Manual for SIENA version 1.98

Tom A.B. Snijders
Mark Huisman *

ICS, Department of Sociology
Grote Rozenstraat 31, 9712 TG Groningen, The Netherlands

t.a.b.snijders@ppsw.rug.nl

February 6, 2003

Abstract

SIENA (for Simulation Investigation for Empirical Network Analysis) is a computer program that carries out the statistical estimation of models for the evolution of social networks according to the dynamic actor-oriented model of Snijders (2001, 2003). It also carries out MCMC estimation for the exponential random graph model according to the procedures described in Snijders (2002). This manual gives some information about SIENA version 1.98.

[*] We are grateful to Peter Boer and Evelien Zeggelink for their cooperation in the development of the StOCNET and SIENA programs.

Contents

1 General information

SIENA [1] , for Simulation Investigation for Empirical Network Analysis, is a computer program that carries out the statistical estimation of models for repeated measures of social networks according to the dynamic actor-oriented model of Snijders and van Duijn (1997) and Snijders (2001). Some examples are presented in van de Bunt (1999) and van de Bunt, van Duijn, and Snijders (1999).

The program also carries out MCMC estimation for the exponential random graph model, also called p^* model, of Frank and Strauss (1986), Frank (1991), and Wasserman and Pattison (1996). This procedure is described in Snijders (2002). For this model, the estimation procedure does not always perform satisfactorily for reasons described in Snijders (2002) and investigated further in Snijders and van Duijn (2002).

This manual is about SIENA version 1.98 (February, 2003). The program and this manual can be downloaded from the web site,
http://stat.gamma.rug.nl/stocnet/. The best way to run SIENA is as part of the StOCNET program collection (Boer, Huisman, Snijders, & Zeggelink, 2003), which can also be downloaded from this website. For the operation of StOCNET, the reader is referred to the corresponding manual. If desired, SIENA can be operated also independently of StOCNET.

This manual consists of two parts: the user's manual and the programmer's manual. There are two parallel pdf versions: s_man_s.pdf for screen viewing and s_man_p.pdf for printing. They were made with the LaTeX pdfscreen.sty package of C.V. Radhakrishan which made it possible, e.g., to insert various hyperlinks within the manual. Both versions can be viewed and printed with the Adobe Acrobat reader. This is the print version. Note that section numbering may differ between the two versions.

[1] This program was first presented at the International Conference for Computer Simulation and the Social Sciences, Cortona (Italy), September 1997, which originally was scheduled to be held in Siena. See Snijders & van Duijn (1997).

Part I

User's manual

The user's manual gives the information for using SIENA.
It is advisable also to consult the user's manual of StOCNET because normally the user will operate SIENA from within StOCNET.

2 Changes compared to earlier versions

The main changes in version 1.90 compared to version 1.70 are

1. possibility to use more than two observation moments;
2. inclusion of the exponential random graph ("p^*") model, corresponding to one observation moment;
3. possibility to have changes of composition of the network (actors leaving and/or entering);
4. changing actor covariates;
5. arbitrary codes allowed for missing data (instead of the automatic use of 6 and 9 as codes for missing data, the user now has to supply these codes explicitly);
6. small improvements in the user interface.

The main changes in version 1.95 compared to version 1.90 are

1. for the exponential random graph model some extra simulation options were added, and inversion steps were added;
2. some effects (3-star and 4-star counts) added to the exponential random graph model;
3. for changing covariates, the global rather than the periodwise mean is subtracted;
4. the program SIENA02 for data description was added.

The main changes in version 1.98 compared to version 1.95 are

1. the advanced option modeltype is added, implementing methods in Snijders (2003);
2. maximum number of actors increased to 500.

3 Parts of the program

The operation of the SIENA program is comprised of four main parts:

1. input of basic data description,
2. model specification,
3. simulation of the model with given and fixed parameter values,
4. estimation of parameter values using stochastic simulation.

The normal operation is to start with data input, then specify a model and estimate parameters, and then continuing with new model specifications followed by estimation or simulation.

The program is organized in the form of *projects*. A project consists of data and the current model specification. All files internally used in a given project have the same root name, which is called the project name, and indicated here by *pname*.

The main output is written to the text file *pname*.out.

4 Input data

The main statistical method implemented in SIENA is for the analysis of repeated measures of social networks, and requires network data collected at two or more time points. Therefore, *two or more data files with digraphs* are necessary: the observed networks, one for each time point. The number of time points is denoted M. For the exponential random graph model, one observed network data set is required. In addition, various kinds of covariates are allowed:

1. *actor-bound* or *individual covariates*, also called *actor attributes*, which can be symbolized as v_i for each actor i; these can be constant over time or changing;

2. *dyadic covariates*, which can be symbolized as w_{ij} for each ordered pair of actors (i, j); they are allowed only to have integer values ranging from 0 to 255.

The data files and the names of the variables are made available to SIENA through specification of these files and variable names in StOCNET.

Names of variables must be composed of at most 14 characters. This is because they are used as parts of the names of effects which can be included in the model, and the effect names should not be too long.

4.1 Digraph data files

Each digraph must be contained in a separate input file in the form of an adjacency matrix, i.e., n lines each with n integer numbers, separated by blanks, each line ended by a hard return. The diagonal values are meaningless but must be present.

The data matrices for the two digraphs must be coded in the sense that their values are converted by the program to the 0 and 1 entries in the adjacency matrix. A set of code numbers is required for each digraph data matrix; these codes are regarded as the numbers representing a present arc in the digraph, i.e., a 1 entry in the adjacency matrix; all other numbers will be regarded as 0 entries in the adjacency matrix. Of course, there must be at least one such code number. All code numbers must be in the range from 0 to 9.

This implies that if the data are already in 0-1 format, the single code number 1 must be given. As another example, if the data matrix contains values 1 to 5 and only the values 4 and 5 are to be interpreted as present arcs, then the code numbers 4 and 5 must be given.

Code numbers for missing numbers also must be indicated. These must, of course, be different from the code numbers representing present arcs.

4.2 Dyadic covariates

Each dyadic covariate also must be contained in a separate input file with a square data matrix, i.e., n lines each with n integer numbers, separated by blanks, each line ended by a hard return. The diagonal values are meaningless but must be present.

The reasons for restricting dyadic covariates to integer values from 0 to 255 has to do with how the data are stored internally. If the user wishes to use a dyadic covariate with a different range, this variable first must be transformed to integer values from 0 to 255. E.g., for a continuous variable ranging from 0 to 1, the most convenient way probably is to multiply by 100 (so the range becomes 0–100) and round to integer values. In the present implementation, this type of recoding cannot easily be carried out within StOCNET, but the user must do it in some other program.

4.3 Individual covariates

Individual (i.e., actor-bound) covariates can be combined in one or more files. If there are k covariates in one file, then this data file must contain n lines, with on each line k numbers which all are read as real numbers (i.e., a decimal point is allowed). The numbers in the file must be separated by blanks and each line must be ended by a hard return.

A distinction is made between constant and changing covariates, which refers to changes over time. Each constant covariates has one value per actor valid for all observation moments. Changing covariates can change between observation moments, but are assumed to have constant values from one observation moment to the next. If observation moments for the network are $t_1, t_2, ..., t_M$, then the changing covariates should refer to the $M - 1$ moments t_1 through t_{M-1} , and the m-th value of the changing covariates is assumed to be valid for the period between moment t_m and moment t_{m+1} . Changing covariates are meaningful only if there are 3 or more observation moments. Each changing covariate must be given in one file, containing $k = M - 1$ variables.

The mean is always subtracted from the covariates. See the section on *Centering* below.

4.4 Missing data

Missing data are not allowed in the covariate data. In the network data it is allowed that there are some missing data. These must be indicated by a missing data code, not by blanks in the data set.

In the current implementation of SIENA, missing data are treated in a simple way, trying to minimize their influence on the estimation results. The simulations are carried out over all actors. Missing data are treated separately for each period between two consecutive observations of the network. In the initial observation for each period, missing entries in the adjacency matrix are set to 0. In the course of the simulations, however, these values are allowed to become 1. For the calculation of the statistics used for the parameter estimation and the simulations, if a given element is missing in the adjacency matrix for the observation at the start and/or the observation at the end of this period, this element in set to 0 in both of these observations.

4.5 Composition change

SIENA can also be used to analyze networks of which the composition changes over time, because actors join or leave the network between the observations, as described in Huisman and Snijders (2002). For this case, a data file is needed in which the *times of composition change* are given. For networks with constant composition (no entering or leaving actors), this file is omitted and the current subsection can be disregarded.

Network composition change, due to actors joining or leaving the network, is handled separately from the treatment of missing data. The digraph data files must contain all actors who are part of the network at any observation time (denoted by n) and each actor must be given a separate (and fixed) line in these files, even for observation times where the actor is not a part of the network (e.g., when the actor did not yet join or the actor already left the network). In other words, the adjacency matrix for each observation time has dimensions $n \times n$. At these times, where the actor is not in the network, the entries of the adjacency matrix can be specified in two ways. First as missing values using missing value code(s). In the estimation procedure, these missing values of the joiners before they joined the network are regarded as 0 entries, and the missing entries of the leavers after they left the network are fixed at the last observed values. This is different from the regular missing data treatment. Note that in the initial data description the missing values of the joiners and leavers are treated as regular missing observations. This will increase the fractions of missing data and influence the initial values of the density parameter.

A second way is by giving the entries a regular observed code, representing the absence or presence of an arc in the digraph (as if the actor was a part of the network). In this case, additional information on relations between joiners and other actors in the network before joining, or leavers and other actors after leaving can be used if available. Note that this second option of specifying entries always supersedes the first specification: if a valid code number is specified this will always be used.

For joiners and leavers, crucial information is contained in the times they join or leave the network (i.e., the times of composition change), which must be presented in a separate input file. This data file must contain n lines, each line representing the corresponding actor in the digraph files, with on each line four numbers. The first two concern joiners, the last two concern leavers: 1) the last observation moment at which the actor is not yet observed, 2) the time of joining (expressed as a fraction of the length of the period), 3) the last observation moment at which the actor is observed, 4) the time of leaving (also expressed as a fraction). Also actors who are part of the network at all observation moments must be given values in this file. In the following example, the number of observation moments is considered to be $M = 5$, which means there are four periods; period m starts at observation moment m and ends at $m + 1$ for $m = 1, 2, ..., 4 = M - 1$.

Example of file with times of composition change				
Present at all five observation times	0	1.0	5	0.0
Joining in period 2 at fraction 0.6 of length of period	2	0.6	5	0.0
Leaving in period 3 at fraction 0.4 of length of period	0	1.0	3	0.4
Joining in per. 1 (0.7) and leaving in per. 4 (0.2)	1	0.7	4	0.2
Leaving in per. 2 (0.6) and joining in per. 3 (0.8)	3	0.8	2	0.6

Note that for joining, the numbers 0 1.0 have a different meaning than the numbers 1 0.0. The former numbers indicate that an actor is observed at time 1 (he/she joined the network right before the first time point), the latter indicate that an actor is not observed at observation time 1 (he/she joined just after the first time point). The same holds for leavers: 5 0.0 indicates that an actor is observed at time point 5, whereas 4 1.0 indicates that an actor left right before he/she was observed at time point 5.

From the example it follows that an actor is only allowed to join, leave, join and then leave, or leave and then join the network. The time that the actor is part of the network must be an uninterrupted period. It is not allowed that an actor joins twice or leaves twice. When there is no extra information about the time at which an actor joins or leaves (in some known period), there are three options: set the fraction equal to 0.0, 0.5, or 1.0. The second option is thought to be least restrictive.

4.6 Centering

Individual as well as dyadic covariates are centered by the program in the following way.

For individual covariates, the mean value is subtracted immediately after reading the variables. For the changing covariates, this is the global mean (averaged over all periods). The values of these subtracted means are reported in the output.

For the dyadic covariates and the similarity variables derived from the individual covariates, the grand mean is calculated, stored, and subtracted during the program calculations. (Thus, dyadic covariates are treated by the program differently than individual covariates in the sense that the mean is subtracted at a different moment, but the effect is exactly the same.)

The formula for balance is a kind of dissimilarity between rows of the adjacency matrix. The mean dissimilarity is subtracted in this formula and also reported in the output. This mean dissimilarity is calculated by a formula given in Section 11.

5 Model specification

After defining the data, the next step is to specify a model. In StOCNET, this is done by going to the Model menu, defining the (two or more) networks to be used as the repeated observations of the evolving network, possibly choosing one or more dyadic covariates and possibly a file with actor attributes, clicking first on the Apply button, and then clicking on the Specifications button.

The model specification consists of three kinds of effects (see Snijders, 2001):

- *objective function*
  This is the main focus of model selection.
  The objective function normally should always contain the 'density', or 'out-degree' effect, to account for the observed density. It is also advisable usually to include the reciprocity effect, this being one of the most fundamental network effects.
- *rate function*
  Advice: start modeling with a constant rate function, indicated in the screen as the 'basic rate parameter', without additional rate function effects.
- *gratification function*
  Advice: start modeling without a gratification function.

The output file contains a list of all available effects, given after the report of the data input.

In addition, the model specification comprises the current parameter values and the Model Type (see Section 5.2). After data input, the constant rate parameter and the density effect in the objective function have default initial values depending on the data. All other parameter values initially are 0.

The estimation process changes the current value of the parameters to the estimated values. Values of effects not included in the model are not changed by the estimation process.

It is possible for the user to change parameter values and to request that some of the parameters are fixed in the estimation process at their current value.

5.1 Effects associated with covariates

For each individual covariate, five effects can be included in the model:

1. the effect on the actor's activity in the objective function (out-degree);
2. the effect on the actor's popularity in the objective function (in-degree);
3. the dissimilarity effect in the objective function;
4. the effect on the rate of change of the actor;
5. the dissimilarity effect on dissolution of relations
   (part of the gratification function).

The usual order of importance of these effects is: 3 is most important, then 1 and 2, then 4 and 5.

For each dyadic covariate, three effects can be included in the model:

1. the effect in the objective function;
2. the interaction effect of this covariate with reciprocity (part of the objective function);
3. the effect on dissolution of relations (part of the gratification function).

The first of these three is usually the most important.

5.2 Model Type

The Model Type is part of what is specified in the advanced options as the Model Code. This distinguishes between the model of Snijders (2001) (Model Type 1) and that of Snijders (2003) (Model Type 2). In the latter model, the 'decisions' by the actors to increase or decrease their number of outgoing ties are determined on the basis of only their current degree; the probabilities of increasing or decreasing the out-degree are expressed by the distributional tendency function ξ (indicated in the output as *xi*) and the volatility function ν (indicated as *nu*). Which new tie to create, or which existing tie to withdraw, depends in the usual way on the objective and gratification functions. Thus, the outdegree distribution is governed by parameters that are not connected to the parameters for the structural dynamics. The use of such an approach in statistical modeling minimizes the influence of the observed degrees on the conclusions about the structural aspects of the network dynamics. This is further explained in Snijders (2003).

For Model Type 2, in the rate function, effects connected to these functions xi and nu are included. On the other hand, effects in the objective function that depend only on the out-degrees are canceled from the model specification, because they are not meaningful in Model Type 2. To evaluate whether Model Type 1 or Model Type 2 gives a better fit to the observed degree distribution, the output gives a comparison between the observed out-degrees and the fitted distribution of the out-degrees (as exhibited by the simulated out-degrees). For Model Type 2 this comparison is always given. For Model Type 1, this comparison is given by specifying the Model Code in the advanced options as 3. (For LaTeX users: the log file contains code that can be used to make a graph of the type given in Snijders, 20003).

6 Estimation

The model parameters are estimated under the specification given during the model specification part, using a stochastic approximation algorithm. In the following, the number of parameters is denoted by p. The algorithm is based on repeated (and repeated, and repeated...) simulation of the evolution process of the network. These repetitions are called 'runs' in the following.

Note that the estimation algorithm is of a stochastic nature, so the results can vary! This is of course not what you would like. For well-fitting combinations of data set and model, the estimation results obtained in different trials will be very similar. It is good to repeat the estimation process at least once for the models that are to be reported in papers or presentations, to confirm that what you report is a stable result of the algorithm.

The initial value of the parameters is the current value (that is, the value that the parameters have immediately before you start the estimation process). Usually, a sequence of models can be fitted without any problems occurring. Sometimes, however, problems may occur during the estimation process, which will normally be indicated by some kind of warning in the output file. In such cases the current parameter estimates may have been determined in an unsatisfactory way, and using them as initial values for the new estimation process may again lead to difficulties in estimation. Therefore, it is advisable before starting the estimation algorithm to use a *standard initial value* when the current parameter values are unlikely and also when they were obtained after a divergent estimation algorithm. The use of standard initial values is one of the advanced options.

6.1 Algorithm

During the estimation process, StOCNET transfers control to the SIENA program. The estimation algorithm has three phases:

1. In phase 1, the parameter is held constant at its initial value. This phase is for estimating the matrix of derivatives. In the case of longitudinal data, each run requires p simulations.

2. Phase 2 consists of several subphases. More subphases means a greater precision. The default number of subphases is 4. The parameter values changes from run to run, reflecting the deviations between generated and observed values of the statistics. The changes in the parameter values are smaller in the later subphases.
   The program searches for parameter values where these deviations average out to 0. This is reflected by what is called the 'quasi-autocorrelations' in the output screen. These are averages of products of successively generated deviations between generated and observed statistics. It is a good sign for the convergence of the process when the quasi-autocorrelations are negative (or positive but close to 0), because this means the generated values are jumping around the observed values.

3. In phase 3, the parameter is held constant again, now at its final value. This phase is for estimating the covariance matrix and the matrix of derivatives used for the computation of standard errors. In the case of longitudinal data, each run again requires p simulations.
   The default number of runs in phase 3 is 500. This requires a lot of computing time, but when the number of phase 3 runs is too low, the standard errors computed are rather unreliable.

The number of subphases in phase 2, and the number of runs in phase 3, can be changed in the advanced options.

The user can break in and modify the estimation process in three ways:

1. it is possible to terminate the estimation;

2. in phase 2, it is possible to terminate phase 2 and continue with phase 3;
3. in addition, it is possible to change the current parameter values and restart the whole estimation process.

6.2 Output

There are three output files. All are ASCII files which can be read by any text editor. The main output is given in the *pname*.out file (recall that pname is the project name defined by the user). A brief history of what the program does is written to the file *pname*.log. Some diagnostic output containing a history of the estimation algorithm which may be informative when there are convergence problems is written to the file siena.chk ('chk' for 'check'). This file is overwritten for each new estimation. Normally, you only need to look at *pname*.out.

The output is divided into sections indicated by a line `@1`, subsections indicated by a line `@2`, subsubsections indicated by `@3`, etc. For getting the main structure of the output, it is convenient to have a look at the `@1` marks first.

The primary information in the output of the estimation process consists of the following three parts. Results are presented here which correspond to Table 2, column "t_1, t_3" of Snijders (2001). The results were obtained in an independent repetition of the estimation for this data set and this model specification; since the repetition was independent, the results are slightly different, illustrating the stochastic nature of the estimation algorithm.

1. Convergence check

In the first place, a convergence check is given, based on phase 3 of the algorithm. This check considers the deviations between simulated values of the statistics and their observed values (the latter are called the 'targets'). Ideally, these deviations should be 0. Because of the stochastic nature of the algorithm, when the process has properly converged the deviations are small but not necessarily exactly 0. The program calculates the averages and standard deviations of the deviations and combines these in a t-statistic (in this case, average divided by standard deviation). For longitudinal modeling, convergence is excellent when these t-values are less than 0.1 in absolute value, good when they are less than 0.2, and satisfactory when they are less than 0.3. The corresponding part of the output is the following.

```
Total of 1857 iterations.
Parameter estimates based on 1357 iterations,
basic rate parameter as well as
covariance and derivative matrices based on 500 iterations.

Information for convergence diagnosis.
Averages, standard deviations, and t statistics for deviations from targets:
  1.     -0.236    7.006   -0.034
  2.      0.204    7.059    0.029
  3.     -1.592   22.242   -0.072

Good convergence is indicated by the t-statistics being close to zero.
```

In this case, the t-statistics are -0.034, -0.029, and -0.072, which is less than 0.1 in absolute value, so the convergence is excellent. In data exploration, if one or more of these t-statistics are larger in absolute value than 0.3, it is advisable to restart the estimation process. For results that are to be reported, it is advisable to carry out a new estimation when one or more of the t-statistics are larger in absolute value than 0.1. Large values of the averages and standard deviations are not at all a reason for concern.

For the exponential random graph (or p^*) model, the convergence of the algorithm is more problematic than for longitudinal modeling. A sharper value of the t-statistics must be found before the user may be convinced of good convergence. It is advisable to try and obtain t-values which are less than 0.15. If, even with repeated trials, the algorithm does not succeed in producing t-values less than 0.15, then the estmation results are of doubtful value.

2. Parameter values and standard errors

The next crucial part of the output is the list of estimates and standard errors. For this data set and model specification, the following result was obtained.

```
@3
Estimates and standard errors

 0. Rate parameter                                   5.4292  (   0.6920)
Other parameters:
 1. f: density (out-degree)                         -0.7648  (   0.2957)
 2. f: reciprocity                                   2.3071  (   0.5319)
 3. f: number of distances 2                        -0.5923  (   0.1407)
```

The rate parameter is the parameter called ρ in section 11.2 below. The value 5.4292 indicates that the estimated number of changes per actor (i.e., changes in the choices made by this actor, as reflected in the row for this actor in the adjacency matrix) between the two observations is 5.43 (rounded in view of the standard error 0.69). Note that this refers to unobserved changes, and that some of these changes may cancel (make a new choice and then withdraw it again), so the average observed number of differences per actor will be somewhat smaller than this estimated number of unobserved changes.

The other three parameters are the weights in the objective function. The terms in the objective function in this model specification are the density effect defined as s_{i1} in section 11.1, the reciprocity effect s_{i2}, and the number of distances 2 (indirect relations) effect, defined as s_{i5}. Therefore the estimated objective function here is

$$-0.76\, s_{i1}(x) \,+\, 2.31\, s_{i2}(x) \,-\, 0.59\, s_{i5}(x) \ .$$

The standard errors can be used to test the parameters. For the rate parameter, testing the hypothesis that it is 0 is meaningless because the fact that there are differences between the two observed networks implies that the rate of change must be positive. The weights in the objective function can be tested by t-statistics, defined as estimate divided by its standard error. (Do not confuse this t-test wth the t-test for checking convergence; these are completely different although both are t ratios!) Here the t-values are, respectively, -0.7648/0.2957 = -2.59, 2.3071/0.5319 = 4.34, and -0.5923/0.1407 = -4.21. Since these are larger than 2 in absolute value, all are significant at the 0.05 significance level. It follows that there is evidence that the actors have a preference for reciprocal relations and for networks with a small number of other actors at a distance 2. The value of the density parameter is not very important; it is important that this parameter is included to control for the density in the network, but as all other statistics are correlated with the density, the density is difficult to interpret by itself.

When for some effects the parameter estimate as well as the standard error are quite large, say, when both are more than 2, and certainly when both are more than 5, then it is possible that this indicates poor convergence of the algorithm: in particular, it is possible that the effect in question does have to be included in the model to have a good fit, but the precise parameter value is poorly defined (hence the large standard error) and the significance of the effect cannot be tested with the t-ratio. This can be explored by estimating the model without this parameter, and also with this parameter fixed at some large value (see section 10.1) – whether the value is large positive or large negative depends on the direction of the effect. For the results of both model fits, it is advisable to check the fit by simulating the resulting model and considering the statistic corresponding to this particular parameter.

3. Collinearity check

After the parameter estimates, some matrices are presented. The most important is the covariance matrix of the estimates. In this case it is

```
Covariance matrix of estimates (correlations below diagonal):
     0.087     -0.036      0.003
    -0.230      0.283     -0.033
     0.078     -0.440      0.020
```

The diagonal values are the variances, i.e., the squares of the standard errors (e.g., 0.087 is the square of 0.2957). Below the diagonal are the correlations. E.g., the correlation between the estimated density effect and the estimated reciprocity effect is -0.230. These correlations can be used to see whether there is an important degree of collinearity between the effects. Collinearity means that several effects can represent the same data pattern, in this case, the same values of the network statistics. When one or more of the correlations are very close to -1.0 or +1.0, this is a sign of collinearity. This will also lead to large standard errors of those parameters. It is then advisable to omit one of the corresponding effects from the model, because it may be redundant given the other (strongly correlated) effect. It is possible that the standard error of the retained effect becomes much smaller by omitting the other effect, which can also mean a change of the t-test from non-sigificance to significance.

6.3 Other remarks about the estimation algorithm

6.3.1 Changing initial parameter values for estimation

When you wish to change initial parameter values for running a new estimation procedure, this can be done in StOCNET as an advanced option.

6.3.2 Automatic fixing of parameters

If the algorithm encounters computational problems, sometimes it tries to solve them automatically by fixing one (or more) of the parameters. This will be noticeable because a parameter is reported in the output as being fixed without your having requested this. This automatic fixing procedure is used, when in phase 1 one of the generated statistics seems to be insensitive to changes in the corresponding parameter.

This is a sign that there is little information in the data about the precise value of this parameter, when considering the neighborhood of the initial parameter values. However, it is possible that the problem is not in the parameter that is being fixed, but is caused by an incorrect starting value of one of the other parameters.

When the warning is given that the program automatically fixed one of the parameter, try to find out what is wrong.

In the first place, check that your data were entered correctly and the coding was given correctly, and then respecify the model or restart the estimation with other (e.g., 0) parameter values. Sometimes starting from different parameter values (e.g., the default values implied by the advanced option of "standard initial values") will lead to a good result. Sometimes, however, it works better to delete this effect altogether from the model.

It is also possible that the parameter does need to be included in the model but its precise value is not well-determined. Then it is best to give the parameter a large (or strongly negative) value and indeed require it to be fixed (see section 10.1).

6.3.3 Conditional and unconditional estimation

SIENA has two methods for estimation and simulation: conditional and unconditional. They differ in the *stopping rule* for the simulations of the network evolution. In unconditional estimation, the simulations of the network evolution in each time period carry on until the predetermined time length (chosen as 1.0) has elapsed. In conditional estimation, the simulations run on until the number of different entries between the initially observed network of this period and the simulated network is equal to the number of entries in the adjacency matrix that differ between the initially and the finally observed networks of this period. 'Conditional' means here 'conditional on the observed number of changes'.

Conditional estimation is slightly more stable and efficient, because the basic rate parameter is not estimated by the Robbins Monro algorithm, so this method decreases by 1 the number of parameters estimated by this algorithm. Therefore this is the default. The possibility to choose between conditional and unconditional estimation is an advanced option.

If there are changes in network composition (see Section 4.5), only the unconditional estimation procedure is available.

6.3.4 Automatic changes from conditional to unconditional estimation

Even though conditional estimation is the default and slightly more efficient than unconditional estimation, there is one kind of problem that sometimes occurs with conditional estimation and which is not encountered by unconditional estimation.

It is possible (but fortunately rare) that the initial parameter values were chosen unfortunately in such a way that the conditional simulation does not succeed in getting to the condition required by its stopping rule (see Section 6.3.3). This is detected by SIENA, which then switches automatically to unconditional estimation; after some time it switches back again to conditional estimation.

7 Simulation

The simulation option simulates the network evolution for fixed parameter values. This is meaningful mainly at the point that you have already estimated parameters, and then either want to check again whether the statistics used for estimation have expected values very close to their observed values, or want to compute expected values of other statistics. The statistics to be simulated can be specified in a special screen in StOCNET.

The number of runs is set at a default value of 1,000, and can be changed in the advanced options. The user can break in and terminate the simulations early.

The output file contains means, variances, covariances, and correlations of the selected statistics. The output file also contains t-statistics for the various statistics; these can be regarded as tests for the simple null hypothesis that the model specification with the current parameter values is correct.

The simulation feature can be used in the following way. Specify a model and estimate the parameters. After this estimation (supposing that it converged properly), add a number of potential effects. This number might be too large for the estimation algorithm. Therefore, do not Estimate but choose Simulate instead. The results will indicate which are the statistics for which the largest deviations (as measured by the t-statistics) occurred between simulated and observed values. Now go back to the model specification, and return to the specification for which the parameters were estimated earlier. The effects corresponding to the statistics with large t-values are candidates for now being added to the model. One should be aware, however, that such a data-driven approach leads to capitalization on chance. Since the selected effects were chosen on the basis of the large deviation between observed and expected values, the t-tests, based on the same data set, will tend to give significant results too easily.

The generated statistics for each run are also written to the file *pname*.sdt ('sdt' for 'simulation data'), so you can inspect them also more precisely. This file is overwritten each time you are simulating again. A brief history of what the program does is again written to the file *pname*.log.

7.1 Conditional and unconditional simulation

The distinction between conditional and unconditional simulation is the same for the simulation as for the estimation option of SIENA.

If the conditional simulation option was chosen (which is the default) and the simulations do not succeed in achieving the condition required by its stopping rule (see Section 6.3.3), then the simulation is terminated with an error message, saying *This distance is not achieved for this parameter vector.* In this case, you are advised to change to unconditional simulation.

8 One observation: exponential random graph models

By choosing only one observation moment, the user specifies that not a model for network evolution is studied, but an exponential random graph model, also called the p^* model (Frank & Strauss, 1986; Frank, 1991; Wasserman & Pattison, 1996). SIENA carries out Markov chain Monte Carlo (MCMC) estimation for this model, as described in Snijders (2002). If the algorithm works properly, the computed estimate is an approximation of the maximum likelihood estimate. However, it is discussed in Snijders (2002) that there often occur problems for estimating parameters of this distribution, and for many data sets it is impossible to achieve satisfactory estimates (i.e., good convergence) with this method. In any case, it is advisable always to chose the conditional estimation/simulation option, which means here that the total number of ties is kept fixed. To use SIENA for one observation moment, it is advised first to read Snijders (2002). A further exploration of the possibilities of estimating parameters ofr this model is presented in Snijders & van Duijn (2002).

For conditional estimation in this situation, the total number of arcs remains constant. For unconditional estimation, the total number of arcs is a random variable. The choice between these two is made in the advanced options.

The program recognizes automatically if the data set is symmetric (an undirected graph, with $x_{ij} = x_{ji}$ for all i, j) or anti-symmetric (a tournament, with $x_{ij} \neq x_{ji}$ for all $i \neq j$). In such cases, the MCMC estimation respects this and the exponential random graph model is considered only on the set of all symmetric or all antisymmetric graphs, respectively.

The program has six possibilities for the definition of the steps in the MCMC procedure:

1. Gibbs steps for single relations x_{ij};

2. Gibbs steps for dyads (x_{ij}, x_{ji});

3. Gibbs steps for triplets (x_{ij}, x_{jh}, x_{ih}) and (x_{ij}, x_{jh}, x_{hi});

4. Metropolis Hastings steps for single relations x_{ij};

5. Metropolis Hastings steps for dyads (x_{ij}, x_{ji}) in which symmetric dyads remain symmetric and asymmetric dyads remain asymmetric. This is appropriate for symmetric and antisymmetric graphs.

6. Metropolis Hastings steps keeping the in-degrees and out-degrees fixed; see Snijders and van Duijn (2002).

The choice between these types of steps is made in the advanced options. Some other options are available by modifying the *pname*.mo3 file as indicated in Section below.

In the conditional option (where the number of arcs is fixed), options 1 and 4 exchange values of arcs x_{ij} and x_{hk} with $(i, j) \neq (h, k)$ with probabilities defined by the Gibbs and Metropolis-Hastings rules, respectively; option 2 changes values of dyads (x_{ij}, x_{ji}) and (x_{hk}, x_{kh}) with $(i, j) \neq (h, k)$, keeping $x_{ij} + x_{ji} + x_{hk} + x_{kh}$ constant; and option 3 changes the value of one triplet (x_{ij}, x_{jh}, x_{ih}) or (x_{ij}, x_{jh}, x_{hi}), keeping the sum $x_{ij} + x_{jh} + x_{ih}$ or $x_{ij} + x_{jh} + x_{hi}$ constant.

The number of steps for generating one exponential random graph is $rn^2/2d(1-d)$, where r is a constant which can be changed in the advanced options, and called *Run length* in the advanced options screen; n is the number of actors; and d is the density of the graph, truncated to lie between 0.05 and 0.95. The default value of r is 10. This value can be increased when it is doubted that the run length is sufficient to achieve convergence of the MCMC algorithm.

9 Advanced options

There are some advanced options available in Siena. The main advanced options determine the following.

1. There is a choice between conditional and unconditional simulation and estimation.
2. The use of standard initial values (suitable estimates for the density and reciprocity parameters and zero values for all other parameters) rather than the default use of the current parameter values as initial values for estimating new parameter values.
3. The Model Code.
   This defines the Model Type and an associated output option. The meaning of this code is:
   Model Code = 1: Model Type 1, default output;
   Model Code = 2: Model Type 2, extra output for evaluating the fit of the out-degree distribution;
   Model Code = 3: Model Type 1, extra output for evaluating the fit of the out-degree distribution.
4. The number of subphases in phase 2 of the estimation algorithm.
   This determines the precision of the estimate. Advice: 3 for quick preliminary investigations, 4 or 5 for serious estimations.
5. The number of runs in phase 3 of the estimation algorithm.
   This determines the precision of the estimated standard errors (and covariance matrix of the estimates), and of the t-values reported as diagnostics of the convergence. Advice: 200 for preliminary investigations when precise standard errors and t-values are not important, 500 or 1000 for serious estimations.
6. The initial gain value, which is the step size in the starting steps of the Robbins-Monro procedure, indicated in Snijders (2001) by a_1 .
7. The number of runs in the straight simulations.
   Advice: the default of 1000 will usually be adequate.

These options are used through StOCNET in the Advanced window of the Estimation specification.

For one observation moment, where SIENA tries to estimate parameters of an exponential random graph model, the Model Code defines the kind of steps made in the MCMC algorithm. Further there is in this case an additional advanced option.

8. The multiplication factor r for the run length used in the MCMC algorithm.

10 Getting started

For getting a first acquaintance with the model, one may use the data set collected by Gerhard van de Bunt, discussed extensively in van de Bunt (1999) and van de Bunt, van Duijn, and Snijders (1999). The data files are provided with the program. The digraph data files used are the two networks vrnd32t2.dat, vrnd32t4.dat. The networks are coded as 0 = unknown, 1 = best friend, 2 = friend, 3 = friendly relation, 4 = neutral, 5 = troubled relation, 6 = item non-response, 9 = actor non-response. In the Transformations screen of StOCNET, choose the values '1 2 3' as the values to be coded as 1 for the first as well as the second network. Choose '6 9' as missing data codes.

The actor attributes are in the file vars.dat. Variables are, respectively, gender ($1 = F$, $2 = M$), program, and smoking (1 = yes, 2 = no). See the references mentioned above for further information about this network and the actor attributes.

Specify the data in StOCNET by using subsequently the Data, Transformation, and Model menus (do not forget to click Apply when finishing each of these parts) and then choose the Specifications button in the Model menu.

You will be requested to make some choices for the specification, the meaning of which should be clear given what is explained above. In the specification of the rate function, choose a constant rate; in the specification of the objective function, choose the out-degree effect, the reciprocity effect, and one other effect. In the specification of the gratification function, choose no effects at all. Leave the specification of the rate function as it is (see Section 5, in which it was advised to start modeling with a constant rate function).

Then let the program estimate the parameters. You will see a screen with intermediate results: current parameter values, the differences ('deviation values') between simulated and observed statistics (these should average out to 0 if the current parameters are close to the correct estimated value), and the quasi-autocorrelations discussed in Section 6.

It is possible to intervene in the algorithm by clicking on the appropriate buttons: the current parameter values may be altered or the algorithm may be restarted or terminated. In most cases this is not necessary.

Some patience is needed to let the machine complete its three phases. After having obtained the outcomes of the estimation process, the model can be respecified: non-significant effects may be excluded (but it is advised always to retain the density and the reciprocity effects) and other effects may be included.

10.1 Model choice

For the selection of an appropriate model for a given data set it is best to start with a simple model (including, e.g., 2 or 3 effects), delete non-significant effects, and add further effects in groups of 1 to 3 effects. Like in regression analysis, it is possible that an effect that is non-significant in a given model may become significant when other effects are added or deleted!

When you start working with a new data set, it is advisable first to investigate the main endogenous network effects (reciprocity, transitivity, etc.) to get an impression of what the network looks like, and later add effects of covariates.

10.1.1 Fixing parameters

Sometimes an effect must be present in the model, but its precise numerical value is not well-determined. E.g., if the network at time t_2 would contain only reciprocated choices, then the model should contain a large positive reciprocity effect but whether it has the value 3 or 5 or 10 does not make a difference. This will be reflected in the estimation process by a large estimated value and a large standard error, a derivative which is close to 0, and sometimes also by lack of convergence of the algorithm. (This type of problem also occurs in maximum likelihood estimation for logistic regression and certain other generalized linear models; see Geyer and Thompson (1992, Section 1.6) and Albert and Anderson (1984).) In such cases this effect should be fixed to some large value and not left free to be estimated. This can be specified in the model specification under the Advanced button. As another example, when the network observations are such that ties are formed but not dissolved (some entries of the adjacency matrix change from 0 to 1, but none or hardly any change from 1 to 0), then it is possible that the density parameter must be fixed at some high positive value.

10.1.2 Exploring which effects to include

For an exploration of further effects to be included, the following steps may be followed:

1. Estimate a model which includes a number of basic effects;
2. Simulate the model for these parameter values but also include some other relevant statistics;
3. Look at the t-values for these other statistics; effects with large t-values are candidates for inclusion in a next model.

It should be kept in mind, however, that this exploratory approach may lead to capitalization on chance, and also that the t-value obtained as a result of the straight simulations is conditional on the fixed parameter values used, without taking into account the fact that these parameter values are estimated themselves.

It is possible that for some model specifications the data set will lead to divergence, e.g., because the data contains too little information about this effect, or because some effects are 'collinear' with each other. In such cases one must find out which are the effects causing problems, and leave these out of the model. Simulation can be helpful to distinguish between the effects which should be fixed at a high positive or negative value and the effects which should be left out because they are superfluous.

When the distribution of the out-degrees is fitted poorly (which can be investigated by selecting Model Code 3 in the advanced options), an improvement usually is possible either by including non-linear effects of the out-degrees in the objective function, or by changing to Model Type 2 (see Section 5.2).

10.2 Convergence problems

If there are convergence problems, this may have several reasons.

- The data specification was incorrect (e.g., because the coding was not given properly).

- The starting values were poor. Try restarting from the default values calculated at the initial data input (a certain non-zero value for the density parameter, and zero values for the other parameters); or from values obtained as the estimates for a simpler model that gave no problems. You can obtain the initial default parameter values by choosing the advanced option "standard initial values".
  When starting estimations with Model Type 2 (see Section 5.2), there may be some problems to find suitable starting values. For Model Type 2, it is advised to start with unconditional estimation (see the advanced options) and a simple model, and to turn back to conditional estimation, using the current parameter values as initial estimates for new estimation runs, only when satisfactory estimates for a simple model have been found.

- Too many weak effects are included. Use a smaller number of effects, delete non-significant ones, and increase complexity step by step. Retain parameter estimates from the last (simpler) model as the initial values for the new estimation procedure, provided for this model the algorithm converged without difficulties.

- Two or more effects are included that are almost collinear in the sense that they can both explain the same observed structures. This will be seen in high absolute values of correlations between parameter estimates.

- An effect is included that is large but of which the precise value is not well-determined (see above: section on fixing parameters). This will be seen in estimates and standard errors both being large and often in divergence of the algorithm. Fix this parameter to some large value. (Note: large here means, e.g., more than 5 or less than -5; depending on the effect, of course.)

If there are problems you don't understand, you could take a look at the file *pname*.log; and, if the problems occur in the estimation algorithm, at the file *pname*.chk. These files give information about what the program did, which may be helpful in diagnosing the problem. E.g., you may look in the *pname*.chk file to see if some of the parameters are associated with positive values for the so-called quasi-autocorrelations. The parameters for which this happens from subphase 2.2 onward are parameters that may have led to problems in the estimation algorithm (e.g., because the corresponding effect is collinear with other effects; or because they started from unfortunate starting values; or because the data set contains too little information about their value).

10.3 Composition change

Example data files for a network of changing composition are also provided with the program. These files are called vtest2.dat, vtest3.dat, and vtest4.dat. They contain the same network data as the friendship data files of van de Bunt (for these three observation times and with the same coding), except that in these data some joiners and leavers were artificially created. These actors were given the code '9' for the observation moment at which they were not part of the network. The attribute file vtestexo.dat contains the times at which the network composition changes (see also the example in Section 4.5). This file is necessary for the program to correctly include the times at which actors join or leave the network. For example, the first line of the file contains the values

```
1 0.7 3 0.0
```

which indicates that the first actor joins the network at fraction 0.7 of period 1 (the period between the first and second observation moments) and leaves the network right after the beginning of the third period, i.e., he/she does not leave the network before the last observation at the third time point. Thus, the first actor joins the network and then stays in during the whole period being analysed.

11 Mathematical definition of effects

The mathematical formulae for the definition of the effects are the following. (See Snijders, 2001, for further background to these formulae.) They are listed in the order in which they appear in SIENA .

11.1 Objective function

The potential effects in the objective function, denoted f in Snijders (2001), are the following. (Those which are a function only of the out-degree of actor i are excluded for Model Type 2.)

1. *density effect*, defined by the out-degree
   $s_{i1}(x) = x_{i+} = \sum_j x_{ij}$;

2. *reciprocity effect*, defined by the number of reciprocated ties
   $s_{i2}(x) = \sum_j x_{ij}\, x_{ji}$;

3. *transitivity effect*, defined by the number of transitive patterns in i's relations (ordered pairs of actors (j,h) to both of whom i is tied, while also j is tied to h),
   $s_{i3}(x) = \sum_{j,h} x_{ij}\, x_{ih}\, x_{jh}$;

4. *balance*, defined by the similarity between the outgoing ties of actor i and the outgoing ties of the other actors j to whom i is tied,

   $$s_{i4}(x) = \sum_{j=1}^{n} x_{ij} \sum_{\substack{h=1 \\ h\neq i,j}}^{n} (b_0 - \mid x_{ih} - x_{jh} \mid) \ ,$$

   where b_0 is a constant included to reduce the correlation between this effect and the density effect, defined by

   $$b_0 = \frac{1}{(M-1)n(n-1)(n-2)} \sum_{m=1}^{M-1} \sum_{i,j=1}^{n} \sum_{\substack{h=1 \\ h\neq i,j}}^{n} \mid x_{ih}(t_m) - x_{jh}(t_m) \mid \ .$$

5. *number of distances two effect*, or indirect relations effect, defined by the number of actors to whom i is indirectly tied (through one intermediary, i.e., at sociometric distance 2),
   $s_{i5}(x) = \#\{j \mid x_{ij} = 0,\ \max_h(x_{ih}\, x_{hj}) > 0\}$;

6. *popularity effect*, defined by $1/n$ times the sum of the in-degrees of the others to whom i is tied,
   $s_{i6}(x) = \frac{1}{n} \sum_j x_{ij}\, x_{+j} = \frac{1}{n} \sum_j x_{ij} \sum_h x_{hj}$

7. *activity effect*, defined by $1/n$ times the sum of the out-degrees of the others to whom i is tied,
   $s_{i7}(x) = \frac{1}{n} \sum_j x_{ij}\, x_{j+} = \frac{1}{n} \sum_j x_{ij} \sum_h x_{jh}$

8. *out-degree up to c*, where c is some constant, defined by
   $s_{i8}(x) = \max(x_{i+}\, ,\, c)$;

9. *square root out-degree - $c\times$ o.d.*, where c is some constant, defined by
   $s_{i9}(x) = \sqrt{x_{i+}} - c x_{i+}$,
   where c is chosen to diminish the collinearity between this and the density effect;

10. *squared (out-degree - c)*, where c is some constant, defined by
    $s_{i10}(x) = (x_{i+} - c)^2$,
    where c is chosen to diminish the collinearity between this and the density effect.

11. *number of 3-cycles*,
    $s_{i11}(x) = \sum_{j,h} x_{ij}\, x_{jh}\, x_{hi}$;

The constants c can be chosen and changed by the user.

The main effect for a dyadic covariate w_{ij} is

12. *covariate (centered)*, defined by the sum of the values of w_{ij} for all others to whom i is tied,
    $s_{i12}(x) = \sum_j x_{ij}\, (w_{ij} - \bar{w})$
    where $\bar{w}$ is the mean value of w_{ij} .

For each actor-dependent covariate v_j (recall that these are centered), there are three potential effects in the objective function, indicated in the following list.

13. *covariate-related popularity*, defined by the sum of the covariate over all actors to whom i has a tie,
    $s_{i13}(x) = \sum_j x_{ij}\, v_j$;

14. *covariate-related activity*, defined by i's out-degree weighted by his covariate value,
    $s_{i14}(x) = v_i\, x_{i+}$;

15. *covariate-related dissimilarity*, defined by the sum of absolute covariate differences between i and the others to whom he is tied,
    $s_{i15}(x) = \sum_j x_{ij}\, \big(|\, v_i - v_j \,| - \bar{d}\big)$
    where $\bar{d}$ is the mean of all $|\, v_i - v_j \,|$ values.

The interaction effect of a dyadic covariate w_{ij} with reciprocity is

16. *covariate (centered)*, defined by the sum of the values of w_{ij} for all others to whom i is tied,
    $s_{i16}(x) = \sum_j x_{ij}\, x_{ji}\, (w_{ij} - \bar{w})$
    where $\bar{w}$ again is the mean value of w_{ij} .

11.2 Rate function

The rate function λ (lambda) is defined for Model Type 1 (which is the default Model Type) as a product

$$\lambda_i(\rho, \alpha, x, m) = \lambda_{i1} \lambda_{i2} \lambda_{i3}$$

of factors depending, respectively, on period m, actor covariates, and actor position (see Snijders, 2001, p. 383). The corresponding factors in the rate function are the following:

1. The dependence on the period can be represented by a simple factor

   $$\lambda_{i1} = \rho_m$$

   for $m = 1, ..., M - 1$. If there are only $M = 2$ observations, the basic rate parameter is called ρ.

2. The effect of actor covariates with values v_{hi} can be represented by the factor

   $$\lambda_{i2} = \exp(\sum_h \alpha_h\, v_{hi})\,.$$

3. The dependence on the position of the actor can be modeled as a function of the actor's out-degree, in-degree, and number of reciprocated relations. Define these by

$$x_{i+} = \sum_j x_{ij},\ x_{+i} = \sum_j x_{ji},\ x_{i(r)} = \sum_j x_{ij} x_{ji}$$

(recalling that $x_{ii} = 0$ for all i).
Denoting the corresponding parameter by α_1, the dependence on the out-degree is represented by

$$\lambda_{i3} = \frac{x_{i+}}{n-1} \exp(\alpha_1) \ + \ \left(1 - \frac{x_{i+}}{n-1}\right) \exp(-\alpha_1).$$

This formula is motivated in Snijders and Van Duijn (1997). This defines a linear function of the out-degree, parametrized in such a way that it is necessarily positive.
For a general dependence on the out-degree, in-degree, and number of reciprocated relations, one can use an average of such terms, the second and third one depending in the analogous way on x_{+i} and $x_{i(r)}$, respectively.

11.3 Gratification function

The gratification function, denoted g in Snijders (2001), is the way of modeling effects which operate in different strengths for the creation and the dissolution of relations. The potential effects in this function are the following.

1. $\gamma_1 \, x_{ij} \, x_{ji}$: indicator of a reciprocated relation; a negative value of γ_1 reflects the costs associated with breaking off a reciprocated relation.
2. $\gamma_2 \, (1 - x_{ij}) \sum_h x_{ih} x_{hj}$: the number of actors through whom i is indirectly related to j; a positive value of γ_2 reflects that it is easier to establish a new tie to another actor j if i has many indirect ties to j via others who can serve as an introduction;
3. $\gamma_3 \, x_{ij} \, w_{ij}$: the value w_{ij} for another actor to whom i has a tie; e.g., a negative value of γ_3 reflects the costs for i associated with breaking off an existing tie to other actors j with a high value for w_{ij} .

11.4 Rate function for Model Type 2

For Model Type 2 (see Section 5.2), the rate function is defined according to Snijders (2003) by

$$\begin{aligned} \rho_m \, \lambda_{i+}(s) &= \rho_m \, \frac{\nu(s) \, \xi(s)}{1 \, + \, \xi(s)} \, , \\ \rho_m \, \lambda_{i-}(s) &= \rho_m \, \frac{\nu(s-1)}{1 \, + \, \xi(s-1)} \ , \end{aligned}$$

where $\rho_m \, \lambda_{i+}(s)$ and $\rho_m \, \lambda_{i-}(s)$ represent, respectively, the rate at which an actor of current out-degree s increases, or decreases, his out-degree by 1. The parameter ρ_m is a multiplicative effect of the observation period.

Function ξ (xi) is called the distributional tendency function and is represented according to Snijders (2003, formula (17)) by

$$\xi(s) \ = \ \exp\left(\alpha_1 \ - \ \alpha_2 \log(s+1) - \frac{\alpha_3}{s+1}\right) \ .$$

where the names given in SIENA are

- α_1 : out-degrees effect;
- α_2 : logarithmic out-degree effect;
- α_3 : factorial out-degree effect.

The reasons for these names and interpretation of the effects can be found in Snijders (2003). To the exponent also effects of actor covariates can be added.

The so-called volatility function ν (nu) is defined as

$$\nu(s) \;=\; \left(1 \,+\, \alpha_4 \, \frac{1}{s+1}\right) \;.$$

Also to this exponent effects of actor covariates can be added.

11.5 Exponential random graph distribution

The exponential random graph distribution, which is used if there is only one observation, is defined by the probability function

$$\mathrm{P}_\theta\{X = x\} = \exp\left(\theta' u(x) - \psi(\theta)\right) \;,$$

where $u(x)$ is a vector of statistics. The following statistics are available. Note that SIENA will note whether the observed graph (x_{ij}) is symmetric or not, and choose accordingly between the statistics for undirected and directed graphs.

1. For undirected graphs, the number of edges $\sum_{i<j} x_{ij}$;

   for directed graphs, the number of arcs $\sum_{i \neq j} x_{ij}$.
2. The number of reciprocated relations $\sum_{i<j} x_{ij}\, x_{ji}$.
3. The number of out-twostars $\sum_i \sum_{h<k} x_{ih}\, x_{ik}$.
4. The number of in-twostars $\sum_i \sum_{h<k} x_{hi}\, x_{ki}$.
5. The number of two-paths (mixed twostars) $\sum_i \sum_{h \neq k} x_{hi}\, x_{ik}$.
6. For undirected graphs, the number of transitive triads $\frac{1}{6} \sum_{i,j,h} x_{ij}\, x_{ih}\, x_{jh}$;

   for directed graphs, the number of transitive triplets $\sum_{i,j,h} x_{ij}\, x_{ih}\, x_{jh}$.
7. The number of three-cycles $\frac{1}{3} \sum_{i,j,h} x_{ij}\, x_{jh}\, x_{hi}$.
8. For each dyadic covariate w_{ij}, the sum $\sum_{i,j} x_{ij}\, w_{ij}$.
9. For each dyadic covariate w_{ij}, the associated reciprocity effect defined by $\sum_{i,j} x_{ij}\, x_{ji}\, w_{ij}$.
10. For each individual covariate v_i (changing or constant; recall that all covariates are centered), three effects are included.
    The first is the v_i-related popularity effect $\sum_i x_{+i}\, v_i$;
11. next is the v_i-related activity effect $\sum_i x_{i+}\, v_i$;
12. finally the v_i-related dissimilarity effect $\sum_{i<j} x_{ij} \left(|\, v_i - v_j \,| - \bar{d} \right)$
    where $\bar{d}$ is the mean of all $|\, v_i - v_j \,|$ values.

12 Limitations and time use

The estimation algorithm, being based on iterative simulation, is time consuming. The time needed is approximately proportional to $p^2 n^a C$ where p is the number of parameters, n is the number of actors, the power a is some number between 1 and 2, and C is the number of relations that changed between time m and time $m+1$, summed over $m = 1$ to $M-1$. For data sets with 30 to 40 actors and something like 5 parameters, the estimation process takes a few minutes on a fast PC. The number of actors n should not give a problem up to, say, 200. For large data sets and models, the estimation process may take more minutes up to several hours.

Section 17 indicates the constants in the program that define limitations for the data sets used.

Part II
Programmer's manual

The programmer's manual will not be important for most users. It is intended for those who wish to run SIENA outside of the StOCNET environment, for those who want to know what all the *pname.** files are all about, and for those who want to have a look inside the source code.

The program consists of a basic computation part programmed by the author in Turbo Pascal and Delphi; and the StOCNET windows shell, programmed by Peter Boer in Delphi, with first Evelien Zeggelink and then Mark Huisman as the project leader. The computational part can be used both directly and from the windows shell. The StOCNET windows shell is much easier for data specification and model definition.

13 Executable programs

The present computational part is composed of 4 executable programs. The programs are:

1. SIENA01.EXE for the basic data input, using an existing basic information file (see below);
2. SIENA02.EXE for data description;
3. SIENA04.EXE for confirmation of the model specification;
4. SIENA05.EXE for simulations with fixed parameter values;
5. SIENA07.EXE for parameter estimation.

In these executable programs, the project name *must* be given in the command line, e.g.

```
SIENA01 bunt
```

if `bunt` is the name of the project, and there exists a `bunt.in` file. This `bunt` is called a *command line parameter*. There are the following three ways to specify a command with a command line parameter in Windows. The command line can be given at the DOS prompt (in a Windows environment); it can be given in the Windows "Run" command (for Windows 98 and higher); and it can be indicated in the "target" in the "properties" of a shortcut.

14 SIENA files

Internally the following files are used. Recall that *pname* is the name of the project, which the user can choose at will.

14.1 Basic information file

The basic information file is called *pname*.in, and contains the definition of the numbers of cases and variables, the names of the files in which data are initially stored and their codes, and the names of the variables. This file is written by StOCNET when the data are defined, and can also be written by any text editor that can produce ASCII files. It is read by SIENA01.EXE. It must have the following contents.

1. First a line with six numbers:
   number of observations (2 or more; denoted by M) of the network;
   number of vertices (denoted further by n);
   number of files with constant individual covariates;
   number of files with changing individual covariates;
   number of dyadic covariates;
   indicator of file with times of composition change (0 means no change of network composition, 1 means composition changes).

2. For each of the M network observations, the following three lines:
   a line with the name of the data file;
   a line with the codes that are regarded as a present arc in the digraph;
   a line with the codes that are regarded as missing data.
   (All codes should be in the range from 0 to 9.)

3. If there are 1 or more files with constant actor covariates, for each file there must be the following lines:
   a line with the name of the data file;
   a line with the number of variables in this file;
   lines with the names of these variables, used in the output of the program; for each variable name a separate line.

4. If there are 1 or more files with changing actor covariates, for each file there must be the following lines:
   a line with the name of the data file;
   a line with the name of this variable, used in the output of the program.

5. If there are 1 or more dyadic covariates, for each of them there must be the following two lines:
   a line with the name of the data file;
   a line with the name of this variable, used in the output of the program.

6. If there is a file with times of composition change, a line with the name of this file must be included.

If there are problems in reading the basic input file, delete blanks that may be present after the last number in the lines containing the codes. See to it that the basic input file is an ASCII text file, with numbers separated by blanks, lines separated by hard returns.

An example for the basic input file is the following. This refers to data files that are included with the program, collected by Gerhard van de Bunt. This example, which contains a file with three covariates, is used in van de Bunt (1999) and in van de Bunt, van Duijn, and Snijders (1999).

```
2 32 1 0 0 0
vrnd32t2.dat
1 2 3
6 9
vrnd32t4.dat
1 2 3
6 9
vars.dat
3
gender
program
smoke
```

The basic data input is carried out by executing SIENA01.EXE. This programs reads the basic information file. Some preliminary output is given in the *pname*.out file.

14.2 Definition files

The program writes and reads for internal use the following four definition files.

- *pname*.mo1 Defines numbers of actors and variables, and variable names.
- *pname*.mo2 Defines names of effects.
- *pname*.mo3 Defines model specification and parameter values.
- *pname*.mo4 Contains parameter values and names.

The four *pname*.mo∗ files are read in a format where certain lines are skipped entirely and other lines are skipped after reading a certain number. These skipped parts are between square brackets [...]. Their purpose is to give information to the human reader about the meaning of the lines. Note, however, that SIENA does not check for the brackets, but skips information on the basis of line numbers and reading numerical information.

File *pname*.mo1 is written at the initial project definition, and not changed further. Files *pname*.mo2 and *pname*.mo4 are used for model specification, and converted internally to *pname*.mo3. All these files must correspond (they contain some overlapping information). File *pname*.mo3 is read and changed in the computational part of SIENA.

14.3 Model specification through the mo3 file

To change the model specification outside the StOCNET shell, you can change the *pname*.mo3 file by an ASCII text editor. In this way you can used advanced SIENA options which are not yet available through StOCNET; whether such options exist, will depend on the versions of SIENA and StOCNET.

When looking at the *pname*.mo3 file, with some good will you can see that this file contains lines corresponding to the various effects; each line has a 0/1 entry denoting that the effect is excluded/included, another 0/1 entry denoting that the effect is not fixed, or fixed, during the estimation process; and the value of the parameter. For effects in the objective function, also a 0/1 entry indicates if the effect contains a (constant) parameter value; if so, this parameter value also is included. Such parameters are constant within SIENA runs, but can be changed by the user.

If you change a constant in the *pname*.mo3 file, you must run SIENA04.EXE to get the correct corresponding *pname*.mo2 and *pname*.mo4 files.

The end of the *pname*.mo3 file contains specifications of various estimation options. Most of these are accessible in StOCNET, e.g., through the advanced options mentioned above in Section 9. The consecutive options are the following.

1. Estimation method:
   0 for unconditional, 1 for conditional estimation;
   for exponential random graphs, another available option is to include incidental vertex parameters: 10 for unconditional, 11 for conditional estimation with incidental vertex parameters.

2. Initial value for estimation: 0 current value, 1 standard. Standard means that a good starting value is chosen for the density effect and in the one-observation (exponential random graph) case also for the reciprocity effect; the other effects then have a 0.0 starting value.

3. A code for the type of model.
   For longitudinal data this is the Model Code described in the section on advanced options.
   For exponential random graph models, this code defines the steps used in the one-observation case for simulating a random (di)graph (see also the description above):
   1. Gibbs steps for single relations;
   2. Gibbs steps for dyads;
   3. Gibbs steps for triplets;
   4. Metropolis Hastings steps for single relations;
   5. Metropolis Hastings steps for dyads in which symmetric dyads remain symmetric and asymmetric dyads remain asymmetric (appropriate for undirected graphs and for tournaments, i.e., antisymmetric graphs);
   6. Metropolis Hastings steps conditional on in-degrees and out-degrees.
   To each of these values, the number 10 may be added (so the values become 11–16). In that case a *continuous* chain is used: i.e., the last generated graph is used as the intial value in the MCMC sequence for simulating the next graph. Otherwise (i.e., for the values 1–6), the initial value is an independently generated random graph.

4. A number r proportional to the number of steps used for generating one graph in the one-observation case. The number of steps is $r\, n^2/2d(1-d)$ where n is the nunber of actors and d is the observed density of the graph; if the observed density is less than .05 or more than .95, the value $d = .05$ is used.

5. The number of subphases in phase 2 of the estimation algorithm (advice: 4).

6. The number of phase 3 iterations for the estimation algorithm (advice: 500 for longitudinal data, 1000 for modeling one-observation data by an exponential random graph).

7. The initial value of the gain parameter in the estimation algorithm (advice: 0.1).

8. The default number of simulations for straight simulation (advice: 1000).

14.4 Data files

After the initial project definition the original data files are not used any more, but the project data files are used. These are the following.

pname.d01	Data file time 1.
pname.d02, etc.	Data file time 2, etc.
pname.m01, etc.	Data file missings time 1, etc.
pname.dav	Data files constant actor-dependent covariates (centered).
pname.dac	Data files changing actor-dependent covariates (centered).
pname.z01, etc.	Data files dyadic covariates.
pname.dex	Data file times of composition change.

The user does not need to care about these data files (but should not delete them either).

14.5 Output files

The output for the user goes to *pname*.out. Extra output is written to *pname*.log, which is a log file of what the program did. The estimation procedure also writes a file siena.chk, containing a more detailed report of the estimation algorithm. The latter two files are for diagnostic purposes only. The siena.chk file is overwritten with each new estimation procedure.

15 Parameters and effects

In the source code there are two kinds of parameters: alpha and theta. The alpha parameters are used in the stochastic model, and each alpha parameter corresponds to one effect, independently of whether this effect is included in the current model specification. Their values are stored in the *pname*.mo3 file, which also indicates (by 0-1 codes) whether these variables are included in the model, and whether they are fixed at their current value in the estimation process. The theta parameters are the statistical parameters that correspond to the effects in the current model specification. These are stored in the *pname*.mo4 file.

The distinction between these two types of parameters in principle also allows linear (or other) restrictions between the alphas. In the present version, the possibility of such restrictions is not implemented, but the distinction between alpha and theta allows to elaborate this possibility in a later version.

The names of the effects are defined in procedures DefineModel_lnames, DefineModel_fnames, and DefineModel_gnames in unit S_BASE. The numbers of these effects must correspond, and are defined in functions MaxEffects_l, MaxEffects_f, and MaxEffects_g in unit S_DAT.

The following procedures are all in unit S_BASE. The rate function is defined in procedures Transform_l and Lambda, the objective function in contr_f, and the gratification function in contr_g. For Model Type 2, the rate functions also depends on the functions xiand nu.

The names of the statistics are defined in DefineFunctionNames and the computation of the statistics is defined in Statistics.

16 Units and executable files

The basic computational parts of SIENA are contained in the following units.

First, there are four basic units.

1. S_DAT contains the basic data structures, and procedures for reading and writing a project using .mo1, .mo3, and data files.
2. S_BASE contains the basic model definition and procedures to write .mo2 and .mo4 files.
3. S_SIM contains the procedure for straight simulation.
4. S_EST contains the procedure for estimation.
5. S_DESC contains procedures for data description.

Then there is an intermediate unit.

6. S_START contains the procedure ReadWriteData to start a project by reading the *pname*.in file and the initial data files, and writing the internally used files. It uses only S_DAT.
   Since S_START was programmed so that it uses only S_DAT, it does not contain the default parameters included in some of the effects and it writes only files .mo1 and .mo3, not .mo2 and .mo4. Procedure ReadWriteData from S_START therefore must be followed always by procedure BeforeFirstModelDefinition from S_BASE. This writes files .mo2 and .mo4.

Further there are three units containing specific kinds of utilities.

7. SLIB is a library of various computational and input/output utilities.
8. RANGEN is a library for generation of random variables. It uses the URNS suite for random numbers generation.
9. EIGHT is a unit for storing te data. Its name was chosen for historical reasons.

16.1 Executable files

The basic data input is carried out by executing SIENA01.EXE. This program executes ReadWriteData and BeforeFirstModelDefinition, thereby reading the basic information file.

Data description is carried out by SIENA02.EXE which executes Describe.

In the StOCNET operation the model specification is carried out by StOCNET changing the .mo3 file and then running SIENA04.EXE.

SIENA04.EXE reads the .mo3 file and writes the corresponding .mo2 and .mo4 files. If you change *pname*.mo3 by an text editor outside of StOCNET, it is advisable to run SIENA04.EXE before proceeding.

The simulation is carried out by SIENA05.EXE which executes Simulate.

The estimation is carried out by executing SIENA07.EXE which executes Estimate.

16.2 Some essential procedures

Unit S_BASE contains the simulation procedures. The main procedures in this unit are the following.

1. Function FRAN which generates the required statistics and is called by procedure Simulate in S_SIM and by Estimate in S_EST.

2. Procedure Statistics which calculates the statistics from a generated network (or adjacency matrix), and which is called by FRAN.

3. Procedure Runepoch which generates a random network for given parameter values and a given initial situation by simulating the actor-oriented evolution model for one period between two observations. This procedure is called by procedure FRAN.

4. Procedure Runstep which makes one stochastic step according to the actor-oriented evolution model, i.e., it choosed stochastically one entry (i, j) of the adjacency matrix to be changed. This procedure is called by procedure Runepoch.

5. Procedure ChangeTie which, called at the end of procedure RunStep, carries out the required change of the adjacency matrix and the associated updates of various statistics.

6. Function Lambda, which is the rate function for each actor, and is used in procedure Runstep. For Model Type 2, it uses functions xi and nu.

7. Function contr_f, which defines the contribution $s_{ih}(x)$ of each given effect h to the objective function, and is used in procedure Runstep.

8. Function contr_g, which defines the contribution of each given effect to the gratification function, and is also used in procedure Runstep.

In unit S_EST, the Robbins-Monro algorithm is contained in the procedure POLRUP (for Polyak-Ruppert, see Snijders, 2001).

17 Constants

The program contains the following constants. Trying to use a basic information file that implies a data set going beyond these constants leads to an error message in the output file and stops the further operation of SIENA.

name	meaning	in unit
$nmax$	maximum number n of actors	EIGHT
nrg	maximum array size for random number generation	RANGEN
$pmax$	maximum number p of included effects	SLIB
$ccmax$	maximum number of possible statistics	SLIB
$nzmax$	maximum number nz of individual covariates	EIGHT
$nzzmax$	maximum number nzz of dyadic covariates	EIGHT

The constant individual covariates and the changing individual variables both have $nzmax$ as their maximum.

Reasonable values for these constants are the following:
$nmax = 500$;
$nrg = nmax$;
$pmax = 30$;
$ccmax = 16 + M + 12 \times nzmax + 3 \times nzzmax$ where M is the number of repeated observations (in the current version of SIENA, the number of statistics is $16 + M + 6 \times (nz + nzc) + 3 \times nzz$; this can become higher in future specifications);
$nzmax = 10$;
$nzzmax = 10$.

The number M of observations may not be higher than 99. Since the number of observations is dealt with by a dynamic array, this is not reflected by some constant. The only reason for the upper bound of 99 is that the index number of the observation is used in the internal data file extension names and may not have more than two digits. But 99 seems quite a high upper bound for practical data sets.

18 References

Albert, A., and J.A. Anderson. 1984. On the existence of the maximum likelihood estimates in logistic regression models. *Biometrika*, **71**, 1 – 10.

Boer, P., Huisman, M., Snijders, T.A.B., and E.P.H. Zeggelink. 2001. *StOCNET: An open software system for the advanced statistical analysis of social networks.* Groningen: Pro*GAMMA*/ICS. Available from http://stat.gamma.rug.nl/stocnet/.

Frank, O. 1991. Statistical analysis of change in networks. *Statistica Neerlandica*, **45**, 283–293.

Frank, O., and D. Strauss. 1986. Markov graphs. *Journal of the American Statistical Association*, **81**, 832 – 842.

Geyer, C.J., and E.A. Thompson. 1992. Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society*, ser. B, **54**, 657 – 699.

Huisman, M.E., and T.A.B. Snijders. 2003. Statistical analysis of longitudinal network data with changing composition. *Submitted for publication.*

Snijders, T.A.B. 2001. The statistical evaluation of social network dynamics. Pp. 361-395 in *Sociological Methodology – 2001*, edited by M.E. Sobel and M.P. Becker. Boston and London: Basil Blackwell.

Snijders, T.A.B. 2002. Markov Chain Monte Carlo Estimation of Exponential Random Graph Models. *Journal of Social Structure*, Vol. 3 (2002), No. 2.
Available from http://www2.heinz.cmu.edu/project/INSNA/joss/index1.html.

Snijders, T.A.B. 2003. Accounting for Degree Distributions in Empirical Analysis of Network Dynamics. *Proceedings of the National Academy of Sciences USA*, to be published.
Available from http://stat.gamma.rug.nl/snijders/siena.html.

Snijders, T.A.B., and M.A.J. Van Duijn. 1997. Simulation for statistical inference in dynamic network models. Pp. 493 – 512 in *Simulating Social Phenomena*, edited by R. Conte, R. Hegselmann, and P. Terna. Berlin: Springer.

Snijders, T.A.B., and van Duijn, M.A.J. 2002. Conditional maximum likelihood estimation under various specifications of exponential random graph models.
Pp. 117–134 in Jan Hagberg (ed.), *Contributions to Social Network Analysis, Information Theory, and Other Topics in Statistics; A Festschrift in honour of Ove Frank.* University of Stockholm: Department of Statistics.

Van de Bunt, G.G. 1999. *Friends by choice. An actor-oriented statistical network model for friendship networks through time.* Amsterdam: Thesis Publishers, 1999.

Van de Bunt, G.G., M.A.J. van Duijn, and T.A.B. Snijders. 1999. Friendship networks through time: An actor-oriented statistical network model. *Computational and Mathematical Organization Theory*, **5**, 167 – 192.

Wasserman, S., and P. Pattison. 1996. Logit models and logistic regression for social networks: I. An introduction to Markov graphs and p^*. *Psychometrika*, **61**, 401 – 425.