Siena Advanced Users' Meeting 2016

Tom A.B. Snijders

University of Oxford
University of Groningen

Zürich, February 2016

Overview

1. Where to look?

Siena is an evolving endeavour, which may be hard to follow.

1. Where to look?

Siena is an evolving endeavour, which may be hard to follow.

- Follow the Siena/Stocnet discussion list!
- The website
  `http://www.stats.ox.ac.uk/~snijders/siena/`
  notes important matters at the '*News*' page:
  list of incompatibilities and bugs;
  new developments; some interesting papers.
- Most recent versions can be downloaded from
  R-Forge and '*Downloads*' page of website,
  and are announced at the Siena/Stocnet discussion list.
- Website '*News*' page, and Appendix B in the manual,
  give description of changes in the new versions.

Where to look? (2)

- Website '*Literature*' page has a section 'Presentations (teaching material)' including (e.g.) these slides.
- Recent (since late 2014) changes in manual:
  ⇒ elementary effects (treated below);
  ⇒ more about user-defined interaction effects;
  ⇒ changed section about convergence and how to use the algorithm options.
- Siena_algorithms.pdf now is at the Siena website (partial explanation of algorithms and code).
- The available effects of 'myeff' are given by effectsDocumentation(myeff).

2. New convergence criterion

The usual convergence criterion is tmax
the absolute maximum of the *t*-ratios for convergence,
considering simultaneously all parameters in the model.

It has appeared that for some models
(e.g., with non-centered actor covariates)
the usual criterion

$$\text{tmax} \leq 0.10$$

is not sufficient.

Therefore, the overall maximum convergence ratio
(included as tconv.max in *sienaFit* objects since some time)
gets a new importance.

2a. Overall maximum convergence ratio

This is defined as the maximum t-ratio for convergence for any linear combination of the parameters,

$$\text{tconv.max} = \max_b \left\{ \frac{b'\left(\bar{s}_j - s^{\text{obs}}\right)}{\sqrt{b'\Sigma b}} \right\} .$$

This is equal to (use Cauchy-Schwarz inequality)

$$\max_c \left\{ \frac{c'\Sigma^{-1/2}\left(\bar{s}_j - s^{\text{obs}}\right)}{\sqrt{c'c}} \right\} = \sqrt{\left(\bar{s}_j - s^{\text{obs}}\right)'\Sigma^{-1}\left(\bar{s}_j - s^{\text{obs}}\right)} .$$

The definition implies that

$$\text{tconv.max} \geq \text{tmax} .$$

Studies comparing results of siena07() with the 'true estimate' (robust mean of many estimations) show:

1. Distance from true estimate is much better indicated by tconv.max than by tmax.
2. When tconv.max exceeds 0.30, distances d_i from the true value are too large.

New criterion

$$\text{tmax} \leq 0.10 \ \underline{\text{and}} \ \text{tconv.max} \leq 0.25 \ .$$

Further options for siena07()

To improve the possibilities of `siena07()` to indeed produce estimates satisfying this new criterion,
some new options were developed since version 1.1-285 (2015-05-20);
see ?sienaAlgorithmCreate and manual, Section 6.1.3;
also see Siena_algorithms.pdf.

Since version 1.1-289 (2015-09-10), new defaults for MoM:

1. doubleAveraging = 0
   (i.e., use double averaging right from subphase 1)
2. diagonalize = 0.2

What is double averaging?

The regular Robbins-Monro update step is

$$\hat{\theta}_{N+1} = \hat{\theta}_N - a_N \tilde{D}^{-1} (S_N - s)$$

The algorithm with double averaging is

$$\hat{\theta}_{N+1} = \bar{\theta}_N - N a_N \tilde{D}^{-1} (\bar{S}_N - s) \, ,$$

where

$$\bar{\theta}_N = \frac{1}{N} \sum_{n \leq N} \hat{\theta}_n \, , \quad \bar{S}_N = \frac{1}{N} \sum_{n \leq N} s_n \, .$$

See Siena_algorithms.pdf.

Achieving this more stringent convergence criterion may require several repeated runs of siena07() linked by using the *prevAns* parameter.

The following page is an extension of siena07() like in the manual Section 6.1.3.

```
siena07ToConvergence <- function(alg, dat, eff, ans0=NULL,
                                 threshold, ...){
 numr <- 0
 ans <- siena07(alg, data=dat, effects=eff, prevAns=ans0, ...) #
 repeat {
   save(ans, file=paste("ans",numr,".RData",sep="")) # to be safe
   numr <- numr+1            # count number of repeated runs
   tm <- ans$tconv.max       # convergence indicator
   cat(numr, tm,"\n")        # report how far we are
   if (tm < threshold) {break}   # success
   if (tm > 10) {break}      # divergence without much hope
                             # of good return
   if (numr > 100) {break}   # now it has lasted too long
   ans <- siena07(alg, data=dat, effects=eff, prevAns=ans, ...)
   }
 ans
}
```

Results for a moderately complicated data set & model with a low `threshold=0.1` :

```
vdb.algo4 <- sienaAlgorithmCreate(seed=54321, nsub=4)

ans012 <- siena07ToConvergence(vdb.algo4,
     vdb.data012, vdb.eff012, threshold=0.1,
     useCluster=TRUE, nbrNodes=2)
```

(note the use of the dots ... parameter)

1 0.1979708	9 0.1471743	17 0.1157999	25 0.1904938
2 0.2461148	10 0.134066	18 0.10688	26 0.1478382
3 0.1748373	11 0.1402179	19 0.1368976	27 0.1509909
4 0.1445431	12 0.1298273	20 0.1147648	28 0.1306809
5 0.1442089	13 0.1447915	21 0.1453033	29 0.157657
6 0.133533	14 0.1548974	22 0.1220962	30 0.1727815
7 0.1211533	15 0.1522713	23 0.1181492	31 0.09992554
8 0.1203132	16 0.1177088	24 0.2007752	

With 5 subphases it goes more quickly:

```
vdb.algo5 <- sienaAlgorithmCreate(seed=54321, nsub=5)

ans012 <- siena07ToConvergence(vdb.algo5,
     vdb.data012, vdb.eff012, threshold=0.1,
     useCluster=TRUE, nbrNodes=2)
```

1 0.1318489
2 0.1417612
3 0.0894502

And with 6:

```
vdb.algo6 <- sienaAlgorithmCreate(seed=54321, nsub=6)

ans012 <- siena07ToConvergence(vdb.algo6,
     vdb.data012, vdb.eff012, threshold=0.1,
     useCluster=TRUE, nbrNodes=2)
```

1 0.111879
2 0.08501629

Conjecture :

If the initial value (*'prevAns'*) is reasonably near the solution (say, $tconv.max < 0.4$), the successive (*'prevAns'*) values of results of the Robbins-Monro procedure of siena07() are almost independent, with the distribution of *tconv.max* having an average value depending on the length of the last subphase.

Default length of subphase k is $N_{max} = (p + 7) \times (2.52)^k$, with p = number of parameters.
This means for the length of the last phase:

k_{max}	N_{max}
4	685
5	1727
6	4353

```
algo700 <- sienaAlgorithmCreate(seed=54321,
      nsub=1, n2start=700)
ans012.0 <- siena07(vdb.algo4, data=vdb.data012,
      effects=vdb.eff012, useCluster=TRUE, nbrNodes=2)
ans012r <- siena07ToConvergence(algo700, vdb.data012,
      vdb.eff012, threshold=0.01,
      useCluster=TRUE, nbrNodes=2, ans0=ans012.0)
```

1 0.1566975	11 0.1043422
2 0.111309	12 0.1077261
3 0.1476088	13 0.1504494
4 0.1310093	14 0.1630273
5 0.1121832	15 0.1494122
6 0.1202841	16 0.09694043
7 0.1483798	17 0.09847209
8 0.08566426	18 0.1935148
9 0.117654	19 0.1862205
10 0.1345501	20 0.1186548

3 smaller than 0.10; mean 0.13.

```
algo2000 <- sienaAlgorithmCreate(seed=54321,
      nsub=1, n2start=2000)
ans012r <- siena07ToConvergence(algo2000, vdb.data012,
      vdb.eff012, threshold=0.01,
      useCluster=TRUE, nbrNodes=2, ans0=ans012.0)
```

1	0.1437418	11	0.3063748
2	0.1356098	12	0.07639885
3	0.1023418	13	0.127548
4	0.1790526	14	0.07809783
5	0.1562154	15	0.1082213
6	0.1009786	16	0.157643
7	0.1109657	17	0.08581463
8	0.09316735	18	0.07484955
9	0.08950204	19	0.1212121
10	0.1508801	20	0.1179153

6 smaller than 0.10; mean 0.13 (one outlier...)

```
algo4000 <- sienaAlgorithmCreate(seed=54321,
      nsub=1, n2start=4000)
ans012r <- siena07ToConvergence(algo4000, vdb.data012,
      vdb.eff012, threshold=0.01,
      useCluster=TRUE, nbrNodes=2, ans0=ans012.0)
```

1	0.08891164	11	0.08027072
2	0.1013756	12	0.09768049
3	0.142784	13	0.1286098
4	0.1038731	14	0.0895482
5	0.1082759	15	0.08794522
6	0.1026858	16	0.1011986
7	0.1429859	17	0.09113767
8	0.07912083	18	0.09408743
9	0.06697233	19	0.07530766
10	0.1144773	20	0.1145916

10 smaller than 0.10; mean 0.10.

```
algo10000 <- sienaAlgorithmCreate(seed=54321,
       nsub=1, n2start=10000)
 ans012r <- siena07ToConvergence(algo10000, vdb.data012,
       vdb.eff012, threshold=0.01,
       useCluster=TRUE, nbrNodes=2, ans0=ans012.0)
```

1	0.06661866	11	0.1237608
2	0.07738042	12	0.1458485
3	0.110977	13	0.1231905
4	0.0775174	14	0.09870046
5	0.07146001	15	0.06263102
6	0.099844	16	0.1059729
7	0.1162311	17	0.07406779
8	0.1294587	18	0.1081187
9	0.07587712	19	0.09084813
10	0.1355451	20	0.1250927

10 smaller than 0.10; mean 0.10.

With larger and larger number of runs for estimation,
for really large numbers of runs
the values of tconv.max do not get convincingly smaller.

What is limiting further decrease?

With larger and larger number of runs for estimation,
for really large numbers of runs
the values of tconv.max do not get convincingly smaller.

What is limiting further decrease?

The length of phase 3.

Therefore now, a series of experiments with n2start = n3.

Recall that for n2start = n3 = 700,
we had 3 smaller than 0.10; mean 0.13.

```
algo2000 <- sienaAlgorithmCreate(seed=54321,
      nsub=1, n2start=2000, n3=2000)
ans012r <- siena07ToConvergence(algo2000, vdb.data012,
      vdb.eff012, threshold=0.01,
      useCluster=TRUE, nbrNodes=2, ans0=ans012.0)
```

1 0.1154304	11 0.1346359
2 0.08335217	12 0.08379924
3 0.1090684	13 0.06819761
4 0.07606212	14 0.09366772
5 0.1121068	15 0.06833264
6 0.09108999	16 0.0710875
7 0.1344702	17 0.0623443
8 0.1006035	18 0.1140778
9 0.1010324	19 0.1103848
10 0.1177485	20 0.08337979

10 smaller than 0.10; mean 0.10.

```
algo4000 <- sienaAlgorithmCreate(seed=54321,
      nsub=1, n2start=4000, n3=4000)
ans012r <- siena07ToConvergence(algo4000, vdb.data012,
      vdb.eff012, threshold=0.01,
      useCluster=TRUE, nbrNodes=2, ans0=ans012.0)
```

1	0.07859669	11	0.06140544
2	0.04700076	12	0.0761522
3	0.08937357	13	0.05954446
4	0.05405991	14	0.07722966
5	0.06668956	15	0.07653949
6	0.06828561	16	0.09723554
7	0.07328622	17	0.04959409
8	0.07626721	18	0.09825669
9	0.05873874	19	0.0545597
10	0.06239783	20	0.08574505

all smaller than 0.10; mean 0.07.

```
algo10000 <- sienaAlgorithmCreate(seed=54321,
       nsub=1, n2start=10000, n3=10000)
 ans012r <- siena07ToConvergence(algo10000, vdb.data012,
       vdb.eff012, threshold=0.01,
       useCluster=TRUE, nbrNodes=2, ans0=ans012.0)
```

1	0.04208639	11	0.04782997
2	0.03915605	12	0.03918571
3	0.04588603	13	0.05879899
4	0.05221235	14	0.05735232
5	0.04504675	15	0.05569969
6	0.04463104	16	0.04880436
7	0.07108124	17	0.04088348
8	0.06551668	18	0.07608058
9	0.07060039	19	0.06220432
10	0.06158192	20	0.04426156

all smaller than 0.08; mean 0.05.

Conclusion

If a low value of tconv.max is not easily achieved,
for getting better convergence:

⇒ Use 5 or 6 subphases;

or

⇒ Starting from a decent *prevAns*,
use an algorithm with nsub=1, n2start='large',
noting that 'large' > $(p+7) \times (2.52)^k$
with default $k = 4$;
use a smaller firstg (e.g., 0.02);

⇒ If tconv.max still too big, further increase n2start.

3. sienacpp()

RSiena has two rooms:

1. front office: user interface in R
2. back office: simulations going on in C++

In siena07(), only the simulations are done in C++;
the further calculations for the Robbins-Monro estimation algorithm are done in R.

Starting from version 1.1-290 (2016-01-31), RSienaTest contains sienacpp() which produces the same as siena07(), but with all calculations in C++.

(Some options are not yet included, e.g., multigroup data.)

Parallellization options may be different.

sienacpp() has a small efficiency advantage,
which is relatively important only for
small data sets / small amounts of total change.

4. Specification; effects

1. GWESP
2. Structural equivalence: Jaccard distances
3. Multivariate degree effects on behaviour
4. Distance-two effects
5. Elementary effects
6. Influence effects
7. Influence from incoming alters
8. Miscellaneous

Effects (1): GWESP

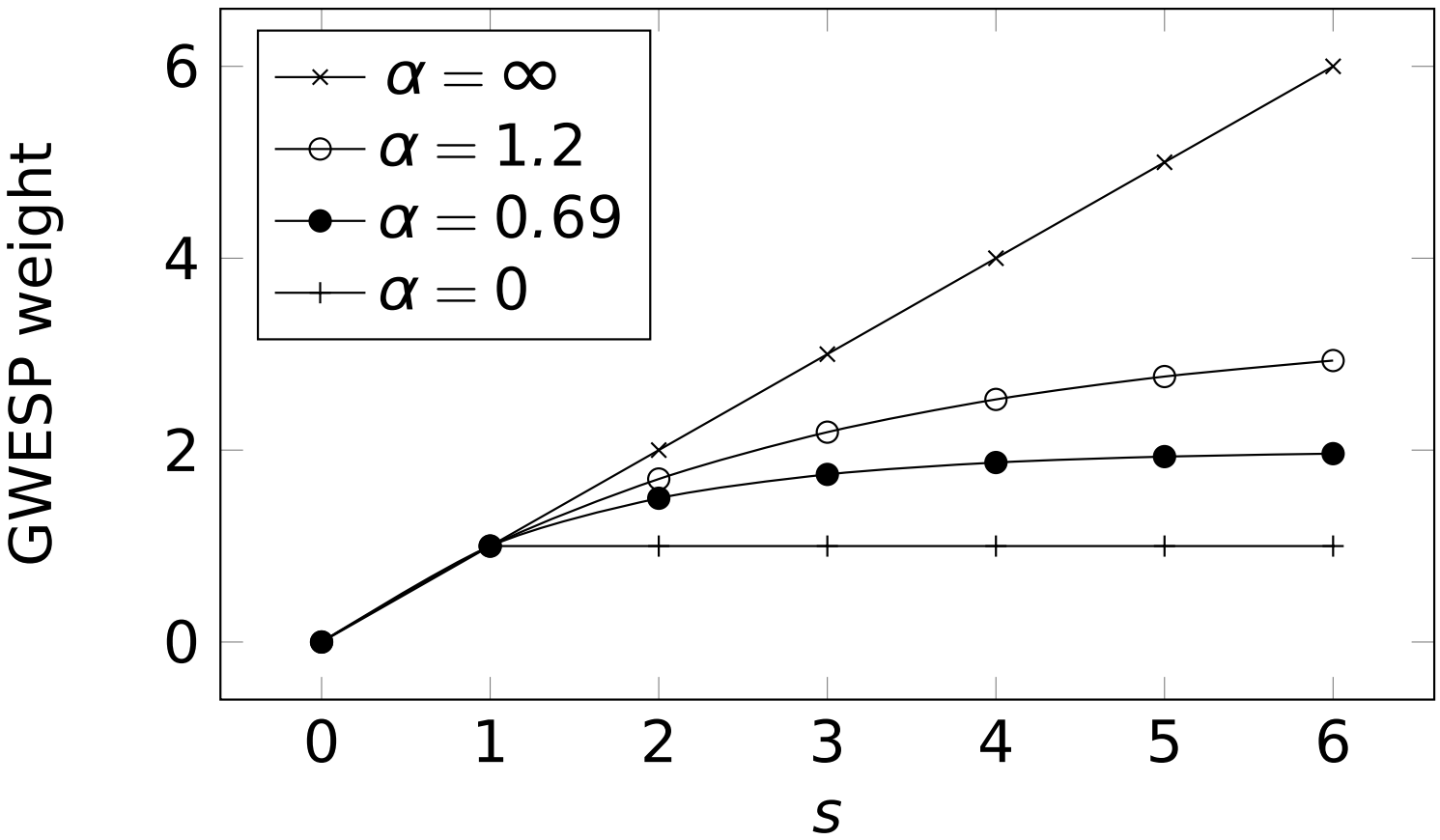
GWESP (*geometrically weighted edgewise shared partners*) (cf. ERGM) is intermediate between transTrip and transTies.

$$\text{GWESP}(i, \alpha) = \sum_j x_{ij}\, e^{\alpha} \left\{ 1 - \left(1 - e^{-\alpha}\right)^{\sum_h x_{ih} x_{hj}} \right\} .$$

for $\alpha \geq 0$ (effect parameter $= 100 \times \alpha$).

Default $\alpha = \log(2)$, parameter = 69.

GWESP (contd.)



Weight of tie $i \to j$ for $s = \sum_h x_{ih} x_{hj}$ two-paths.

The implementation of GWESP is an *elementary* effect:

For creation of a new tie,
only its role as $i \to j$ in the formula is counted,
not its role as $i \to h$.

GWESP sometimes yields better fit than transTrip or transTies.

The implementation of GWESP is an *elementary* effect:

For creation of a new tie,
only its role as $i \to j$ in the formula is counted,
not its role as $i \to h$.

GWESP sometimes yields better fit than transTrip or transTies.

The GWESP effect exists also for multivariate networks:
gwespFFMix etc.

New effects (1): Structural equivalence

A good way of expressing structural equivalence,
i.e., being connected to the same others,
is the *Jaccard similarity* between rows, or columns:

$$J_{\text{out}}(i,j) = \frac{\sum_h x_{ih}\, x_{jh}}{x_{i+} + x_{j+} - \sum_h x_{ih}\, x_{jh}}$$

$$J_{\text{in}}(i,j) = \frac{\sum_h x_{hi}\, x_{hj}}{x_{+i} + x_{+j} - \sum_h x_{hi}\, x_{hj}}$$

Based on these (by summing over the outgoing ties of i),
the effects Jout and Jin are defined.

For multivariate networks: JoutMix, JinMix.

Specification

Basic specification for the usual type of networks:

- outdegree, reciprocity
- transitive closure: gwespFF *or* transTrip *or* ...
  or perhaps Jin and Jout could do just as well?
- interactions between this and reciprocity (Per Block):
  transRecTrip, gwespFF × recip
  (possible because gwespFF is an elementary effect!)
- inPop; outAct; inAct or outPop (or ...sqrt)
- (if available) representation of meeting opportunities

New effects (2): Multivariate degree effects

Combined degrees affect behavior.

Number to whom i is tied in network X_1 and network X_2:
F = 'Forward', B = 'Backward', R = 'Reciprocal'

1. *double outdegree effect* (FFDeg),
   $s_{i1}^{\text{beh}}(x,z) = z_i \sum_j x_{1ij} x_{2ij}$;
2. *double indegree effect* (BBDeg),
   $s_{i2}^{\text{beh}}(x,z) = z_i \sum_j x_{1ji} x_{2ji}$;
3. *combined out-indegree effect* (FBDeg),
   $s_{i3}^{\text{beh}}(x,z) = z_i \sum_j x_{1ij} x_{2ji}$;
4. *combined out-reciprocated degree effect* (FRDeg),
   $s_{i4}^{\text{beh}}(x,z) = z_i \sum_j x_{1ij} x_{2ij} x_{2ji}$;
5. *combined in-reciprocated degree effect* (BRDeg),
   $s_{i5}^{\text{beh}}(x,z) = z_i \sum_j x_{1ji} x_{2ij} x_{2ji}$.

New effects (3): Influence

The triple avSim – totSim – avAlt
now is a quartet with a 2×2 structure:
{ sim , alt } × { av, tot }

totAlt was implemented for regular influence effects,
influence from reciprocated alters, and
influence from other covariates (non-dependent / exogenous).

New effects:

1. totAlt (next to avAlt, totSim, avSim)
2. totRecAlt (next to avRecAlt)
3. totXAlt (next to avXAlt, the old AltsAvAlt)

Incoming influence effects

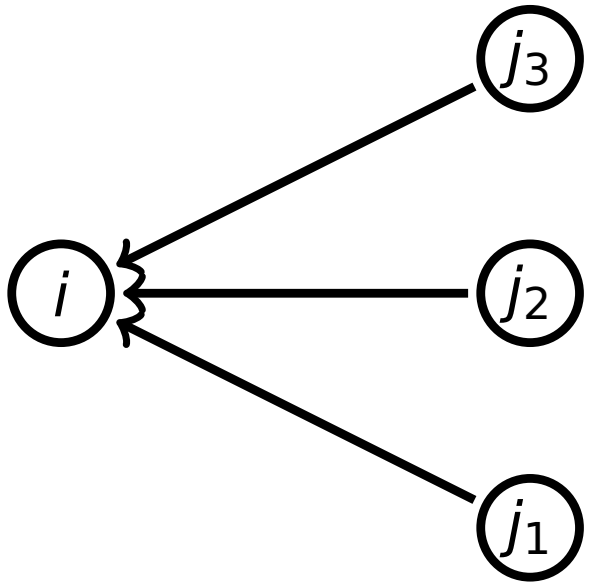
The effects avAlt – totAlt – avXAlt – totXAlt

now also have analogues for influence from incoming ties:

4. avInAlt
5. totInAlt
6. avXInAlt
7. totXInAlt

i is influenced by

incoming ties $j_1 - j_3$



Extreme influence effects

8. maxAlt
9. minAlt

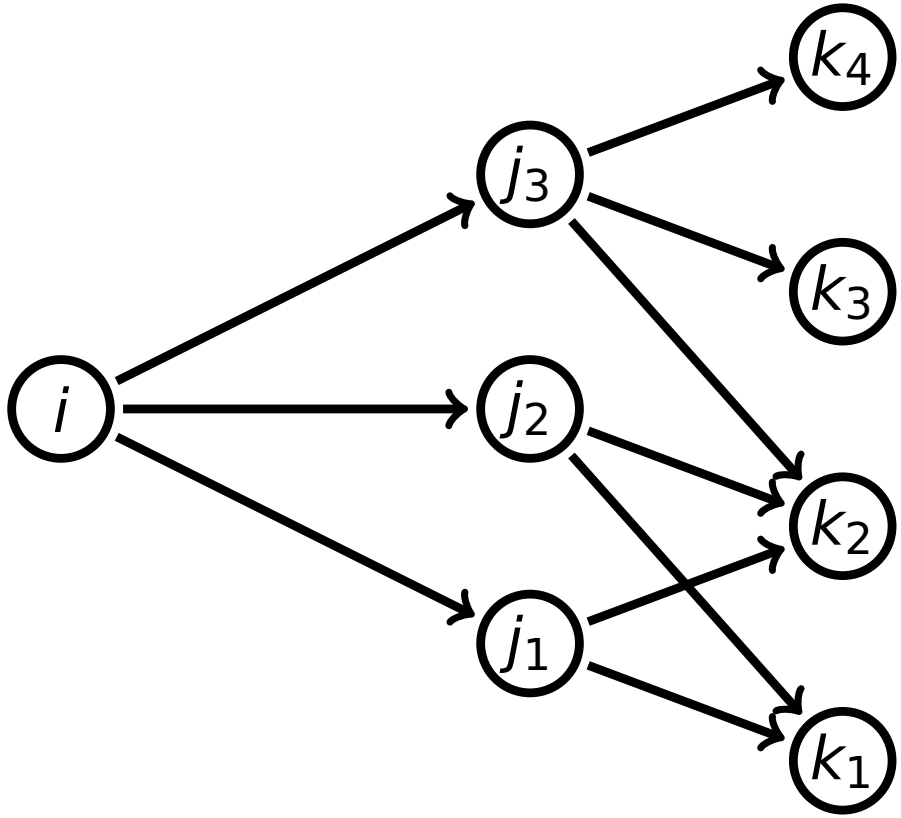
New effects (4): Distance-two

There now is the possibility to express influence at distance 2.

With the distinction average/total this leads to 4 possibilities: average vs. total at step 1 or step 2.

10. avAltDist2
11. totAltDist2
12. avTAltDist2
13. totAAltDist2

i is influenced by
the average/total of the
alter averages/totals of $j_1 - j_3$



New effects (4a)

14. The formula for avAltDist2 (average at both steps) uses

$$\breve{z}_j^{(-i)} = \begin{cases} \dfrac{\sum_{h \neq i} x_{jh} z_h}{x_{j+} - x_{ji}} & \text{if } x_{j+} - x_{ji} > 0 \\ 0 & \text{if } x_{j+} - x_{ji} = 0. \end{cases}$$

The effect is

$$s_{i14}^{\text{beh}}(x, z) = z_i \times \frac{\sum_j x_{ij} \breve{z}_j^{(-i)}}{\sum_j x_{ij}}$$

(and the mean behavior, i.e. 0, if the ratio is 0/0).

New effects (4b)

15. totAltDist2 (total at both steps) is defined by

$$s^{\mathrm{beh}}_{i15}(x,z) = z_i \sum_j x_{ij} \sum_{h \neq i} x_{jh}\, z_h = z_i \sum_j x_{ij}\,(x_{j+} - x_{ji})\,\breve{z}^{(-i)}_j \ .$$

New effects (4c)

16. avTAltDist2 (average of totals) is defined by

$$s^{\text{beh}}_{i16}(x,z) = z_i \times \frac{\sum_j x_{ij}\,(x_{j+} - x_{ji})\,\breve{z}^{(-i)}_j}{\sum_j x_{ij}}$$

$$= z_i \times \frac{\sum_j x_{ij} \sum_{h\neq i} x_{jh}\, z_h}{\sum_j x_{ij}}$$

and the mean behavior, i.e. 0, if the ratio is 0/0.

17. totAAltDist2 (total of averages) is defined by

$$s^{\text{beh}}_{i17}(x,z) = z_i \times \Big(\sum_j x_{ij}\,\breve{z}^{(-i)}_j\Big)\,.$$

New effects (5)

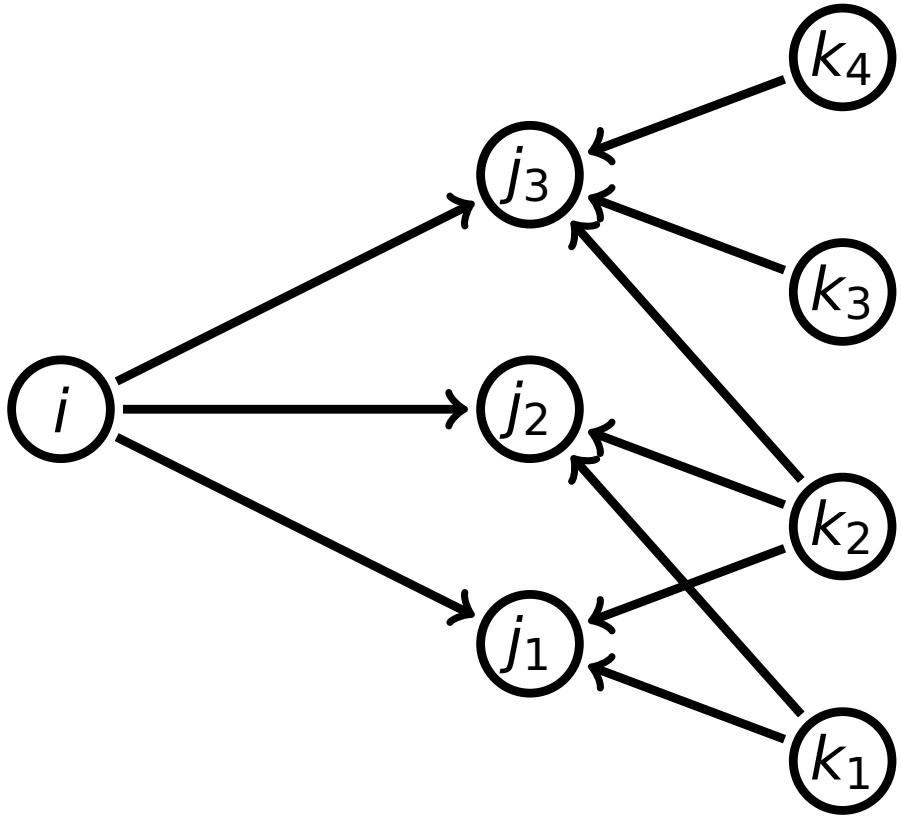
The same for distance-2 averages and totals of covariates:

18. avXAltDist2
19. totXAltDist2
20. avTXAltDist2
21. totAXAltDist2

New effects (6): outgoing - incoming

The same for distance-2 averages and totals where the second step is for incoming ties:

22. avInAltDist2
23. totInAltDist2
24. avTInAltDist2
25. totAInAltDist2
26. avXInAltDist
27. totXInAltDist2
28. avTXInAltDist2
29. totAXInAltDist2

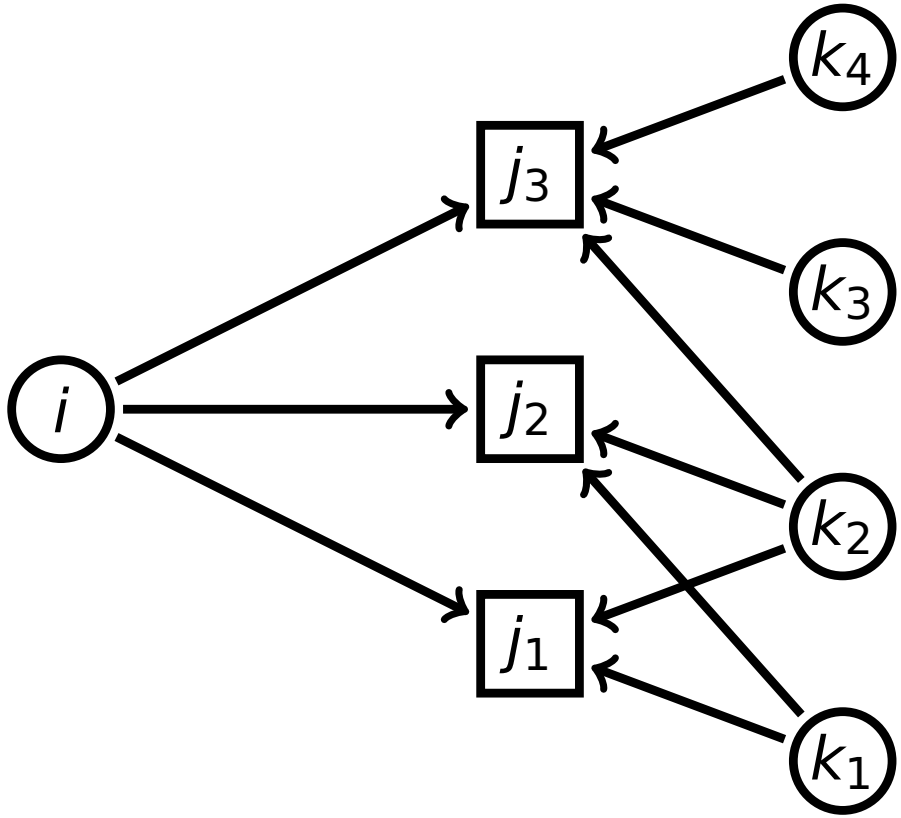


i is influenced by the incoming alter averages of $j_1 - j_3$.

Also ‘sim’ versions (simEgoInDist2 etc.)

New effects (6a)

The *InAltDist2 effects are
also available
for two-mode networks.



This means that it is now possible to model influence
from those out-alters
who have the same affiliations as the focal actor.

Structural equivalence again

These distance-two outgoing–incoming effects
can be regarded as representing influence from
actors who are structurally equivalent (w.r.t. outgoing ties).

An alternative would be to use Jaccard measures (cf. Jin, Jout)
for defining influence effects.
This is still for future consideration.

Elementary effects

SAOM effects have been framed in the triple

1. evaluation
2. maintenance/endowment
3. creation

effects.

If the parameters for a creation and corresponding maintenance effect are the same, then it can be represented just as well by an evaluation effect.

These kinds of effects differ in how they contribute to the probability of a particular choice in the ministep.

The contributions to probabilities are based on
evaluation function f^{ev}
maintenance function f^{mt}
creation function f^{cr} .

Evaluation function plays a role for any step;
creation function only for upward change;
maintenance function only against downward change.

The definition is on the following page.

The probability that, given a current network x and actor i making the ministep, the network changes to $x^{\pm ij}$, is

$$\frac{\exp\left(u_i\left(x, x^{\pm ij}\right)\right)}{1 + \sum_{h \neq i} \exp\left(u_i\left(x, x^{\pm ih}\right)\right)}$$

where the objective function is

$$u_i(x, x^*) = f_i^{\text{ev}}(x^*) - f_i^{\text{ev}}(x) + \Delta^+(x, x^*)\left(f_i^{\text{cr}}(x^*) - f_i^{\text{cr}}(x)\right) + \Delta^-(x, x^*)\left(f_i^{\text{mt}}(x^*) - f_i^{\text{mt}}(x)\right)$$

and

$$\Delta^+(x, x^*) = \begin{cases} 1 & \text{if tie is created } (x^* = x^{+ij}) \\ 0 & \text{if tie is dropped, or no change} \end{cases}$$

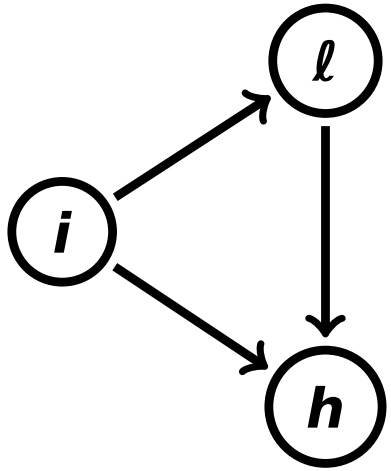
$$\Delta^-(x, x^*) = \begin{cases} 1 & \text{if tie is dropped } (x^* = x^{-ij}) \\ 0 & \text{if tie is created, or no change.} \end{cases}$$

However, not all probabilities of change can be based on changes in some (evaluation-type) function.

Example : transitive triplets

The transitive triplets effect is defined as

$$s_i(x) = \sum_{j,k} x_{ij}\, x_{ik}\, x_{kj}$$



with change statistic
(change when adding tie $i \to j$)

$$\delta_{ij}(x) = \sum_{k} x_{ik}\big(x_{kj} + x_{jk}\big)\ .$$

The first part refers to creating the tie $i \to j = h$,
the second part to creating the tie $i \to j = \ell$.

But one could be interested in only transitive closure,
as defined by closing of an open two-path ($i \rightarrow j = h$),
as distinct from creating ties
to those with the same out-choices,
which is a kind of structural equivalence ($i \rightarrow j = \ell$).

This cannot be represented
as a change in an evaluation function.

But one could be interested in only transitive closure,
as defined by closing of an open two-path ($i \rightarrow j = h$),
as distinct from creating ties
to those with the same out-choices,
which is a kind of structural equivalence ($i \rightarrow j = \ell$).

This cannot be represented
as a change in an evaluation function.

Therefore we need a different kind of effect:

But one could be interested in only transitive closure,
as defined by closing of an open two-path ($i \rightarrow j = h$),
as distinct from creating ties
to those with the same out-choices,
which is a kind of structural equivalence ($i \rightarrow j = \ell$).

This cannot be represented
as a change in an evaluation function.

Therefore we need a different kind of effect:
elementary effect

Elementary effect

An elementary effect is a term of the objective function $u_i\left(x, x^*\right)$ used to define change probabilities for ministeps, referring to creation and/or maintenance of a tie $i \to j$, without being necessarily a difference $f_i(x^{\pm ij}) - f_i(x)$ of some function f_i (or similar with multiplication by Δ^+ or Δ^-).

Elementary effect

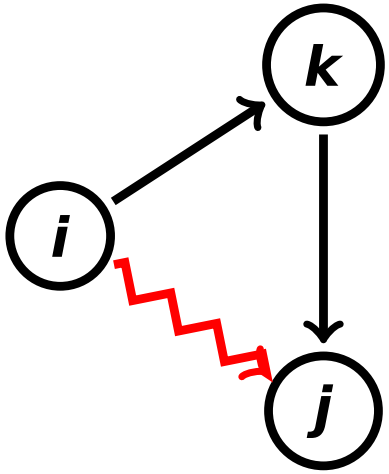
An elementary effect is a term of the objective function $u_i\left(x, x^*\right)$ used to define change probabilities for ministeps, referring to creation and/or maintenance of a tie $i \to j$, without being necessarily a difference $f_i(x^{\pm ij}) - f_i(x)$ of some function f_i (or similar with multiplication by Δ^+ or Δ^-).

Evaluation function is only about the result; elementary effect can express the detailed process / step that leads to a given configuration.

Example : transTrip1 and transTrip2

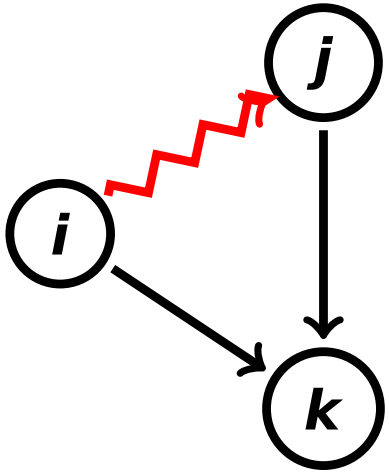
transTrip1 (transitive closure)

$$s_{ij}(x) = x_{ij} \sum_k x_{ik} x_{kj}$$



transTrip2
(structural equivalence outgoing ties)

$$s_{ij}(x) = x_{ij} \sum_k x_{ik} x_{jk}$$



Elementary effects can lead to the same configuration and therefore have the same target statistic
(such as transTrip1 and transTrip2).

In such cases they cannot be distinguished empirically by estimation by the Method of Moments.

Elementary effects can lead to the same configuration and therefore have the same target statistic
(such as transTrip1 and transTrip2).

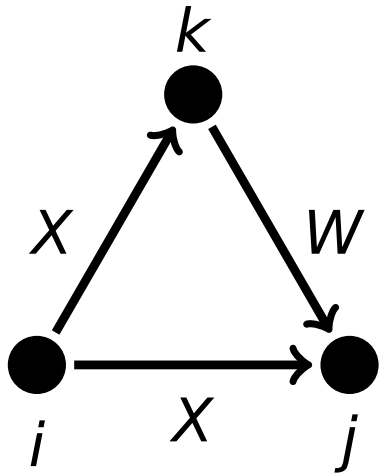
In such cases they cannot be distinguished empirically
by estimation by the Method of Moments.

However, they can be be distinguished empirically
by estimation by the Generalized Method of Moments
(under development)
and by likelihood-based methods
(Maximum Likelihood, Bayes).

The use of elementary effects can give a more fine-grained representation of the process of network change;
but this will require more data;
like also distinction creation-maintenance requires more data.

Other example of elementary effects

30. XWX1: like XWX, dependent variable is only one of the XWX ties: $i \to j$.
31. XWX2: dependent variable here is $i \to k$.



XWX1 and XWX2 are elementary effects.

Still other elementary effects

32. cl.XWX1: like XWX1 but for dependent network.
33. cl.XWX2: like XWX2 but for dependent network.

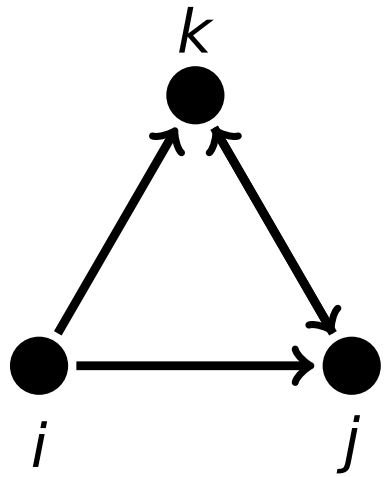
34 sameXInPop, *indegree popularity from same covariate*
number of incoming ties received by those
to whom i is tied and sent by others
who have the same covariate value as i,

$$s^{\text{net}}_{i34}(x) = \sum_j x_{ij} \sum_h x_{hj} I\{v_i = v_h\} \ .$$

35 altXOutAct, *outd. activity weighted by alter's covariate*
squared sum of ties weighted by alter's covariate values,
$s^{\text{net}}_{i35}(x) = \left(\sum_j x_{ij} v_j\right)^2$;
makes sense especially for non-centered covariates.

36. transRecTrip2, another

    reciprocity × transTrip interaction.



37. reciPop: reciprocal degree popularity
38. reciAct: reciprocal degree activity
39. gwesp.. effects have endowment and creation effects. They also are allowed to interact with other effects (interactionType = "dyadic") .
40. And various others
    (e.g., interactions between networks and covariates).

5. Co-evolution

Evolution of multiple networks is studied more and more.

Various new effects have been constructed for this purpose: see Section 12.1.2 of the manual.

5. Co-evolution

Evolution of multiple networks is studied more and more.

Various new effects have been constructed for this purpose:
see Section 12.1.2 of the manual.

When a monadic or dyadic variable is regarded
as a control variable,
it still may be advisable to use it as a dependent variable
in the SAOM analysis, rather than as a covariate,
because this will allow the ‘control’ variable much better
to maintain its correspondence during the simulations
with the focal dependent variables.

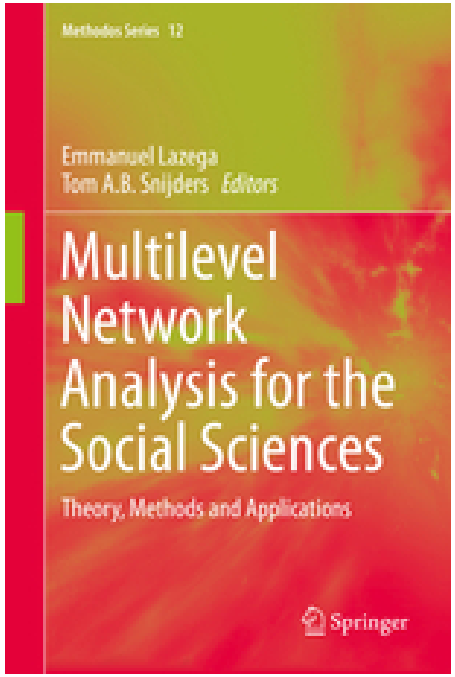
Results using a ‘control network’ as a covariate
will differ quite appreciably from results obtained
while using it as a co-evolving dependent network;
and similarly for monadic variables.

Example: acquaintance or communication
as a control network variable for advice
to study the properties of the ‘purified’ advice relation,
conditional on the condition of acquaintance.

6. Multilevel Analysis of Networks

See MultiMetaSAOM_s.pdf, at website.

Emmanuel Lazega and Tom A.B. Snijders (eds).
Multilevel Network Analysis
for the Social Sciences.
Cham: Springer, 2016.



Special issue of *Social Networks* 'Multilevel Social Networks', edited by Alessandro Lomi, Garry Robins, and Mark Tranmer, vol. 44 (January 2016).

Analysis of Multilevel Networks

Multilevel network (Wang, Robins, Pattison, Lazega, 2013):

Network with nodes of several types,
distinguishing between types of ties
according to types of nodes they connect.

Thus, if types of nodes are A, B, C,
distinguish between $A-A$, $B-B$, $C-C$ ties, etc., (*within-type*)
and between $A-B$, $A-C$, etc., ties (*between-type*).

Some may be networks of interest,
others may be fixed constraints,
still others may be non-existent or non-considered.

This generalizes two-mode networks
and multivariate one mode – two mode combinations.

See paper

Tom A.B. Snijders, Alessandro Lomi, and Vanina Torlò (2013).
A model for the multiplex dynamics of two-mode and one-mode networks, with an application to employment preference, friendship, and advice.
Social Networks, 35, 265-276;

Analysis of longitudinal multilevel networks in RSiena is possible by a trick (thanks to James Hollway).

Consider multilevel network with two node sets, A and B.

There are two one-mode networks internal to A and B,
and two two-mode networks X_1 from A to B; X_2 from B to A.

Specification for RSiena possible by employing
one joint node set $A \cup B$ and two dependent networks:

$$\begin{array}{c} \\ A \\ B \end{array} \begin{array}{c} \begin{array}{cc} A & B \end{array} \\ \begin{pmatrix} \text{internal } A & 0 \\ 0 & \text{internal } B \end{pmatrix} \end{array} \quad \begin{array}{c} \begin{array}{c} A \end{array} \\ \begin{pmatrix} 0 \\ \text{two-mode } B \times A \end{pmatrix} \end{array} \quad \begin{array}{c} \begin{array}{c} B \end{array} \\ \begin{pmatrix} \text{two-mode } A \times B \\ 0 \end{pmatrix} \end{array}$$

networks $A,\ B$ | network X_2 | network X_1

For example:

A a set of organizations, B a set of individuals,
X_2 is a fixed membership relation, X_1 is not there;

networks A and B could be taken apart
in two distinct networks;

if there are only ties between individuals within organizations,
B will be a network of diagonal blocks
and structural zeros between different organizations;

if there are essential differences between individual ties
within organizations or across organizations,
B can be decomposed in two further distinct networks.

For the ‘Analysis of Multilevel Networks’ using RSiena, possibilities exist in principle, as indicated above;

a first example is Snijders, Lomi, Torlò (2013) mentioned above;

the research program has been continued by James Hollway in his DPhil thesis (Oxford – Zürich – Genève);

further relevant effects have to be elaborated;

and the field is open!

7. Missing Data in RSiena

The internal treatment of missing tie values in RSiena is simple:

- Impute missing tie variables in wave 1 by 0.
- Impute missing tie variables in later waves by Last Observation Carried Forward.
- Exclude these imputed values from the calculation of the statistics used for estimation in the MoM.

This can be improved if you have more knowledge of the data and also if you are willing to take more effort.

Missing Data: improvements

⇒ Sometimes there is enough information to make some imputations,
based on knowledge of the data,
with a high degree of confidence.
If possible, do this!

⇒ There was an error in the treatment of missings in *non-centered* monadic covariates
until and including version 1.1-284.

Missing Data (contd.)

⇒ New option *imputationValues* in *coCovar*, *varCovar* :
these values will be used for imputation of missings
for the simulations,
but (like always happens for missings) are not taken into account for the statistics used for estimation.

Can be used if there are reasonable, not completely reliable values for imputation.

Missing Data (contd. further)

⇒ Papers about treatment of missing data in *Social Networks* by Hipp, Wang, Butts, Jose, Lakon (2015) and Wang, Butts, Hipp, Jose, Lakon (2016) criticize missing data treatment by RSiena; but they disregard the fact that imputed values are not used for the statistics for estimation, only for simulations. Thus the effect of these imputations is only indirect.

⇒ In Wang et al. (2016) it is proposed to do multiple imputations by ERGMs for treating missing data in SAOMS. This might be an improvement of the current defaults, but it disregards the longitudinal dependence!

Intermezzo:

Multiple imputation – how does it work?

Multiple stochastic imputation was developed by Don Rubin.

For a given incomplete data set,
the missing data is imputed independently D times
by drawing from the conditional distribution
of the missing data given the observed data.

This leads to D complete data sets,
that differ only with respect to the imputed values.

Intermezzo:

Multiple imputation – how does it work?

Multiple stochastic imputation was developed by Don Rubin.

For a given incomplete data set,
the missing data is imputed independently D times
by drawing from the conditional distribution
of the missing data given the observed data.

This leads to D complete data sets,
that differ only with respect to the imputed values.

For each complete data set the desired analysis is executed;
standard errors of parameters are a combination
of the within-data set standard errors,
and the variability of estimates between the data sets.

How to combine the multiple imputations

The parameter of interest is denoted γ.

Suppose that the d'th randomly imputed data set leads to estimates $\hat{\gamma}_d$ and estimated variances W_d ('Within'),

$$W_d = \text{var}\{\hat{\gamma}_d \mid \text{ data set } d\} \ .$$

Note that W_d underestimates true uncertainty, because it treats imputed data as real data.

The combined estimate is the average

$$\bar{\gamma}_D = \frac{1}{D} \sum_{d=1}^{D} \hat{\gamma}_d \ .$$

Combine multiple imputations....

Compute the average within-imputation variance

$$\overline{W}_D = \frac{1}{D}\sum_{d=1}^{D} W_d \ ,$$

and the between-imputation variance

$$B_D = \frac{1}{D-1}\sum_{d=1}^{D}\left(\hat{\gamma}_d - \bar{\gamma}_D\right)^2 \ .$$

Estimated total variability for $\bar{\gamma}_D$ is

$$T_D = \widehat{\text{var}}\left(\bar{\gamma}_D\right) = \overline{W}_D + \frac{D+1}{D}B_D \ , \ s.e.\left(\bar{\gamma}_D\right) = \sqrt{T_D} \ .$$

Another kind of multiple imputation

The ML option in RSiena will give a model-based simulation of the missings in the second wave,
if the first wave has complete data.

This can be used for getting model-based longitudinal imputations:

1. If the first wave has any missings, estimate an ERGM and impute the missings in the first wave using this.
2. Estimate the SAOM parameters provisionally using the default treatment of missing data.

3. For each wave m, $m = 1, \ldots, M-1$:
   given the completed data set for wave m, produce a model-based random draw from the missings in wave $m+1$ from an ML simulation.
   This is not as time-consuming as full ML estimation, because only one simulation is required.
4. Use this complete data set to obtain one estimate $\hat{\gamma}_d$.
5. Repeat this procedure D times and use Rubin's rules for combining the estimates and standard errors.

The main disadvantage is that the future values are not used for the imputations.

This assumes 'missingness at random': i.e., observed data are sufficient for randomly generating missing data.

Example

Waves 2-3-4 of the van de Bunt students data.

Wave 0 is complete, so no ERGM imputation is needed!

Number of missing actors in waves 0–4 are
0; 2; 3; 5; 6, out of 32.

Impute wave 1 – then 2 – then 3 – then 4.

Effect	default par.	default (s.e.)	multiple imputation par.	multiple imputation (s.e.)	m.f.
Rate 1	4.207	(0.640)			
Rate 2	5.063	(0.668)			
outdegree	–1.728***	(0.317)	–1.804***	(0.343)	.16
reciprocity	2.024***	(0.233)	2.100***	(0.260)	.18
trans. trip.	0.324***	(0.048)	0.329***	(0.049)	.12
indeg. - pop.	0.002	(0.038)	0.024	(0.039)	.16
outdeg. - pop.	–0.132***	(0.027)	–0.155***	(0.031)	.11
outdeg. - act.	0.014	(0.014)	0.013	(0.014)	.09
sex alter	0.409*	(0.200)	0.323	(0.204)	.08
sex ego	–0.386†	(0.208)	–0.282	(0.218)	.13
same sex	0.379*	(0.189)	0.362*	(0.193)	.07
program sim.	0.604**	(0.205)	0.687**	(0.213)	.09

par. = estimate; s.e. = standard error; m.f. = missing fraction;
† $p < 0.1$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$;

convergence t ratios all < 0.06; overall maximum convergence ratio 0.08.

Note:
in waves 3 and 4 the proportion of missing actors is 0.15;
proportion missing information is of about this size.

Standard errors of the two approaches are similar;
estimates sometimes (3 cases) differ by about half s.e.,
in other cases differ hardly.

Further studies are needed to see
how this procedure performs.

8. Relative Importance of Effects

Natalie Indlekofer has contributed the function `sienaRI()`, which assesses the relative importance of effects.
From version 1.1-270.

Natalie Indlekofer and Ulrik Brandes (2013).
Relative importance of effects
in stochastic actor-oriented models.
Network Science 1.3, 278–304.

`sienaRI()` also gives (not explicitly used in her paper)
the raw/total importance of effects.

`sienaRIDynamics()` still has difficulties
(temporarily withdrawn).

Expected importance of a parameter is defined as
the change in choice probabilities
if this parameter would be changed to the value 0.

Expected relative importance is the same,
relative to all effects
(i.e., rescaled to have sum = 1).

`sienaRI()` also produces entropies
(cf. Snijders, *Maths. and Soc. Sci.*, 2004).

Indlekofer & Brandes (2013), formulae (3, 4):

π_i is the vector of probabilities for actor i in next ministep, and $\pi_i^{(-k)}$ is the same if effect k obtains a weight of 0;

$$I_k(X,i) = \frac{\left\|\pi_i - \pi_i^{(-k)}\right\|_1}{\sum_{\ell=1}^{K}\left\|\pi_i - \pi_i^{(-\ell)}\right\|_1} \;;$$

expected relative importance then is

$$\frac{1}{N}\sum_{i=1}^{N} I_k(X,i) \;.$$

Expected (raw / total) importance can then be defined as

$$\frac{1}{N}\sum_{i=1}^{N}\left\|\pi_i - \pi_i^{(-k)}\right\|_1 \;.$$

Example: Results for Glasgow data

Effect	par.	(s.e.)
basic rate parameter friendship	11.207	(1.025)
outdegree (density)	–2.023***	(0.249)
reciprocity	2.563***	(0.190)
transitive recipr. triplets	–0.323***	(0.086)
GWESP I -> K -> J (69)	2.172***	(0.145)
indegree - popularity	–0.016	(0.031)
outdegree - popularity	–0.135†	(0.076)
outdegree - activity	–0.146***	(0.026)
sex alter	–0.101	(0.118)
sex ego	0.076	(0.150)
same sex	0.691***	(0.118)

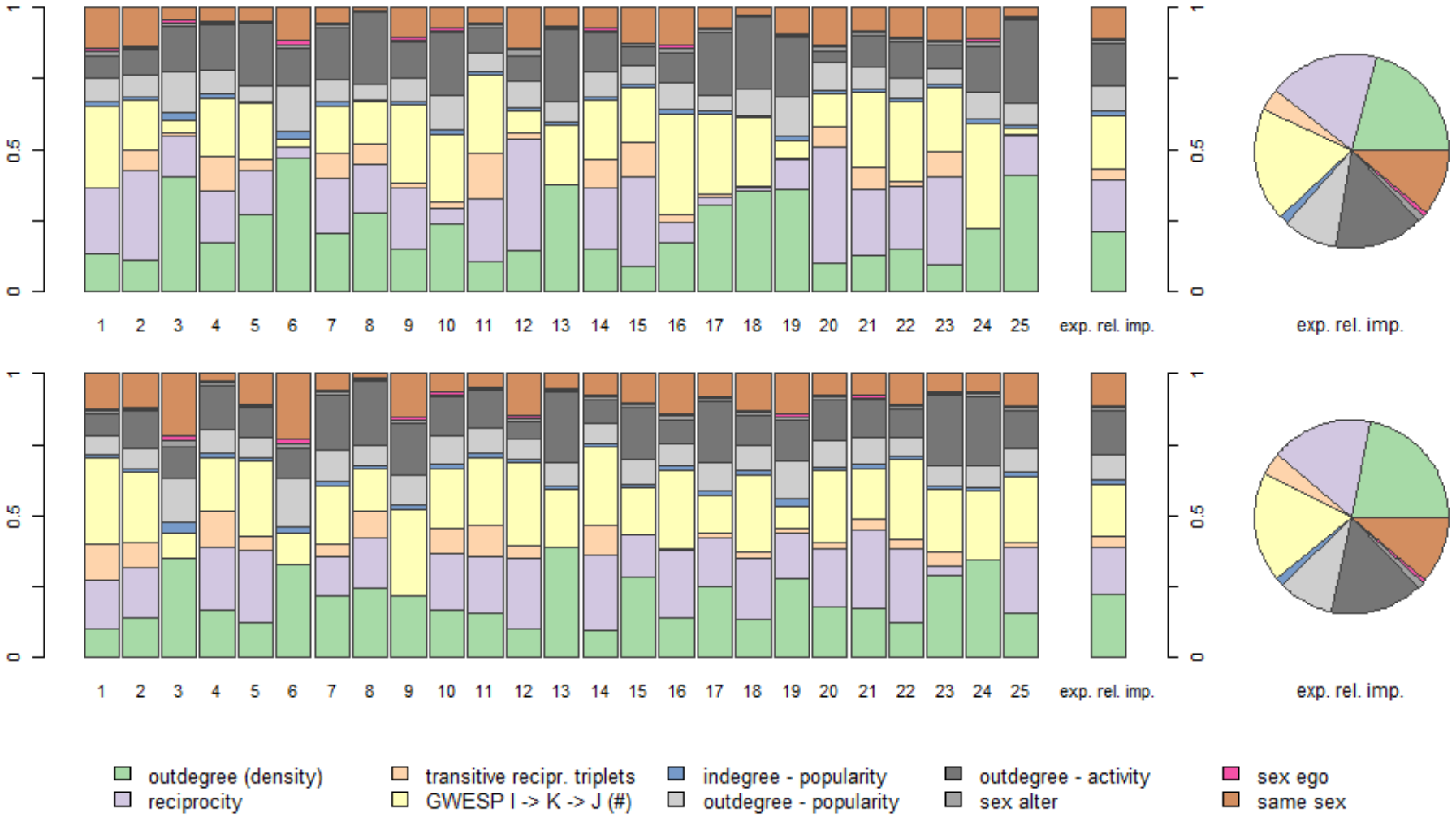
† $p < 0.1$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$;

convergence t ratios all < 0.07.

Overall maximum convergence ratio 0.15.

Example: Results for Glasgow data

Effect	Exp. rel. importance		Exp. importance	
	wave 1	wave 2	wave 1	wave 2
outdegree (density)	0.2075	0.2193	0.8656	0.9122
reciprocity	0.1857	0.1691	0.7154	0.6701
transitive recipr. triplets	0.0369	0.0381	0.1650	0.1696
GWESP I -> K -> J (69)	0.1889	0.1831	0.8079	0.7839
indegree - popularity	0.0145	0.0149	0.0543	0.0551
outdegree - popularity	0.0900	0.0922	0.3361	0.3500
outdegree - activity	0.1486	0.1541	0.6608	0.6791
sex alter	0.0113	0.0109	0.0373	0.0365
sex ego	0.0062	0.0063	0.0244	0.0248
same sex	0.1104	0.1121	0.3798	0.3860
Entropy			0.3632	0.3941



Plot of relative importance of effects for first 25 actors and averaged for all actors (pie-chart).

The graph was produced by

```
plot(RI, actors=1:25, addPieChart = TRUE, legendColumns=5)
```

where RI was the object produced by `sienaRI()`;
`plot.sienaRI()` was slightly improved in version 1.1-288,
with a new argument *actors*,
and better proportions of the pie chart.

Note: you can get the code of such a function by
`RSIenaTest:::plot.sienaRI`
(no parentheses!) and then, if you know enough R,
modify as desired.

9a. Developments in current models

There still is much more to do and explore within the confines of what has already been developed and implemented.

1. The topics mentioned above are open for application / elaboration.
2. Evaluation / creation / maintenance / elementary effects
3. Evaluation / creation / maintenance / effects for behaviour
4. Variants of non-directed models.
5. Comparability of effects across models, data sets

   $\rightsquigarrow$ 'marginal' effects

Developments in current models (contd.)

6. Model selection
7. Importance of GoF for validity of results
8. Extended auxiliary functions for GoF
9. avAlt $\Leftrightarrow$ avSim $\Leftrightarrow$ totAlt $\Leftrightarrow$ totSim
10. Diffusion of innovations – event history analysis
11. Two-mode networks
12. Multivariate (e.g., signed) networks
13. Ordered networks

9b. Hot Issues

- Analysis of Multilevel Networks (see above!)
- Comparison SAOM $\longleftrightarrow$ ERGM (Per Block et al)
- JSiena (Felix Schönenberger)
- Generalized Method of Moments (Viviana Amati)
- Continuous dependent actor variables (Nynke Niezink)
- Settings model (Tom Snijders)
- Marginal effects
- Stable standard errors (Nynke Niezink)
- CUP Books!


The horserace of Siena,
Il palio, was originated in the 13th