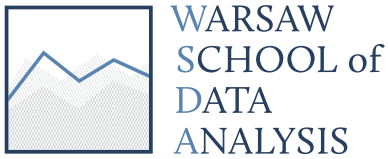
Further Topics in Social Network Analysis

Fitting Exponential Random Graph Models with `statnet`

Dominik Batorski Michał Bojanowski Bartosz Chroł Kamil Filipek

Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw



Contents

HUMAN CAPITAL NATIONAL COHESION STRATEGY

UNIVERSITY OF WARSAW

ICS RUG / UU / RU

EUROPEAN UNION EUROPEAN SOCIAL FUND

Project co-financed by the European Union under the European Social Fund

1 Introduction

This tutorial supplements the topics covered on WSAD Summer School workshop "Introduction Statistical Social Network Analysis". The workshop presented how Exponential Random Graph Models (ERGMs) can be fit using program PNet (Wang, Robins, and Pattison 2009). Here we provide examples how ERGMs can be simulated and fitted using `statnet` suite of R packages for network analysis (Handcock et al. 2003).

In this tutorial we will use network data in the form of objects of class `network` instead of objects of class `igraph`. In principle, you can use both packages at the same time. However, we recommend to use only one at a time to avoid function name conflicts.[1] You can easily convert `igraph` objects to `network` objects (or vice versa) using `asIgraph` and `asNetwork` functions from `intergraph` package.

2 Exponential Random Graph Models in a nutshell

ERG models, as they are currently known, stem from the work by Holland and Leinhardt (1976) and Frank and Strauss (1986), and were further developed by, among others, Frank (1991), Wasserman and Pattison (1996), Pattison and Robins (2002), Snijders et al. (2006). ERGMs have established their position as one of the most important tools in social networks analysis. Comprehensive description of ERG models is provided in the book by Lusher, Koskinen, and Robins (2012).

Exponential Random Graph Models are a family of statistical models for understanding processes that shape the global structure of a network. This goal is achieved by assigning probability to networks according to network statistics – summary measures describing selected features of the network.

Formally, let A represent a network in the form of an adjacency matrix of given size n. Let $\mathcal{A}$ be a collection of all possible networks of size n that can be constructed. An ERG model postulates that a probability distribution defined over all networks in $\mathcal{A}$ can be represented as:

$$P_\theta(A = a) = \frac{\exp(\theta^T g(a))}{\kappa(\theta, \mathcal{A})}, \qquad a \in \mathcal{A},$$

where

- $g(a)$ is a vector of network statistics,
- θ is a vector of weights associated with these statistics,
- $\kappa(\theta, \mathcal{A})$ is a normalizing constant ensuring that all the probabilities sum-up to 1.

$$\kappa(\theta, \mathcal{A}) = \sum_{a \in \mathcal{A}} \exp(\theta^T g(a)).$$

In words, the probability that we observe a particular network a out of the set $\mathcal{A}$ of all possible networks of size n is a function of chosen network statistics $g(a)$, associated parameters θ, and the normalizing constant κ.

The model can also be written as a model for conditional log-odds of tie existence between a pair of nodes. This makes it a little bit similar to a logistic regression model (see section Dyadic independence ERGMs for more details):

$$\text{logit}(Y_{ij} = 1 | y^c_{ij}) = \theta \delta(y_{ij})$$

where:

- Y_{ij} is a binary variable capturing the state of a pair of actors i and j, whether there is a tie ($Y_{ij} = 1$) or not ($Y_{ij} = 0$).

[1] For more details see for example http://bc.bojanorama.pl/2010/08/namespaces-and-name-conflicts/

HUMAN CAPITAL NATIONAL COHESION STRATEGY
UNIVERSITY OF WARSAW
ICS RUG / UU / RU
EUROPEAN UNION EUROPEAN SOCIAL FUND
Project co-financed by the European Union under the European Social Fund

- y_{ij} is a particular realization of variable Y_{ij}.
- y_{ij}^c is the complement of y_{ij}, i.e. all other actor pairs in the network apart from (i, j).
- $\delta(y_{ij})$ is a vector of *change statistics* for each term in the model.
- θ is a vector of parameters, as before.

In words, the log-odds that actors i and j are connected with a tie, given the particular configuration of all the remaining ties, is a function of change statistics for each term in the model.

The change statistics capture how a particular term $g(y)$ changes if the y_{ij} tie is formed or not:

$$\delta(y_{ij}) = g(y_{ij}^+) - g(y_{ij}^-)$$

where y_{ij}^+ represents y^c and y_{ij} set to 1, while y_{ij}^- represents y^c and y_{ij} set to 0.

As a consequence, model parameters θ can be interpreted as log-odds of a presence of a network tie conditional on the state of all other dyads in the network.

2.1 Network statistics

Different ERG models can be specified by choosing different network statistics. A set of network statistics is chosen on the basis of theoretical premises concerning a particular research problem at hand. However, there are a few basic configurations often included in the model. Example of network statistics include:

1. Number of edges in the network (`edges`)
$$S_1(a) = \sum_{1 \le i \le j \le n} a_{ij},$$
2. Number of k-stars (for $k \ge 2$). A k-star consists of a central node and k neighbors. (`kstar`).
$$S_k(a) = \sum_{1 \le i \le n} \binom{k_i}{k}.$$
3. Number of triangles (`triangles`).
$$T(a) = \sum_{1 \le i < j < h \le n} a_{ij} a_{ih} a_{jh}.$$

ERGMs could easily incorporate additional information about actors attributes. To indicate extra information function $g(a)$ could be replaced with $g(a, X)$, where X is a matrix containing attributes. This leads to another statistics, which measure the effect of actor's attributes. y_i denotes value of an attribute y of the i-th node:

4. Attribute-based activity (`nodeofactor`)
$$\sum_{1 \le i \le j \le n} a_{ij}(y_i + y_j),$$
5. Homophily (binary attribute) (`nodematch`).
$$\sum_{1 \le i \le j \le n} a_{ij} y_i y_j,$$
6. Homophily (continuous attribute) (`absdif`)
$$\sum_{1 \le i \le j \le n} a_{ij} |y_i - y_j|.$$

See `?ergm.terms` for a complete list of network statistics implemented in `ergm` or view it on the www.rdocumentation.org here.

2.2 Calculating network statistics

The workhorse for estimating ERG models is the ergm function from the ergm package. We will describe it in detail later. Function ergm uses a formula interface to specify the network statistics, or terms, to be included in the model. If you simply want to calculate the values of network statistics in a given network you can use the function summary with a single argument – an ERGM formula. Such formula should have an object of class network on the left hand side, and a sum of network statistics on the right hand side.

As an example, let us take the classroom network we used in other tutorials. We need it now as a network object:

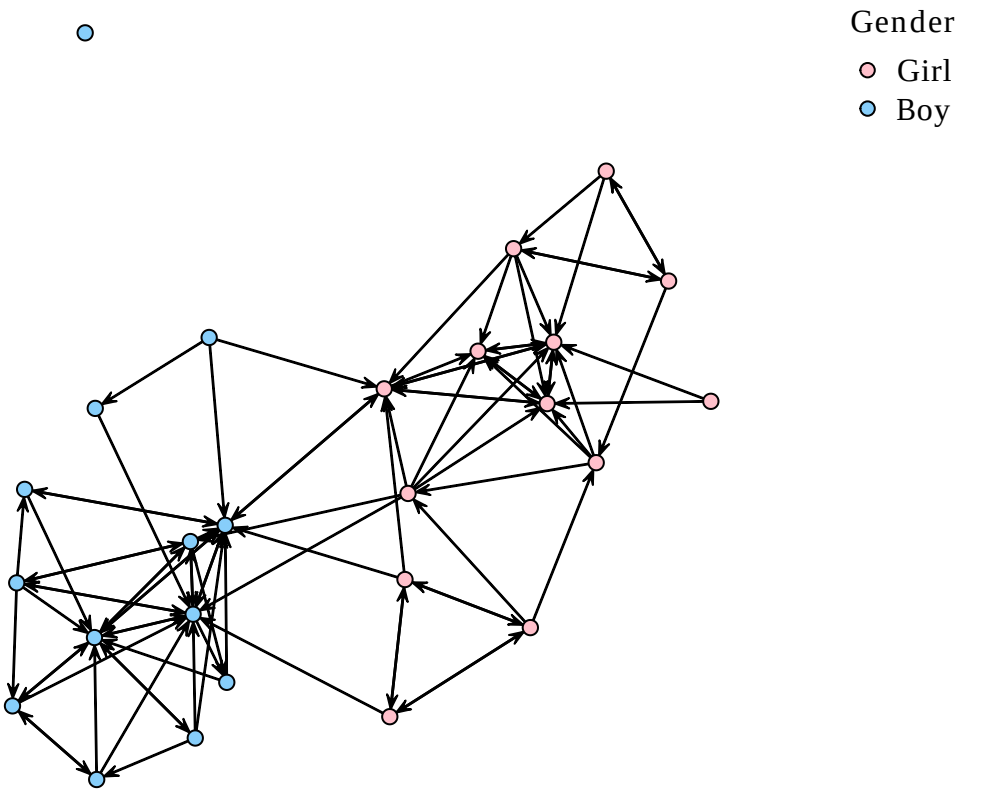
```
library(network)
```

```
## network: Classes for Relational Data
## Version 1.13.0 created on 2015-08-31.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##                     Mark S. Handcock, University of California -- Los Angeles
##                     David R. Hunter, Penn State University
##                     Martina Morris, University of Washington
##                     Skye Bender-deMoll, University of Washington
##  For citation information, type citation("network").
##  Type help("network-package") to get started.
```

```
# load data
data(IBE121, package="isnar")
# select the "would like to play with" network
# by dropping other edges
playnet <- igraph::delete.edges(IBE121, igraph::E(IBE121)[question != "play"])
# convert to 'network' object
ibe <- intergraph::asNetwork(playnet)
ibe
```

```
##  Network attributes:
##   vertices = 26
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   school = 121
##   class = B
##   total edges= 88
##     missing edges= 0
##     non-missing edges= 88
##
##  Vertex attribute names:
##     female isei08_f isei08_m vertex.names wraven
##
##  Edge attribute names:
##     question
```

```
plot(ibe, vertex.col=ifelse(ibe %v% "female", "pink", "lightskyblue"))
legend("topright", pch=21, pt.bg=c("pink", "lightskyblue"), col="black",
       legend=c("Girl", "Boy"), title="Gender", bty="n")
```



We can now use summary to calculate some network statistics of interest. For example, let's calculate

- number of ties
- number of reciprocated ties
- Gender-based activity (out-degree)

```
library(ergm)
```

```
## Loading required package: statnet.common

##
## ergm: version 3.5.1, created on 2015-10-18
## Copyright (c) 2015, Mark S. Handcock, University of California -- Los Angeles
##                     David R. Hunter, Penn State University
##                     Carter T. Butts, University of California -- Irvine
##                     Steven M. Goodreau, University of Washington
##                     Pavel N. Krivitsky, University of Wollongong
##                     Martina Morris, University of Washington
##                     with contributions from
##                     Li Wang
##                     Kirk Li, University of Washington
##                     Skye Bender-deMoll, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("ergm").

## NOTE: If you use custom ERGM terms based on 'ergm.userterms'
## version prior to 3.1, you will need to perform a one-time update
## of the package boilerplate files (the files that you did not write
## or modify) from 'ergm.userterms' 3.1 or later. See
## help('eut-upgrade') for instructions.
```

```
summary(ibe ~ edges + mutual + nodeofactor("female"))
```

```
##                   edges                  mutual nodeofactor.female.TRUE
##                      88                      22                      46
```

Consequently, this network has 88 ties, 22 of which are reciprocated, and girls send ties 46 times.

2.3 Fitting ERGMs

Before we go into more complex examples let us illustrate how ERG models are fit with statnet using two very simple models.

2.3.1 Homogenous Bernoulli model

First let's estimate the simplest possible model. The model assumes that the network is purely random: every tie in the network appears with the same unknown probability p. It is also known as *homogeneous Bernoulli model* as each tie is sampled independently with a given probability p. Appropriate ERGM formulation is:

```
model.ibe0 <- ergm(ibe ~ edges)
```

```
## Evaluating log-likelihood at the estimate.
```

```
summary(model.ibe0)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   ibe ~ edges
##
## Iterations:  5 out of 20
##
## Monte Carlo MLE Results:
##       Estimate Std. Error MCMC % p-value
## edges  -1.8542     0.1146      0  <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 901.1  on 650  degrees of freedom
##  Residual Deviance: 515.4  on 649  degrees of freedom
##
## AIC: 517.4    BIC: 521.9    (Smaller is better.)
```

The model consists of a single term edges: number of edges in the network.

The ibe network has 26 nodes. This means that the number of all possible ties in this network is equal to $26 \times 25 = 650$. Odds for tie existence are then equal to $88/(650 - 88) = 0.1565836$. If we calculate log-odds we will get $\log(0.156583) = -1.854165$ which is exactly the value of the edges term in the model above. This corresponds to a constant tie probability of $88/650 = 0.1353846$. This, in turn, is obviously equal to network density:

```
network.density(ibe)
```

```
## [1] 0.1353846
```

As network density is usually not of our primary research interest, the edges term is usually included in models and interpreted in a similar manner like the intercept in linear regression.

2.3.2 Are ties reciprocated?

As a little bit more complex example, let's assess the extent, to which children nominations are reciprocated, i.e., if A nominated B is it more likely for B to nominate A back, instead of nominating someone else. We need a model with two terms:

- edges which is the number of edges in the network. With this term we model overall network density.
- mutual which is the number of symmetrical (reciprocated) dyads in the network.

To fit the model to our classroom network we use the function ergm and provide a model formula as an argument. The formula has a network object on the left hand side and a sum of the two terms on the right hand side:

```
model.ibe1 <- ergm(ibe ~ edges + mutual, control=control.ergm(seed=666))
```

```
## Starting maximum likelihood estimation via MCMLE:
## Iteration 1 of at most 20:
## The log-likelihood improved by 0.003151
## Step length converged once. Increasing MCMC sample size.
## Iteration 2 of at most 20:
## The log-likelihood improved by 0.004904
## Step length converged twice. Stopping.
## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 1
##
## This model was fit using MCMC.  To examine model diagnostics and check for degeneracy, use the mcmc.diag
```

```
summary(model.ibe1)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   ibe ~ edges + mutual
##
## Iterations:  2 out of 20
##
## Monte Carlo MLE Results:
##        Estimate Std. Error MCMC % p-value
## edges   -2.4636     0.1679      0  <1e-04 ***
## mutual   2.4587     0.3864      0  <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 901.1  on 650  degrees of freedom
##  Residual Deviance: 473.1  on 648  degrees of freedom
##
## AIC: 477.1    BIC: 486    (Smaller is better.)
```

From the results we see that the edges effect is negative and significant while the mutual term is positive and significant too. Both effects are roughly equal in absolute size, so cancel each other out, which means that odds for a mutual tie are about 1 and the probability about 0.5. Conditional odds for a non-mutual tie are $\exp(-2.466) = 0.0849$ so the probability about 0.08. Indeed, there is a strong tendency to reciprocate ties.

2.4 Goodness of fit

2.4.1 Model comparison

ERG models do not have associated goodness of fit measures like R^2 in linear regression. To compare different models a Akaike Information Criterion (AIC) measure (Akaike 1998) is used. Models with smaller AIC should be preferred. AIC can be used to compare non-nested models. Its design penalizes models with more parameters, so adding more terms to the ERG model does not necessarily lead to better fitting models according to AIC.

AIC can be calculated with function AIC. For the two models estimated in previous sections we obtain:

```
AIC(model.ibe0, model.ibe1)
```

```
##             df      AIC
## model.ibe0   1 517.4447
## model.ibe1   2 477.0731
```

So the "reciprocity" model seems to fit the data better than homogeneous Bernoulli model.

2.4.2 Goodness of fit through simulation

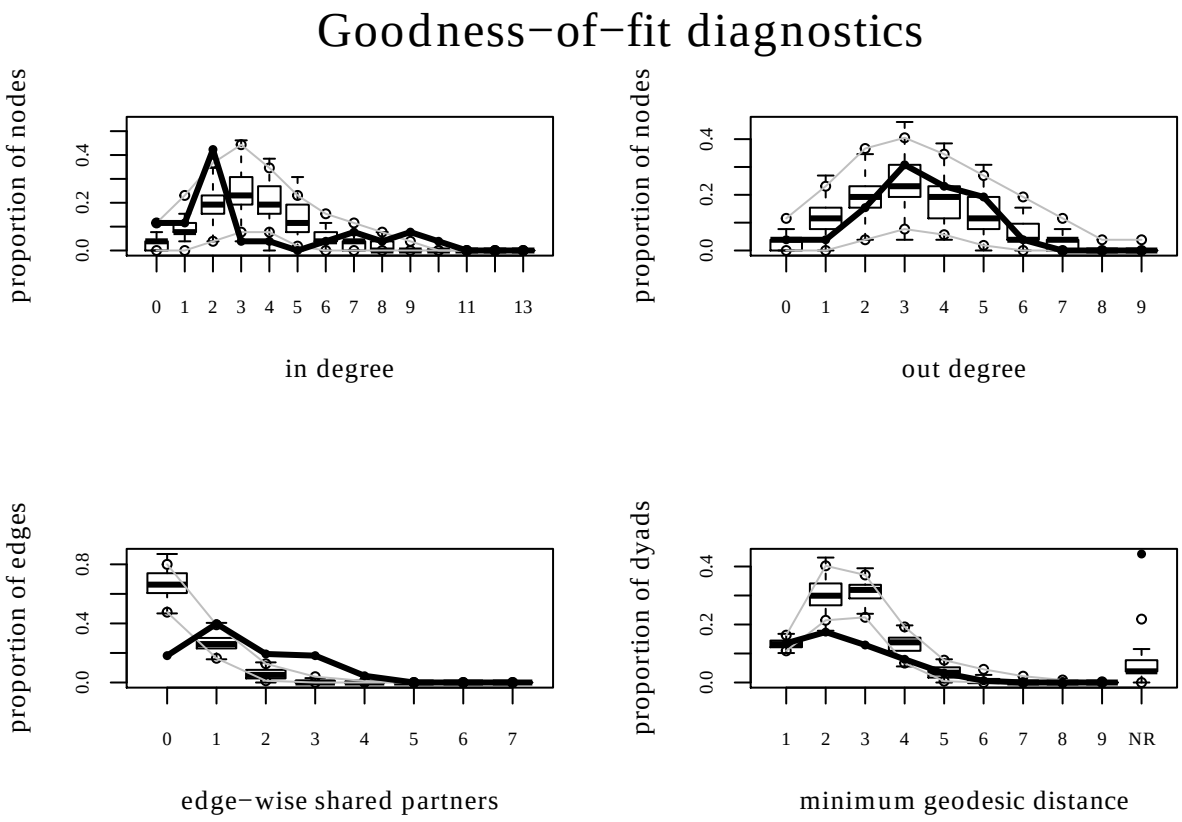
To examine how well a given ERG model fits the data a simulation-based methods are used. Given an estimated model we can simulate networks consistent with it. Once we simulate a lot of networks from the model, we can check, whether these simulated networks posses the same kind of global network properties as the observed network.

Global properties usually analyzed in this way include:

- (in-/out-)degree distributions.
- distribution of shortest path lengths (minimum geodesic distances).
- distribution of the number of edge-wise shared partners.

This can be done with the function gof. Let us examine the homogeneous Bernoulli model fitted earlier.

```
fit0 <- gof(model.ibe0)
layout(matrix(1:4, 2, 2, byrow=TRUE))
plot(fit0)
```



We are presented with four charts. Each chart presents two types of data:

First, solid thick black lines represent properties (distributions) of our observed data (the `ibe` network). We have in-degree distribution, out-degree distribution, distribution of the number of edge-wise shared partners, and the distribution of shortest path lengths.

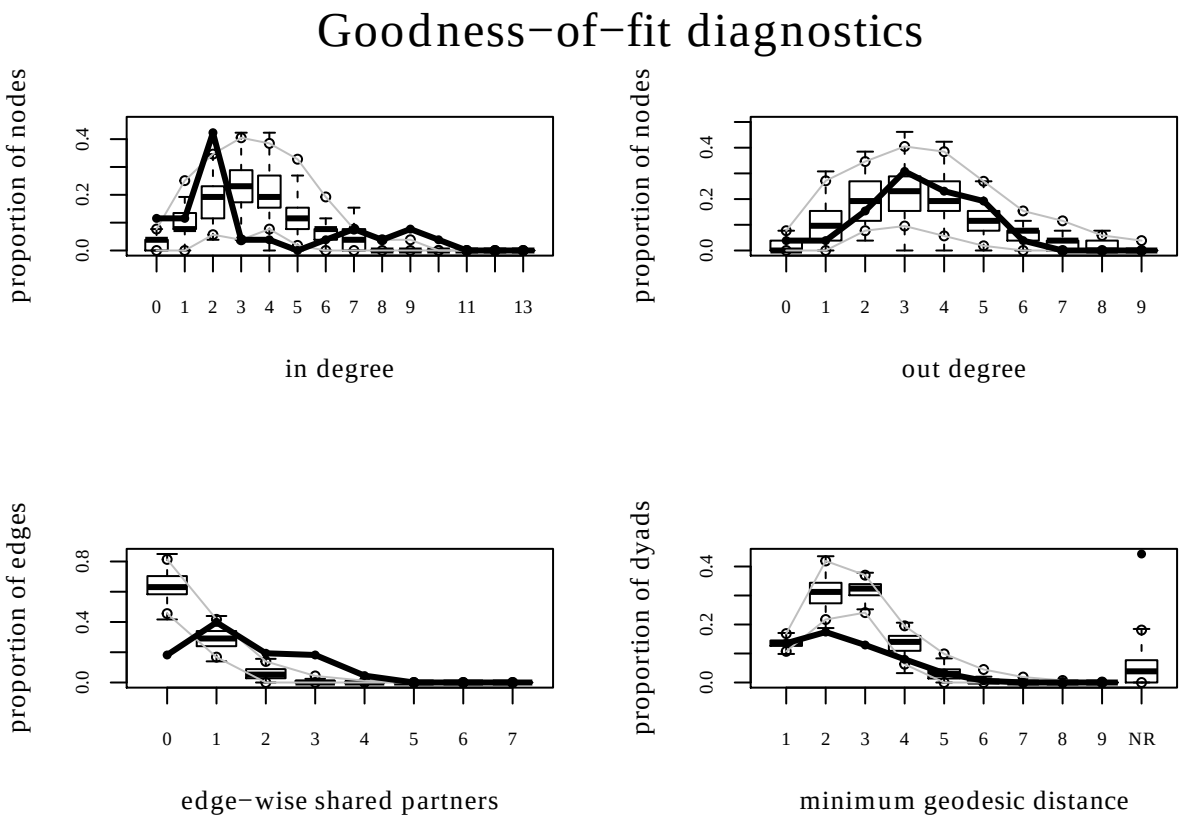
Second, boxplots summarize the properties of the model-based simulated networks. By default 100 networks are simulated. Due to the random nature of the simulation procedure each simulated network will be a little bit different from the others. That's why boxplots are used to visualize how the simulated distributions vary.

To examine the fit, we have to compare visually empirical data (solid thick black lines) to simulated distributions (boxplots). Median values of simulated distributions are shown with black horizontal lines within the boxes. If our model would fit the data perfectly, the empirical lines should follow the medians of all the boxes. If, on the other hand, the empirical line strays far away, even beyond the gray lines symbolizing 95% confidence intervals, it is an indication of a very poor fit.

From the charts above we see that the Bernoulli model fits the data very poorly indeed. While the out-degree distribution the model is able to recover, the remaining three statistics are not modeled in a satisfactory manner.

Let us examine the goodness of fit of the "reciprocity" model.

```
fit1 <- gof(model.ibe1)
layout(matrix(1:4, 2, 2, byrow=TRUE))
plot(fit1)
```



Again, we see that this model also does not fit the data very well. This means that there must be other social process, apart from reciprocity, that govern the structure of the network in this classroom. We will explore other possibilities in the sections below.

3 (In)dependence assumptions

You are probably familiar with Ordinary Least Squares (OLS) regression:

$$Y_i = \alpha + \beta X_i + \varepsilon_i$$

If you recall, on of the assumption of OLS is that observations are independent, which translates to the requirement $\text{Cor}(\varepsilon_i, \varepsilon_j) = 0$. In other words, that the correlation between errors for any two observations is 0. Some generalizations of OLS regression go in the direction of relaxing this assumption, by allowing some observations to be correlated. In particular, multilevel models (e.g. Snijders and Bosker 2011) can be considered such a generalization as, while we assume zero correlation between observations belonging to different groups, observations belonging to the same group can be correlated.

Part of the challenge in modeling the structure of social networks is in dependence. Social networks represent relationships between actors, which make these actors and their actions dependent in some respect. In particular, the process through which they form relations with others may be dependent. Consequently, in social networks, in principle, every tie may depend on other ties in the network. To approach this "Hell of Dependence" in a tractable way, proposed models can be characterized with *independence assumptions* that they pose. These statements specify what kind of independence, and between which ties, is assumed behind a given model. Much like in OLS regression, a lot of new developments in statistical analysis of social networks have been made by postulating new assumptions that relax some independence assumptions made by previous ones.

In this section we review the following independence assumptions.

1. The Bernoulli assumption.

2. Dyadic independence assumption.
3. Markov dependence assumption.
4. Other forms of dependence.

Practical examples showing ERG models that conform to particular assumptions are presented in the section with practical examples.

3.1 Bernoulli assumption

The Bernoulli assumption is the simplest and the strongest. We have already discussed it in the section with the homogeneous Bernoulli model example above. According to this model every *tie* is assumed to be a realization of an independent Bernoulli Trial, a flip of a coin. The probability that an arc from i to j exist does not depend in any way on the structure of the remaining ties of the network. In particular, it does not depend on how many ties i and j already have, whether they have any network neighbors in common, and so on. It also does not depend on whether or not there is an arc from j to i.

3.2 Dyadic independence assumption

Recall that a dyad is a pair (Y_{ij}, Y_{ji}) that is the state of ties between actors i and j. According to the Bernoulli assumption all Y_{ij}, for all i and j, are independent from one another. In particular, Y_{ij} is assumed independent from Y_{ji}, as mentioned in the previous section.

The assumption of *dyadic independence* relaxes the assumption of independence *within dyads*. Consequently, Y_{ij} and Y_{ji} can be dependent.

The reciprocity model discussed previously is a dyad-independent model. The presence or absence of a tie from i to j affects the probability of tie from j to i because of the presence of the `mutual` term. However, it does not affect the probabilities of any other ties in the network.

There are also other important dyad-independent models:

- Models that include node covariates. Sending or receiving ties might be affected by a value of a nodal attribute, e.g., gender, age, and so on.
- Models that include dyadic covariates. Ties may be more likely between nodes that are similar to one another (e.g. same gender, smaller age difference) or are in some way closer to one another, e.g., are separated by a smaller geographical distance.

We will explore both types of models when analyzing homophily in the subsequent sections.

Some dyad-independent ERGMs simplify to Generalized Linear Models, which we will show in section Dyad-independent ERGMs.

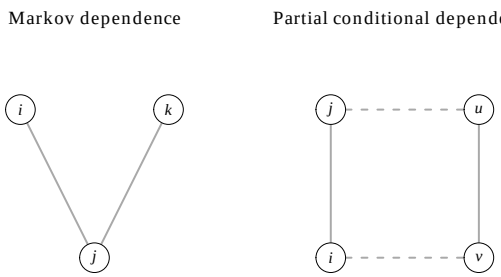
3.3 Markov dependence

Assumption of *Markov dependence* was proposed by Frank and Strauss (1986) and further relaxes the dyadic independence assumption.

Dyadic independence implies that all dyads are independent, including those involving the same actor, i.e.:

- Y_{ij} is independent from Y_{ik} (the same tie sender)
- Y_{ij} is independent from Y_{kj} (the same tie receiver)

Markov dependence assumes that dyads are independent *unless* share an actor on one of the endpoints. See left on the figure below. The probability of tie ij might not be independent from the probability of tie jk because these are both ties of actor j.



ERG models conforming to the Markov dependence assumption allow to model the shape of degree distribution and processes like preferential attachment or transitivity (e.g. meeting friends through friends, etc.).

Further relaxations of independence assumptions lead to the so-called "Socal Circuit Model" in which two ties ij and vu are assumed independent *unless* there is a link iu or jv. See Koskinen and Daraganova (2012) for further details.

4 ERG models and logistic regression

Dyadic independence is a very strong assumption which rarely (if ever) holds in real social networks. Dyadic independence ERGMs treat every dyad as a single object, independent from its surrounding. Therefore, in such models we could use only such measures, that are dyadic independent. That means that we could calculate change statistic knowing only the state of a given dyad and maybe some attributes of nodes (in this dyad). For instance number of edges is dyadic independent – if we toggle an edge from 0 to 1 we are sure that number of edges will increase by 1, no matter how the rest of the network looks like. Another example is the homophily effect – we could compute change in the number of edges between nodes of the same type knowing only attributes of these two specific nodes in a dyad. On the other hand, number of 2-stars is dyadic dependent - we need two know structure of the neighborhood if we want to calculate the change in the number of twopaths.

Assuming that our statistics are dyadic independent we could rewrite edge probability to logit form:

$$\text{logit}\left(P_\theta(A_{ij}=1)\right) = \theta^T \delta[g(a)]_{ij}$$

That looks like a logistic regression model. Indeed, dyad-independent ERGMs simplify to a logistic regression for data in which dyads are observations, the dependent variable is binary and equals to 1 if an edge exists and 0 otherwise. Independent variables are the change statistics (one for each term).

5 Practical examples of ERG models

In the following subsections we elaborate some example ERG models fit to the classroom data introduced above. The primary goal of these examples is to illustrate

- how to specify different ERG models matching different research questions
- show ERG models pretending to different independence assumptions (see section on (in)dependence assumptions).

Below we present brief examples of modeling homophily and triadic closure with ERGMs.

5.1 Actor relation effects: gender homophily in a school class

The first example shows how ERG models can be used to model homophily. Topics of homophily and segregation in social networks are covered in a separate tutorial.

Recall that homophily is a tendency for ties to form between similar actors. Similarity need to be specified according to a specific node attribute, which can be continuous or categorical. Let us focus on categorical attribute first.

The key ERGM term in modeling homophily on a categorical attribute is `nodematch`.

Consider the following example of fitting a dyad-independent ERGM to classroom data.

```
model.ibe1 <- ergm(ibe ~ edges + nodematch("female"))
```

```
## Evaluating log-likelihood at the estimate.
```

```
summary(model.ibe1)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   ibe ~ edges + nodematch("female")
##
## Iterations:  6 out of 20
##
## Monte Carlo MLE Results:
##                  Estimate Std. Error MCMC % p-value
## edges             -3.8562     0.3819      0  <1e-04 ***
## nodematch.female   2.8082     0.4032      0  <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 901.1  on 650  degrees of freedom
##  Residual Deviance: 425.5  on 648  degrees of freedom
##
## AIC: 429.5    BIC: 438.4    (Smaller is better.)
```

Let us now try to fit the same model using logistic regression. We need to prepare the data first.

```
mm <- isnar::mixingm(playnet, "female", full=TRUE)
mm
```

```
## , , tie = FALSE
##
##        alter
## ego     FALSE TRUE
##   FALSE   116  167
##   TRUE    164  115
##
## , , tie = TRUE
##
##        alter
## ego     FALSE TRUE
##   FALSE    40    2
##   TRUE      5   41
```

Object `mm` is a so-called *mixing matrix* that cross-classifies all dyads in the network according to three characteristics:

1. Gender of first node.

2. Gender of second node.
3. Whether there is a tie from first node to second node or not.

For example, there are 115 girl-girl pairs that are connected with a tie and only 2 ties that are send by a boy towards a girl, and so on.

We now transform the mixing matrix to a data frame:

```
d <- as.data.frame(as.table(mm))
d
```

```
##     ego alter   tie Freq
## 1 FALSE FALSE FALSE  116
## 2  TRUE FALSE FALSE  164
## 3 FALSE  TRUE FALSE  167
## 4  TRUE  TRUE FALSE  115
## 5 FALSE FALSE  TRUE   40
## 6  TRUE FALSE  TRUE    5
## 7 FALSE  TRUE  TRUE    2
## 8  TRUE  TRUE  TRUE   41
```

Columns `ego` and `alter` mark whether, respectively, tie sender or tie receiver is a female. Variable `tie` is `TRUE` for connected pairs. Finally, variable `Freq` contains the frequency.

We now add a variable `match` differentiating same-gender pairs: `match` is `TRUE` whenever `ego` and `alter` are equal:

```
d$match <- with(d, ego == alter)
d
```

```
##     ego alter   tie Freq match
## 1 FALSE FALSE FALSE  116  TRUE
## 2  TRUE FALSE FALSE  164 FALSE
## 3 FALSE  TRUE FALSE  167 FALSE
## 4  TRUE  TRUE FALSE  115  TRUE
## 5 FALSE FALSE  TRUE   40  TRUE
## 6  TRUE FALSE  TRUE    5 FALSE
## 7 FALSE  TRUE  TRUE    2 FALSE
## 8  TRUE  TRUE  TRUE   41  TRUE
```

We are now ready to fit our logit model. Our binary dependent variable is `tie` (is there a tie or not). Our only independent variable is `match`, also binary, representing whether dyad involves actors of the same sex. We weight the cases with variable `Freq`.

```
ibe.logit <- glm( tie ~ match, data=d, weight=Freq, family=binomial("logit"))
ibe.logit
```

```
##
## Call:  glm(formula = tie ~ match, family = binomial("logit"), data = d,
##     weights = Freq)
##
## Coefficients:
## (Intercept)    matchTRUE
##      -3.856        2.808
##
## Degrees of Freedom: 7 Total (i.e. Null);  6 Residual
## Null Deviance:       515.4
## Residual Deviance: 425.5     AIC: 429.5
```

As you can see, the coefficients are identical:

```
cbind( ERGM=coef(model.ibe1), Logit=coef(ibe.logit))
```

```
##                        ERGM     Logit
## edges             -3.856208 -3.856208
## nodematch.female   2.808240  2.808240
```

In the next section we show how homophily can be modeled in some more detail.

5.2 Gender homophily example

In our attempt to model the `ibe` network let's pursue the topic of gender *homophily*. Network homophily is a pattern in which ties are more likely to exist between nodes similar to each other according to some attribute, e.g. gender [McPherson, Smith-Lovin, and Cook (2001); bojanowski_corten_2014].

ERG models for homophily are also dyad-independent models, similar to the one from the previous section. Conditional probability of a tie exist between two nodes can be represented as a function of attributes of the first node, attributes if the second node, and possible interaction effects.

```
model.ibe2a <- ergm(ibe ~ edges + mutual + nodeofactor("female") +
                      nodeifactor("female"),
                    control=control.ergm(seed=666))
```

```
## Starting maximum likelihood estimation via MCMLE:
## Iteration 1 of at most 20:
## The log-likelihood improved by 0.02143
## Step length converged once. Increasing MCMC sample size.
## Iteration 2 of at most 20:
## The log-likelihood improved by 0.001766
## Step length converged twice. Stopping.
## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 1
##
## This model was fit using MCMC.  To examine model diagnostics and check for degeneracy, use the mcmc.diag
```

```
summary(model.ibe2a)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   ibe ~ edges + mutual + nodeofactor("female") + nodeifactor("female")
##
## Iterations:  2 out of 20
##
## Monte Carlo MLE Results:
##                         Estimate Std. Error MCMC % p-value
## edges                    -2.4848     0.2149      0  <1e-04 ***
## mutual                    2.4645     0.3839      0  <1e-04 ***
## nodeofactor.female.TRUE   0.1455     0.2517      0   0.563
## nodeifactor.female.TRUE  -0.1105     0.2455      0   0.653
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##      Null Deviance: 901.1  on 650  degrees of freedom
##  Residual Deviance: 473.0  on 646  degrees of freedom
##
## AIC: 481    BIC: 498.9    (Smaller is better.)
```

```
model.ibe2b <- ergm(ibe ~ edges + mutual +  nodeofactor("female") + nodeifactor("female")
                    + nodematch("female"),
                  control=control.ergm(seed=666))
```

```
## Starting maximum likelihood estimation via MCMLE:
## Iteration 1 of at most 20:
## The log-likelihood improved by 0.003509
## Step length converged once. Increasing MCMC sample size.
## Iteration 2 of at most 20:
## The log-likelihood improved by 0.001904
## Step length converged twice. Stopping.
## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 1
##
## This model was fit using MCMC.  To examine model diagnostics and check for degeneracy, use the mcmc.diag
```

```
summary(model.ibe2b)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   ibe ~ edges + mutual + nodeofactor("female") + nodeifactor("female") +
##     nodematch("female")
##
## Iterations:  2 out of 20
##
## Monte Carlo MLE Results:
##                          Estimate Std. Error MCMC % p-value
## edges                     -4.0556     0.4127      0  <1e-04 ***
## mutual                     1.7755     0.3975      0  <1e-04 ***
## nodeofactor.female.TRUE    0.5158     0.4477      0   0.250
## nodeifactor.female.TRUE   -0.4932     0.4525      0   0.276
## nodematch.female           2.4057     0.4309      0  <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 901.1  on 650  degrees of freedom
##  Residual Deviance: 403.8  on 645  degrees of freedom
##
## AIC: 413.8    BIC: 436.2    (Smaller is better.)
```

Compare Bernoulli model, reciprocity model, and the two homophily models

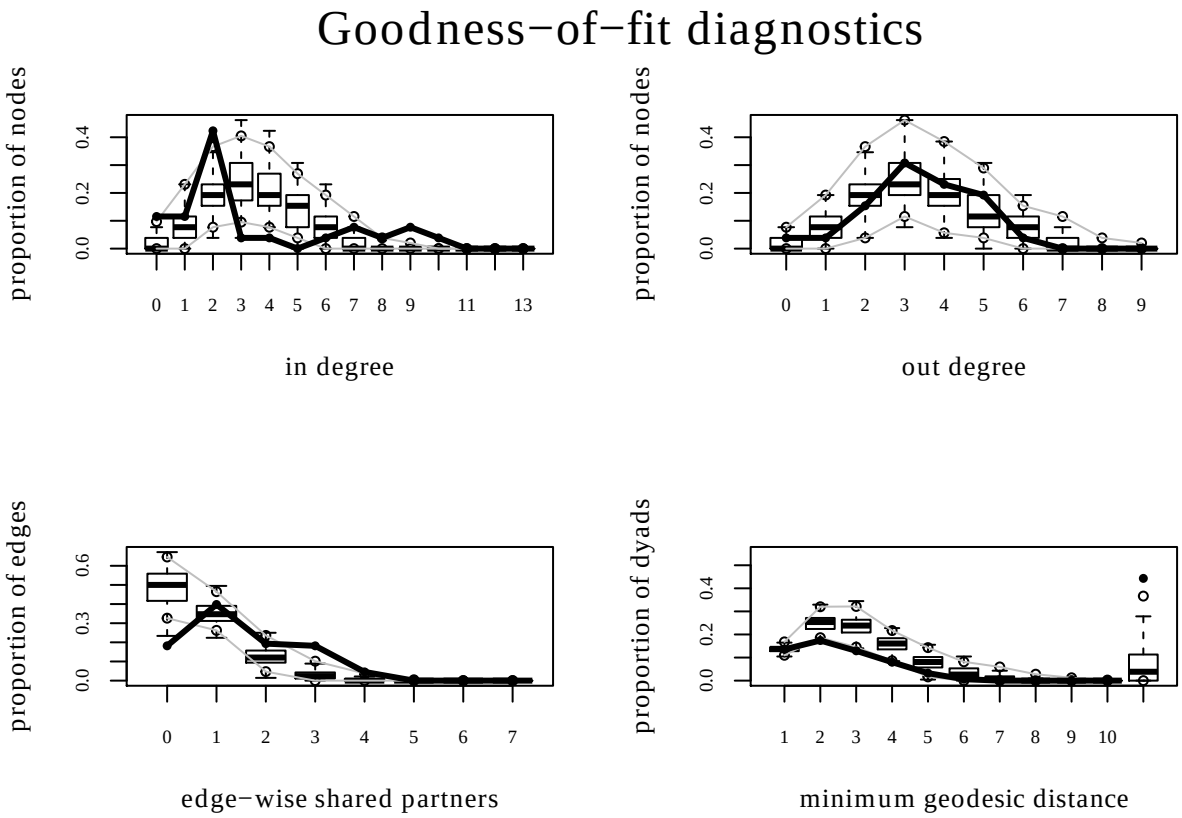
```
AIC(model.ibe0, model.ibe1, model.ibe2a, model.ibe2b)
```

```
##             df      AIC
## model.ibe0   1 517.4447
```

```
## model.ibe1   2 429.4702
## model.ibe2a  4 481.0273
## model.ibe2b  5 413.7863
```

Homophily model seems to fit best. Let's examine the (lack of) fit in some more detail with the gof function.

```
fit2b <- gof(model.ibe2b)
layout(matrix(1:4, 2, 2, byrow=TRUE))
plot(fit2b)
```



5.3 Social circuit models

Here we present an example of an ERGM which contains a statistic which subsumes to the "Social Circuit" dependence assumptions presented earlier. It is Geometrically Weighted Edgwise Shared Partners (GWESP). With this statistic we model the distribution of the number of network partners of nodes, that are themselves connected (hence "edge" in the name of the statistic). Positive effect indicates the tendency for creating transitive triplets.

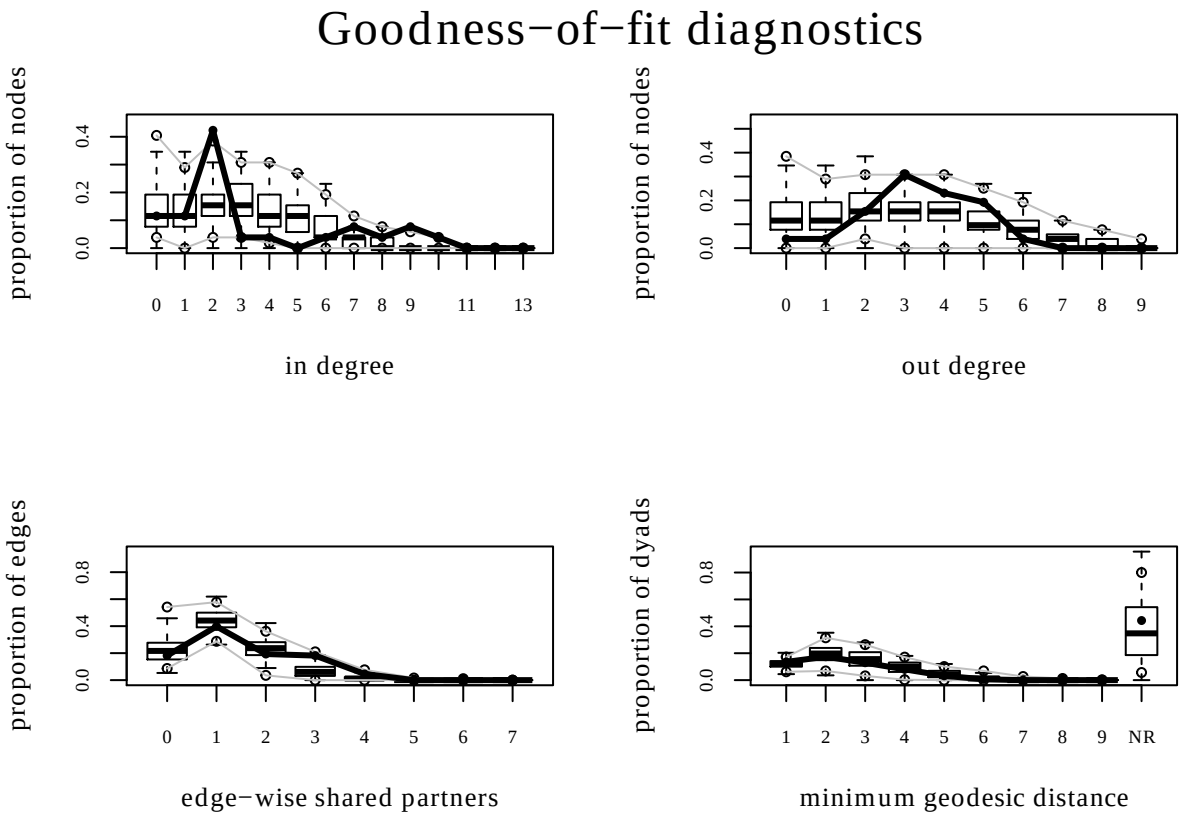
Let us fit an ERGM with the GWESP statistic together with gender homophily effects investigated earlier.

```
model.ibe4 <- ergm(ibe ~ edges + mutual +
                     nodeofactor("female") + nodeifactor("female") + nodematch("female") +
                     gwesp(alpha=0.2, fixed=TRUE),
                   control=control.ergm(seed=666))
```

```
## Starting maximum likelihood estimation via MCMLE:
## Iteration 1 of at most 20:
## The log-likelihood improved by 4.455
## Iteration 2 of at most 20:
```

```
## The log-likelihood improved by 3.277
## Step length converged once. Increasing MCMC sample size.
## Iteration 3 of at most 20:
## The log-likelihood improved by 1.472
## Step length converged twice. Stopping.
## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 1
##
## This model was fit using MCMC.  To examine model diagnostics and check for degeneracy, use the mcmc.diag
```

```
fit4 <- gof(model.ibe4)
layout(matrix(1:4, 2, 2, byrow=TRUE))
plot(fit4)
```



Model fit still not ideal. In particular, the models does not explain the spike in the in-degree distribution for degree 2. Nevertheless, the fit is much better than in models estimated earlier. Let us have a look at the estimates:

```
summary(model.ibe4)
```

```
##
## ==========================
## Summary of model fit
## ==========================
##
## Formula:   ibe ~ edges + mutual + nodeofactor("female") + nodeifactor("female") +
##     nodematch("female") + gwesp(alpha = 0.2, fixed = TRUE)
##
## Iterations:  3 out of 20
##
## Monte Carlo MLE Results:
```

```
##                          Estimate Std. Error MCMC % p-value
## edges                     -4.5169     0.3551      0 < 1e-04 ***
## mutual                     1.1776     0.3885      0 0.00253 **
## nodeofactor.female.TRUE    0.3474     0.3326      0 0.29653
## nodeifactor.female.TRUE   -0.3859     0.3289      0 0.24103
## nodematch.female           1.4969     0.3136      0 < 1e-04 ***
## gwesp.fixed.0.2            1.1503     0.2668      0 < 1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 901.1  on 650  degrees of freedom
##  Residual Deviance: 372.0  on 644  degrees of freedom
##
## AIC: 384    BIC: 410.8    (Smaller is better.)
```

From the results we can observe:

- Friendship nominations are reciprocated
- There is a strong gender homophily effect
- GWESP effect is positive, hence friendship nominations are transitive. In other words, ceteris paribus, friend of a friend is likely to be a friend too.

6 More technical details for interested and technically inclined readers

In the following two sections we present a somewhat more detailed discussion of how an ERG model can be interpreted as:

1. A model for a probability distribution over all networks of a given size.
2. A model for conditional probabilities of tie existence given the structure of the remainder of the network.

Interpretation (1) can be called *global* while interpretation (2) can be called *local*.

6.1 Model-based network probabilities

This is a more advanced topic.

As mentioned, ERGMs assign probabilities to all networks with given size (number of nodes) according to some statistic (measure/attribute). To see how it works let's consider undirected network of size 4. There are 64 such networks in total, but some of them differ only in node permutation, so there are only 11 topologically different networks.

Assume we have three models: null model, model with one parameter (number of edges) and with two parameters (number of edges and 2-stars/twopaths). Parameters are equal $\theta_1 = -0.5$ for number of edges (both models) and $\theta_2 = 0.2$ for number of twopaths. We will calculate probability of each type of network under all three models.

First we need to generate all possible networks of size 4 and select one representative for each canonical form.

```
# all 4-node networks
adj <- as.matrix(expand.grid(0:1, 0:1, 0:1, 0:1, 0:1, 0:1))
full_edgelist <- subset(expand.grid(1:4, 1:4), Var1 > Var2)
nets <- lapply(seq(nrow(adj)), function(i) {
  network.edgelist(full_edgelist[as.logical(adj[i,]), ],
                   network.initialize(4, directed = FALSE))
})
```

```
nets <- nets[order(sapply(nets, network.edgecount))]

# unique canonical permutation (based on degree sequence, works for 4 nodes)
degrees <- as.data.frame(t(sapply(nets, function(net) summary(net ~ degree(0:3)))))
degrees <- data.frame(degrees, id = apply(degrees, 1, paste, collapse = ""))
canonical_count <- table(degrees$id)
nets_unique <- nets[match(unique(degrees$id), degrees$id)]
```

Now we calculate the numerator in equation for probability for every canonical form and each model. Under null model every network is equally probable, so we don't have to compute anything. For other functions small function will come in handy.

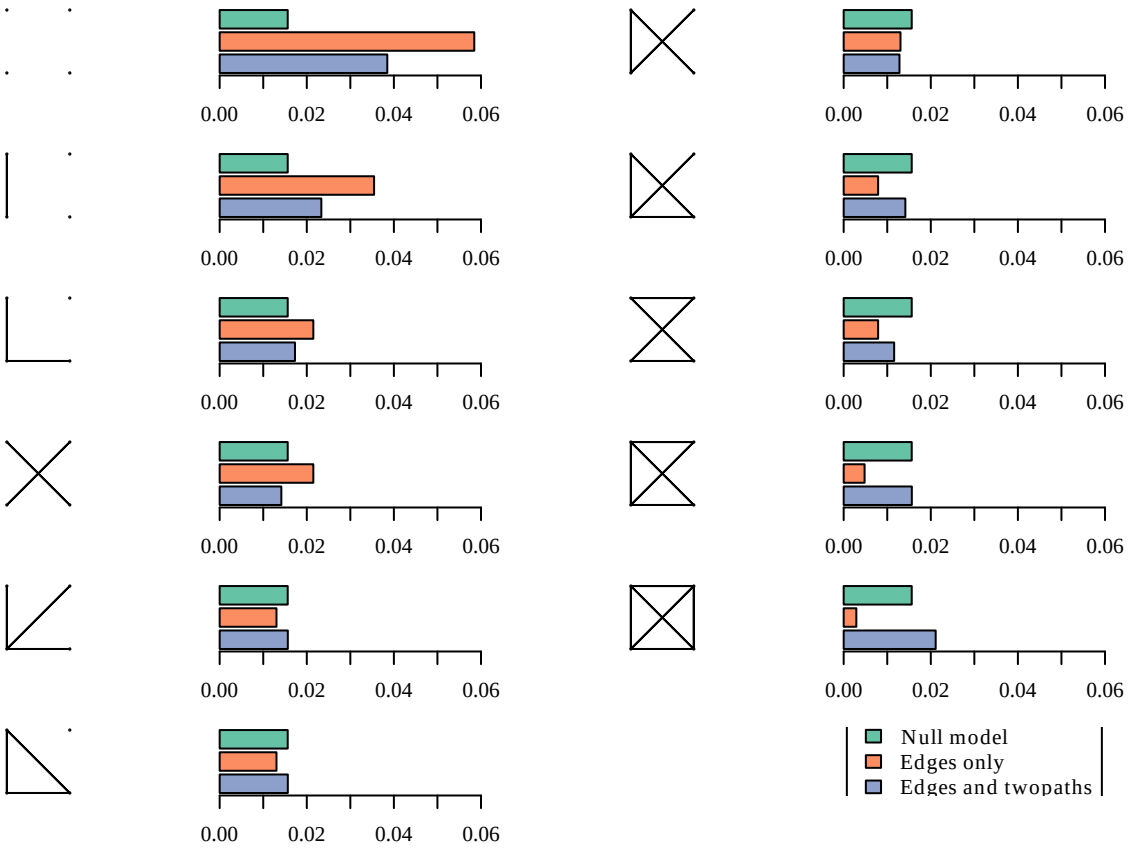
```
model0 <- rep(1, 11)

prob <- function(net, coeff) {
  form <- as.formula(paste("net ~", paste(names(coeff), collapse = "+"), collapse =" "))
  z <- summary(form)
  exp(sum(z * coeff))
}
model1 <- sapply(nets_unique, prob, coeff = c(edges = -0.5))
model2 <- sapply(nets_unique, prob, coeff = c(edges = -0.5, twopath = 0.2))
```

Next step is calculating proper probabilities. To do so we sum numerators over all canonical forms taking the size of each group into account.

```
model0 <- model0 / sum(model0 * canonical_count)
model1 <- model1 / sum(model1 * canonical_count)
model2 <- model2 / sum(model2 * canonical_count)
```

And plot them.

HUMAN CAPITAL NATIONAL COHESION STRATEGY
UNIVERSITY OF WARSAW
ICS RUG / UU / RU
EUROPEAN UNION EUROPEAN SOCIAL FUND
Project co-financed by the European Union under the European Social Fund



All networks has the same probabilities under null model obviously. It is also clear that under model with edges only probability is decreasing as density (or edge count) is increasing. This is caused by negative edge parameter - less dense networks are preferable. Last model is the most interesting one - we could notice that probability decreases in the beginning as density increases, but afterwards it starts to increase again. Edge parameter is still negative so dense networks are penalised. However, second parameter is positive and number of twopath (2-stars) increases more or less faster than number of edges. Therefore probabilities of dense networks are increasing.

6.2 Conditional edge probabilities

This a more advanced topic.

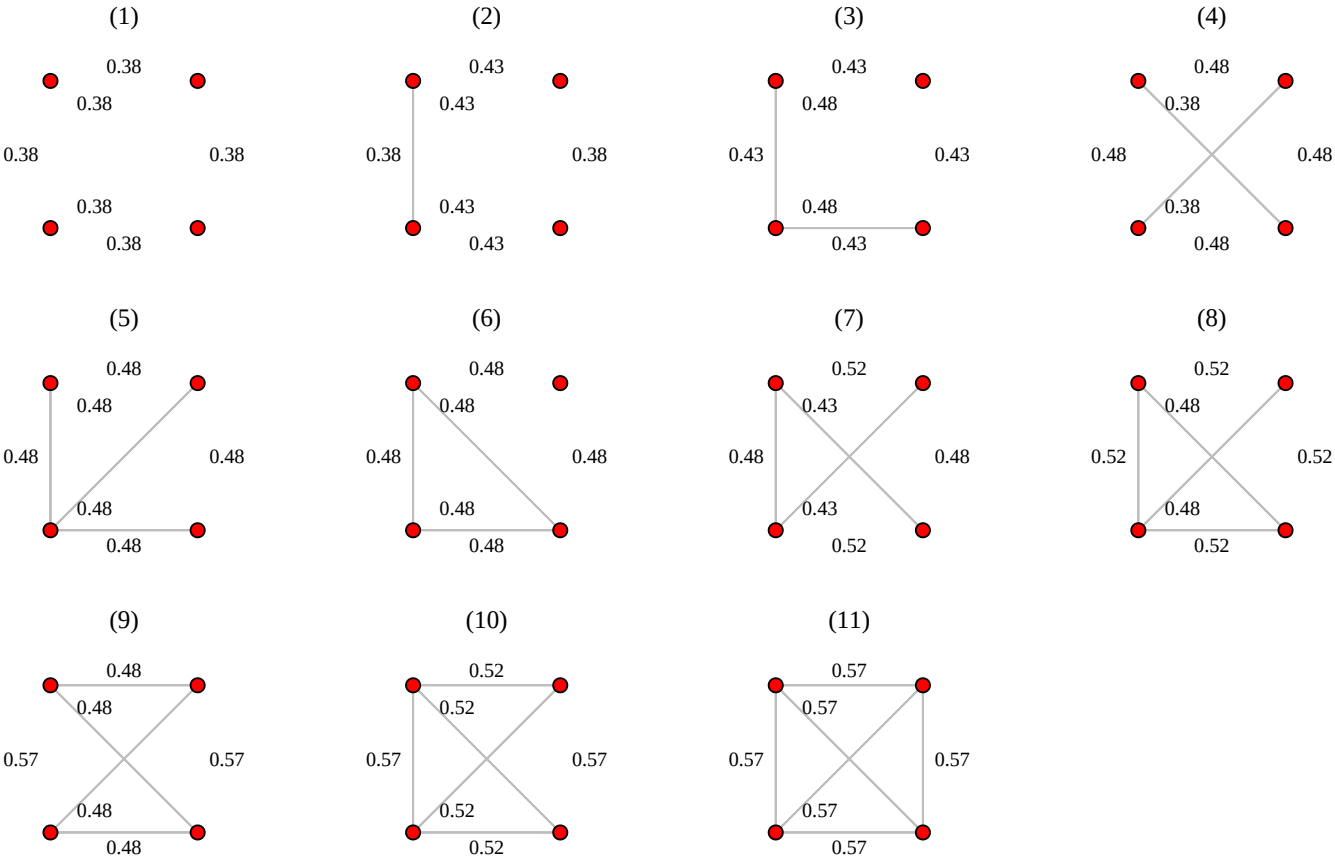
The model could be equivalently described by conditional probabilities that an edge exists given the rest of the network. Conditional log-odds could be easily derived from the general form of the model

$$\frac{P_\theta(A_{ij}=1|A^c_{ij}=a^c_{ij})}{P_\theta(A_{ij}=0|A^c_{ij}=a^c_{ij})} = \exp\{\theta^T\delta[g(a)]_{ij}\}$$

where $\delta[g(a)]_{ij}$ is the change of $g(a)$ when a_{ij} is changed from 0 to 1. So the probability of an edge is equal to

$$\begin{aligned} P_\theta(A_{ij}=1|A^c_{ij}=a^c_{ij}) &= \frac{\exp\{\theta^T\delta[g(a)]_{ij}\}}{1+\exp\{\theta^T\delta[g(a)]_{ij}\}} \\ &= (1+\exp\{-\theta^T\delta[g(a)]_{ij}\})^{-1}, \end{aligned}$$

Coming back to 4-nodes networks. Let's say we have a model with two statistics defined above: number of edges and number of 2-stars (twopaths) with corresponding parameters $\theta_1 = -0.5$ and $\theta_2 = 0.2$.



Look close on 7th network. Where has probability 0.52 come from? So assume that we create an edge between top two nodes. Then number of edges is obviously increased by 1, but number of twopaths is increased by 3 – new edge creates twopath with every other edges. Now we have change statictics so we could compute probability from equation

$$\frac{\exp\{-0.5 \cdot 1 + 0.2 \cdot 3\}}{1 + \exp\{-0.5 \cdot 1 + 0.2 \cdot 3\}} = 0.5249792$$

7 References

Akaike, Hirotogu. 1998. "Information Theory and an Extension of the Maximum Likelihood Principle." In *Selected Papers of Hirotugu Akaike*, 199–213. Springer.

Frank, Ove. 1991. "Statistical Analysis of Change in Networks." *Statistica Neerlandica* 45 (3): 283–93.

Frank, Ove, and David Strauss. 1986. "Markov Graphs." *Journal of the American Statistical Association* 81 (395): 832–42.

Handcock, Mark S., David R. Hunter, Carter T. Butts, Steven M. Goodreau, and Martina Morris. 2003. *Statnet: Software Tools for the Statistical Modeling of Network Data*. Seattle, WA. http://statnetproject.org.

Holland, Paul W., and Samuel Leinhardt. 1976. "Local Structure in Social Networks." In *Sociological Methodology: 1976*, edited by D. Heise, 1–45. San Francisco: Jossey-Bass.

Koskinen, Johan, and Galina Daraganova. 2012. "Dependence Graphs and Sufficient Statistics." In *Exponential Random Graphs Models for Social Networks*, edited by Dean Lusher, Johan Koskinen, and Garry Robins. Cambridge University Press.

HUMAN CAPITAL NATIONAL COHESION STRATEGY
UNIVERSITY OF WARSAW
ICS RUG / UU / RU
EUROPEAN UNION EUROPEAN SOCIAL FUND
Project co-financed by the European Union under the European Social Fund

Lusher, Dean, Johan Koskinen, and Garry Robins. 2012. *Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press.

McPherson, Miller, Lynn Smith-Lovin, and James M Cook. 2001. "Birds of a Feather: Homophily in Social Networks." *Annual Review of Sociology*. JSTOR, 415–44.

Pattison, Philippa, and Garry Robins. 2002. "Neighborhood–Based Models for Social Networks." *Sociological Methodology* 32 (1): 301–37.

Snijders, Tom A. B., and Roel Bosker. 2011. *Multilevel Analysis*. Springer.

Snijders, Tom A. B., Philippa E. Pattison, Garry L. Robins, and Mark S. Handcock. 2006. "New Specifications for Exponential Random Graph Models." *Sociological Methodology* 36 (1): 99–153.

Wang, Peng, Garry Robins, and Philippa Pattison. 2009. *PNet: Program for the Simulation and Estimation of Exponential Random Graph (P*) Models. User Manual*. Melbourne, Australia: Department of Psychology. School of Behavioral Science. University of Melbourne. http://sna.unimelb.edu.au/PNet.

Wasserman, Stanley, and Philippa Pattison. 1996. "Logit Models and Logistic Regressions for Social Networks: I. an Introduction to Markov Graphs and P*." *Psychometrika* 61 (3): 401–25.

HUMAN CAPITAL NATIONAL COHESION STRATEGY
UNIVERSITY OF WARSAW
ICS RUG / UU / RU
EUROPEAN UNION EUROPEAN SOCIAL FUND
Project co-financed by the European Union under the European Social Fund