Advanced Network Analysis

Stochastic Actor-Oriented Models

Shahryar Minhas [s7minhas.com]

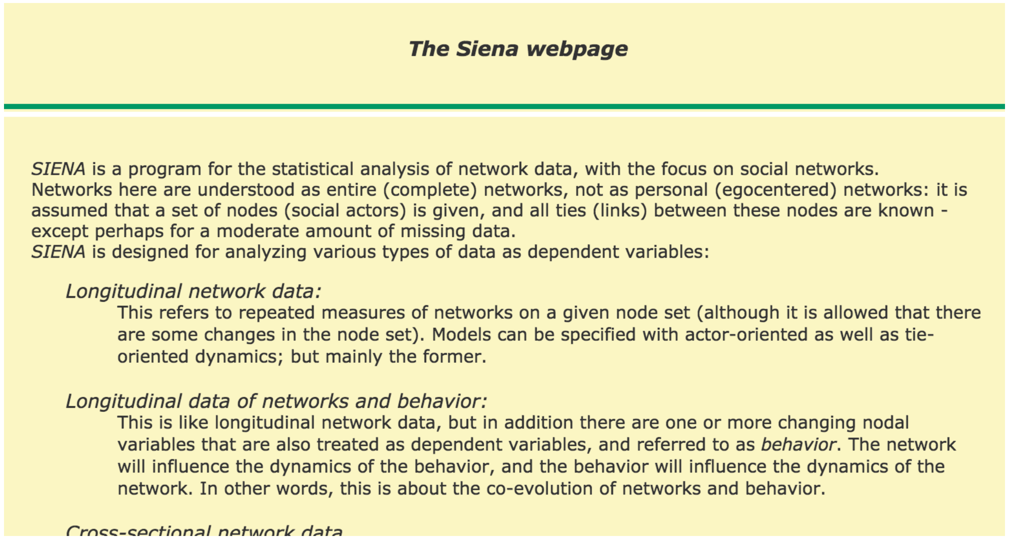
Why Use SOAM?

- Until now, we modeled networks as a function of covariates (endogenous and exogenous).
- Suppose we want to know whether network ties affect behavior? (e.g., friendships and bad habits)
- Or whether ties in two different networks affect each other (e.g., work and romantic relationships)
- SOAMs allow for modeling network(s) and nodal behavior as two mutually constitutive outcome variables.
- Can you think of other examples of mutually-constitutive networks and behavior?

SAOM

Stochastic actor oriented model developed primarily by Snijders is implemented in the RSiena package on CRAN:

- https://www.stats.ox.ac.uk/~snijders/siena/
- Recent overview piece by Snijders



The Siena webpage

SIENA is a program for the statistical analysis of network data, with the focus on social networks. Networks here are understood as entire (complete) networks, not as personal (egocentered) networks: it is assumed that a set of nodes (social actors) is given, and all ties (links) between these nodes are known - except perhaps for a moderate amount of missing data.
SIENA is designed for analyzing various types of data as dependent variables:

Longitudinal network data:
This refers to repeated measures of networks on a given node set (although it is allowed that there are some changes in the node set). Models can be specified with actor-oriented as well as tie-oriented dynamics; but mainly the former.

Longitudinal data of networks and behavior:
This is like longitudinal network data, but in addition there are one or more changing nodal variables that are also treated as dependent variables, and referred to as *behavior*. The network will influence the dynamics of the behavior, and the behavior will influence the dynamics of the network. In other words, this is about the co-evolution of networks and behavior.

Cross-sectional network data

SAOM Assumptions

- Actors control their outgoing ties and have full knowledge of the broader network
- Changes in network ties and actor behavior happen in continuous time
- Continuous time is modeled as a series of microsteps
- During a microstep, a randomly chosen actor can change only one of its ties or the value of its behavior variable
- Tie change only depends on the present network

SAOM Broadstrokes

- The simulation starts out at the network observed at the first time point t_0
  - Examples: students in the same school
- An actor is chosen randomly using a rate function
- The identified actor gets the opportunity to set a micro step. The actor's choice is determined by their objective function
- Model time is updated and simulation proceeds at step 2
- The simulation terminates once modified network resembles network at t_1

TABLE 1
Schematic Overview of the Model Components

	Occurrence	Rule of Change
Network changes	Network rate function	Network objective function
Behavioral changes	Behavioral rate function	Behavioral objective function

Rate Functions

- *Micro steps* are opportunities for a randomly chosen actor to change one of his/her outgoing ties or behavior.
- The frequency of these opportunities for change is modeled using *rate functions*, one for each type of change (network, behavior).
- Waiting time between steps follows an exponential distribution with parameter $\lambda_t N$ (N refers to number of actors in the network)
    - Values of λ_t are estimated by calculating the number of edge differences between networks:
    - The higher λ_t is the greater the number of changes between observation moments
- Probability that an actor i has the opportunity to make a change is equal to $1/N$

Actor's Objective Function

- Micro steps can be of two kinds: those that involve network changes or behavior changes
- For network changes, the micro step consists of the change of one tie by a given actor.
- Suppose $\mathbf{x}$ is the current network, and actor $\mathbf{i}$ has an opportunity to make a network change.
- The next network $\mathbf{x}'$ then must either equal to $\mathbf{x}$ or deviate from $\mathbf{x}$ by just one row element.
- Hence, there are $\mathbf{N}$ possible outcomes, and $\mathbf{i}$ chooses the one that maximizes his/her utility function, called *objective function,*

$$f_i^{net}(\mathbf{x}, \mathbf{x\prime}, \mathbf{z}) + \epsilon_i^{net}(\mathbf{x}, \mathbf{x\prime}, \mathbf{z}),$$

  - where $\mathbf{z}$ is the current vector of behavior scores, f^{net} is a measure of actor's satisfaction with the result of the network decision, and ϵ^{net} is the random component.

Multinomial Choice Model

- The choice probabilities can be expressed in multinomial logit shape:

$$\exp\left(\mathbf{f}_{\mathbf{i}}^{\text{net}}(\mathbf{x}', \mathbf{z})\right) \Big/ \sum_{\mathbf{x}''} \exp\left(\mathbf{f}_{\mathbf{i}}^{\text{net}}(\mathbf{x}'', \mathbf{z})\right),$$

- where the sum in the denominator extends over all possible next network states $\mathbf{x}''$

Behavior Micro Steps

- A given actor increments or decrements his score on the behavioral variable by one unit, provided that this change does not step outside the range of this variable. The score can also stay the same.
- Let $\mathbf{z}$ denote the current vector of behavior scores for all actors, and $\mathbf{z}'$ is the vector resulting from the actor's action in the micro step.
- Actor $\mathbf{i}$ makes the choice by maximizing their *behavior objective function*,

$$f_i^{beh}(\mathbf{x}, \mathbf{z}, \mathbf{z}') + \epsilon_i^{beh}(\mathbf{x}, \mathbf{z}, \mathbf{z}'),$$

  - where f_i^{beh} models the actor's satisfaction with the result of the behavior decision, and ϵ^{beh} is the random error.
- The behavior choice probability can be expressed as:

$$\exp\left(\mathbf{f}_\mathbf{i}^{\mathrm{beh}}(x, z')\right) \Big/ \sum_{z''} \exp\left(\mathbf{f}_\mathbf{i}^{\mathrm{beh}}(\mathbf{x}, \mathbf{z}'')\right),$$

Parameterization

For network changes, the objective function has the general shape

$$f_i^{net}(\mathbf{x}, \mathbf{x}', \mathbf{z}) = \sum_h \beta_h^{net} \mathbf{s}_h^{net}(i, \mathbf{x}, \mathbf{x}', z),$$

- where statistics s_h^{net} are the effects, weighted by parameters β_h^{net}.

For behavior changes:

$$f_i^{beh}(\mathbf{x}, \mathbf{z}, \mathbf{z}') = \sum_h \beta_h^{beh} s_h^{beh}(\mathbf{i}, \mathbf{x}, \mathbf{z}, \mathbf{z}'),$$

- Can specify exogenous and endogenous (network) effects.

Estimation

- A stochastic simulation algorithm that generates network and behavioral data according to the postulated dynamic process.
- Start with some network--behavior configuration, **x**(t), **z**(t).
- Draw a waiting time and increment the parameter **t** by this waiting time.
- Determine whether the next event is a network or a behavior change and select an actor who is making this change.
- Determine the action of the selected actor.
- Iterate until the end of the period is reached and evaluate the resulting simulated network--behavior configuration.

Example: Friendship Networks

```
library(RSiena)
    friend.data.w1 <- s501
    friend.data.w2 <- s502
    friend.data.w3 <- s503
    drink <- s50a
    smoke <- s50s
```

Specify the Network DV:

```r
friendship <- sienaDependent(
                     array( c( friend.data.w1, friend.data.w2, frienc
                     dim = c( 50, 50, 3 ) ) )
```

```r
friendship
```

```
## Type         oneMode
## Observations 3
## Nodeset      Actors (50 elements)
```

```r
class(friendship)
```

```
## [1] "sienaDependent"
```

```r
dim( friendship)
```

```
## [1] 50 50  3
```

```r
attributes(friendship)
```

```
## $dim
## [1] 50 50  3
##
## $class
## [1] "sienaDependent"
##
## $type
## [1] "oneMode"
##
## $sparse
## [1] FALSE
##
## $nodeSet
## [1] "Actors"
##
```

Specify the Behavior DV:

```r
drinkingbeh <- sienaDependent( drink, type = "behavior" )
drinkingbeh
```

```
## Type         behavior
## Observations 3
## Nodeset      Actors (50 elements)
```

Specify IVs:

```
smoke1 <- coCovar( smoke[ , 1 ] )

# Put the variables together in the data set for analysis
NBdata <- sienaDataCreate( friendship, smoke1, drinkingbeh)
NBdata
```

```
## Dependent variables:  friendship, drinkingbeh
## Number of observations: 3
##
## Nodeset                          Actors
## Number of nodes                      50
##
## Dependent variable friendship
## Type               oneMode
## Observations       3
## Nodeset            Actors
## Densities          0.046 0.047 0.05
##
## Dependent variable drinkingbeh
## Type               behavior
## Observations       3
## Nodeset            Actors
## Range              1 - 5
```

Possible Types of IVs

- `coCovar`--constant node-level covariate (does not change between time periods)
- `varCovar`--time-variable node-level covariate
- `coDyadCovar`--constant edge-level covariate
- `varDyadCovar`--time-varying edge-level covariate
- `sienaCompositionChange`--over time changes in node set (e.g., some actors leave the network)

```
?coCovar
```

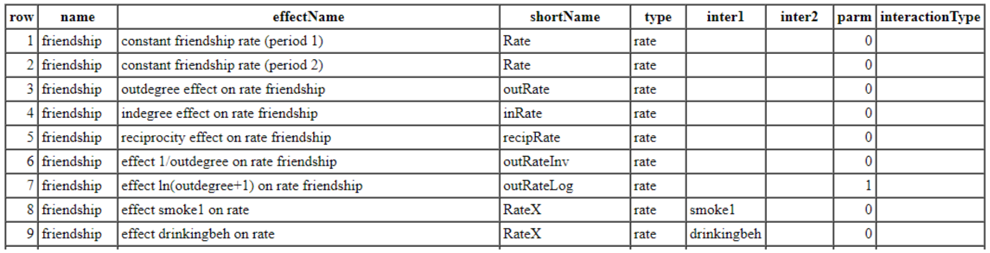
Specify Endogenous Effects

```
NBeff <- getEffects( NBdata )
NBeff
```

```
##   name        effectName                          include fix   test
## 1 friendship  constant friendship rate (period 1) TRUE    FALSE FALSE
## 2 friendship  constant friendship rate (period 2) TRUE    FALSE FALSE
## 3 friendship  outdegree (density)                 TRUE    FALSE FALSE
## 4 friendship  reciprocity                         TRUE    FALSE FALSE
## 5 drinkingbeh rate drinkingbeh (period 1)         TRUE    FALSE FALSE
## 6 drinkingbeh rate drinkingbeh (period 2)         TRUE    FALSE FALSE
## 7 drinkingbeh drinkingbeh linear shape            TRUE    FALSE FALSE
## 8 drinkingbeh drinkingbeh quadratic shape         TRUE    FALSE FALSE
##   initialValue parm
## 1    4.69604   0
## 2    4.32885   0
## 3   -1.46770   0
## 4    0.00000   0
## 5    0.70571   0
## 6    0.84939   0
## 7    0.32237   0
## 8    0.00000   0
```

Effects Description

```
effectsDocumentation(NBeff)
```



row	name	effectName	shortName	type	inter1	inter2	parm	interactionType
1	friendship	constant friendship rate (period 1)	Rate	rate			0	
2	friendship	constant friendship rate (period 2)	Rate	rate			0	
3	friendship	outdegree effect on rate friendship	outRate	rate			0	
4	friendship	indegree effect on rate friendship	inRate	rate			0	
5	friendship	reciprocity effect on rate friendship	recipRate	rate			0	
6	friendship	effect 1/outdegree on rate friendship	outRateInv	rate			0	
7	friendship	effect ln(outdegree+1) on rate friendship	outRateLog	rate			1	
8	friendship	effect smoke1 on rate	RateX	rate	smoke1		0	
9	friendship	effect drinkingbeh on rate	RateX	rate	drinkingbeh		0	

Specify Effects

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

```
##   effectName                 include fix   test  initialValue parm
## 1 transitive triplets        TRUE    FALSE FALSE          0    0
## 2 transitive recipr. triplets TRUE    FALSE FALSE          0    0
```

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

```
##   effectName           include fix   test  initialValue parm
## 1 smoke1 alter         TRUE    FALSE FALSE          0    0
## 2 smoke1 squared alter TRUE    FALSE FALSE          0    0
## 3 smoke1 ego           TRUE    FALSE FALSE          0    0
## 4 smoke1 diff. squared TRUE    FALSE FALSE          0    0
```

```
NBeff
```

```
##    name        effectName                          include fix   test
## 1  friendship  constant friendship rate (period 1) TRUE    FALSE FALSE
## 2  friendship  constant friendship rate (period 2) TRUE    FALSE FALSE
## 3  friendship  outdegree (density)                 TRUE    FALSE FALSE
## 4  friendship  reciprocity                         TRUE    FALSE FALSE
```

Define the Model:

```
myalgorithm1 <- sienaAlgorithmCreate( projname = 's50_NB' )

# Estimate using the second algorithm right from the start.
NBans <- siena07(myalgorithm1, data = NBdata, effects = NBeff)
NBans <- siena07(myalgorithm1, data = NBdata, effects = NBeff, batch=
```

Look at results

```
NBans
```

```
## Estimates, standard errors and convergence t-ratios
##
##                                                 Estimate   Standard   Conve
##                                                              Error      t-r
## Network Dynamics
##    1. rate constant friendship rate (period 1)  6.2738  ( 1.2186   )    0.
##    2. rate constant friendship rate (period 2)  5.0845  ( 0.8572   )   -0.
##    3. eval outdegree (density)                 -2.6246  ( 0.2222   )    0.
##    4. eval reciprocity                          2.7737  ( 0.2664   )    0.
##    5. eval transitive triplets                  0.8915  ( 0.1360   )    0.
##    6. eval transitive recipr. triplets         -0.5130  ( 0.2160   )    0.
##    7. eval smoke1 alter                         0.2548  ( 0.2914   )   -0.
##    8. eval smoke1 squared alter                -0.2197  ( 0.2489   )    0.
##    9. eval smoke1 ego                           0.0873  ( 0.3034   )    0.
##   10. eval smoke1 squared ego                   0.0104  ( 0.2501   )    0.
##   11. eval smoke1 diff. squared                -0.0981  ( 0.0679   )   -0.
##
## Behavior Dynamics
##   12. rate rate drinkingbeh (period 1)          1.1712  ( 0.3042   )   -0.
##   13. rate rate drinkingbeh (period 2)          1.6518  ( 0.4074   )    0.
##   14. eval drinkingbeh linear shape             0.3664  ( 0.1425   )    0.
```

Interpretation of Parameters

- The parameters on the rate functions are interpreted as related to the speed of the evolution process. These parameters reflect the frequencies of the opportunities for change.
- The parameters on the objective functions may be interpreted in terms of their direction and statistical significance or by calculating predicted probabilities of ties in hypothetical scenarios (i.e., use the reverse logistic transformation).
- The linear and the quadratic shape parameters of the behavior objective function model the shape of the long-term distribution of the behavior variable.
- Can also use the reverse logit transformation to interpret the parameters on the behavior effects.

Example

Suppose we estimated the following model:

$$f_i^{net} = -2\sum_j \mathbf{x}'_{ij} + 2.5\sum_j \mathbf{x}'_{ij}\mathbf{x}_{ji} + 1\sum_j \mathbf{x}'_{ij}\mathrm{sim}_i j$$

and

$$f_i^{beh}(\mathbf{x},\mathbf{z},\mathbf{z}') = -1(\mathbf{z}'_i - \mathbf{z}) - .5(\mathbf{z}'_i - \mathbf{z})^2 + 2.5(\sum_j \mathbf{x}_{ij}\mathrm{sim}'_{ij})/(\sum_j x_{ij})$$

The primes indicate the elements in the formulae that are under control of actor **i** and may be changed in a micro step.

- Note that the network objective function contains outdegree, reciprocity, and the similarity effect, and the behavior function includes the linear and quadratic shape of the distribution of the behavior variable and an effect of average similarity to neighbors.

Interpretation of Parameters: Network Function

- The parameter estimate of -2 on the outdegree tells us that, if we ignore all other variables, the baseline probability of a tie in this network is $\exp{(-2)}/(1+\exp{(-2)}) = 0.12$.

- The parameter estimate of 2.5 on the reciprocity tells us that the probability of a reciprocated tie is $\exp{(-2+2.5)}/(1+\exp{(-2+2.5)}) = 0.62$.

- The positive similarity effect indicates that actors tend to form ties with similar others rather than to dissimilar ones. The probability of a tie to a maximally similar actor is $\exp{(-2+1)}/(1+\exp{(-2+1)}) = 0.26$ vs that of the baseline probability of 0.12.

Interpretation of Parameters: Behavior Function

- Suppose our behavior variable ranges from 1 to 5 and has an average of 3. When mean centered, this variable then ranges from -2 to 2.
- The first two parameters define a parabolic shape. We can calculate the inflection point of this parabola by solving $-1 - 0.5 * 2x = 0$ for x. The result is x=-1, which corresponds to the value of 2 on the original scale of this variable. This means that in the long run, the distribution of the behavior variable is unimodal with a maximum at score value 2.
- Can also calculate the probability of moving from one score to the other. For example, the probability of moving from 3 to 2 is

$$\begin{aligned}&\exp(-1(2-3) - 0.5(2-3)^2)/(1+\exp(-1(2-3) - 0.5(2-3)^2)) \\ &\quad = 0.62\end{aligned}$$

The probability of staying at 3 rather than moving to 2 is 1-0.62=0.38.

Interpretation of Parameters: Behavior Function Cont'd

- The third parameter of 2.5 on the average similarity indicates that actors tend to act in the same manner as their friends.
- Consider an actor with a 3 on their behavior and assume he has five friends, four of whom score 3 and 1 scoring 2 or lower.
- If this actor moves to 2, his average similarity to four of his friends will decrease by -1*4 and his similarity to his fifth friend will increase by 1.
- The average similarity will decrease from 1/5 to 4/5 for a change of -0.6. Then, accounting for the similarity effect, the probability of moving from 3 to 2 for this actor is

$$\begin{aligned}&\exp(-1(2-3)-0.5(2-3)^2\\&+2.5(-0.6))/(1+\exp(-1(2-3)-0.5(2-3)^2\\&+2.5(-0.6))=0.26\end{aligned}$$

Let's Specify a Model That...

1. accounts for homophily effect for smoking

```
NBeff <- getEffects( NBdata )

NBeff <- includeEffects(NBeff, simX,
         interaction1 = "smoke1" )
```

```
##   effectName        include fix   test  initialValue parm
## 1 smoke1 similarity TRUE    FALSE FALSE            0    0
```

Let's Specify a Model That...

1. Parses out whether there is a selection or influence (or both) effect for drinking behavior--- include sender, receiver and homophily effects of drinking for friendship formation, and vice versa.

```
NBeff <- includeEffects(NBeff, egoX, altX, simX,
            interaction1 = "drinkingbeh" )
```

```
##   effectName             include fix   test  initialValue parm
## 1 drinkingbeh alter      TRUE    FALSE FALSE            0    0
## 2 drinkingbeh ego        TRUE    FALSE FALSE            0    0
## 3 drinkingbeh similarity TRUE    FALSE FALSE            0    0
```

```
NBeff <- includeEffects(NBeff, name = "drinkingbeh",
      avAlt,indeg,outdeg,
      interaction1 = "friendship" )
```

```
##   effectName                include fix   test  initialValue parm
## 1 drinkingbeh indegree      TRUE    FALSE FALSE            0    0
## 2 drinkingbeh outdegree     TRUE    FALSE FALSE            0    0
## 3 drinkingbeh average alter TRUE    FALSE FALSE            0    0
```

Results

```
## Estimates, standard errors and convergence t-ratios
##
##                                               Estimate   Standard   Conve
##                                                            Error      t-r
## Network Dynamics
##    1. rate constant friendship rate (period 1)  5.9052  ( 0.9341   )   -0.
##    2. rate constant friendship rate (period 2)  4.5123  ( 0.7046   )   -0.
##    3. eval outdegree (density)                 -2.4804  ( 0.1415   )    0.
##    4. eval reciprocity                          2.7742  ( 0.2659   )    0.
##    5. eval smoke1 similarity                    0.2967  ( 0.2100   )    0.
##    6. eval drinkingbeh alter                   -0.0144  ( 0.1147   )   -0.
##    7. eval drinkingbeh ego                      0.1293  ( 0.1177   )    0.
##    8. eval drinkingbeh similarity               1.3202  ( 0.6778   )   -0.
##
## Behavior Dynamics
##    9. rate rate drinkingbeh (period 1)          1.2619  ( 0.3548   )    0.
##   10. rate rate drinkingbeh (period 2)          1.7245  ( 0.4400   )    0.
##   11. eval drinkingbeh linear shape            -1.2946  ( 2.8020   )    0.
##   12. eval drinkingbeh quadratic shape         -1.5840  ( 2.6522   )   -0.
##   13. eval drinkingbeh indegree                -0.8568  ( 2.1755   )    0.
##   14. eval drinkingbeh outdegree                1.7257  ( 3.6961   )    0.
##   15. eval drinkingbeh average alter            3.3212  ( 5.7404   )   -0.
##
## Overall maximum convergence ratio:    0.1387
```

Convergence Ratios

- Note that the "convergence t-ratio" is the t-ratio for convergence checking, not the t statistic for testing the significance of this effect (See Section 6.3 of the RSiena manual).
- For good convergence, the t-ratios for convergence all should be less than .1 in absolute value, and the overall maximum convergence ratio should be less than 0.25.
- If this is not yet the case, you should try again, starting from the last estimate as the previous answer (`prevAns` argument), e.g.:

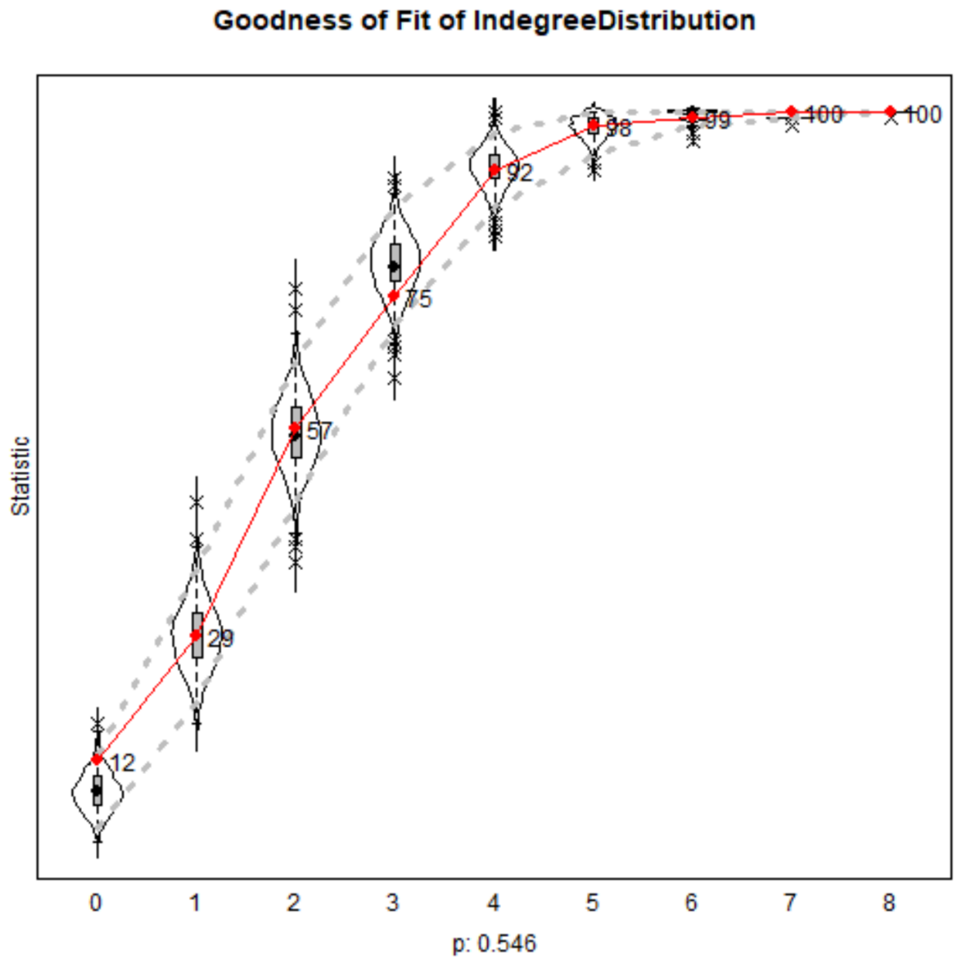
```
m3 <- siena07(myalgorithm2, data = NBdata,
      effects = NBeff, prevAns = m2 )
```

Goodness of Fit: Indegree

- Can plot distributions of network statistics from simulated networks
- Note: must specify `returnDeps=TRUE` in the `siena07` function to save these networks.

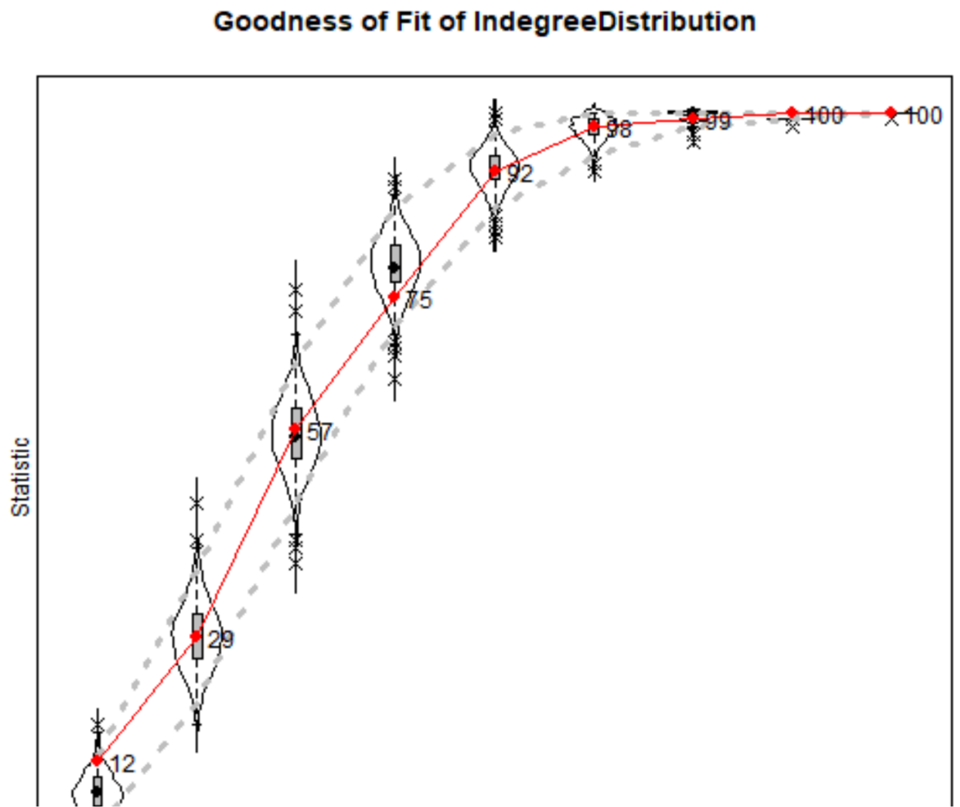
```
myalgorithm2 <- sienaAlgorithmCreate(projname = 's50CoEv_2' )
m2 <- siena07(myalgorithm2, data = NBdata,
      effects = NBeff, batch=TRUE, returnDeps=TRUE)
gofI <- sienaGOF(m2, IndegreeDistribution, verbose=TRUE, join=TRUE,
     varName="friendship")
plot(gofI)
saveRDS(gofI,"./data/gofI")
```

Goodness of Fit: Indegree



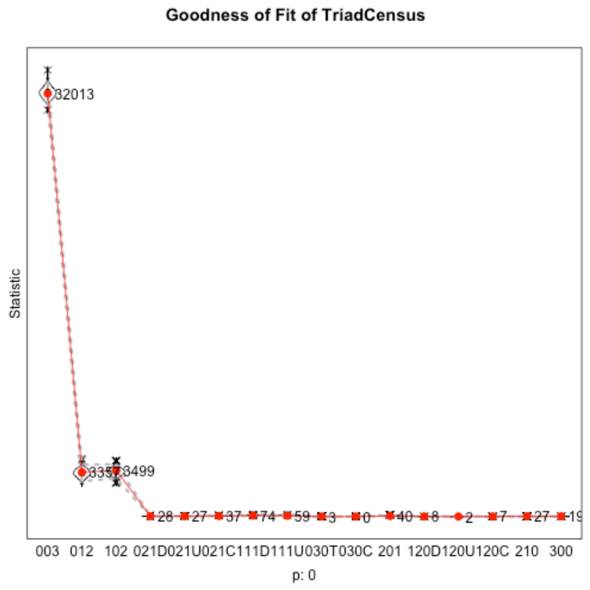
Goodness of Fit: Outdegree

```r
gofO <- sienaGOF(m2, OutdegreeDistribution, verbose=TRUE, join=TRUE,
     varName="friendship")
plot(gofO)
saveRDS(gofO,"data/gofO")
```



Goodness of Fit: Triad Census

```r
gofTC <- sienaGOF(m2, TriadCensus, verbose=TRUE, join=TRUE,
     varName="friendship")
plot(gofTC)
saveRDS(gofTC,"data/gofTC")
```



Example: Knecht Data

```r
library(btergm)
data(knecht)
```

- Longitudinal classroom friendship network and behavior (Andrea Knecht)
- `friendship`--a list of adjacency matrices at four time periods
- `demographics`--node-level covariates: sex, age, ethnicity, religion
- `delinquency`--number of delinquencies

Actors Entering and Exiting the Network

- Treatment by structural zero coding
- Treatment by composition change directives
- What to use?
- Loosely related to structural zeros: structural ones

Treatment by structural zero coding

- When actors are not part of the group at a given measurement point, code their outgoing and incoming ties as “10”, meaning “absent, and could not possibly have been present”.
- When running simulations, this is handled as follows:
    - A tie value “10” at the beginning of a period implies that the tie will remain structurally absent throughout the period, no matter what the tie’s value at the end of the period is.
    - A tie value “10” at the end of a period implies that no matter what the tie’s simulated value at the end of the period is, it is overwritten by “10” before any statistics are evaluated.
- See RSiena manual section 4.1.2.

Treatment by structural zero coding

- When information is known about the exact time when actors left or entered the group in continuous time between observation moments, this information can be made use of.
- In simulations, joiners enter at the indicated time point and then are simultaneously connected to the rest of the actors according to the data provided for the period begin (so, they do not necessarily have to ‘start from scratch’ but can inherit ties!)
- Leavers just exit and cannot change their ties any more from this time point on; their last connection data can be provided for the period end.
- Joiner and leaver data need to be provided in an additional file; see RSiena manual sections 2.1.2 and 4.7.

What to Use

- Composition change directives allow to make use of more information. If information is scarce, this may be the better option.
- Structural zero treatment is quite crude, if results can be obtained this way, they will likely be robust. But under scarce information conditions, it can happen that no results can be obtained.

Structural Ones

- Sometimes, ties can be “present, and could not possibly have been absent”.
    - Studying a communication network among employees, where Loosely related to structural zeros:
    - Studying a communication network among employees, where some people are forced to communicate anyway (by their job contract).
- Studying a growing network where ties once formed cannot be dissolved again.
- In such situations, tie variables can be coded as “11”.
- See RSiena manual section 4.1.2.

Example: Duque Data

```r
library(devtools)
#install_github("ochyzh/networkdata")
data("duqueData")
dim(dipl_ties[[1]])
```

```
## [1] 134 134
```

```r
dim(dipl_ties[[2]])
```

```
## [1] 148 148
```

- Remember that in these data, time periods have varying numbers of observations, as states enter and leave the system.
- In order to use RSiena, we must have the same number of actors in each time period. If an actor is missing, their tie values are coded using the structural zero code `10`. Alternatively, you can provide an additional file that details when actors enter and leave the data.

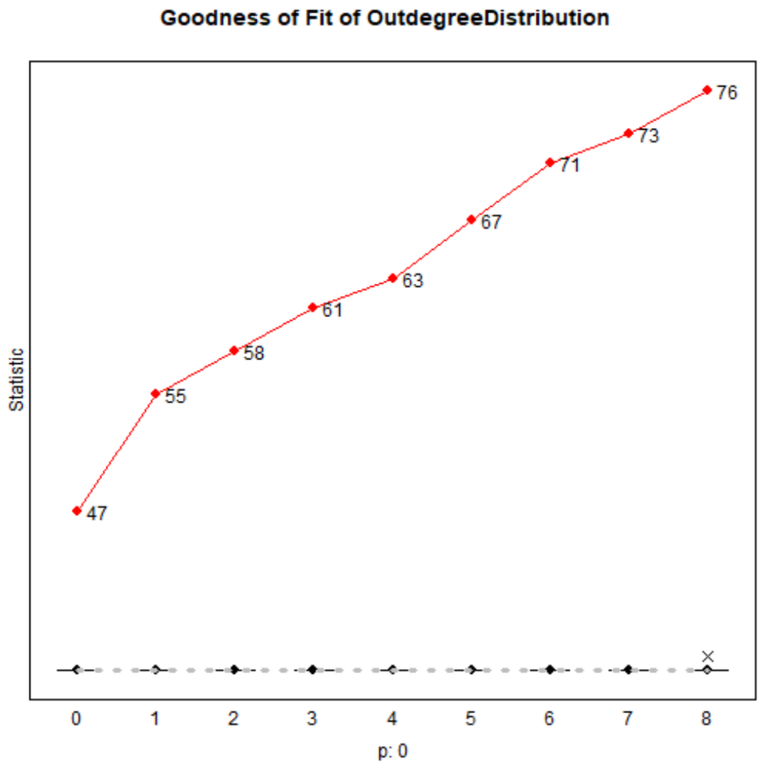
Example: Duque Data

```
library(tidyverse)
#get the full list of actors:
myactors<-sort(as.numeric(unique(do.call("c",lapply(dipl_ties[1:3],na
n<-length(myactors)

dipl<-array(10, dim = c(n,n, 3 ),
    dimnames=list(myactors,myactors,seq(from=1970,to=1980,by=5)))
dipl[1:10,1:10,1]

for(t in 1:3){
    d<-dipl_ties[[t]]
    for(i in 1:nrow(d)){
      for (j in 1:ncol(d)){
        a1 = names(d)[i]
        a2 = colnames(d)[j]
        val = as.numeric(as.character(d[i,j]))
        dipl[i,j,t] <- val
        dipl[j,i,t] <- val
      }}
    }

dipl <- sienaDependent(dipl)
```

Results

```
## Estimates, standard errors and convergence t-ratios
##
##                                     Estimate   Standard   Convergence
##                                                  Error      t-ratio
##
## Rate parameters:
##   0.1      Rate parameter period 1 59.0072  ( 3.7459   )
##   0.2      Rate parameter period 2 55.7625  ( 2.8318   )
##
## Other parameters:
##   1.  eval degree (density)        -0.3857  ( 0.3191   )   -0.0117
##   2.  eval transitive ties         -0.2815  ( 0.3132   )   -0.0083
##   3.  eval cont                     0.6214  ( 0.0600   )   -0.0381
##   4.  eval ally                     0.9287  ( 0.0347   )    0.0433
##   5.  eval dem similarity          -0.3061  ( 0.0221   )    0.0352
##
## Overall maximum convergence ratio:    0.1007
##
##
## Total of 2267 iteration steps.
##
## Covariance matrix of estimates (correlations below diagonal)
##
##        0.102       -0.100       -0.001        0.000       -0.002
```

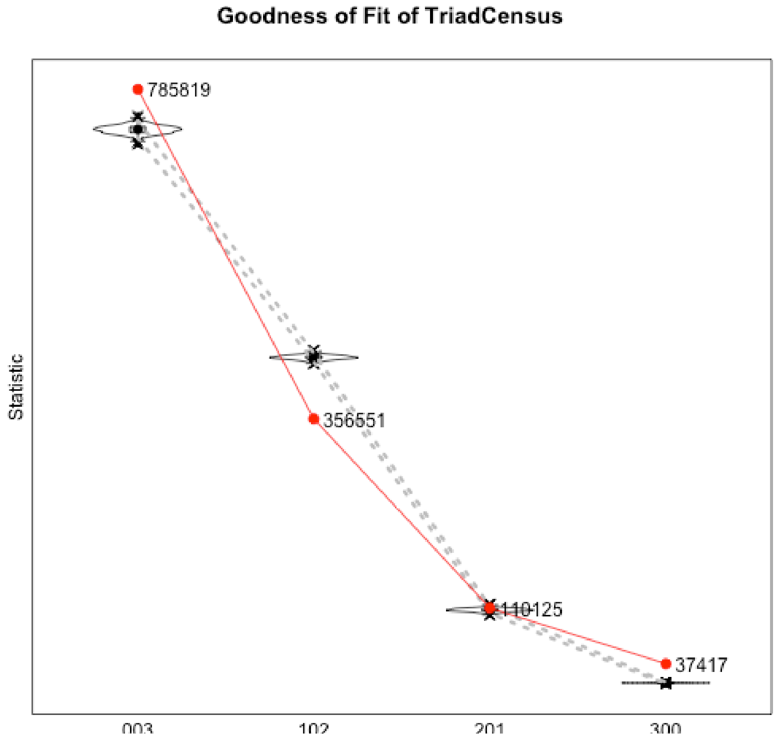
Goodness of Fit: Outdegree



Goodness of Fit: Triad Census

```
gofTC_dipl <- sienaGOF(ans, TriadCensus, verbose=TRUE, join=TRUE,
     varName="dipl")
plot(gofTC_dipl)
```

```
## Note: some statistics are not plotted because their variance is 0.
## This holds for the statistics: 012 021D 021U 021C 111D 111U 030T 030C 120D
```



TERGM vs. SAOM

- Block et al. 2017
- Block et al. 2018
- Leifeld & Cranmer 2018