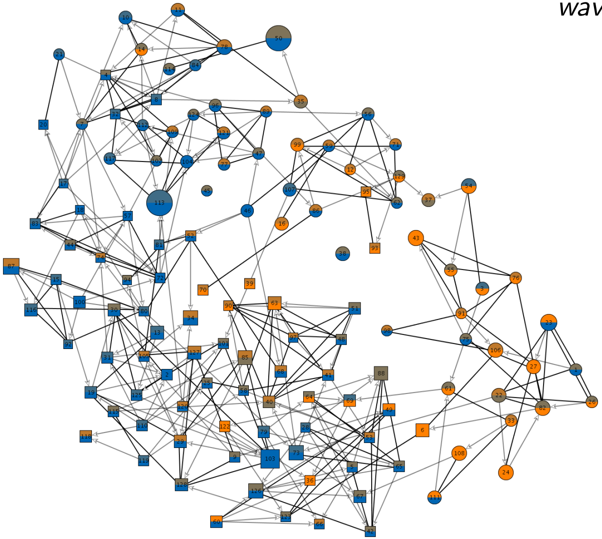


wave 2

girls: circles
boys: squares

node size: pocket money

color: top = drinking
bottom = smoking
(orange = high)



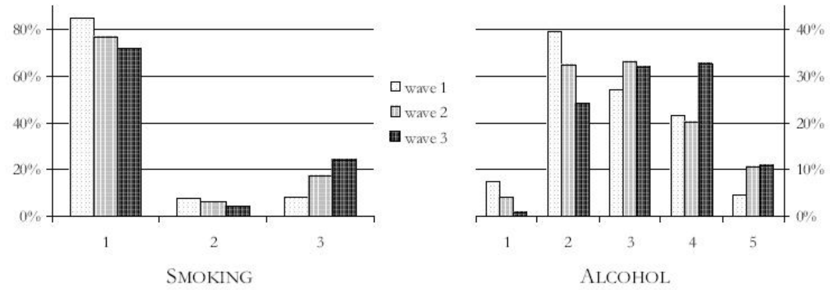
wave 3

girls: circles
boys: squares

node size: pocket money

color: top = drinking
bottom = smoking
(orange = high)

FIGURE 2. — OBSERVED DISTRIBUTION OF SUBSTANCE USE IN THE THREE WAVES.



More realistic model

Friendship dynamics	Rate 1	18.67	(2.17)
	Rate 2	12.42	(1.30)
	Outdegree	–1.57	(0.27)
	Reciprocity	2.04	(0.13)
	Transitive triplets	0.35	(0.04)
	Transitive ties	0.84	(0.09)
	Three-cycles	–0.41	(0.10)
	In-degree based popularity ($\sqrt{}$)	0.05	(0.07)
	Out-degree based popularity ($\sqrt{}$)	–0.45	(0.16)
	Out-degree based activity ($\sqrt{}$)	–0.39	(0.07)
	Sex alter	–0.14	(0.08)
	Sex ego	0.08	(0.10)
	Sex similarity	0.66	(0.08)
	Romantic exp. similarity	0.10	(0.06)
	Money alter (unit: 10 pounds/w)	0.11	(0.05)
	Money ego	–0.06	(0.06)
	Money similarity	0.98	(0.27)

More realistic model (continued)

Friendship dynamics	Drinking alter	–0.01	(0.07)
	Drinking ego	0.09	(0.09)
	Drinking ego $\times$ drinking alter	0.14	(0.06)
	Smoking alter	–0.08	(0.08)
	Smoking ego	–0.14	(0.09)
	Smoking ego $\times$ smoking alter	0.03	(0.08)

Smoking dynamics	Rate 1	4.74	(1.88)
	Rate 2	3.41	(1.29)
	Linear tendency	–3.39	(0.45)
	Quadratic tendency	2.71	(0.40)
	Ave. alter	2.00	(0.95)
	Drinking	–0.11	(0.24)
	Sex (F)	–0.12	(0.35)
	Money	0.10	(0.20)
	Smoking at home	–0.05	(0.29)
	Romantic experience	0.09	(0.33)

Alcohol consumption dynamics	Rate 1	1.60	(0.32)
	Rate 2	2.50	(0.42)
	Linear tendency	0.44	(0.17)
	Quadratic tendency	–0.64	(0.22)
	Ave. alter	1.34	(0.61)
	Smoking	0.01	(0.21)
	Sex (F)	0.04	(0.22)
	Money	0.17	(0.16)
	Romantic experience	–0.19	(0.27)

Conclusion:

In this case, the conclusions from a more elaborate model
– i.e., with better control for alternative explanations –
are similar to the conclusions from the simple model.

There is evidence for friendship selection based on drinking,
and for social influence with respect to smoking and drinking.

Parameter interpretation for behaviour change

Omitting the non-significant parameters yields
the following objective functions.
For smoking

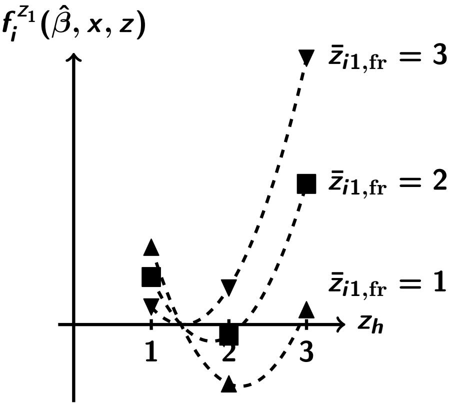
$$f_i^{z_1}(\hat{\beta}, x, z) =$$

$$-3.39\,(z_{i1} - \bar{z}_1) + 2.71\,(z_{i1} - \bar{z}_1)^2 + 2.00\,(z_{i1} - \bar{z}_1)\,(\bar{z}_{i1,\mathrm{fr}} - \bar{z}_1)\ ,$$

where z_{i1} is smoking of actor i: values 1–3, mean 1.4.
$\bar{z}_{i1,\mathrm{fr}}$ is the average smoking behavior of i's friends.

Convex function – consonant with addictive behavior.

$$-3.39\,(z_{i1}-\bar{z}_1)\;+\;2.71\,(z_{i1}-\bar{z}_1)^2\;+\;2.00\,(z_{i1}-\bar{z}_1)\,(\bar{z}_{i1,\mathrm{fr}}-\bar{z}_1)$$



For drinking the objective function (significant terms only) is

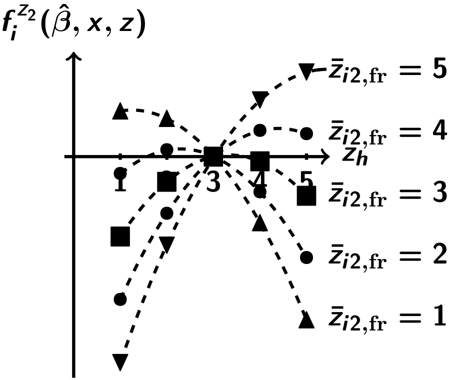
$$f_i^{z_2}(\hat{\beta}, x, z) =$$

$$0.44\,(z_{i2} - \bar{z}_2) \,-\, 0.64\,(z_{i2} - \bar{z}_2)^2 \,+\, 1.34\,(z_{i2} - \bar{z}_2)\,(\bar{z}_{i2,\mathrm{fr}} - \bar{z}_2)\ ,$$

where z_{i2} is drinking of actor i: values 1–5, mean 3.0.

Unimodal function – consonant with non-addictive behavior.

$$0.44\,(z_{i2} - \bar{z}_2) \;-\; 0.64\,(z_{i2} - \bar{z}_2)^2 \;+\; 1.34\,(z_{i2} - \bar{z}_2)\,(\bar{z}_{i2,\mathrm{fr}} - \bar{z}_2)$$



Testing parameters using score-type test

We might not be able to fit everything (no, really ...)
How test multiple parameters without estimation?

Score (-type) test

If MoM estimate, then

$$\hat{z} - z \approx 0$$

If this holds for the statistic z_K for a parameter $\theta_K = 0$, then $\theta_K = 0$ is a good value
Test non-estimated parameters θ^*, with the statistic

$$(\hat{z} - z)^\top D(\theta^*)^{-1}(\hat{z} - z)$$

where $\hat{z}$ is a vector of expected statistics for parameters θ^*, and D is a suitably scaled covariance matrix.

Trouble shooting: non-convergence

What stochastic approximation algorithm does

1. Gauging sensitivity of (estimation) statistics Z to parameters θ;
2. Robbins-Monro updates for θ
   - *nsub* subphases (usually 4)
   - decreasing step sizes, determined by *firstg*
3. Final: *n3* runs, θ constant at $\hat{\theta}$
   - Check deviations from targets

$$E_{\hat{\theta}}\{Z\} - z$$

   - estimating standard errors

Trouble shooting: non-convergence - bad start!

Initial values:

1. `sienaAlgorithmCreate`
   - `useStdInits=FALSE`: parameter values in effects object
     - starting with standard values
     - can be modified by functions `setEffect` and `updateTheta`
   - `useStdInits=TRUE`:
     - standard initial values
     - the values put in the effects object by `getEffects`.
2. With arg `prevAns` passed to `siena07`
   - initial values used from existing `sienaFit` object,
   - Skipping Phase 1 if mods identical

Trouble shooting: non-convergence - when?

Standard initial values mostly fine but for

- non-directed networks
- two-mode networks
- monotonic dependent variables
- multivariate networks with constraints
- data sets with many structurally determined values.

You may try

- start with only rate and density (-effects)
- `updateTheta` $\Rightarrow$ restart

Trouble shooting: non-convergence - normal

Typically, for `tconv.max > 0.25`,

- repeat estimation,
- using the `prevAns` parameter in `siena07`,
- until `tconv.max < 0.25`

Warning sign

- estimation *diverges right away*
  - check data and model specification;
  - perhaps use a simpler model.
- estimation still *diverges right away*, either:
  - estimate a simpler model, and use the result for `prevAns` with the intended model, or
  - use a smaller value for `firstg` (default: 0.2)

NB: `siena07` will **tell you** if effects *co-linear* - so don't worry about that

Trouble shooting: non-convergence - brute force

If model resits converging (`tconv.max > 0.25` after many restarts)

- Brute force: increase e.g. n2start and/or n3, with smaller `firstg`
- Better model
- Check for time-heterogeneity
- Better data
    - Do you miss important covariates?
    - Do your variables need transformations?
    - etc