



PNet: Program for the Simulation and Estimation of
(p^*) Exponential Random Graph Models for
Multilevel Networks

USER MANUAL

Peng Wang

Alex Stivala

Garry Robins

Philippa Pattison

Johan Koskinen

Alessandro Lomi

Swinburne University of Technology

Università della Svizzera italiana

The University of Melbourne

The University of Sydney

©2022

Table of Content

1	<i>Introduction</i>	5
1.1	Acknowledgements	5
1.2	System Requirements	6
2	<i>Setup MPNet</i>	6
3	<i>MPNet sessions</i>	6
3.1	Start MPNet	7
3.2	MPNet main user interface	8
4	<i>MPNet for ERGMs</i>	9
4.1	Simulating one-mode networks	9
4.1.1	Simulation Options	13
4.1.2	Simulation Output	14
4.2	Simulating two-mode networks	15
4.3	Simulating two level networks	16
4.4	Estimation	18
4.5	Estimating ERGMs for one-mode networks	18
4.6	Estimating ERGMs for two mode networks	19
4.7	Estimating ERGMs for combined one- and two- mode networks	20
4.8	Estimating ERGMs for two-level networks	20
4.9	Estimating ERGMs with nodal attributes as covariates	20
4.10	Estimating conditional ERGMs	21
4.11	Options for the estimation algorithm:	21
4.12	Estimation Output	22
4.13	Goodness of Fit	23
4.13.1	Goodness of Fit Setup	23
4.13.2	Goodness of Fit Output	24

4.14	Bayesian estimation	25
4.14.1	Bayesian estimation settings	26
4.14.2	Bayesian estimation outputs	27
4.14.3	Bayesian estimations with missing network data	28
5	<i>MPNet for ALAAMs</i>	30
5.1	Simulating ALAAMs for one-mode networks	30
5.1.1	Simulation Options	33
5.1.2	Simulation Output	33
5.2	ALAAM Estimation	35
5.3	ALAAM Goodness of Fit Test	36
6	<i>References</i>	38
7	<i>Appendix A – Sample Files</i>	40
7.1	Sample Input Files	40
8	<i>Appendix B – Model Configurations</i>	42
8.1	Non-directed one-mode networks (A & B)	42
8.2	Bipartite networks (X)	42
8.3	Directed one-mode networks (A & B)	43
8.4	Non-directed one- and two-mode interactions (A & X, or B & X)	45
8.5	Directed one- and two-mode interactions (A & X, or B & X)	46
8.6	Non-directed cross-level interactions (A, B & X)	47
8.7	Directed cross-level interactions (A, B & X)	48
8.8	Non-directed one-mode social selection models with binary attributes	49
8.9	Two-mode social selection models with binary attributes	49
8.10	Non-directed one-mode social selection models with continuous attributes	50
8.11	Two-mode Social selection models with continuous attributes	50
8.12	One- and two-mode social selection models with categorical attributes	51
8.13	Directed one-mode social selection models with binary attributes	51
8.14	Directed cross level social selection models with binary attributes	52
8.15	Directed one-mode social selection models with continuous attributes	52
8.16	Directed cross level social selection models with continuous attributes	54

8.17	Directed cross level social selection models with categorical attributes	56
9	<i>Appendix C – R utility functions for mpnet</i>	58
9.1	reading in simulated networks	58
9.2	reading in simulated statistics	58

Introduction

MPNet is a program for statistical analysis of exponential random graph models (ERGMs) for multilevel networks and Auto-Logistic Actor Attribute Models (ALAAMs). It has three major functionalities:

Simulation:

Simulating network or attribute distributions with specified model parameter values.

Estimation:

Estimating specified ERGM/ALAAM parameters for a given network using Markov Chain Monte Carlo Maximum likelihood estimations (Snijders, 2002), or Bayesian approximation algorithms with and without missing data (Caimo and Friel, 2011; Koskinen et al, 2011; 2013).

Goodness of Fit:

Testing the goodness of fit of a specified model to a given network/attribute with a specified set of parameters.

MPNet is capable of modelling one-mode and bipartite networks. This documentation will illustrate how to model one-mode networks, bipartite networks, then two-level networks which are combinations of one- and two-mode networks. The model specifications largely follow Wang et al (2013). The whole list of ERGM and ALAAM specifications implemented in MPNet is summarized in the Appendices.

For a description of ERGMs, ALAAMs and their applications, see:

Lusher, D., Koskinen, J., & Robins, G. (2013). *Exponential random graph models for social networks: Theory, methods and applications*. Cambridge University Press.

Acknowledgements

MPNet contains code and ideas from many contributors. We would like to thank the following people for contributing to this program.

Emmanuel Lazega, Galina Daraganova, Dean Lusher, Tom A.B. Snijders, Lei Xing, and Yu Zhao.

System Requirements

Operating system	Microsoft® Windows or Macintosh with Windows parallels
Software	Microsoft .NET Framework Version 4.0+

The Software required is freely available from Microsoft web site.

<http://www.microsoft.com/en-au/download/details.aspx?id=17851>

MPNet can be made to run native in Macintosh environment either by following James Hollways instructions on <http://www.jameshollway.com/mpnet-for-mac/> or by downloading his bottled version from <http://www.jameshollway.com/mpnet-for-mac/mpnet/>.

Setup MPNet

MPNet.exe is a Windows executable program available from PNet website at

www.sna.unimelb.edu.au/pnet/pnet.html

New versions of MPNet will be available from the PNet website. To update MPNet, simply download the most recent version, and discard the early versions

MPNet sessions

Simulations and estimations of ERGMs and ALAAMs under MPNet are organized by *sessions*. A session typically consists of the following steps:

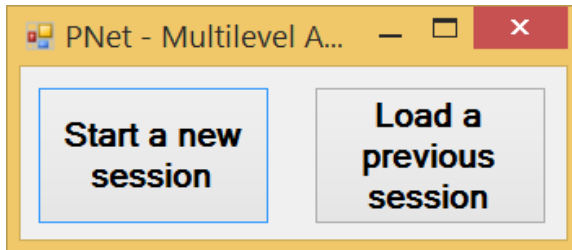
1. network or attribute data specifications
2. ERGM/ALAAM model specifications
3. simulation or estimation runs
4. analysing and interpreting the simulation/estimation outputs

MPNet keeps track of session setting in a PNet session file, e.g. "session.pnet" which contains all settings, such as network data and model specifications in the most recent MPNet session.

Users can start a new session or load a previous session when start up MPNet.

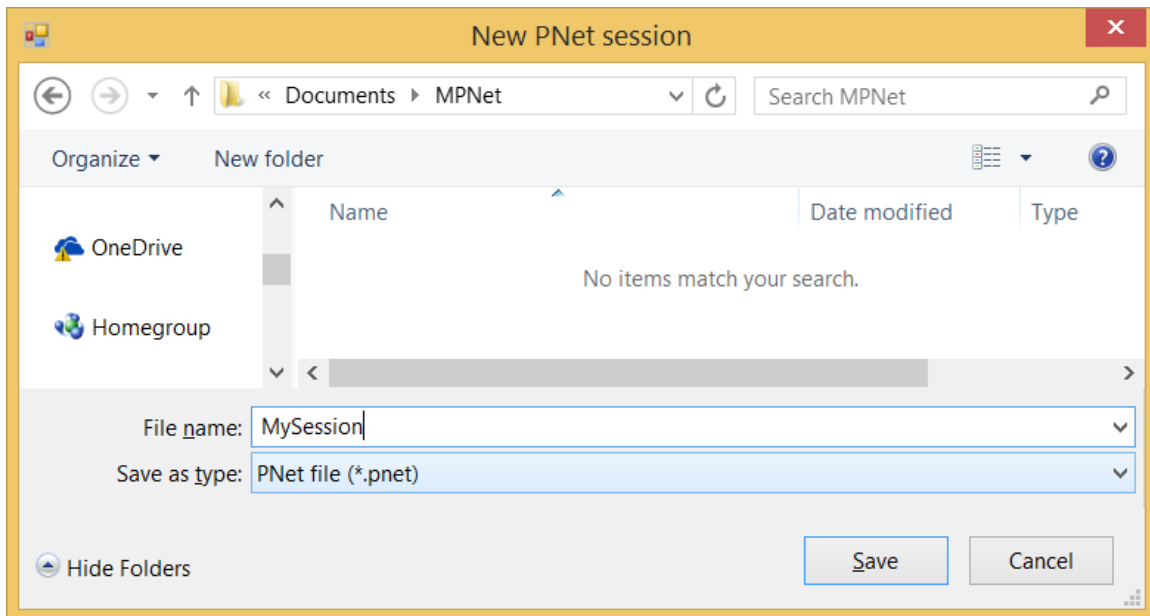
Start MPNet

Double click on the MPNet.exe program to start the program with the option of beginning a new session or load a previous session.



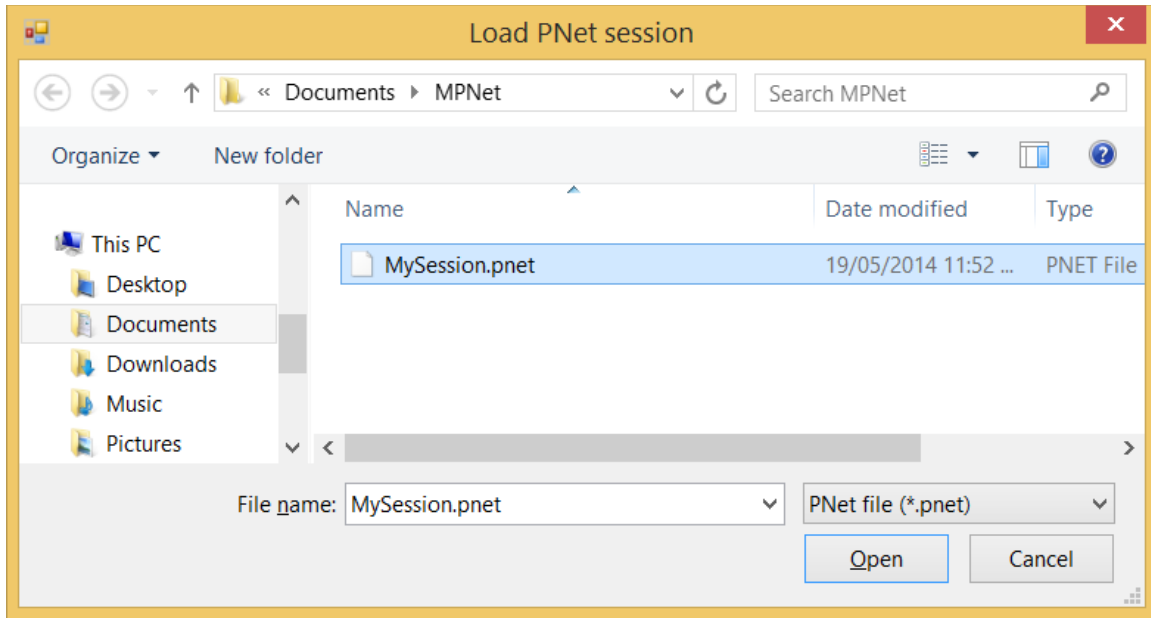
To start a new session, click on the “**Start a new session**” button. A file saving dialog will appear and ask for a MPNet session file name. Specify a session name (e.g. MySession) and click on “**Save**”, the MySession.pnet file will be created by MPNet in the user specified folder, and all MPNet output for this session will appear in this folder.

All MPNet output files will have file names that end with the session file name you provided here (e.g. if you have a session name MySession under simulation, you will have an output file named “simulation_MySession.txt.”)



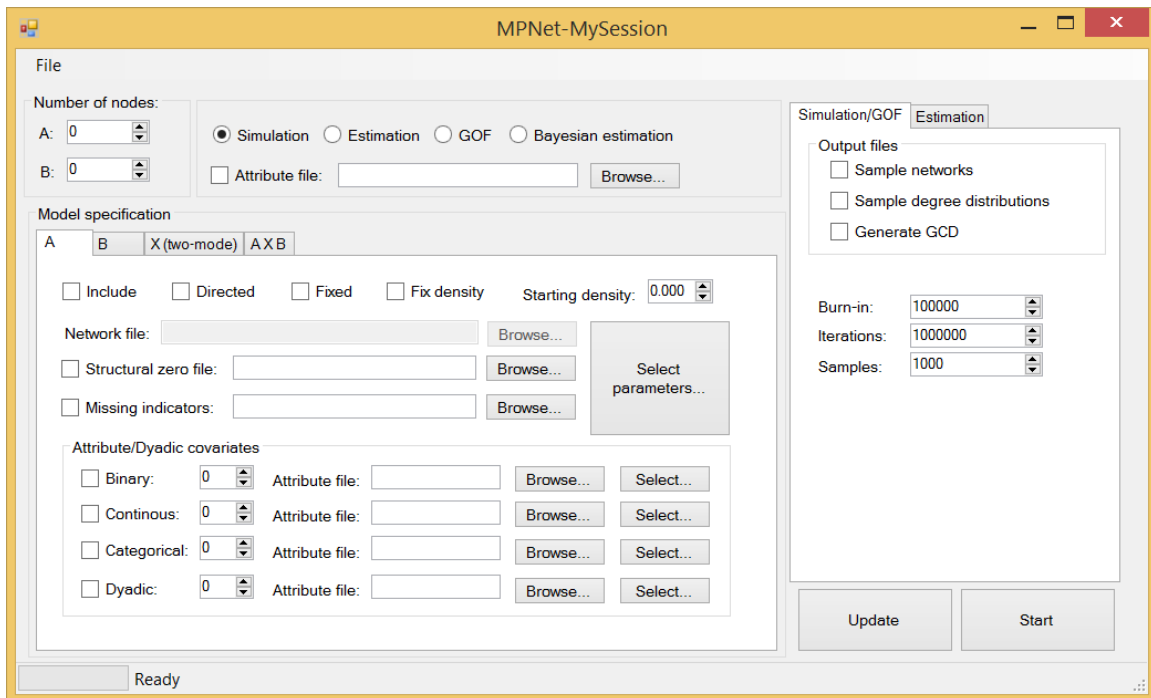
The MySession.pnet file records all session settings in the most recent session and allows the user to reload the session after closing the program. The MPNet main user interface will appear once we saved the session file.

To load a previous session, click on “**Load a previous session**” button when start up MPNet. A open file dialog will apparar, and ask for an MPNet session file.



Select a session file and click on **Open** will load the previous session settings into the MPNet main user interface.

MPNet main user interface

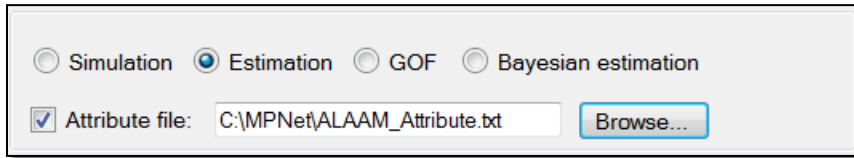


MPNet treats a two-level network as a combination of two within level one-mode networks, labelled as network A and B, and a two-mode cross level network labelled as X. The overall two-level network is labelled as network M in the MPNet output files. This means that you may have two distinct node-sets, N_A and N_B , in which case A is the network (directed or undirected) among

the nodes in N_A ; B is the network (directed or undirected) among the nodes in N_B ; and M is the bipartite network of ties between nodes in N_A and nodes in N_B .

The top left section of the user interface specifies the number of nodes involved in the levels A and B, i.e. the number of nodes in N_A and N_B , respectively.

The top middle section specifies the functions the current session is performing, that is, Simulation, Estimation, Model Goodness of Fit (GOF) test, or Bayesian estimations. MPNet uses ERGMs by default. To simulate or estimate ALAAMs, an attribute file, which contains the attributed dependent variables, needs to be specified by choosing the check box:



The screenshot shows a control panel with four radio buttons: Simulation, Estimation (selected), GOF, and Bayesian estimation. Below them is a checked checkbox for 'Attribute file:' followed by a text box containing 'C:\MPNet\ALAAM_Attribute.txt' and a 'Browse...' button.

Using examples, Section 4 of the manual provides instructions on how to use MPNet to simulate or estimate ERGMs. Detailed instructions on MPNet for ALAAMs are described in Section 5.

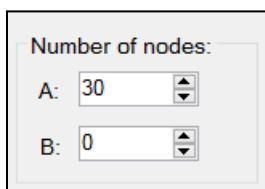
The tabbed panel on the right specifies function specific settings, such as setting for a simulation section, including the number of simulation burn-ins, number of simulation iterations and number of sample networks, etc.

The bottom left tabbed panels are interfaces designed for specifying network and attribute data involved in an MPNet session.

MPNet for ERGMs

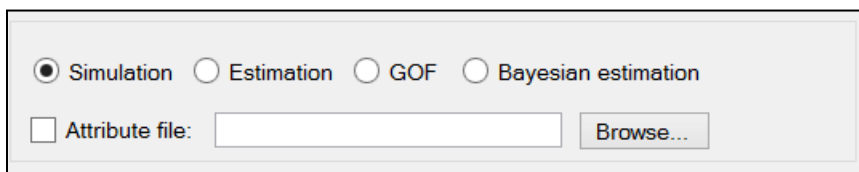
Simulating one-mode networks

Simulating one-mode networks in MPNet will only involve network A, i.e. all data settings are under the network A tab. The following settings or information are required for simulations



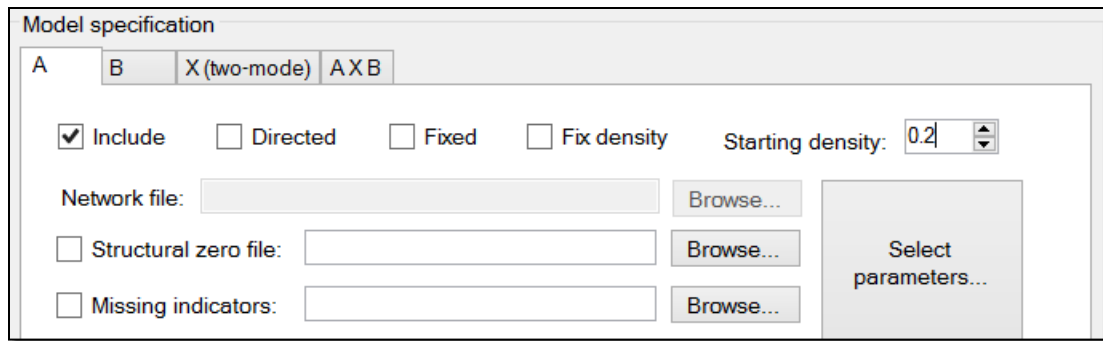
The dialog box is titled 'Number of nodes:'. It contains two spinners: 'A:' with the value '30' and 'B:' with the value '0'.

Number of nodes: Type in the number of actors in the one mode network A.



The screenshot shows a control panel with four radio buttons: Simulation (selected), Estimation, GOF, and Bayesian estimation. Below them is an unchecked checkbox for 'Attribute file:' followed by an empty text box and a 'Browse...' button.

Select "Simulation" radio button to perform model simulation.



Under the Model specification tabs, select network A, and click on the “Include” check box to include the one-mode network A in this simulation session.

Network A can be directed or non-directed. Tick the “Directed” check box if we are simulating a directed network. (Network A can also be treated as a fixed covariate in two-level network models. Click the “Fixed” check box, if we want to treat network A as fixed covariate.)

We can also simulate or estimate conditional models by ticking the “Fix density” check box which will force the network density to be fixed, i.e. addition or deletion of network ties are not possible under such condition. This option is useful in investigating properties of ERGM parameters (and for estimation – below - when unconditional model convergence is hard to reach.)

The starting density (between 0 and 1) is the density of a random network generated by MPNet as the starting graph in the simulation. With fixed density, this will be the density of all simulated graphs. If the density of the graph is not fixed, the value here will not affect the longer term result of the simulation.

Click on the “Select parameters...” button to specify the model effects (parameters) and graph statistics to be collected during the simulation. A parameter selection dialog will show up with a list of implemented ERGM configurations for user selection.

- The “include” column is a list of check boxes where if ticked, the corresponding graph statistics will be included in the simulation.
- The “Fixed” column is not implemented yet (which is designed for explore model properties when certain effect is fixed). Please ignore for now.
- The “ λ ” column provides the weighting parameter for the alternating statistics introduced by Snijders et al. (2006). They are not in use for other statistics.
- The “Value” column specifies parameter values for included effects. Here a Markov model is specified with (EdgeA = -2; Star2A = 0.5; Star3A = -0.3; and TriangleA = 1.0). We may include other graph statistics but leave the parameter values at 0 for MPNet to generate the corresponding graph statistic distributions.

There are several buttons at the bottom of the parameter selection dialog:

- “Clear All” will unselect all included configurations and set their parameter values to 0.

- “Select All” will select all available parameters implemented under the current dialog.
- “Reset to 0s” will set all parameter values to 0.
- “Exclude $\theta=0$ s” will unselect all statistics with parameter values as 0s.

The “Select All” and “Exclude $\theta=0$ ” buttons become particularly useful for model GOF testing.

Click on “OK” to finalize the parameter selection.

The screenshot shows a dialog box titled "ParameterForm" with a table of effects and their parameters. The table has five columns: "Effects", "Include", "Fixed", " λ ", and "Value". The "Include" column contains checkboxes, and the "Value" column contains spinners. Below the table are several buttons: "Clear All", "Select All", "Reset to 0s", "Exclude $\theta = 0$ ", "OK", and "Cancel".

Effects	Include	Fixed	λ	Value
EdgeA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	-2.0000
Star2A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.5000
Star3A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	-0.2000
Star4A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Star5A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
TriangleA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	1.0000
Cycle4A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
IsolatesA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
IsolateEdgesA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
ASA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
ASA2	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
ATA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
A2PA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000

Buttons: Clear All, Select All, Reset to 0s, Exclude $\theta = 0$, OK, Cancel

Structural zero files: Part of a network can be treated as exogenous especially in the cases of ERGM estimations where in some observed networks it makes empirical sense to fix part of the network and estimate the structural features of the rest of the network given the fixed part. To fix (or forbid the creation or deletion of) some of the network ties, one can apply structural zero files in MPNet. For network of size n , the structural zero file contains an n by n adjacency matrix of “1/0” indicators where 1 indicates the network tie is NOT fixed, and 0 otherwise. Please check Appendix A for structural-zero file format.

Missing indicators: MPNet can estimate ERGM with missing network data following Koskinen et al (2013). It is only used under Bayesian estimations. More detailed description can be found in the Bayesian estimation section of this user manual.

The screenshot shows a dialog box with three rows of input fields and buttons. The first row is 'Network file:' with a text box and a 'Browse...' button. The second row is 'Structural zero file:' with a checked checkbox, a text box containing 'C:\Users\Peng\Documents\MPNet\...', and a 'Browse...' button. The third row is 'Missing indicators:' with an unchecked checkbox and a text box, followed by a 'Browse...' button. To the right of these rows is a larger button labeled 'Select parameters...'.

Attribute/Dyadic covariates: Nodal attributes such as gender, age, performance, etc. can be used as covariates in ERGMs to form social selection models (SSMs) (Robins et al, 2001). Attributes or measurements on the tie variables or dyads (e.g. distance, strength) can be treated as covariate under ERGMs. MPNet can handle binary, continuous, categorical nodal attributes, as well as dyadic attributes for network ties as covariates for ERGMs.

The screenshot shows a section titled 'Attribute/Dyadic covariates'. It contains four rows, each with a checked checkbox, a dropdown menu for the number of attributes, and an 'Attribute file:' text box with a 'Browse...' button and a 'Select...' button. The rows are: 'Binary: 2', 'Continuous: 3', 'Categorical: 2', and 'Dyadic: 2'. All 'Attribute file:' text boxes contain the path 'C:\Users\Peng\Do'.

Tick the corresponding types of **attribute check boxes** to enable attribute covariates. Following the check boxes enter the number of attributes to be included in the simulation or estimation. The covariate values are stored in **Attribute files** for **binary, continuous and categorical** attribute: tab delimited text files where the first row of the file contains the names of the attributes (e.g. gender, age, etc.), and each column contains the attribute values in the same order as the nodes listed in the network matrices. The number of columns must be the same as the number of attributes specified on the MPNet interface, or MPNet will provide an error message. Attribute files for **dyadic** attribute are valued adjacency matrices start with the attribute name; then stack one upon another depending on the number of such attributes. See example attribute file formats in the Appendices.

Click on the “**Browse...**” button to specify the attribute file, and click on the “**Select...**” button to select corresponding attribute configurations. The attribute names specified in the attribute files

will be loaded into the parameter selection dialog. The parameter selection dialog follows the same format as the ERGM parameter selection dialog.

Effects	Include	Fixed	λ	Value
income_ActivityA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
income_SumA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
income_DifferenceA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
income_ProductA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
age_ActivityA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
age_SumA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
age_DifferenceA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
age_ProductA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
performance_ActivityA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
performance_SumA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
performance_DifferenceA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
performance_ProductA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000

Buttons: Clear All, Select All, Reset to 0s, Exclude $\theta = 0$, OK, Cancel

Simulation Options

Simulation/GOF Estimation

Output files

- Sample networks
- Sample degree distributions
- Generate GCD

Burn-in: 100000

Iterations: 1000000

Samples: 1000

The output files options enable us to pick sample graph matrices and sample degree distributions as tab delimited files for further analysis using other software such as SPSS or R.

Tick the “**Sample networks**” option will let the program generate each sample graph in adjacency matrix format together with some graph statistics such as degree distributions and global clustering coefficient, etc. The sample files are readable by the Pajek program for the

ease of visualizing the simulated samples. Be careful about the size of the sample ('Samples') if you check this box because it can take a long time for the computer to write out, for instance, 1000 files.

Tick the "**Sample degree distribution**" option to allow MPNet generating degree distributions of simulated samples in tab delimited output together with the standard deviations and skewness of the degree distributions.

The "**Generate GCD**" option will be implemented in a future release. It is only used in model estimation or GOF testing with generalized Cook's distances (GCDs) for each node as a measure of how extreme or important each node is in contributing towards the network structure (see Koskinen et al 2013 for more details)

Burn in is the first period of a simulation during which the simulation move towards the desired graph distribution implied by the specified parameter values. Depending on the size of the network and number of parameter values, the required burn-in can vary a lot. The larger the network, or the more parameters, the longer burn-in is needed. Examination of the output files can indicate whether the simulation has reached a consistent state and the burn-in is sufficient. For instance, the number of edges in a stationary graph distribution should vary consistently around a mean and not be consistently increasing or decreasing.

The **Iterations** box contains the number of proposed simulation updates after burn-in.

Samples expresses the number of graphs sampled from the simulation. Note that the number of iterations between graph samples is calculated as the division between the number of iterations and the number of samples to pick up.

Clicking on the **Start** button will start the simulation. Once the simulation finished, MPNet will open the simulated graph statistics output file using your default text editor. You can find all the output files in the session folder, i.e. where the MPNet session setting file is allocated. More detailed descriptions of the output files are in the next section.

Simulation Output

MPNet will generate several output files upon finishing simulating the specified model. Some of the output files are optional depending on the simulation settings described above. Here is a list of possible output files and their content information. Note that depending on the simulation settings, not all output file listed below would appear.

"**MySession_Network_A_0.txt**" is the initial or starting (Level A) graph for the simulation. In a simulation session, it will be a random graph with the user specified density, i.e. the **starting density**. It contains the adjacency matrix for the network and graph statistics such as: density, mean degree, standard deviation and skewness of the degree distribution and global clustering coefficient. For directed graphs, the output file will list statistics for in- and out- degree distributions separately. This file can be read by Pajek for visualization. The nodes will be plotted as blue squares.

“**MySession_Network_A_1001000.txt**” is the 1,001,000th simulated graph in the simulation. The output file name depends on the number of interactions and samples, and it ends with the last simulated network id. It has the same format as MySession_Network_A_0.txt.

“**MySession_Network_B_0.txt**” is the initial or starting (Level B) graph which follows the same format as MySession_Network_A_0.txt. The nodes will be plotted as red circles in Pajek. When only a unipartite graph distribution is simulated (i.e. the “Include” box is not ticked for network B), this output will not appear.

“**MySession_Network_X_0.txt**” is the initial or starting meso-level two-mode graph which list the two-mode network in edge-list format followed by some two-mode graph statistics. Level A nodes will be plotted as blue squares, and level B nodes as red circles. When the “Include” box is not ticked for network X, this output will not appear.

“**MySession_Network_M_0.txt**” contains the overall multilevel network in edge list format.

If the “**Sample networks**” option is selected under the Simulation/GOF tab, sample network files following the same format as described above will be generated by MPNet.

“**MySession.clu**” is a Pajek cluster or partition file where the partitions are defined based on the levels. Nodes in level A are in partition 0, and level B nodes are in partition 1. One may use the cluster file to plot the meso or the overall two-level network in layers under Pajek. Again, this requires the “Include” box to be ticked for network B.

“**MySession_sim.txt**” is the file opened by MPNet at the end of simulation which contains the selected graph statistics. The statistics are listed in columns separated by tabs.

“**MySession_spss.sps**” is an SPSS script to plot the scatter-plot and histogram of the simulated graph statistics using SPSS version 12.0 and above. It will read in the statistics in MySession_sim.txt.

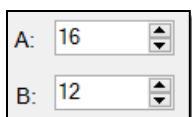
If the “**Sample degree distributions**” option is ticked under the Simulation/GOF tab, the degrees of each node will be listed as tab delimited columns in the output files

“**MySession_degreeA.txt**”, “**MySession_degreeB.txt**” and “**MySession_degreeX.txt**”.

“**MySession_model.txt**” lists the parameter/statistic names, the lambda values, and the parameter values used in the simulation.

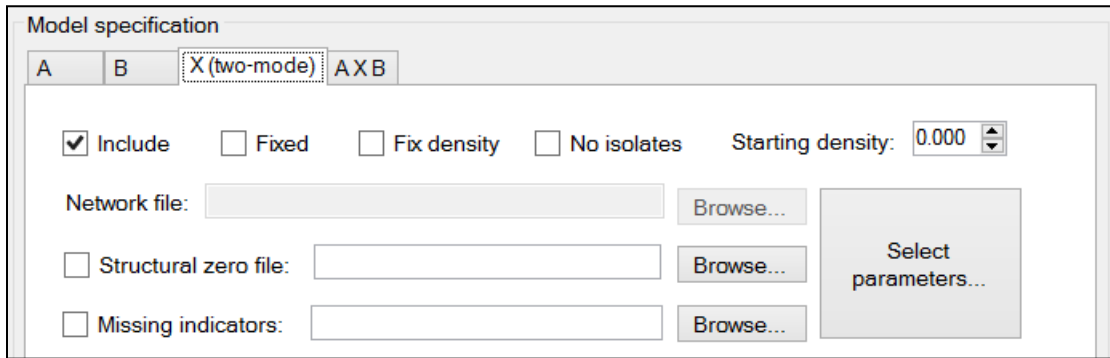
Simulating two-mode networks

To simulate two-mode networks, we need to specify the number of nodes in each modes (A and B), e.g. to simulate a 16 by 12 bipartite network



(Note that A and B here can be seen as the numbers of rows and columns in the network file for X.)

The under “Model specifications”, click on “**X (two-mode)**” tab, and tick the **Include** check box (make sure only network X is included. Inclusion of A or B networks will simulate the corresponding one-mode networks together with the two-mode network.).



Most of the model specification settings are the same as in one mode networks (A or B), except there is one more option as “No isolates”. Ticking such option in simulations or estimations will ensure all nodes in the bipartite network to have a degree at least 1.

Similar to one-mode attribute files, the bipartite attribute covariate files contains attribute values in tab delimited columns with attribute headers in the first row. However, as two sets of nodes are involved, attribute values should be listed for A nodes first followed by B nodes. For attribute that are only applicable to one set of nodes, 0s should be used for the other set of nodes, and only relevant graph statistics or parameters should be selected during simulation or estimation. For example, a 16 people (A) by 12 club (B) bipartite graph with the gender as binary attribute for people, the binary attribute file should start with “gender” as header followed by a column of 28 attribute values where the first 16 is defined by the gender of people, the rest 12 should be listed as 0s.

Other simulation settings and output files are very similar to the setting and outputs in simulations for one-mode networks as described in the previous section.

Simulating two level networks

Simulating two-level networks will require the number of nodes in both levels (A and B), and the inclusion of all three networks (A, B and X) by ticking the “Include” check boxes under each of the tabs (A, B and X) under model specifications. The within- (A and B) and cross-level (X) model parameters/statistics can be selected under the corresponding tabs. The statistics involving network ties from different networks can be selected under the “A X B” tab:

Model specification

A B X (two-mode) AXB

A and X

Structure

Binary

Continuous

Categorical

B and X

Structure

Binary

Continuous

Categorical

A, B and X

Structure

Binary

Continuous

Categorical

Attribute covariates

Binary 0 Attribute file:

Continuous 0 Attribute file:

Categorical 0 Attribute file:

Click on the “Structure” buttons to open the corresponding parameter selection dialogs with configurations representing interactions among ties across the levels. Using “A and X” as an example

ParameterForm

Effects	Include	Fixed	λ	Value
Star2AX	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
StarAA1X	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
StarAX1A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
StarAXAA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
TriangleXAX	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
L3XAX	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
ATXAX	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
EXTA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000

The dialog shares the same format as parameter selection dialogues for one or two-mode network simulations/estimations.

For multilevel social selection models, MPNet require attribute files before the user can select attribute parameters. The attribute file format follows the format as described in the section for simulating two-mode networks, i.e. tab delimited columns with headers in the first row, and

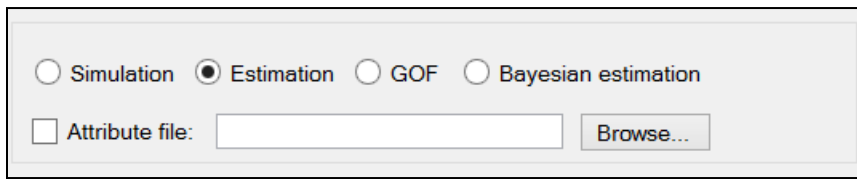
attribute values from nodes of type A followed by nodes of type B; using 0 as values for attributes that do not apply to either types of nodes.

Other simulation settings and simulation output files are similar to simulations for one- or two-mode networks as described in previous sections.

Estimation

Estimating ERGM parameters under MPNet require the user to specify the network data to be modelled, the ERGM specification and some estimation options. MPNet implements Markov Chain Monte Carlo Maximum Likelihood estimation algorithm as proposed by Snijders (2002) based on the Robbins-Monro procedure (1951). MPNet can model one- (A) or two-mode (X) networks, a combination of one and two-mode networks (A and X), and two-level networks (A, B and X).

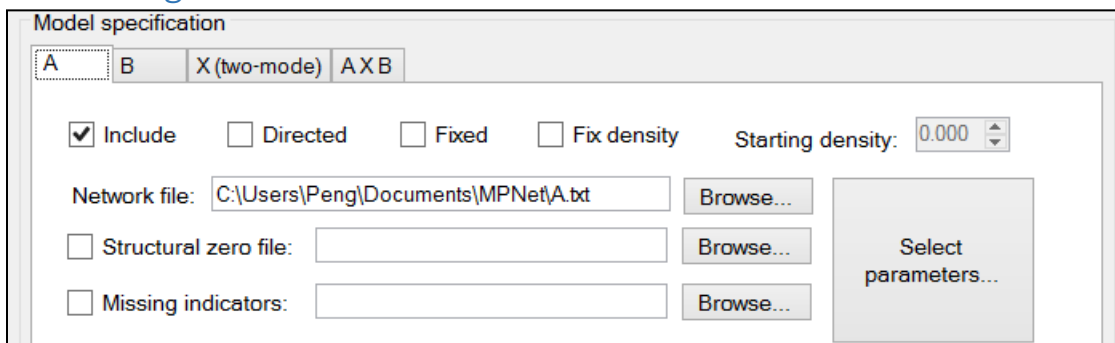
To estimate a model, start MPNet and provide a session name for a new session. Select the “Estimation” radio button. You may also continue from a previous session. Note that in contrast to PNet, you may change data set and specifications in an active or saved session.



The screenshot shows a dialog box with four radio buttons: "Simulation", "Estimation" (which is selected), "GOF", and "Bayesian estimation". Below the radio buttons is a checkbox labeled "Attribute file:" followed by a text input field and a "Browse..." button.

Upon selecting the “Estimation” option, the “Network File” text box is enabled under the “Model specification” tabs for user to specify the network data. Click on the “Browse...” button to specify the network file which has the format of a raw adjacency matrix. The number of rows and columns of the matrix must be the same as the number of nodes specified. Please refer to the Appendices for an example network file.

Estimating ERGMs for one-mode networks



The screenshot shows the "Model specification" dialog box. At the top, there are four tabs: "A", "B", "X (two-mode)", and "A X B". The "A" tab is selected. Below the tabs are several options: "Include" (checked), "Directed" (unchecked), "Fixed" (unchecked), "Fix density" (unchecked), and "Starting density:" (0.000). There are three "Browse..." buttons for "Network file:", "Structural zero file:", and "Missing indicators:". A "Select parameters..." button is located on the right side of the dialog.

To estimate models for one-mode networks, only inclusion of network A is required. Tick the “Directed” option if the network is directed. For estimations of models conditioning on the density of the network, tick the “Fix density” option. Click on “Select parameters...” to open the parameter selection dialog.

ParameterForm

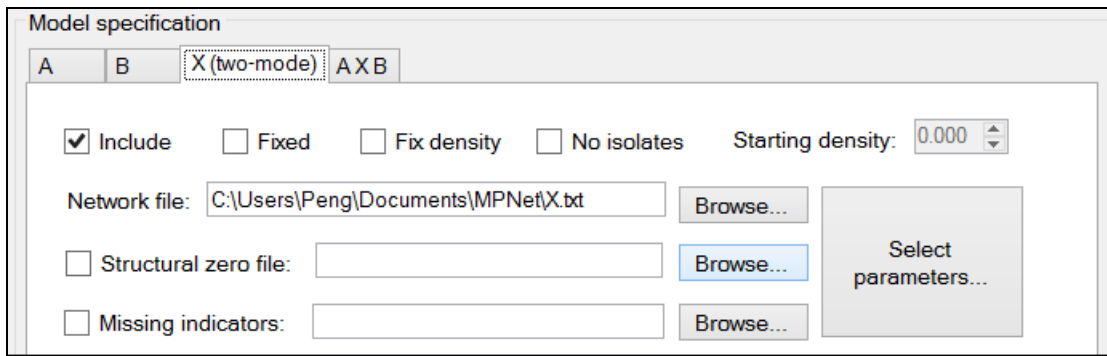
Effects	Include	Fixed	λ	Value
EdgeA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Star2A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Star3A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Star4A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Star5A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
TriangleA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Cycle4A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
IsolatesA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
IsolateEdgesA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
ASA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
ASA2	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
ATA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
A2PA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000

Clear All Select All Reset to 0s Exclude $\theta = 0$ OK Cancel

Select the effects to be included in the model under estimation by ticking the check boxes under the “**Include**” column. The “**Value**” column contains the starting parameter values. If we leave all parameters at 0s, MPNet will start estimation with an Edge or Arc parameter calculated based on the density of the network. Note that if we are estimating a model conditioning on the density of the network, please do not select Edge or Arc parameter. The model specification implemented in MPNet follows the Markov (Frank and Strauss, 1986) and the social circuit (Snijders et al, 2006; Robins et al, 2009) assumptions. Some higher order configurations are also implemented based on Pattison and Snijders (2013). Please refer to the Appendices for a list of implemented model configurations.

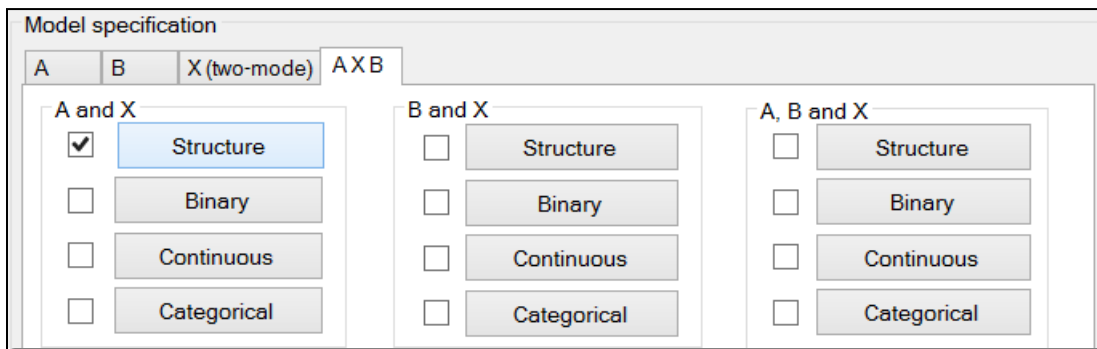
Estimating ERGMs for two mode networks

To estimate ERGMs for two mode networks, we need to specify the number of nodes in set A and set B. Then, only include network X under the Model specification tabs. The network file is a n by m rectangular matrix if we have n nodes in set A and m nodes in set B. Possible conditional ERGMs including fixing the density of the network (the “**Fix density**” option), or enforce nodes to have degrees at least 1 (the “**No isolates**” option). Click on the “Select parameters...” button to open the parameter selection dialog. The implemented two-mode configurations follows the model specifications proposed in (Wang et al, 2009; 2013) as shown in the Appendices.



Estimating ERGMs for combined one- and two- mode networks

Estimating ERGMs for a combined one- and two-mode networks require inclusion of network A, X and their corresponding network files. The within one- or two-mode network effects are the same as in separate models for network A or B. The interaction effects between network A and X can be selected under the “A X B” tab by ticking the check box next to the “**Structure**” button under “**A and X**”. The interaction configurations can be selected by click on the “Structure” button, and they follow the specifications proposed in Wang et al (2013). See the appendices for a list of configurations.



Estimating ERGMs for two-level networks

ERGMs for two-level networks require network files for all networks A, B and X. The possible within- and meso-level model configurations follow the same specifications as in models for individual one- or two-mode networks. The interaction effects among the networks A, B and X can be selected by the “Structure” buttons for the corresponding interactions under the “A X B” tab. The implemented model configurations follow Wang et al (2013), and they are listed in the Appendices.

Estimating ERGMs with nodal attributes as covariates

MPNet can model network structures with nodal attributes as covariates. The attribute file inputs are the same as described in the Simulation section. Note that separate attribute files are required for each of the networks under the Model specification tabs. The attribute file format for the interactions among networks A, B and X are the same as attribute file for bipartite network (X), i.e. columns of attribute values starting with attribute names, then attribute values

for A nodes then B nodes, with 0s represent attribute values that do not apply to either set of nodes.

Estimating conditional ERGMs

Besides using nodal attributes as covariates, we may also treat one or more of the three networks involved in the two-level network as fixed and exogenous. The research question is then about how one given network affects the structures of the other networks. For example, how club membership (fixed two-mode network X) may affect friendship (one-mode network A), or vice versa. Snijders and Van Duijn (2002) has a detailed discussions on conditional estimations for ERGMs with covariates. To fix one or more networks as covariates, tick the “Fixed” option under the tabs for the corresponding networks; and make sure no parameters are selected for fixed networks.

Model specification

A B X (two-mode) AXB

Include Directed Fixed Fix density Starting density: 0.000

Network file: C:\Users\Peng\Documents\MPNet\A.txt Browse...

Structural zero file: Browse...

Missing indicators: Browse...

Select parameters...

Options for the estimation algorithm:

The MCMCMLE algorithm has several customizable settings or options modifying which may help model convergence.

Simulation/GOF Estimation

Subphases: 5

Gaining factor: 0.010

Multiplication factor: 10

Iterations in phase 3: 500

Max. estimation runs: 1

Do GOF at convergence 500

Generate GCD at convergence

Subphases: Each sub-phase refines the parameter values, but more sub-phases do not guarantee convergence. The default value is 5. If a good set of starting parameter values is available, a smaller number of sub-phases may help reduce time required for the estimation.

Gaining Factor is a multiplier that affects the sizes of parameter updates. It is halved after each sub-phase to refine the parameter values as the model converges. The default a-value is 0.01. Smaller a-values may be used, if a good set of starting parameter values is available.

Multiplication Factor is a multiplier that determines the number of simulation iterations between network samples during estimations (other factors including the size and the density of the network). The larger the multiplication factor, the greater the distances between network samples, and hence the smaller the auto-correlations between samples which may yield a more reliable model. Networks with greater number of nodes may require greater multiplication factors to achieve model convergence. However, greater multiplication factor will also result in longer estimation time. The default value is 10 but for directed networks and larger networks a larger multiplication factor is generally needed. It is rare that estimation requires a larger multiplication factor than 100. If the SACF (see OUTPUT below) is greater than 0.4 you will need to increase the multiplication factor.

Iterations in phase 3: In phase 3, MPNet simulates network graphs using estimated parameters obtained from phase 2, and produces t-statistics based on comparisons between the simulated graph distribution and the observed graph statistics. The default value is 500 samples. Note that the number of simulation updates between samples is the same as in the estimation which is determined by the network size, network density, and the multiplication factor.

Max. estimation runs: As default, the program will perform one estimation and stop. Multiple estimations runs in sequence can be performed such that each new run uses the parameter values obtained from the end of the previous run. An improved parameter estimate may be obtained as the new estimation may start with a set of parameter values closer to convergence. MPNet will stop and ignore the subsequent estimation runs as soon as the model is converged; otherwise the maximum number of estimation runs will be performed.

Do GOF at convergence: PNet can perform a goodness of fit (GOF) examination once the model under estimation has converged. The GOF output file will be located in the session folder. See detailed description of the GOF test in the next section.

Click on **Start** button to start the estimation. Upon completion of the estimation, MPNet will show you whether the model has converged or not, and open the estimated model with the default text editor.

After first estimation run, the **Update** button will be enabled. It is used when you want to start the next estimation run with previous estimated parameters so that you may start the new estimation from a better set of parameter values.

Estimation Output

For a MPNet estimation session with session name “MySession”, MPNet will generate the following output:

“MySession_Network_A_0.txt”, “MySession_Network_B_0.txt”, “MySession_Network_X_0.txt”, and “MySession_Network_M_0.txt” are the networks that have been modelled in adjacency matrix format for networks A and B, and edge list format for networks X and the overall two-level network M. The content of the files are the same as output from a simulation session as described previously.

“MySession__model.txt” contains the model specification during the estimation session, and the starting parameter values.

“**MySession_update.txt**” contains the model specification during the estimation session, and the most recent parameter estimates. MPNet uses this file for updating parameter values when the **Update** button is clicked.

“**MySession_est.txt**” contains all parameter estimates throughout the entire estimation session, i.e. any estimation runs under this session name will be appended towards the bottom of this file. The most recent estimates are listed at the end of the file. The “Estimation results” section of the output lists the effect names, parameter estimates, estimated standard errors, t-ratio for convergence test, and sample autocorrelation functions (SACF) for a reliability check. They are listed in tab delimited columns which you may copy and paste into table format (e.g. In Excel). Here is an example output:

<i>Effects</i>	<i>Lambda</i>	<i>Parameter</i>	<i>Stderr</i>	<i>t-ratio</i>	<i>SACF</i>	
EdgeA	2	-3.3993	1.421	-0.092	0.065	*
ASA	2	1.028	0.505	-0.078	0.064	*
ATA	2	-0.06	0.249	-0.069	0.077	
A2PA	2	-0.2094	0.251	-0.071	0.061	

When all t-ratios in the estimated model have absolute values smaller than 0.1, we consider the model is well converged. SACFs smaller than 0.4 indicates there are sufficient distances between simulated samples during the estimation, hence the model is more reliable. We consider the absolute value of a parameter estimate greater than twice the size of the estimated standard errors as significant, and they are indicated by “*”.

The variance covariance matrix of the estimated parameters is listed at the end of the estimation output. This may be useful for Bayesian estimations described below.

Goodness of Fit

Once a converged ERGM is obtained, the model goodness of fit (GOF) can be tested by comparing simulated graph statistics of the estimated model against the network that has been modelled. The graph statistics are not limited to the ones that are already included in the model, but also a greater range of configurations representing the network structure. Click on the GOF radio button to specify a model GOF session.

Simulation
 Estimation
 GOF
 Bayesian estimation

Attribute file:

Goodness of Fit Setup

Most settings for Goodness of Fit are the same as in Simulation, except the observed network and parameter values are required. The observed network file can be specified the same as in Estimation. The parameter values can be typed under the corresponding model parameter

selection dialog; or by using the “**Update**” button if the GOF session is for the most recently converged model under Estimation.

During model parameter selection, click on the “**Select All**” button will include all implemented statistics in the GOF simulation.

The screenshot shows a window titled "ParameterForm" with a table of effects and their parameters. The table has five columns: Effects, Include, Fixed, λ , and Value. The "Include" column contains checkboxes, and the "Fixed" column contains checkboxes. The λ and Value columns contain numerical values with spinners. At the bottom of the window, there are buttons for "Clear All", "Select All", "Reset to 0s", "Exclude $\theta = 0$ ", "OK", and "Cancel".

Effects	Include	Fixed	λ	Value
EdgeA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	-3.3993
Star2A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Star3A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Star4A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Star5A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
TriangleA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Cycle4A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
IsolatesA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
IsolateEdgesA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
ASA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	1.0280
ASA2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
ATA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	-0.0600
A2PA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	-0.2094

Other GOF settings are the same as in Simulations. At the end of the GOF simulation, MPNet will calculate t-ratios for all included graph statistics. For configurations that are already included in the model, t-ratios smaller than 0.1 in absolute value reconfirm the model is converged (if there is a discrepancy between the estimation convergence statistics and the GOF t-ratios, you may have to increase the ratio of the number of iterations to the number of samples (See more detailed discussions in Koskinen and Snijders (2012) Chapter 12 of the book on ERGMs). For other statistics, t-ratios smaller than 2.0 in absolute values suggest adequate fit to that particular graph feature. T-ratios greater than 2.0 standard deviation units from the mean indicate poor fit to the data on that particular graph feature.

Goodness of Fit Output

Besides the same sets of output files as in Simulation session, MPNet will generate a GOF result file namely “**MySession_gof.txt**”. It contains a tab delimited table where the first column lists the configurations included in the GOF simulation; the second column contains the counts of the configurations in the observed network; the third column contains the means of the simulated graph statistic distribution; the third column has the standard deviations; the fourth column shows the t-ratios; and the last column shows “#” signs for t-ratios that are greater than 2.0 in

absolute values indicating poor fit to the corresponding statistics. Below is an example output for a GOF test of a one-mode network A.

Besides the user selected configurations, MPNet also includes some global network measurements as part of the output, including the standard deviation and skewness of the degree distributions and the global clustering coefficient. The Mahalanobis distance shown at the end of the file is an overall heuristic measure of model GOF, taking into account the covariance of the included statistics. Smaller Mahalanobis distances indicate better fit to the dat. (Wang et al 2009). Mahalanobis distance should not be tested with standard chi-squared statistics: in this context, it is an indicative measure. If two models have the same configurations in the GOF output, then the one with the lower Mahalanobis distance is a better fit.

<i>Configuration</i>	<i>Observed</i>	<i>Mean</i>	<i>StdDev</i>	<i>t-ratio</i>
EdgeA	22	22.39	3.98	-0.09
Star2A	71	65.84	24.58	0.21
Star3A	62	59.14	36.77	0.07
Star4A	30	36.16	35.31	-0.17
Star5A	8	15.68	24.14	-0.31
TriangleA	7	5.33	3.65	0.45
Cycle4A	16	10.78	9.47	0.55
IsolatesA	2	0.32	0.59	2.84 #
IsolateEdgesA	0	0.02	0.15	-0.15
ASA	46.56	43.62	12.82	0.22
ATA	17.75	13.64	8.13	0.50
A2PA	57.12	56.06	17.34	0.06
AETA	33.62	24.39	19.15	0.48
stddev_degreeA	2.51	2.22	0.33	0.87
skew_degreeA	1.06	1.30	0.15	-1.60
clusteringA	0.29	0.22	0.09	0.77
Mahalanobis distance = 193				

Bayesian estimation

MPNet implements a version of the Bayesian estimation algorithm proposed by Camio and Friel (2009) as specified in Koskinen et al (2013). Instead of obtaining the point estimates as in MCMCMLE, the Bayesian estimation generates the posterior distributions of the model parameters. In lieu of MLEs and standard errors, point estimates and measures of uncertainty are calculated as averages and standards errors of this distribution respectively. The approximations of Phase 3 are thus not necessary. However, as the posterior is generated using an iterative MCMC algorithm it is important to assess ‘mixing’, i.e., how well the algorithm samples from the posterior..

Simulation
 Estimation
 GOF
 Bayesian estimation

Attribute file:

Select “Bayesian estimation” from the main user interface, the Bayesian estimation options will be enabled. The same as in Estimation, network data input file and model parameter selections can be specified under the “Model specification” tabs. The setting options for Bayesian estimations are different from Estimations as shown on the right side of the user interface.

Bayesian estimation settings

Simulation/GOF Estimation

Parameter burn-in:

Proposal scaling:

Multiplication factor:

MCMC Sample size:

Max. estimation runs:

Do GOF at convergence

Generate GCD at convergence

Bayesian estimation options

Maximum lag (SACF):

Scaled identity matrix
 Combined simulation
 Nonconditional simulation
 Covariance file:

Parameter burn-in: similar to burn-in for simulations, the starting parameters may be considered extreme from the posterior parameter distribution. The burn-in will discard the specified number of parameter updates at the beginning of the estimation.

Proposal scaling: similar to a-values in maximum likelihood estimations (MLEs), the proposal scaling (or ‘step-size constant’; Tierney, 1994) is a multiplier for the sizes of parameter updates. Greater scaling will cover greater range for parameter proposals; however, greater scaling may also reduce the number of accepted parameter proposals as part of the posterior. The proposal distribution in the Metropolis algorithm is $N_p(\theta, S)$, where θ is the current value, $S=c/\sqrt{(1+p)\Sigma}$, and Σ is some estimate of the variance-covariance matrix of the posterior distribution. In this expression c is the ‘proposal scaling’.

Multiplication factor: is the same as in MLEs, and determines the number of iterations to be simulated in order to generate a network given a proposed parameter. The multiplication factor generally speaking should be about three times as large as for the non-Bayesian algorithm.

MCMC Sample size: is the number of parameter proposals. If all parameter proposals are accepted, the posterior will contain this number of parameter sets. (Note that achieving 100% acceptance of proposed parameter values are not the goal of the estimation. Acceptance of all proposals suggesting the resulting posterior may only cover part of the actual posterior, and a greater proposal scaling may be required.) The larger the MCMC Sample size, the better the precision of the posterior mean and standard deviation (given a fixed acceptance rate).

Max. estimation runs, Do GOF at convergence and Generate GCD at convergence: they are not applicable in Bayesian estimations.

Maximum lag (SACF): determines the largest lag (distance) for which the sample autocorrelation function for the estimated posterior is. In order for the effective sample size (ESS) to be reliable, the autocorrelation at the Maximum lag has to be sufficiently close to zero (as a rule of thumb, smaller than 0.05 in absolute value). The lag at which the SACF value is approximately zero is gives the number of parameter draws you need to discard in-between every successive parameter value that you base your posterior inference on. For example, if the SACF at lag 100 is approximately zero, then you need an MCNC sample size of 100,000 to get 1000 independent draws from the posterior distribution. If the SACF at lag 100 is greater than, say, 0.4 you need to modify the parameter proposals by increasing the 'proposal scaling'.

There are several possible matrices we may apply to Bayesian estimation which is used for determining the 'direction' of parameter updates. There are four options approximating Σ , which is used to set the proposal variance-covariance matrix through $S=c/\sqrt{(1+p)} \Sigma$.

Scaled identity matrix: An identity matrix that implies no preferred direction of updates. The directions of updates are solely based on the difference statistics between the observed graph and the simulated samples.

Combined simulation: only applicable for Bayesian estimations with missing data as in Koskinen et al (2013). See more detailed instructions in the next section.

Nonconditional simulation: the differences between the observed graph and the simulated samples are refined by a covariance matrix generated based on a simulation with the starting parameters. (This is an analogous procedure to the one employed in Phase 1 of the non-Bayesian estimation)

Covariance file: A user defined covariance matrices of the parameters are used to refine the direction of parameter updates. The covariance file is a p by p matrix if there are p parameters in the model. Such covariance file may be obtained based on previous estimations of the same model. MPNet generates such files at the end of estimations with file name e.g. "MySession_varcov.txt". If this estimate of Σ is close to the true posterior variance-covariance matrix, the proposal scaling c should be in the range of 0.5 to 4.

Bayesian estimation outputs

Bayesian estimation summarize the estimation results in two file

"MySession_posterior_bayesian.txt" contains the estimated posterior. It has p tab delimited columns for models with p parameters; each column contains the accepted parameter values in

the posterior with parameter names on the first row. The posterior can be plotted by software such as R, Excel, etc. Plotting the parameter draws across iterations gives a quick indication of whether the algorithm performs well. If there is ‘drift’, better initial values and a longer burn-in may be needed. If the parameters move slowly (there is great autocorrelation) between different values, a larger proposal scaling is needed.

“**MySession_est_bayesian.txt**”: As shown in the example below, the output summarizes the parameter posterior distributions in terms of means and standard deviation, followed by the covariance matrix of the parameters. Since there is no convergence test for Bayesian estimation, the reliability of the generated posterior is indicated by the sample autocorrelation functions (SACFs) for different lags up to the user defined **maximum lag**. The maximum lag should be set to the lag for which SACF is approximately zero, then the effective sample size (ESS) can be trusted (the calculation of ESS here is based only on lags up to and including the max lag). Increasing the proposal scaling will decrease the SACF. If you use ‘scaled identity’ and the SACF differs a lot between parameters, change to option ‘Covariance file’.

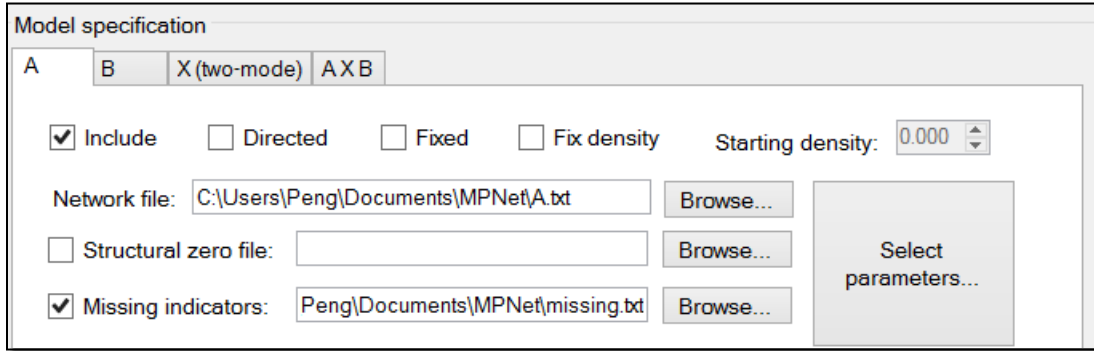
Acceptance rate: 0.38						
Estimation results						
Effects	Lambda	PostMean	Stddev			
EdgeA	2	-2.513	1.173	*		
ASA	2	0.4911	0.492			
ATA	2	0.0097	0.25			
A2PA	2	-0.0568	0.317			
Covariance matrix						
1.3752	-0.3789	0.0932	-0.1083			
-0.3789	0.2423	-0.0467	-0.0736			
0.0932	-0.0467	0.0624	-0.0139			
-0.1083	-0.0736	-0.0139	0.1004			
SACF						
Effect	10	30	50	70	90	ESS(100)
EdgeA	0.993	0.98	0.969	0.96	0.951	51
ASA	0.976	0.935	0.895	0.86	0.825	55
ATA	0.888	0.711	0.572	0.466	0.378	82
A2PA	0.953	0.882	0.824	0.779	0.73	60

Bayesian estimations with missing network data

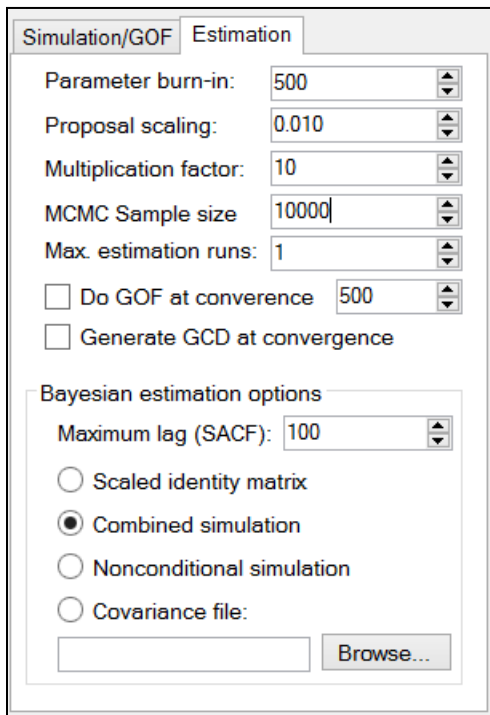
Following Koskinen et al (2013), MPNet implements Bayesian estimations with missing network ties. The assumption is that we have the information about which network ties are missing, and the missing ties follow the same social processes as the observed part of the network. The current implementation of MPNet can only estimate models for one-mode networks (A). Future release will extend the method to two-mode and two-level networks.

The estimation settings mostly follow settings under Bayesian estimation, except it requires a **missing indicator file**, and the use of the “**Combined simulation**” estimation option.

The missing indicator file has the same format as the network file, i.e. an adjacency matrix of 1s and 0s, where 1s indicate ties that are part of the missing data and 0s indicate non-missing ties. The missing indicator file can be specified under Model specification tabs by ticking the “**Missing indicators**” check box, and clicking on the “**Browse...**” button.



It is advisable to start estimation with ‘**Scaled identity matrix**’ but if a short estimation round yields reasonable preliminary estimates, better performance of the algorithm may be had from using ‘**Combined simulation**’ in the options under Bayesian estimation.

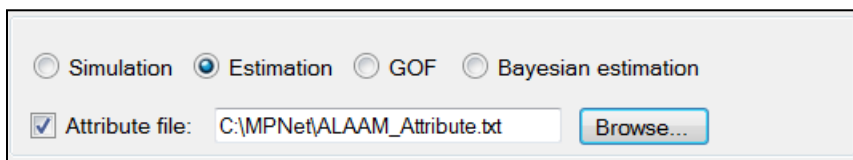


The output of Bayesian missing data estimation is the same as in Bayesian estimations without missing data.

MPNet for ALAAMs

Auto-Logistic Actor Attribute Models (ALAAMs) treat a given network structure as exogenous, and models individuals' outcomes (or attributes) as a result of their network positions and other attributes. MPNet is capable of simulating and estimating ALAAMs for one-mode, two-mode, and multilevel networks. ALAAMs shares similar model constructs as ERGMs. ERGMs model network structure with nodal attributes as covariates, whereas ALAAMs model nodal attributes using network structure as covariates. All ERGM simulation and estimation and goodness of fit (GOF) test algorithms and techniques are equally applicable to ALAAMs, and the implementation in MPNet share the same technical approaches. The model specifications share similar graph configurations/statistics as ERGM, plus some endogenous interactions among nodal outcome variables.

To run ALAAM sessions in MPNet, the only differences on the user interface is the selection of "Attribute file" check box, as shown here:

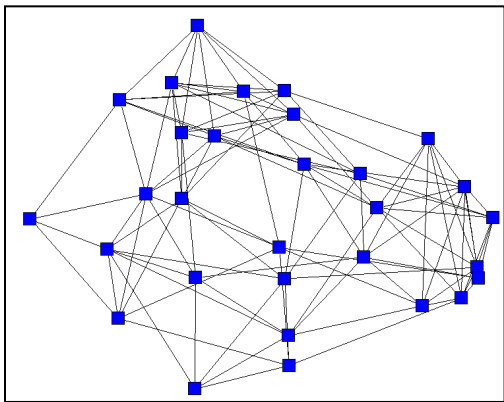


The screenshot shows a software interface with four radio buttons: 'Simulation', 'Estimation', 'GOF', and 'Bayesian estimation'. The 'Estimation' radio button is selected. Below the radio buttons, there is a checked checkbox labeled 'Attribute file:' followed by a text input field containing the path 'C:\MPNet\ALAAM_Attribute.txt' and a 'Browse...' button.

To simulate ALAAMs, select the Simulation radio button, and the Attribute file is not required (but make sure the checkbox is ticked). Whereas Estimation and GOF require the Attribute file to be specified (by clicking on the Browse... button). The content of the Attribute file should start with the outcome variable name as header, followed by a single column of "1/0" indicating each node's outcome in the same order as they are in the network files.

Simulating ALAAMs for one-mode networks

Simulating ALAAMs will require network matrix files as in ERGMs. Here we use a simulated non-directed network with 30 nodes as an example (the ERGM used to simulate this network is based on our simulation examples in part 4.1). Here is a visualization of the network:



Set the number of A nodes to be 30

Number of nodes:
 A: 30
 B: 0

Make sure **Simulation**, and **Attribute file** are selected.

Simulation Estimation GOF Bayesian estimation
 Attribute file:

Leaving the “Attribute file” empty, MPNet will generate a random starting attribute vector with density specified under the A tab. For example, a 30 node attribute vector with starting density set at 0.2 will have $30 \times 0.2 = 6$ nodes coded as 1 in the starting attribute vector.

Model specification
 A B X (two-mode) A X B
 Include Directed Fixed Fix density Starting density: 0.200
 Network file: C:\MPNet\NetA30.txt
 Structural zero file:
 Missing indicators:

Under the Model specification tabs, select network A, and click on the “Include” check box to include the one-mode network A in this simulation session. Specify the network A file. Network A can be directed or non-directed. Tick the “Directed” check box if the given network is directed. (Note that the attributes for nodes in network A can also be treated as a fixed covariate in two-mode or two-level network models. Click the “Fixed” check box, if we want to treat attribute values for A nodes as fixed covariate.)

We can also simulate or estimate conditional ALAAMs by ticking the “Fix density” check box which will force the attribute density to be fixed (here at 0.2).. This option is useful in investigating properties of ALAAM parameters (and for estimation – below - when unconditional model convergence is hard to reach.)

Click on the “Select parameters...” button to specify the model effects (parameters) and graph statistics to be collected during the simulation. A parameter selection dialog will show up with a list of implemented ALAAM configurations for user selection. The layout of the parameter selection panels are the same as under ERGMs.

Here, we simulate an ALAAM (DensityA = -6.8; ActivityA = 0.5; and ContagionA = 1.0). We may include other graph statistics but leave the parameter values at 0 for MPNet to generate the corresponding attribute statistic distributions.

Effects	Include	Fixed	λ	Value
DensityA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	-6.8000
ActivityA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	0.5000
ContagionA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00	1.0000
Ego-2StarA	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Alter-2Star1A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
Alter-2Star2A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
TA1A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
TA2A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000
TA3A	<input type="checkbox"/>	<input type="checkbox"/>	2.00	0.0000

Buttons: Clear All, Select All, Reset to 0s, Exclude $\theta = 0$, OK, Cancel

ALAAMs with **structural zeros** or **missing values** will be implemented in a coming version of MPNet.

Attribute covariates: Similar to ERGMs, nodal attributes such as gender, age, performance, etc. can be used as covariates in ALAAMs to test how other attributes may contribute to the dependent attribute variable besides the network structure. The current implementation of MPNet can handle binary, continuous, categorical nodal attributes. (Please refer to Appendices for a complete list of implemented model configurations).

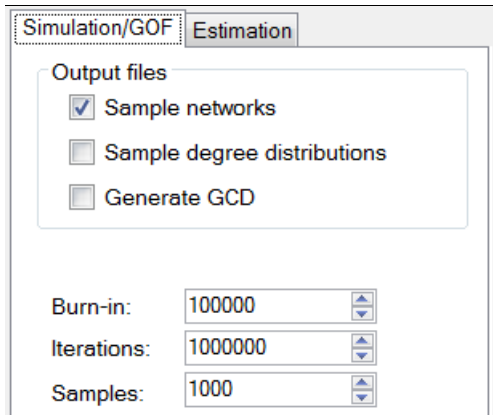
Attribute/Dyadic covariates

<input checked="" type="checkbox"/> Binary:	1	Attribute file:	C:\MPNet\MySess	Browse...	Select...
<input checked="" type="checkbox"/> Continuous:	1	Attribute file:	C:\MPNet\MySess	Browse...	Select...
<input checked="" type="checkbox"/> Categorical:	1	Attribute file:	C:\MPNet\MySess	Browse...	Select...
<input type="checkbox"/> Dyadic:	0	Attribute file:	\LAAM_Attribute.txt	Browse...	Select...

Tick the corresponding types of **attribute check boxes** to enable attribute covariates. Following the check boxes enter the number of attributes to be included in the simulation or estimation. The covariate values are stored in **Attribute files** for **binary, continuous and categorical** attribute: tab delimited text files where the first row of the file contains the names of the attributes (e.g. gender, age, etc.), and each column contains the attribute values in the same order as the nodes listed in the network matrices. The number of columns must be the same as the number of attributes specified on the MPNet interface, or MPNet will provide an error message. See example attribute file formats in the Appendices.

Click on the “**Browse...**” button to specify the attribute file, and click on the “**Select...**” button to select corresponding attribute configurations. The attribute names specified in the attribute files will be loaded into the parameter selection dialog.

Simulation Options



The screenshot shows a dialog box titled "Simulation/GOF Estimation". It has two tabs: "Simulation/GOF" (selected) and "Estimation". Under the "Simulation/GOF" tab, there is a section labeled "Output files" containing three checkboxes: "Sample networks" (checked), "Sample degree distributions" (unchecked), and "Generate GCD" (unchecked). Below this section are three spinners: "Burn-in:" with a value of 100000, "Iterations:" with a value of 1000000, and "Samples:" with a value of 1000.

Tick the “**Sample networks**” option will let the program generate each sample attribute vectors in a format that can be read by Pajek as portion files for visualization. Be careful about the size of the sample (‘Samples’) as it can take a long time for the computer to write out, for instance, 1000 files for 1000 samples.

The “**Sample degree distribution**” option and “**Generate GCD**” option does not apply to ALAAMs

Burn-in is the first period of a simulation during which the simulation move towards the desired attribute distribution implied by the specified parameter values.

The **Iterations** box has the number of proposed simulation updates after the burn-in.

Samples is the number of graphs sampled from the simulation. Note that the number of iterations between graph samples is calculated as the division between the number of iterations and the number of samples to pick up.

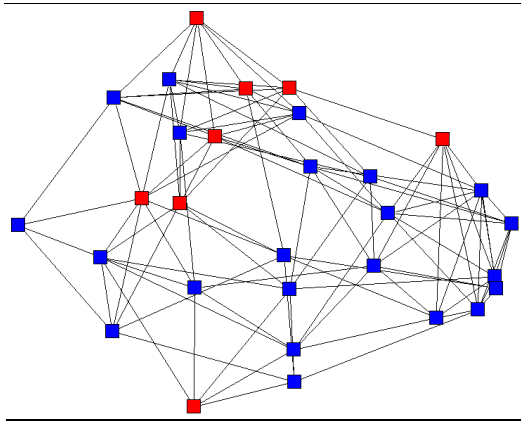
Clicking on the **Start** button will start the simulation. Once the simulation finished, MPNet will open the simulated graph statistics output file using your default text editor. You can find all the output files in the session folder, i.e. where the MPNet session file is allocated. More detailed descriptions of the output files are in the next section.

Simulation Output

MPNet will generate several output files upon finishing simulating the specified model. Some of the output files are optional depending on the simulation settings described above. Here is a list of possible output files and their content information. Note that depending on the simulation settings, not all output file listed below would appear.

“**MySession_Network_A_0.txt**” is the one-mode network covariate (Level A) graph which can be plotted by Pajek.

“**MySession_ALAAM.clu**” is the starting attribute vector. In a simulation session, it will be a random binary vector with the user specified density, i.e. the **starting density**. This file can be read by Pajek as a partition file for visualization. The following is a visualization of the simulated attribute vector plotted as a Pajek partition file, where red squares are coded as 1s (having the attribute), and blue squares are coded as 0s (without the attribute).



“**MySession_ALAAM_1001000.clu**” is the 1,001,000th or the last simulated attribute vector.

“**MySession_Network_B_0.txt**” is the (Level B) network covariate which follows the same format as MySession_Network_A_0.txt..

“**MySession_Network_X_0.txt**” is the cross-level two-mode graph covariate which list the two-mode network in edge-list format followed by some two-mode graph statistics. Level A nodes will be plotted as blue squares, and level B nodes as red circles. When the “Include” box is not ticked for network X, this output will not appear.

“**MySession_Network_M_0.txt**” contains the overall multilevel network in edge list format.

If the “**Sample networks**” option is selected under the Simulation/GOF tab, sample attribute vector files following the same format as described above will be generated by MPNet.

“**MySession.clu**” is a Pajek cluster or partition file where the partitions are defined based on the levels. Nodes in level A are in partition 0, and level B nodes are in partition 1. One may use the cluster file to plot the meso or the overall two-level network in layers under Pajek. Again, this requires the “Include” box to be ticked for network B.

“**MySession_sim.txt**” is the file opened by MPNet at the end of simulation which contains the selected graph statistics. The statistics are listed in columns separated by tabs.

“**MySession_spss.sps**” is an SPSS script to plot the scatter-plot and histogram of the simulated graph statistics using SPSS version 12.0 and above. It will read in the statistics in MySession_sim.txt.

If the “**Sample degree distributions**” option is ticked under the Simulation/GOF tab, the degrees of each node will be listed as tab delimited columns in the output files

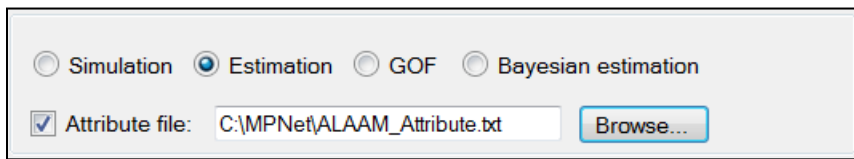
“**MySession_degreeA.txt**”, “**MySession_degreeB.txt**” and “**MySession_degreeX.txt**”.

“MySession_model.txt” lists the parameter/statistic names, the lambda values, and the parameter values used in the simulation.

ALAAM Estimation

Estimating ALAAM parameters requires users to specify the nodal outcome variable data to be modelled, the ALAAM specification and some estimation options. MPNet uses the same estimation algorithm as proposed by Snijders (2002) based on the Robbins-Monro procedure (1951) for ERGM as to ALAAMs, since the two models shares almost identical mechanisms. MPNet can model one- (A) or two-mode (X) networks, a combination of one and two-mode networks (A and X), and two-level networks (A, B and X).

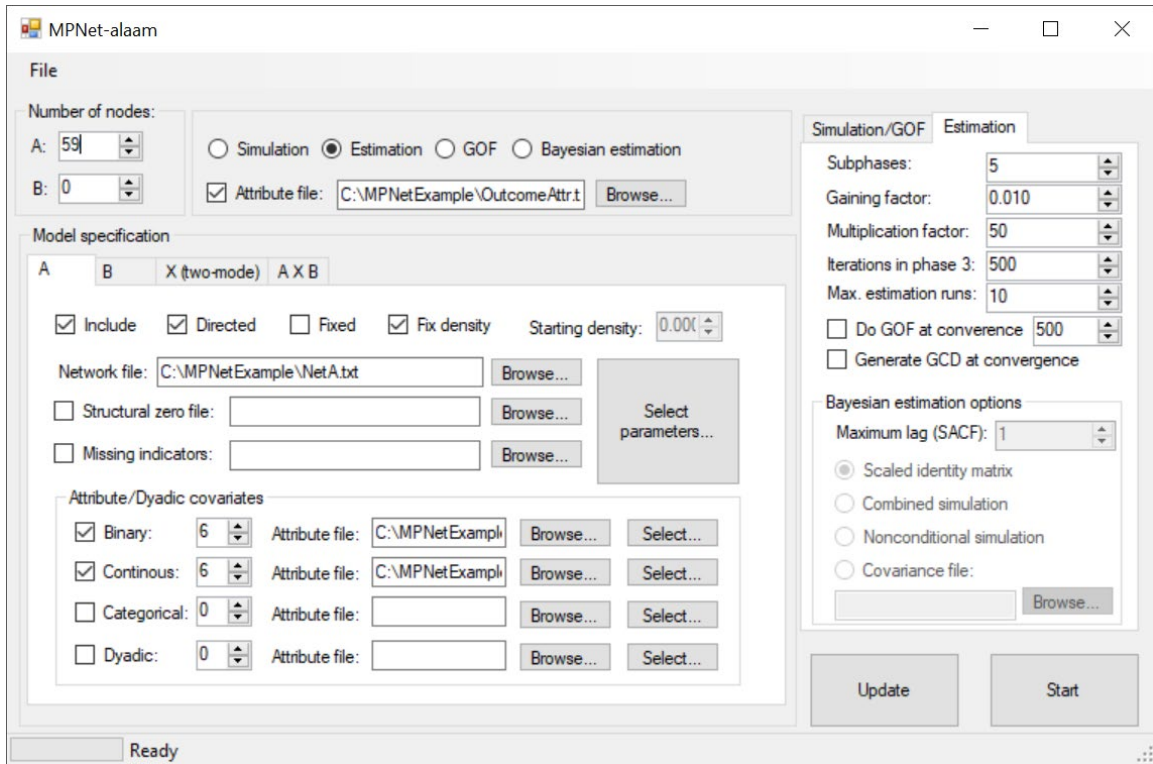
To estimate a model, start MPNet and provide a session name for a new session. Select the “Estimation” radio button. You may also continue from a previous session.



Upon selecting the “Estimation” option, the “Attribute file” becomes effective as it contains the observed outcomes. Attribute file should start with attribute name as header, then a single column of “1” or “0” as outcome indicators. Please refer to the Appendices for an example network file.

The rest of ALAAM set ups are almost identical to ERGM estimations, except the model specifications are tailor to ALAAMs instead of ERGMs. Depending on whether the ALAAM is for one-mode network (A) or two-mode network (X) or two-level network (A, X, B), users can specify the network files (as covariate for ALAAM) as well as attribute files in identical ways as for estimating ERGMs. The estimation configurations are identical to ERGMs. ALAAMs are easier to converge in comparison to ERGMs on the same set of data, since there are only n outcome variables (n is the number of nodes) in ALAAM, but $n(n-1)$ for ERGMs.

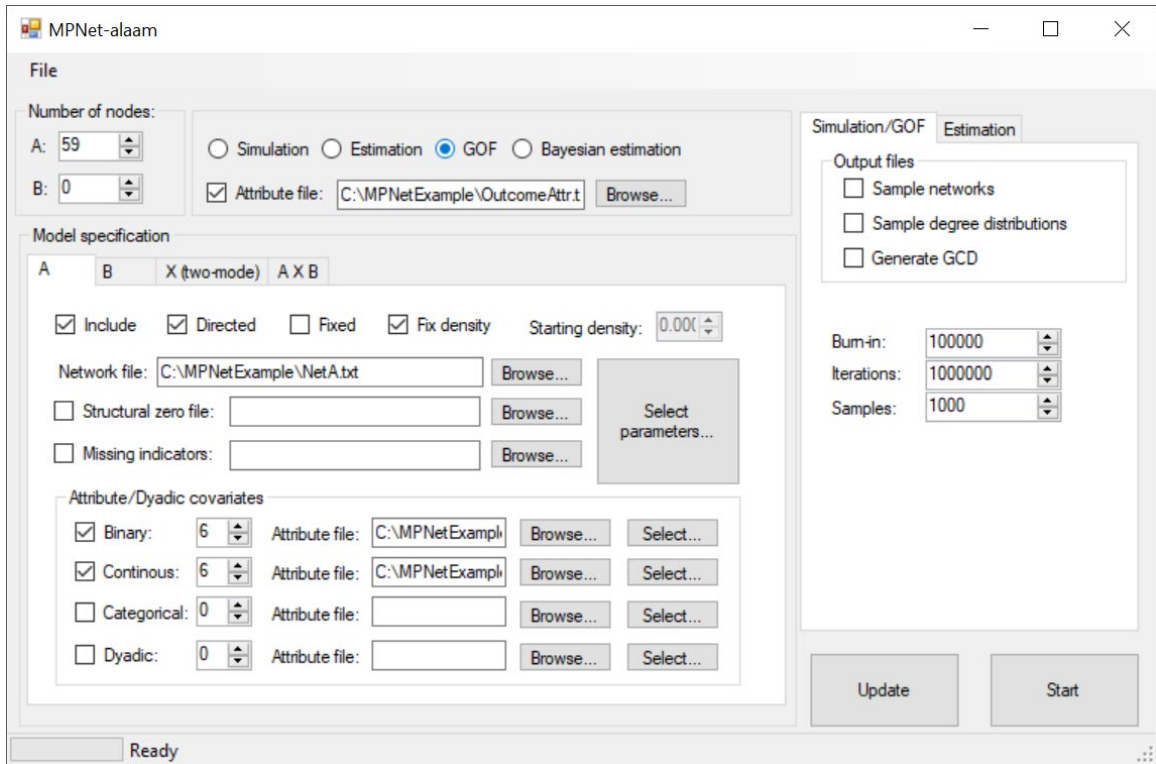
Similar to ERGMs, we can fit conditional ALAAMs by fixing densities where the density here refers to the number of “1” outcomes relative to the number of outcomes (n). This is enabled in the same way as in ERGMs by ticking the “Fix density” checkbox.



For two-mode or two-level networks, as two sets of nodes are involved, the “Attribute file” should contain a single vector of “1/0” after a header, and there should be (n+m) elements in the vector where n is the number of nodes in network A, and m is the number of nodes in network B. We usually have outcomes defined on only one set of nodes (e.g. network A), therefore the outcome should be coded as “0” for the other set of nodes (in network B). The attribute file should be header followed by “1/0” for nodes in network A, then all “0”s for nodes in network B. Hint: You can indeed model A and B at the same time if a common outcome can be defined on both sets of node. Furthermore, you can indeed model two different kind of outcomes for A and B simultaneously (e.g. staff performance and the performance of their affiliated projects) which makes ALAAM rather interesting.

ALAAM Goodness of Fit Test

To test the Goodness of Fit (GOF) of a given ALAAM, we use identical approach as in GOF tests for ERGM, where by comparing simulated graph/outcome statistics of the estimated model against the observed outcomes that has been modelled. Again, the graph statistics are not limited to the ones that are already included in the model, but also a greater range of configurations representing the interactions among the outcome variables and how network structure affect the nodal outcomes. Click on the GOF radio button to specify a model GOF session.



The setup of GOF for ALAAM is the same as ERGM, except that one must also supply the “Attribute file” for ALAAM as the observed outcome vector. You can keep this unchanged, if you have just obtained a converged model from an Estimation session.

Parameters can be entered manually, or be uploaded to MPNet by clicking on “Update” button following an Estimation session with a converged model. You may choose “Select All” or any of the configurations/statistics listed in the parameter selection panel. And the GOF simulation configuration follows identical configurations as in ERGM simulations or GOF.

Most settings for Goodness of Fit are the same as in Simulation, except the observed network and parameter values are required. The observed network file can be specified the same as in Estimation. The parameter values can be typed under the corresponding model parameter selection dialog; or by using the “**Update**” button if the GOF session is for the most recently converged model under Estimation.

At the end of the GOF simulation, MPNet will calculate t-ratios for all included graph statistics. For configurations that are already included in the model, t-ratios smaller than 0.1 in absolute value reconfirm the model is converged (if there is a discrepancy between the estimation convergence statistics and the GOF t-ratios, you may have to increase the ratio of the number of iterations to the number of samples (See more detailed discussions in Koskinen and Snijders (2012) Chapter 12 of the book on ERGMs). For other statistics, t-ratios smaller than 2.0 in absolute values suggest adequate fit to that particular graph feature. T-ratios greater than 2.0 standard deviation units from the mean indicate poor fit to the data on that particular graph feature.

References

- Caimo, A., & Friel, N. (2011). Bayesian inference for exponential random graph models. *Social Networks*, 33(1), 41-55.
- Daraganova, G., Robins, G. Auto-logistic actor-attribute models. (2013) In Lusher, D., Koskinen, J., & Robins, G. (eds). *Exponential Random Graph Models for Social Networks: Theories, Methods and Applications*. New York: Cambridge University Press.
- Erdős, P., & Rényi, A. (1976). On the evolution of random graphs. *Selected Papers of Alfréd Rényi*, vol, 2, 482-525.
- Frank, O., & Strauss, D. (1986). Markov graphs. *Journal of the American Statistical association*, 81(395), 832-842.
- Handcock, M. S., Robins, G., Snijders, T. A., Moody, J., & Besag, J. (2003). Assessing degeneracy in statistical models of social networks (Vol. 39). Working paper.
- Handcock, M. S., Hunter, D., Butts, C. T., Goodreau, S. M., & Morris, M. (2003). statnet: An R package for the Statistical Modeling of Social Networks. Web page <http://www.csde.washington.edu/statnet>.
- Holland, P. W., & Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical association*, 76(373), 33-50.
- Hunter, D. R. (2007). Curved exponential family models for social networks. *Social networks*, 29(2), 216-230.
- Koskinen, J. H., Robins, G. L., & Pattison, P. E. (2010). Analysing exponential random graph (p-star) models with missing data using Bayesian data augmentation. *Statistical Methodology*, 7(3), 366-384.
- Koskinen, J. H., Robins, G. L., Wang, P., and Pattison, P. E. (2013). Bayesian analysis for partially observed network data, missing ties, attributes and actors. *Social Networks*, vol. 35(4), 514-527.
- Koskinen, J. H., & Snijders, T. A. (2012) Simulation, Estimation, and Goodness of Fit. In Lusher, D., Koskinen, J., & Robins, G. (Eds.). *Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications*. Cambridge University Press.
- Lusher, D., Koskinen, J., & Robins, G. (Eds.). (2012). *Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications*. Cambridge University Press.
- Pattison, P., & Robins, G. (2002). Neighborhood-based models for social networks. *Sociological Methodology*, 32(1), 301-337.
- Pattison, P., & Robins, G. (2004). Building models for social space: neighbourhood-based models for social networks and affiliation structures. *Mathématiques et sciences humaines. Mathematics and social sciences*, (168).
- Pattison, P., & Wasserman, S. (1999). Logit models and logistic regressions for social networks: II. Multivariate relations. *British Journal of Mathematical and Statistical Psychology*, 52(2), 169-193.
- Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400-407.

- Robins, G., Pattison, P., Kalish, Y., & Lusher, D. (2007). An introduction to exponential random graph (p^*) models for social networks. *Social networks*, 29(2), 173-191.
- Robins, G., Snijders, T., Wang, P., Handcock, M., & Pattison, P. (2007). Recent developments in exponential random graph (p^*) models for social networks. *Social networks*, 29(2), 192-215.
- Robins, G., Elliott, P., & Pattison, P. (2001). Network models for social selection processes. *Social Networks*, 23(1), 1-30.
- Robins, G.L., Pattison, P. & Elliott, P. (2001b). Network models for social influence processes. *Psychometrika*, 66, 161-190.
- Robins, G., Pattison, P., & Wang, P. (2009). Closure, connectivity and degree distributions: exponential random graph (p^*) models for directed social networks. *Social Networks*, 31, 105-117.
- Snijders, T. A. (2002). Markov chain Monte Carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2), 1-40.
- Snijders, T. A., Pattison, P. E., Robins, G. L., & Handcock, M. S. (2006). New specifications for exponential random graph models. *Sociological methodology*, 36(1), 99-153.
- Snijders, T. A., Van de Bunt, G. G., & Steglich, C. E. (2010). Introduction to stochastic actor-based models for network dynamics. *Social networks*, 32(1), 44-60.
- Tierney, L. (1994) Markov Chains for Exploring Posterior Distributions. *The Annals of Statistics* 22 (4), 1701–1728.
- Wang, P., Robins, G., Pattison, P., & Lazega, E. (2013). Exponential random graph models for multilevel networks. *Social Networks* 35(1), 96-115.
- Wang, P., Robins, G., Pattison, P., & Lazega, E. (under review). Social selection models for multilevel networks.
- Wang, P., Sharpe, K., Robins, G. L., & Pattison, P. E. (2009). Exponential random graph (p^*) models for affiliation networks. *Social Networks*, 31(1), 12-25.
- Wasserman, S. and Pattison, P. (1996). Logit models and logistic regressions for social networks: I. an introduction to Markov graphs and p^* . *Psychometrika*, 61(3):401–425.

Appendix A – Sample Files

Sample Input Files

Sample network file:

Network files contain the observed network in the adjacency matrix format.

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 1 1 0 0
0 0 0 0 1 1 0 0 0 0 0 0 1 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 1 0 0 1 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
```

Sample dyadic attribute file

Dyadic attribute files contain the values of network ties as covariate in the adjacency matrix format with headers as attribute names. Multiple dyadic attributes are listed in the same file each with separate headers.

Note: Examples here omitted some values in the matrices.

```
Distance
0 1 5 0 7 0 4 0 0 5 0 0 9 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
.....
.....
0 0 8 0 4 3 0 1 0 0 0 0 0 0
0 0 0 9 0 1 4 0 2 1 0 0 0 0
Strength
0 1 5 0 7 0 4 0 0 5 0 0 9 0
0 0 0 0 0 0 0 0 3 0 0 0 0 0
.....
.....
0 3 0 0 0 0 0 5 1 0 0 0 0 0
0 6 0 0 0 0 0 1 0 0 0 7 0 0
0 0 7 0 0 1 0 0 7 1 0 0 8 0
```

Sample structural zero file:

The file contains a binary matrix where '1' indicates changeable ties, and '0' indicates fixed ties. Applying this structural zero file example will fix all the tie variables related to node 2 and 5. Ties between node 1 and 13, node 1 and 14, are also fixed.

```
0 0 1 1 0 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 1 0 1 1 1 1 1 1 1 1 1
1 0 1 0 0 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 0 0 1 1 1 1 1 1 1 1
1 0 1 1 0 1 0 1 1 1 1 1 1 1
1 0 1 1 0 1 1 0 1 1 1 1 1 1
1 0 1 1 0 1 1 1 0 1 1 1 1 1
1 0 1 1 0 1 1 1 1 0 1 1 1 1
1 0 1 1 0 1 1 1 1 1 0 1 1 1
1 0 1 1 0 1 1 1 1 1 1 0 1 1
1 0 1 1 0 1 1 1 1 1 1 1 0 1
1 0 1 1 0 1 1 1 1 1 1 1 1 0
```


Attribute file format

Attribute names should be listed in the first line, delimited by tabs. Note that attribute names should not start with numbers to meet the SPSS script requirements for variable names. Each column represents an attribute. Each row corresponds to the same row as in the adjacency matrix

Sample binary attribute file:

Member	gender
1	1
1	1
0	1
1	0
1	1
0	0
1	0
0	0
1	1
1	0
0	1
0	0
0	1
1	0

Sample categorical attribute file:

Department	club
1	1
3	2
2	3
3	2
1	3
2	1
1	2
2	3
3	1
3	3
2	2
3	2
1	1
1	2

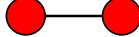
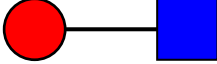
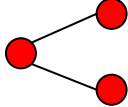
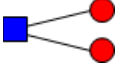
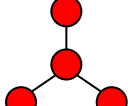
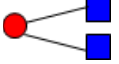
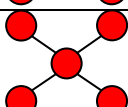
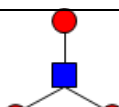
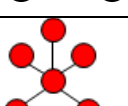
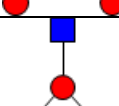
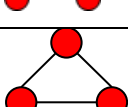
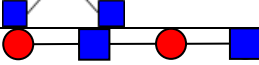
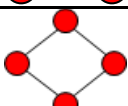
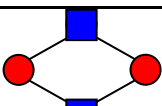

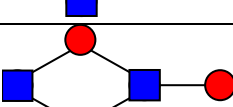
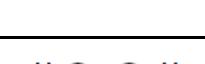
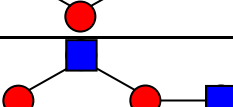
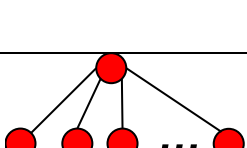
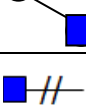
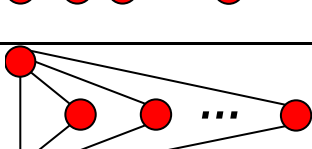
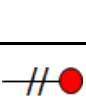
Sample continuous attribute file:

Income	age	performance
1.0	23	2
1.1	34	6
1.1	42	5
0.5	23	4
0.3	24	1
1.1	19	1
1.5	38	2
0.2	49	1
0.1	58	1
0.2	47	2
1.0	24	3
0.2	36	2
0.1	19	4
0.5	20	3

Appendix B – Model Configurations

Non-directed one-mode
networks (A & B)

Bipartite networks (X)

Label	Configuration	Label	Configuration
EdgeA EdgeB		XEdge	
Star2A Star2B		XStar2A	
Star3A Star3B		XStar2B	
Star4A Star4B		XStar3A	
Star5A Star5B		XStar3B	
TriangleA TriangleB		X3Path	
Cycle4A Cycle4B		X4Cycle	
IsolatesA IsolatesB		XECA	
IsolateEdgesA IsolateEdgesB		XECB	
ASA ASA2 ASB ASB2		IsolatesXA	
ATA ATB		IsolatesXB	

Label	Configuration	Label	Configuration
A2PA A2PB		XASA	
AETA		XASB	
		XACA	
		XACB	
		XAECA	
		XAE CB	

Directed one-mode networks (A & B)

Label	Configuration	Label	Configuration
ArcA ArcB		ReciprocityA ReciprocityB	
In2StarA In2StarB		Out2StarA Out2StarB	
In3StarA In3StarB		Out3StarA Out3StarB	
TwoPathA TwoPathB		Transitive-TriadA Transitive-TriadB	

Label	Configuration	Label	Configuration
Cyclic-TriadA Cyclic-TriadB		T1A T1B	
T2A T2B		T3A T3B	
T4A T4B		T5A T5B	
T6A T6B		T7A T7B	
T8A T8B		SinkA SinkB	
SourceA SourceB		IsolateA IsolateB	
AinSA AinSA2AinSB AinSB2		AoutSA AoutSA2 AoutSB AoutSB2	
AinAoutSA AinAoutSB		ATA-T ATB-T	
ATA-C ATB-C		ATA-D ATB-D	
ATA-U ATB-U		A2PA-T A2PB-T	
A2PA-D A2PB-D		A2PA-U A2PB-U	

Non-directed one- and two-mode interactions (A & X, or B & X)

Label	Configuration	Label	Configuration
Star2AX		Star2BX	
StarAA1X		StarAB1X	
StarAX1A		StarAX1B	
StarAXAA		StarAXAB	
TriangleXAX		TriangleXBX	
L3XAX		L3XBX	
ATXAX		ATXBX	
EXTA		EXTB	

Directed one- and two-mode interactions (A & X, or B & X)

Label	Configuration	Label	Configuration
In2StarAX		In2StarBX	
Out2StarAX		Out2StarBX	
AXS1Ain		AXS1Bin	
AXS1Aout		AXS1Bout	
AAinS1X		ABinS1X	
AAoutS1X		ABoutS1X	
TXAXarc		TXBXarc	
TXAXreciprocity		TXBXreciprocity	
ATXAXarc		ATXBXarc	

ATXAXreciprocity		ATXBXreciprocity	
L3XAX		L3XBX	
L3XAXreciprocity		L3XBXreciprocity	

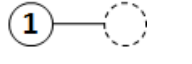
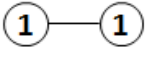
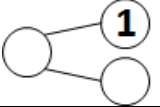
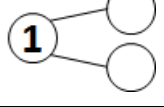
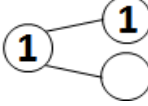
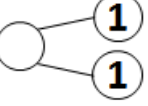
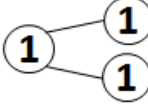
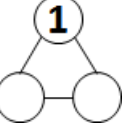
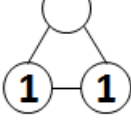
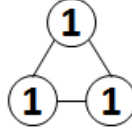
Non-directed cross-level interactions (A, B & X)

Label	Configuration	Label	Configuration
L3AXB		C4AXB	
ASAXASB		AC4AXB	

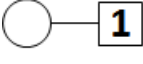
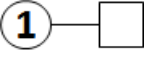
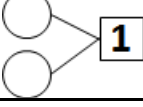
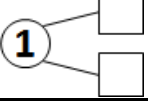
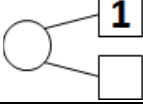
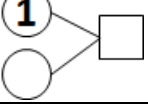
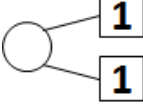
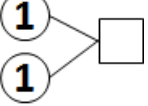
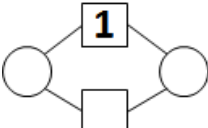
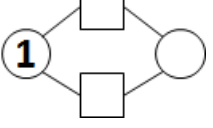
Directed cross-level interactions (A, B & X)

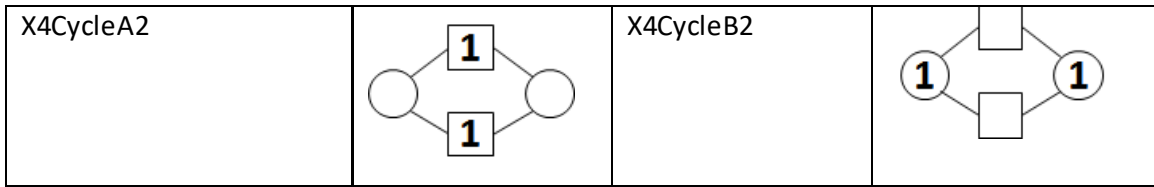
Label	Configuration	Label	Configuration
L3AXBin		L3AXBout	
L3AXBpath		L3BXApath	
C4AXBentrainment		C4AXBexchange	
C4AXBexchangeAreciprocity		C4AXBexchangeBreciprocity	
C4AXBreciprocity		C4AXBexchangeBreciprocity	
AinASXAinBS		AoutASXAoutBS	
AinASXAoutBS		AoutASXAinBS	

Non-directed one-mode social selection models with binary attributes

Label	Configuration	Label	Configuration
ActivityA ActivityB		InteractionA InteractionB	
TwoPath100A TwoPath100B		TwoPath010A TwoPath010B	
TwoPath110A TwoPath110B		TwoPath101A TwoPath101B	
TwoPath111A TwoPath111B		Triangle1A Triangle1B	
Triangle2A Triangle2B		Triangle3A Triangle3B	

Two-mode social selection models with binary attributes

Label	Configuration	Label	Configuration
XEdgeA		XEdgeB	
X2StarA010		X2StarB010	
X2StarA100		X2StarB100	
X2StarA101		X2StarB101	
X4CycleA1		X4CycleB1	



Non-directed one-mode social selection models
with continuous attributes

Label	Configuration	Label	Configuration
ActivityA ActivityB		SumA SumB	
DifferenceA DifferenceB		ProductA ProductB	

Two-mode Social selection models
with continuous attributes

Label	Configuration	Label	Configuration
XEdgeA		XEdgeB	
X2StarA		X2StarB	
X2StarASum		X2StarBSum	
X2StarADifference		X2StarBDifference	
X4CycleASum		X4CycleBSum	

X4CycleADifference		X4CycleBDifference	
XEdgeABSum		XEdgeABDifference	

One- and two-mode social selection models with categorical attributes

Label	Configuration	Label	Configuration
MatchA MatchB		MismatchA MismatchB	
MismatchA MismatchB			
X2StarAMatch		X2StarBMatch	
X2StarAMismatch		X2StarBMismatch	
X4CycleAMatch		X4CycleBMatch	
X4CycleAMismatch		X4CycleBMismatch	
XEdgeMatchAB		XEdgeMismatchAB	

Directed one-mode social selection models with binary attributes

Label	Configuration	Label	Configuration
-------	---------------	-------	---------------



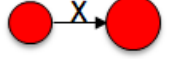



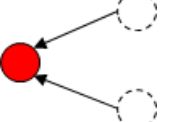
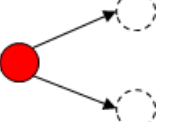
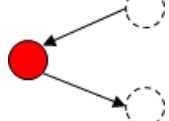
SenderA SenderB		ReceiverA ReceiverB	
InteractionA InteractionB		ActivityReciprocityA ActivityReciprocityB	
InteractionReciprocityA InteractionReciprocityB		In2Star010A In2Star010B	
Out2Star010A Out2Star010B		Mix2Star010A Mixed2Star010B	

Directed cross level social selection models with binary attributes

Label	Configuration	Label	Configuration
L3AXBSenderAB		L3AXBReceiverAB	
L3ASXBRpath		L3ARXBSpath	
C4AXBentrainmentA		C4AXBentrainmentB	
C4AXBexchangeA		C4AXBexchangeB	
C4AXBAReciprocityA		C4AXBAReciprocityB	

Directed one-mode social selection models with continuous attributes

Label	Configuration	Label	Configuration
SenderA		ReceiverA	

SenderB		ReceiverB	
SumA SumB		DifferenceA DifferenceB	
ProductA ProductB		SumReciprocityA SumReciprocityB	
DifferenceReciprocityA DifferenceReciprocityB		ProductReciprocityA ProductReciprocityB	
In2StarA In2StarB		Out2StarA Out2StarB	
Mixed2StarA Mixed2StarB			

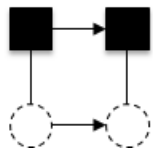
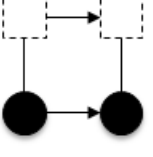
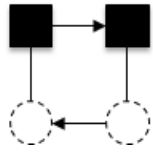
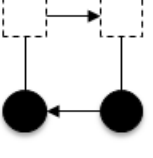
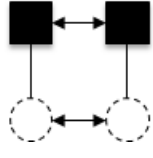
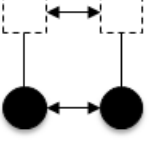
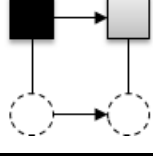
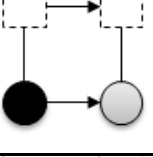
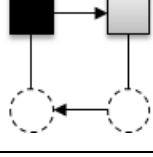
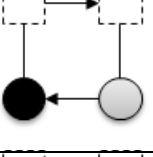
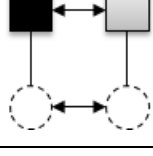
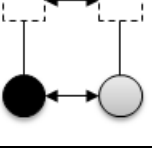
Directed cross level social selection models
with continuous attributes

Label	Configuration	Label	Configuration
Star2AXSender		Star2BXSender	
Star2AXReceiver		Star2BXReceiver	
TXAXSumArc		TXBXSumArc	
TXAXDiffArc		TXBXDiffArc	
TXAXSumReciprocity		TXBXSumReciprocity	
TXAXDiffReciprocity		TXBXDiffReciprocity	
L3XAXSumArc		L3XBXSumArc	
L3XAXDiffArc		L3XBXDiffArc	
L3XAXSumReciprocity		L3XBXSumReciprocity	

L3XAXDiffReciprocity		L3BXDiffReciprocity	
C4AXBSumEntrainmentA		C4AXBSumEntrainmentB	
C4AXBSumExchangeA		C4AXBSumexchangeB	
C4AXBSumReciprocityA		C4AXBSumReciprocityB	
C4AXBDiffEntrainmentA		C4AXBDiffEntrainmentB	
C4AXBDiffExchangeA		C4AXBDiffexchangeB	
C4AXBDiffReciprocityA		C4AXBDiffReciprocityB	

Directed cross level social selection models
with categorical attributes

MatchA MatchB		MismatchA MismatchB	
MatchReciprocityA MatchReciprocityB		MismatchReciprocityA MismatchReciprocityB	
TXAXMatchArc		TXBXMatchArc	
TXAXMismatchArc		TXBXMismatchArc	
TXAXMatchReciprocity		TXBXMatchReciprocity	
TXAXMismatchReciprocity		TXBXMismatchReciprocity	
L3XAXMatchArc		L3BXMatchArc	
L3XAXMismatchArc		L3BXMismatchArc	
L3XAXMatchReciprocity		L3BXMatchReciprocity	
L3XAXMismatchReciprocity		L3BXMismatchReciprocity	

C4AXBMatchEntrainmentA		C4AXBMatchEntrainmentB	
C4AXBMatchExchangeA		C4AXBMatchexchangeB	
C4AXBMatchReciprocityA		C4AXBMatchReciprocityB	
C4AXBMismatchEntrainmentA		C4AXBMismatchEntrainmentB	
C4AXBMismatchExchangeA		C4AXBMismatchexchangeB	
C4AXBMismatchReciprocityA		C4AXBMismatchReciprocityB	

Appendix C – R utility functions for mpnet reading in simulated networks

The function `readPNetStatistics()` lets you read in a simulated network in R

```
readPNetStatistics <- function(filename)
{
  impordata <- scan(filename,what='character',quiet= TRUE)
  n <- as.numeric(impordata[(grep("*vertices",impordata)+1)])
  impordata <-
  impordata[(grep("*matrix",impordata)+1):(grep("*matrix",impordata)+n[1]
  *n[1])]
  AdjMatrix <- matrix(as.numeric(impordata),n,n,byrow=T)
  return(AdjMatrix)
}
```

If your session is called 'mytest' and you have a file with '_Network_A_[iteration].txt' appended - for iteration=1001000 it would be called 'mytest_Network_A_1001000.txt' - in your R current directory (check it using `dir()`) read it into R using

```
SimADJ <- readPNetStatistics('mytest_Network_A_1001000.txt')
```

You can wrap this function to read in all (or part) of the simulated networks (simply done by using the command `paste()`). The resulting variable is of the 'matrix' class and `sna` or `network` can be used to plot the network or calculate summary statistics.

reading in simulated statistics

If your session is called 'mytest' the output from MPNet appended as '_sim.txt', i.e. 'mytest_sim.txt', will be a normal text file that you can read into R using `read.table()`:

```
output <- read.table('mytest_sim.txt',header = TRUE )
```

Each row of the data frame will contain the statistics count for a sample point in the estimation. If you are simulating the unimodal A-network and are saving the number of edges, the number of edges across simulations can be plotted using

```
plot(output$EdgeA)
```