# Introduction to Multilevel analysis of Network Dynamics using `sienaBayes`

Tom A.B. Snijders

University of Groningen
University of Oxford


June, 2024

Traditionally, network analysis tended to consist of cases studies of single networks.

However, it is preferable to generalize to a **population** of networks.

This can be achieved, in principle, by multilevel network analysis in the sense of analyzing multiple similar networks, mutually independent.

This was proposed by Snijders & Baerveldt (*J. Math. Soc., 2003*).

Also see Entwisle, Faust, Rindfuss, & Kaneda (*Am. J. Soc., 2007*) who gave on overview of empirical work until then involving multiple networks.





## Sample from Population of Networks

Suppose we have a sample indexed by $j = 1, \ldots, N$ from a population of networks on disjoint node sets, where the networks are 'replications' of each other in the following sense:

they all are regarded as realizations of processes obeying the same model, but having different parameters $\theta_1, \ldots, \theta_j, \ldots, \theta_N$ .

Each disjoint network is called a **group**.
We assume we have network panel data for each group, and wish to analyze these by fitting Stochastic Actor-Oriented Models ('*SAOMs*', RSiena).

Several approaches are possible for combining such data:

1. Multi-group analysis:
   assume all parameters are identical.
2. Integrated hierarchical approach:
   Assumption: population of networks, normal distribution,
   A. mixed effects,
   some parameters varying, others constant across groups;
   B. random effects, all parameters varying.
3. Meta analysis:
   Assumption: population of networks,
   no distributional assumptions:
   *two-stage meta analysis*.
4. Meta analysis: no population assumption:
   *Fisher combination of independent tests.*





Such data sets have multilevel structure;
all caveats and considerations from usual multilevel analysis apply!
E.g.:

- A low number of groups really gives problems
  in saying something about the population,
  the more so if the groups are rather dissimilar;
- Individual concepts take on a different meaning at the group level
  (‘*ecological fallacy*’), and group-aggregates of individual variables
  can be used as additional model components.

See Snijders & Bosker (textbook 2012);
T.A.B. Snijders, ‘The Multiple Flavours of Multilevel Issues for Networks’,
in Lazega & Snijders (2016).

Of the four approaches, here we treat the integrated hierarchical approach.





# Random effects: hierarchical multilevel structure

1. On the **tie level** there is a dynamic process governed by the SAOM.
2. On the **network level** there is a Stochastic Actor-Oriented Model (SAOM) with parameter vector $\theta_j = (\theta_j^{(1)}, \eta)'$ for group *j*. Here $\theta_j^{(1)} \sim \mathcal{N}(\mu, \Sigma)$.
3. On the **global level** there is a population of networks with either a multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ for the randomly varying parameters $\theta_j^{(1)}$, and a common parameter $\eta$ for the rest; or without distribution assumption.

# Hierarchical multilevel analysis

Assumption: population of networks;
$\theta_j \sim$ multivariate normal distribution.

Some of the parameters will be varying across groups $(\theta_j^{(1)})$ ,
others may be constant across groups $(\eta)$ .

This can be compared to the hierarchical linear model,
the generalization of the linear model to nested non-network data sets;
there some individual variables with have random slopes,
but (usually) not all of them.

Group-level variables can also be used,
both as main effects and in interaction with other effects:
'*cross-level interactions*'.





*Advantage of the random effects approach to multilevel modeling*:
The analysis of the separate networks draws strength from
the total sample of networks by regression to the mean.

*Useful especially for many rather small networks.*

(For large networks, where the model of interest can be estimated
without any problems, a two-stage procedure may be better:
(1) estimation by group, (2) meta-analysis.)

*Integrated procedure:*
Estimate $\mu$, $\eta$, and $\Sigma$ to obtain the distribution of $\theta_j$;
then use the 'posterior' distribution of $\theta_j$ given the data
if you want to say something about group *j*.

# Package multiSiena

For network dynamics according to the SAOM, there is the *multi-level estimation function* `sienaBayes` in package multiSiena.

multiSiena is available from the Siena website.

(Earlier, `sienaBayes` was in package `RSienaTest`;
at some moment it will be integrated in RSiena.)





# Bayesian approach

The approach taken in `sienaBayes` is that of *Bayesian statistics*, as distinct from the more usual *frequentist statistics*.

Bayesian statistics postulates a probability distribution
not only for the data, but also for the parameters.

The summary is that the data $X$ have a conditional distribution
given the parameters $\theta$, which is the statistical model;
and the parameters $\theta$ have a **prior distribution**,
reflecting what the researcher knows about them in advance,
i.e., before having looked at the data;
the statistical analysis then leads to the **posterior distribution**,
which is the conditional distribution of the parameters given the data.

# Bayesian approach (2)

In symbols:

Before the data we have the model $p(x \mid \theta)$
and the prior $\pi(\theta)$,
then given the data we have the posterior $p(\theta \mid x)$
which is defined, according to Bayes' rule, as

$$p(\theta \mid x) = \frac{\text{joint p.d. of } (x, \theta)}{\text{p.d. of } x} = \frac{p(x \mid \theta) \times \pi(\theta)}{\int \big(p(x \mid \theta') \times \pi(\theta')\big)\, d\theta'} .$$





# Bayesian approach (3)

The parameter can then be estimated by its posterior mean

$$\mathrm{E}\big\{\theta \mid x\big\}$$

and its uncertainty is expressed
by the posterior covariance matrix

$$\mathrm{cov}\big\{\theta \mid x\big\} \ ;$$

the posterior standard deviations are the square roots
of its diagonal elements, and have the role of standard errors.

This is all very nice;
the catch is that the researcher has to define a prior.

# The Bayesian model for multilevel SAOM

Recall that the group-dependent parameters are denoted $\theta_j^{(1)}$ (for group *j*), and are assumed to have a multivariate normal distribution with mean vector $\mu$ and covariance matrix $\Sigma$;
$\mu$ is the *global mean* and $\Sigma$ is the *between-groups covariance matrix*.

The prior for $\mu$ and $\Sigma$ is specified for `sienaBayes` by keywords `priorMu, priorSigma, priorKappa, priorDf`.

The basic idea is that our prior guess for $\Sigma$ is `priorSigma`, while our prior guess for $\mu$ is `priorMu`, and our uncertainty about this prior guess for $\mu$ is `(1/priorKappa)`×`priorSigma`.
Since our uncertainty about $\mu$ will be much larger than the size of the between-group differences, `priorKappa` should be chosen much smaller than 1; it might even be 0.





# In mathematical terms

*Model*: for all groups *j* independently, $\theta_j^{(1)} \sim \mathcal{N}(\mu, \Sigma)$.

*Prior*: the prior distribution for $(\mu, \Sigma)$ is the inverse Wishart distribution for $\Sigma$;
and, conditional on $\Sigma$, for $\mu$ a multivariate normal distribution with mean $\mu_0$ and covariance matrix $\Sigma/\kappa_0$. In formula:

$$\Sigma \sim \mathrm{InvWishart}_{p_1}(\Lambda_0^{-1}, \mathrm{df}), \text{ and conditionally on } \Sigma$$
$$\mu \mid \Sigma \sim \mathcal{N}_{p_1}(\mu_0, \Sigma/\kappa_0).$$

Here $\mu_0$ = `priorMu`, $\Lambda_0 =$ `priorDf` × `priorSigma`,
df = `priorDf`, $\kappa_0$= `priorKappa`.

The 'central tendency' of this inverse Wishart distribution is approximately `priorSigma`.

# How to prepare

⇒ Data preparation

⇒ Prior specification





# Data for sienaBayes

- For `sienaBayes`, a `sienaGroup` data set is needed.
  This means you first create the *N* separate data sets,
  which need to have the same variables, with the same names,
  and the same number of waves;
  then you apply `sienaGroupCreate` to combine them.
- It is still useful to read the descriptions given by
  `print01Report`, even though this is rather repetitive.
- If there any groups with 'forbidden changes', e.g.,
  structural zero turning into observed 1, which would run
  into logical errors, the data set needs to be changed;
  e.g., by splitting waves, or changing some values into `NA`.

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

## Very simple prior specification

The default values for the prior specification in `sienaBayes` are based on just 0's and 1's, but not all of it is very good.

If you want a very simple prior specification, you could do as shown on the next page

This assumes basically nothing for the prior means, and it assumes for your prior ideas about the between-group differences in the parameter values, that they are roughly of the order of 0.1. (This has the purpose of the estimation for the groups to help each other.)





## Very simple prior specification (2)

Denote by $p$ the number of randomly varying parameters (including the rate parameters).

- Use `priorKappa=0` ($\kappa_0 = 0$; the default).
- Use `priorMu = rep(0,p)` ($\mu_0 = 0$, vector of length $p$; the default).
- Request

```
Sig <- matrix(0,p,p)
diag(Sig) <- 0.01
```

  and use `priorSigma = Sig` (covariance matrix with covariances 0 and variances 0.01, i.e., standard deviations 0.1).
- Use `priorRatesFromData=2` (the default).

# Estimation by sienaBayes

⇒ The Bayesian MCMC procedure

⇒ Steps in sienaBayes

⇒ Operating sienaBayes

⇒ Results of sienaBayes

⇒ Plotting the results

⇒ Perhaps: continuing sienaBayes





# Bayesian MCMC procedure

The estimation is done by a so-called MCMC[1] procedure
that has three levels.
It works by iteratively updating provisional estimates
for $(\theta_1, \ldots, \theta_G)$ and $\mu, \eta, \Sigma$.

1. At the lowest level, the sequence of ministeps to connect the data for the consecutive waves is simulated;
2. at the intermediate level, groupwise parameters $\theta_j$ are updated to correspond to this sequence of ministeps;
3. at the highest level, global parameters $\mu, \eta, \Sigma$ are updated to correspond to $(\theta_1, \ldots, \theta_G)$.

[1] Markov chain Monte Carlo

## Steps in sienaBayes

The MCMC process in sienaBayes has two phases:

1. a *warming phase* with nwarm warming steps;
2. a *main phase* with nmain main steps.

✠ each warming/main step consists of nrunMHBatches steps, i.e., *thinning* is applied to record the provisional estimates only once every nrunMHBatches steps.

So the total number of steps is nrunMHBatches × (nwarm+nmain).
But in each step,
hundreds or thousands of changes at the lowest level are made
(see previous page).





## Operating sienaBayes

1. Construct the data sets and combine them with sienaGroupCreate.
2. Specify the model and construct an effects object; use, e.g.,

   ```
   print(myeff, includeRandoms=TRUE, dropRates=TRUE)
   ```

3. Create an algorithm object (only the seed really matters).
4. If you wish: estimate a multi-group model ans.mom using siena07 (could be helpful for exploration, not necessary).
5. Create Sig as indicated above.
6. ```
   sienaBayes(myalg, data=mydata, effects=myeff,
              priorSigma=Sig, nwarm=500, nmain=1000,
   # perhaps  prevAns=ans.mom,
              nbrNodes=..., silentstart=...)
   ```

   This implicitly uses the default values priorKappa=0, priorMu=0, nrunMHBatches=20 (and many others).
   ```

## Results of sienaBayes

sienaBayes results in an MCMC sample:
a sequence of (nwarm+nmain) sampled values of the parameters;
this sample is not independent but has a Markovian dependence.

These 'parameters' are $\mu, \eta$ and $\Sigma$ but also $(\theta_1, \ldots, \theta_G)$.

IF the process has converged after the warming phase,
the sample from the main phase is a sample from the

**posterior distribution of the parameters.**

This can be studied by plotting, averaging, etc.

However: the big question is, has the process converged?
... this is not easy to see ...





## Plotting results

The Bayesian MCMC procedure produces
a sample from the posterior distribution of all the parameters,
both the $\theta_j$ referring to the individual sampled networks,
and $\eta$, $\mu$ and $\Sigma$ referring to the population of networks.

The inference is based on these sampled posteriors.

Three kinds of plot are important:

1. *multidimensional scaling plots of the group-wise posterior means*, indicating possible outliers; function plotPostMeansMDS
2. *trace plots*, representing successive draws from the posterior distribution (after *thinning*);
3. *density plots*, representing the plausible values of the parameters, given the observed data.



//www.stats.ox.ac.uk/~snijders/siena/BayesPlots.r

## Continue and glue

A first way to assess convergence of the MCMC process
is to look at plots of the successively drawn values of the parameters,
so-called '*trace plots*'.

If they look pretty stable during the warming phase, this is a good sign.
But perhaps stability sets in only somewhere during the main phase,
or the main phase is too short.

Then the previous MCMC estimation can be prolonged,
using `sienaBayes` with keyword `prevBayes`,
which will skip the warming phase, and
use the last of the previous MCMC estimates as the new starting value.

The previous and the new MCMC sample can be glued together
by function `glueBayes`.





## Assess convergence from multiple sequences

Perhaps the best way to assess convergence is by
using multiple independent MCMC sequences (independence here
is achieved by using algorithms with different random number seeds).

The similarity between these sequences then can be checked
numerically by the so-called R-hat criterion
using function `monitor` of package `rstan` (see the RSiena manual)
or visually by package `bayesplots`.

# Literature

Johan H. Koskinen and Tom A. B. Snijders (2023),
Multilevel Longitudinal Analysis of Social Networks.
*Journal of the Royal Statistical Society, Series A*, 186, 376–400.
DOI: https://doi.org/10.1093/jrsssa/qnac009

The Siena scripts page contains some examples of the use of sienaBayes;
see http://www.stats.ox.ac.uk/~snijders/siena/

RSiena manual: Chapter 11.