

Stable Exponential Random Graph Models with Non-parametric Components for Large Dense Networks

S. Thiemichen^a, G. Kauermann^a.

^a*Institut für Statistik Ludwigs-Maximilians-Universität München, Germany.*

April 18, 2022

Abstract

Exponential Random Graph Models (ERGM) behave peculiar in large networks with thousand(s) of actors (nodes). Standard models containing two-star or triangle counts as statistics are often unstable leading to completely full or empty networks. Moreover, numerical methods break down which makes it complicated to apply ERGMs to large networks. In this paper we propose two strategies to circumvent these obstacles. First, we fit a model to a subsampled network and secondly, we show how linear statistics (like two-stars etc.) can be replaced by smooth functional components. These two steps in combination allow to fit stable models to large network data, which is illustrated by a data example including a residual analysis.

Keywords: Exponential random graph models; Conditional independence; Subsampling; Smooth non-parametric components; Network analysis

1 Introduction

The analysis of network data is an emerging field in statistics. It is challenging both model-wise and computationally. Recently, Goldenberg et al. (2010), Hunter et al. (2012), and Fienberg (2012) published comprehensive survey articles discussing new statistical approaches and developments in network data analysis. We also refer to the monograph of Kolaczyk (2009) for a general introduction to the field, or the recent book of Lusher et al. (2013), which focuses on a specific and widely used class of network models, so-called Exponential Random Graph Models (ERGM).

In its most simple form a network consists of a set of n nodes (actors) which are potentially linked with each other through edges. These edges between the actors are thereby the focus of interest. Notationally a network can be expressed as a $n \times n$ (random) adjacency matrix \mathbf{Y} with entries $Y_{ij} = 1$ if node i and j are connected, and $Y_{ij} = 0$ otherwise. In undirected networks one has $Y_{ij} = Y_{ji}$ while for directed links we have $Y_{ij} = 1$ if a directed edge goes from node i to node j . For the sake of readability and notional simplicity we will concentrate here on undirected networks. The term \mathbf{y} denotes a concrete realisation of \mathbf{Y} .

A common and powerful model for network data \mathbf{Y} was proposed by Frank and Strauss (1986) as Exponential Random Graph Model (ERGM) taking the form

$$\mathbb{P}(\mathbf{Y} = \mathbf{y} | \boldsymbol{\theta}) = \frac{\exp \left\{ \sum_{l=0}^p s_l(\mathbf{y}) \theta_l \right\}}{\kappa(\boldsymbol{\theta})}, \quad (1)$$

with $\boldsymbol{\theta} = (\theta_0, \dots, \theta_p)^t$ as parameter vector and $s(\mathbf{y}) = (s_0(\mathbf{y}), \dots, s_p(\mathbf{y}))^t$ as vector of statistics of the network. In equation (1) the term $\kappa(\boldsymbol{\theta})$ denotes the normalizing constant, that is

$$\kappa(\boldsymbol{\theta}) = \sum_{\mathbf{y} \in \mathcal{Y}} \exp \left\{ \boldsymbol{\theta}^t s(\mathbf{y}) \right\},$$

where \mathcal{Y} is the set of all networks and accordingly the sum is over $2^{\binom{n}{2}}$ terms. It is therefore numerically intractable, except for very small graphs. We denote with $s_0(\mathbf{y}) = \sum_{i=1}^n \sum_{j>i}^n y_{ij}$ the baseline statistic giving the number of edges in the (undirected) network, so that θ_0 serves as intercept. The interpretation of the remaining parameters θ_l , $l = 1, \dots, p$, results through the corresponding conditional model for each single edge Y_{ij} given the remaining network $\mathbf{Y} \setminus Y_{ij}$, since

$$\text{logit} \left[\mathbb{P}(Y_{ij} = 1 | \mathbf{Y} \setminus Y_{ij}; \boldsymbol{\theta}) \right] = \theta_0 + \sum_{l=1}^p \Delta_{ij} s_l(\mathbf{y}) \theta_l, \quad (2)$$

where $\Delta_{ij} s_l(\mathbf{y}) = s_l(\mathbf{y} \setminus y_{ij}, y_{ij} = 1) - s_l(\mathbf{y} \setminus y_{ij}, y_{ij} = 0)$ is the so-called change statistics which is obtained by flipping the edge between nodes i and j from non-existent to existent.

Exponential Random Graph Models are numerically unstable, in particular if the number of actors n gets large. Hence, for large networks one is faced with two relevant problems.

First, the model itself is notoriously unstable leading to either full or empty networks. This issue is usually called degeneracy problem, see, for example, (Schweinberger, 2011), and Chatterjee and Diaconis (2013). Secondly, the estimation is per se numerically demanding or even unfeasible since numerical simulation routines are too time consuming. We aim to tackle both problems in this paper. First, we propose the use of stable statistics which are derived as smooth, non-parametric curves. Secondly, instead of fitting the model to the entire network we propose to draw samples from the network such that estimation in each sample is numerically (very) easy. These two proposals allow to easily analyse network data in large and sufficiently dense networks.

Schweinberger (2011) denotes network statistics (and the corresponding ERGM) as unstable if the statistics is not at least of order $O_p(n)$. In fact he shows that any k -star or triangle statistics is unstable leading to an odd behaviour of model (1). Effectively, unstable networks are either complete (i.e. have all possible edges) or empty (i.e. all nodes are unconnected) unless for a diminishing subspace of the parameter space for n increasing. If n gets large it is therefore advisable to replace the statistics in model (1) by stable statistics of order $O_p(n)$. A first proposal in this direction are alternating star and alternating triangle statistics as proposed in Snijders et al. (2006), or geometrically weighted statistics as proposed in the context of Curved Exponential Random Graph Models, see Hunter and Handcock (2006). Hunter (2007) shows that from a modelling point of view the alternating statistics are equivalent to geometrically weighted degree or geometrically weighted edgewise shared partners, respectively. Both approaches stabilize the models but for the price of less intuitive interpretations of the parameter estimates. We propose an alternative by making use of non-parametric models based and the technique of smoothing (see, e.g., Ruppert et al., 2003). The non-parametric model thereby maintains the interpretability of the ERGM based on the conditional model (2). To motivate our idea we start with the conditional model (2) and replace the linear terms through non-linear smooth components. This leads to the conditional non-parametric model

$$\text{logit}\left[\mathbb{P}\left(Y_{ij} = 1|\mathbf{Y}\setminus Y_{ij}\right)\right] = \theta_0 + \sum_{l=1}^p m_l(\Delta_{ij}s_l(\mathbf{y})), \quad (3)$$

where $m_l(\cdot)$ are smooth functions which need to be estimated from the data. Models of type (3) have been proposed in a simple regression framework as generalized additive models, see, e.g., Hastie and Tibshirani (1990), or Wood (2006), but apparently the structure here is more complex as we are tackling network data. We additionally need to postulate that functions $m_l(\cdot)$ are monotone and bounded which in turn leads to stable network statistics in the definition of Schweinberger (2011). We make use of penalized spline smoothing which also allows to accommodate constraints on the functional shape leading to stable network models. In fact, assuming $m_l(\cdot)$ to be monotone and bounded, we may derive a non-parametric Exponential Random Graph Model from (3) which takes the form

$$\mathbb{P}\left(\mathbf{Y} = \mathbf{y}|\theta_0, m_l(\cdot), l = 1, \dots, p\right) = \frac{\exp\left\{s_0(\mathbf{y})\theta_0 + \sum_{l=1}^p \sum_i \sum_{j>i} y_{ij} m_l(\Delta_{ij}s_l(\mathbf{y}))\right\}}{\kappa(\theta_0, m_l(\cdot), l = 1, \dots, p)} \quad (4)$$

Apparently, model (4) appears rather complex due to its semi-parametric structure and estimation looks like a challenging task. We will argue, however, that smoothing techniques can easily be applied and estimation becomes feasible by making use of sampling strategies in networks leading to numerically simple likelihoods and in fact consistent (though not efficient) estimates.

Estimation in Exponential Random Graph Models is cumbersome and numerically demanding as it requires simulation based routines. Snijders (2002) suggests the calculation of $\partial\kappa(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ in the score equation resulting from (1) using stochastic approximation. Hunter and Handcock (2006) propose to use MCMC methods in order to obtain the maximum likelihood estimate. The approach is extended and improved in Hummel et al. (2012). In a recent paper Caimo and Friel (2011) develop a fully Bayesian estimation routine by incorporating the so-called exchange algorithm from Murray et al. (2006) which circumvents the calculation or approximation of the normalisation constant for the price of extended MCMC sampling. A general survey of available routines for fitting Exponential Random Graph Models is given in Hunter et al. (2012). In fact, if the network is large, MCMC based routines readily become numerically infeasible. As aforementioned, we will therefore make use of subsampling the network data and fit the model to subsamples that allow for simple likelihoods. We follow ideas of Koskinen and Daraganova (2013). In fact, for models with k -stars or triangles only, the edges follow a Markovian independence structure by conditioning on parts of the network (see Frank and Strauss, 1986, or Whittaker, 2009). This is exemplified in a simple network with four nodes in Figure 1. Conditioning on edges Y_{12}, Y_{14}, Y_{23} , and Y_{34} we find that Y_{13} and Y_{24} are conditionally independent, which can be denoted as $Y_{13} \perp\!\!\!\perp Y_{24} | \mathbf{Y} \setminus \{Y_{13}, Y_{24}\}$. The idea is now to make use of this independence property to fit model (4) to a subsample of the network while conditioning on the rest of the network. Hence, exemplary we sample edges Y_{13} and Y_{24} , and condition on $\mathbf{Y} \setminus \{Y_{13}, Y_{24}\}$. Due to the (conditional) independence structure we can easily fit the conditional model (2) with standard software for generalized linear and non-parametric additive models. This will be demonstrated below. Apparently such a strategy is not efficient if the network is small, but if the network is (very) large and (sufficiently) dense, sampling appears as a plausible approach which also maintains numerical feasibility.

The paper is organized as follows. In Section 2 we suggest to estimate large Exponential Random Graph Models through subsampling of the network. In Section 3 we extend the idea towards non-parametric models. Section 4 gives a data example demonstrating the usability of the approach. Finally, a discussion completes the paper in Section 5.

All routines for fitting and analysing the models are written in R (R Core Team, 2016) and will be made available as R package.

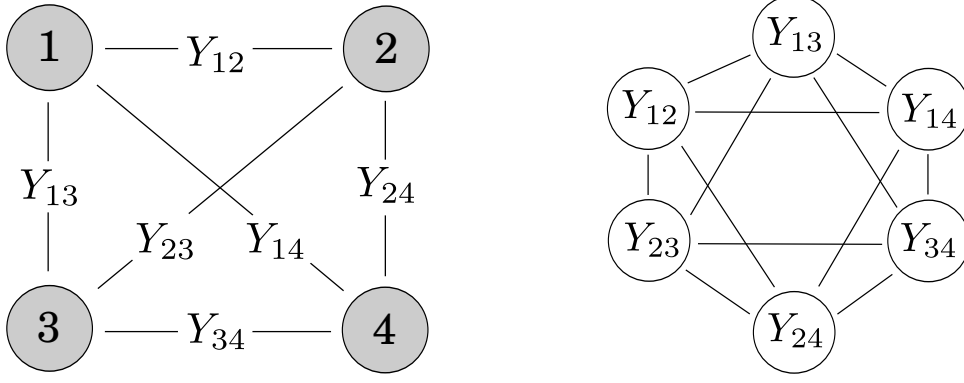


Figure 1 Visualisation of the induced Markov independence graph (right) for a Exponential Random Graph Model for a simple 4-node network (left).

2 Estimation through Subsampling

The general idea proposed in this section is that instead of fitting an ERGM to the entire data, we fit a conditional model to appropriate subsamples of the data. Due to conditional independence this allows for fast and easy computing. We start the presentation with the classical (unstable) ERGM and assume model (1) has statistics like k -star and triangle effects only. That is statistics $s_l(\cdot)$ in (1) for instance has no “4-cycles” of the form $\sum_{i < j < k < l} Y_{ij}Y_{jk}Y_{kl}Y_{li}$ (or higher order cycles). Let us get more specific. For simplicity of presentation let n , the number of nodes in the network, be even. With $\mathcal{D}(n|2)$ we denote a decomposition of the set $\{1, \dots, n\}$ into subsets of size 2, e.g., $\mathcal{D}(n|2) = \{(1, 2), (3, 4), \dots, (n-1, n)\}$. For $A = (i, j) \in \mathcal{D}(n|2)$ we denote $Y_A = Y_{ij}$ and $\mathbf{Y} \setminus Y_{\mathcal{D}(n|2)} = \{Y_{ij}, (i, j) \notin \mathcal{D}(n|2)\}$. Apparently $\mathcal{D}(n|2)$ has $n/2$ elements. We assume now that the statistics $s_l(\mathbf{y})$ in (1) can be decomposed to

$$s_l(\mathbf{y}) = \sum_{A \in \mathcal{D}(n|2)} s_{lA}(y_A, \mathbf{y} \setminus y_{\mathcal{D}(n|2)}). \quad (5)$$

This holds for all k -stars and triangle statistics. It is not difficult to show that with condition (5) density (1) can then be factorized to

$$\mathbb{P}(\mathbf{Y} = \mathbf{y}) = \prod_{A \in \mathcal{D}(n|2)} h_A(y_A, \mathbf{y} \setminus y_{\mathcal{D}(n|2)}), \quad (6)$$

where $h_A(\cdot)$ is some function depending on $\{y_A, \mathbf{y} \setminus y_{\mathcal{D}(n|2)}\}$. The factorization (6) implies that the edges with indices in $\mathcal{D}(n|2)$ are mutually independent conditional on the rest of the network (see Whittaker, 2009).

The conditional independence will be used to fit model (1) not for the entire network but for an appropriately chosen subnetwork. We therefore draw a sample of the network \mathbf{Y} by taking \mathbf{y}_A with $A \in \mathcal{D}(n|2)$ as sampled binary observations accompanied by $\Delta_A s(\mathbf{y}) = (\Delta_A s_1(\mathbf{y}), \dots, \Delta_A s_p(\mathbf{y}))^t$ as corresponding change statistics with obvious definition of $\Delta_A s(\cdot)$. The term $\Delta_A s(\mathbf{y})$ plays the role of covariates and the conditional model (2) takes the form

$$\text{logit} \left[\mathbb{P}(Y_A = 1 | \Delta s(\mathbf{y}), \boldsymbol{\theta}) \right] = \theta_0 + \sum_{l=1}^p \Delta_A s_l(\mathbf{y}) \theta_l = \theta_0 + \sum_{l=1}^p x_l \theta_l,$$

where x_l denotes the change statistics $\Delta_{As_l}(\mathbf{y})$ which is considered as covariate in the logit model. Due to the induced conditional independence the likelihood for the sample results to

$$\mathcal{L}_{\mathcal{D}(n|2)}(\boldsymbol{\theta}) = \prod_{A \in \mathcal{D}(n|2)} \mathbb{P}(Y_A = 1 | \Delta_{As}(\mathbf{y}), \boldsymbol{\theta}), \quad (7)$$

which is easily fitted using standard software for generalized linear models. Note that (7) is the true likelihood for the conditional subsample so that consistent estimates and their variance estimates are easily available. This means, by taking the subsample of edge variables Y_A with $A \in \mathcal{D}(n|2)$ and conditioning in the remaining graph we circumvent numerical estimation problems and remain in the classical generalized linear model framework. It also implies that we can estimate $\boldsymbol{\theta}$ consistently (for n increasing) by maximizing $\mathcal{L}_{\mathcal{D}(n|2)}(\boldsymbol{\theta})$.

	1	2	3	4
1	0	1	2	3
2	1	0	3	2
3	2	3	0	1
4	3	2	1	0

Figure 2 Symmetric Latin Square with unique diagonal.

Apparently, we may draw different samples of edges leading to different estimates. This means using different decomposition sets $\mathcal{D}(n|2)$ leads to different estimates. This leaves us with the question how to combine the different estimates. We may either draw $\mathcal{D}(n|2)$ randomly or make use of a combinatorial approach to cover the entire network Y_{ij} . Let therefore

$$\mathcal{P} = \{\mathcal{D}_k(n|2), k = 1, \dots, n-1\}$$

be a sequence of sets $\mathcal{D}_k(n|2)$ such that each index pair is exactly in one single set $\mathcal{D}_k(n|2)$. That is for Y_{ij} there exists exactly one set $\mathcal{D}_k(n|2) \in \mathcal{P}$ with $(i, j) \in \mathcal{D}_k(n|2)$. The $n-1$ sets $\mathcal{D}_k(n|2)$ in \mathcal{P} can be constructed using a symmetric Latin Square with a unique diagonal (see, e.g., Andersen and Hilton, 1980).¹ For instance for $n = 4$ nodes Figure 2 shows a symmetric Latin Square. As we are focusing on undirected networks, where the corresponding adjacency matrix is symmetric, we use only the upper diagonal of the Latin Square. We may take the entries in the Latin Square as the sample number. For instance, $\mathcal{D}_1(n|2)$ results by taking the pairs with entries 1 in the upper triangle from the

¹A description of a possible algorithm for the construction of such a symmetric Latin square with a unique diagonal is available, e.g., from Bogomolny (2016).

corresponding network adjacency matrix, i.e. $(1, 2), (3, 4)$ and condition on the remaining variables. Accordingly we proceed for entries 2 and 3 in the Latin Square. We denote with $\hat{\boldsymbol{\theta}}_{\langle k \rangle}$ the resulting estimate from sequence set $\mathcal{D}_k(n|2)$. Note that each estimate $\hat{\boldsymbol{\theta}}_{\langle k \rangle}$ is consistent but they are not mutually independent. With $Y_{\langle k \rangle} = \{Y_{ij} : (i, j) \in \mathcal{D}_k(n|2)\}$ we easily get with the asymptotic properties of Maximum Likelihood estimates as $n \rightarrow \infty$ that

$$\mathbb{E}(\hat{\boldsymbol{\theta}}_{\langle k \rangle}) = \mathbb{E}_{\mathbf{Y} \setminus Y_{\langle k \rangle}} \left(\mathbb{E}_{Y_{\langle k \rangle}} (\hat{\boldsymbol{\theta}}_{\langle k \rangle} | \mathbf{Y} \setminus Y_{\langle k \rangle}) \right) \rightarrow \boldsymbol{\theta}.$$

Moreover

$$\begin{aligned} \text{Var}(\hat{\boldsymbol{\theta}}_{\langle k \rangle}) &= \mathbb{E}_{\mathbf{Y} \setminus Y_{\langle k \rangle}} \left(\text{Var}_{Y_{\langle k \rangle}} (\hat{\boldsymbol{\theta}}_{\langle k \rangle} | \mathbf{Y} \setminus Y_{\langle k \rangle}) \right) \\ &\quad + \text{Var}_{\mathbf{Y} \setminus Y_{\langle k \rangle}} \left(\mathbb{E}_{Y_{\langle k \rangle}} (\hat{\boldsymbol{\theta}}_{\langle k \rangle} | \mathbf{Y} \setminus Y_{\langle k \rangle}) \right) \\ &\rightarrow \mathbb{E}_{\mathbf{Y} \setminus Y_{\langle k \rangle}} \left(F_{\langle k \rangle}^{-1}(\boldsymbol{\theta}_{\langle k \rangle}) \right), \end{aligned}$$

where $F_{\langle k \rangle}(\boldsymbol{\theta})$ denotes the (conditional) Fisher matrix corresponding to the likelihood function (7). Apparently $F_{\langle k \rangle}^{-1}(\boldsymbol{\theta}_{\langle k \rangle})$ is an unbiased estimate for $\mathbb{E}_{\mathbf{Y} \setminus Y_{\langle k \rangle}} \left(F_{\langle k \rangle}^{-1}(\boldsymbol{\theta}_{\langle k \rangle}) \right)$. Note that $F_{\langle k \rangle}^{-1}(\hat{\boldsymbol{\theta}}_{\langle k \rangle})$ can be obtained with any software package for fitting logistic regression models. Hence an estimate for the variance is readily available.

3 Non-parametric Exponential Random Graph Models

3.1 Spline-Based Model

We have shown how an appropriate sample of the network allows for simple estimation of the parameters. Apparently this is a recommendable approach only if n , the number of nodes, is large. In this case, however, ERGMs become unstable if the change statistics increase linearly in n . As shown in Schweinberger (2011) this holds for almost all basic models with $\theta_l \neq 0$ for $l > 0$. In other words, even though we are able to estimate the parameters as described before, the resulting network will be either full or empty as n is becoming large. Stability is achieved if the network statistics are of order $O_p(n)$. One intention is therefore to modify the statistics in the model such that they become stable. This is done with non-parametric components so that the change statistics have a bounded influence. To do so we make use of the non-parametric model (3) where we additionally postulate that the smooth functions $m_l(\cdot)$ are monotone and bounded.

To estimate functions $m_l(\cdot)$ we make use of penalized spline smoothing as discussed in detail in Ruppert et al. (2003), and Ruppert et al. (2009), see also Kauermann et al. (2009). The general idea is as follows. First, one replaces the unknown smooth function $m_l(\cdot)$ by a spline basis which is flexible (i.e. high dimensional) enough to capture the underlying true functional relation. As a second step a penalty or regularization is imposed on the

unknown spline coefficients leading to a smooth and numerically stable fit. The third step is to calibrate/estimate the amount of penalization, which is controlled by a smoothing parameter. The original idea goes back to O’Sullivan (1986) and was made popular by the seminal paper of Eilers and Marx (1996). We make use of the idea here, but amend it towards the specific problem of non-stability occurring in large networks. As first step we choose a basis $\mathbf{B}(x) = (B_1(x), \dots, B_K(x))^t$ where $x \in \mathbb{R}^+$ and the basis components $B_q(x)$, for $q = 1, \dots, K$, fulfill the following three properties:

- 1) $B_q(0) \equiv 0$,
- 2) $B_q(x)$ is monotone, and
- 3) $B_q(x)$ is bounded for $x \rightarrow \infty$.

A convenient choice are distribution functions on \mathbb{R}^+ . Here we employ the exponential distribution and set

$$B_q(x) = 1 - \exp(-\gamma_q x), \quad (8)$$

where γ_q are fixed scaling parameters. The set $\{\gamma_1, \dots, \gamma_K\}$ covers a wide range of possible shapes as visualized in Figure 3. We now replace the unknown function $m_l(\cdot)$ in model (3) by the spline representation

$$m_l(x_l) = \mathbf{B}(x_l)^t \mathbf{u}_l, \quad (9)$$

with $\mathbf{B}(x) = (B_1(x), \dots, B_K(x))^t$ and $\mathbf{u}_l = (u_{l1}, \dots, u_{lK})^t$ as the coefficient vector. Note that as long as the coefficients of \mathbf{u}_l are finite we have constructed a bounded and hence stable network statistics. Apparently we need additional constraints on \mathbf{u}_l in order to guarantee monotonicity. This implies for monotonically increasing functions that

$$\mathbf{B}'(x)^t \mathbf{u}_l \geq 0, \quad (10)$$

where $\mathbf{B}'(x) = (\gamma_1 \exp(-\gamma_1 x), \dots, \gamma_K \exp(-\gamma_K x))^t$. This is a linear constraint on the parameters, which for estimation is easily accommodated by quadratic programming. For monotonically decreasing functions we use almost the same constraint. For practical purposes we select the cutpoints of neighbouring basis functions ξ_r with $\gamma_{r+1} \exp(-\gamma_{r+1} \xi_r) = \gamma_r \exp(-\gamma_r \xi_r)$ and set the constraints to $\mathbf{B}'(\xi_r)^t \mathbf{u}_l \geq 0$ for monotonically increasing functions (or to $\mathbf{B}'(\xi_r)^t \mathbf{u}_l \leq 0$ for monotonically decreasing functions), for $r = 1, \dots, K - 1$. Our experiences show a stable behaviour with this setting.

3.2 Penalized Estimation

The second step is now to impose a penalty on the spline coefficients in order to achieve smoothness and numerical stability. For a sample of the network as proposed in the previous section, let $\ell(\theta_0, \mathbf{u})$ be the log-likelihood resulting from model (3) in combination

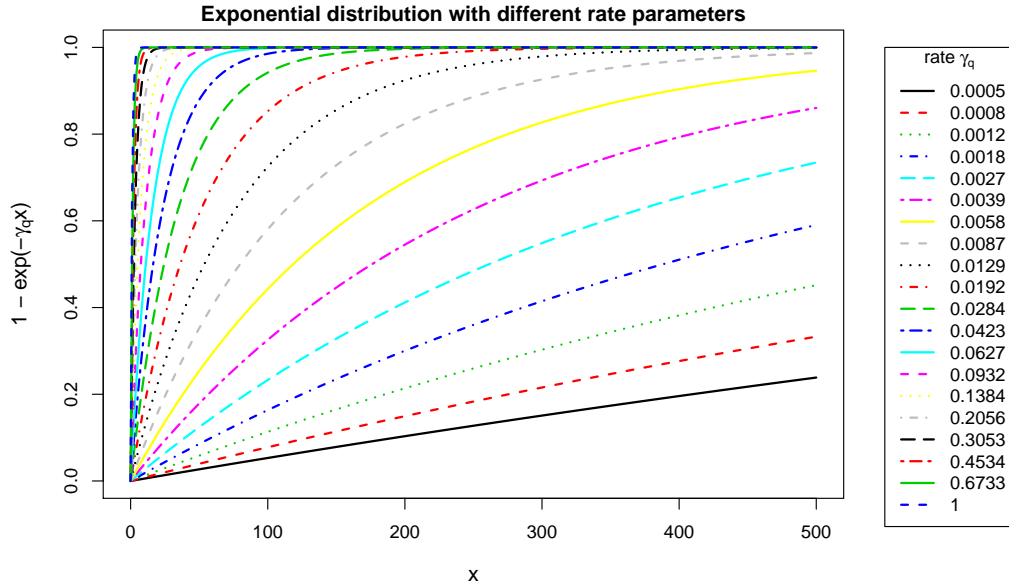


Figure 3 Visualisation of the cumulative distribution function of the exponential distribution with different rate parameters γ_q as example of possible basis functions $B_q(x)$.

with (9), where $\mathbf{u} = (\mathbf{u}_1^t, \dots, \mathbf{u}_p^t)^t$. For notational simplicity we omit the sampling index in this subsection. We emphasize however that the likelihood and hence its estimate do depend on the particular sample of the network. Bear in mind that \mathbf{u} is high dimensional, so that (ML) estimates are unstable and the resulting fits $\mathbf{B}(x)\hat{\mathbf{u}}_l$ would be wiggled. We therefore apply a ridge penalty leading to the penalized log-likelihood

$$\ell_p(\theta_0, \mathbf{u}, \boldsymbol{\lambda}) = \ell(\theta_0, \mathbf{u}) - \frac{1}{2} \sum_{l=1}^p \lambda_l \mathbf{u}_l^t \mathbf{u}_l, \quad (11)$$

where $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_p)^t$ are the penalty parameters. Apparently setting $\lambda_l \rightarrow \infty$ leads to $m_l(\cdot) \equiv 0$ while $\lambda_l \rightarrow 0$ gives an unpenalized fit. It remains therefore to choose $\boldsymbol{\lambda}$ data driven balancing goodness of fit ($\boldsymbol{\lambda} \rightarrow 0$) and parsimony of the model (minimal for $\boldsymbol{\lambda} \rightarrow \infty$). These steps can be carried out with classical cross-validation (see, e.g., Eilers and Marx, 1996) or in a more sophisticated way by comprehending the penalty as normal prior. In this case we follow a Bayesian view and assume $\mathbf{u}_l \sim N(0, \lambda_l^{-1} I_K)$ with I_K as K dimensional unit matrix. Then λ_l is the reciprocal of the a priori variance of \mathbf{u}_l . The connection between penalized estimation and its Bayesian view by imposing normal priors is extensively motivated and discussed in Ruppert et al. (2003). In fact the approach led to a real breakthrough in smooth functional estimation as mirrored in the survey article by Ruppert et al. (2009). Note that the Bayesian approach in our setting here leads to a generalized linear mixed model which is extensively discussed, e.g., in Breslow and Clayton (1993), see also McCulloch et al. (2008). In particular, assuming a normal prior for coefficient vector \mathbf{u}_l we may consider the penalty λ_l as parameter which needs to be estimated. To do so we make use of the procedure of Schall (1991) leading to the following

formulae. With $F(\theta_0, \mathbf{u})$ we denote the Fisher matrix of the conditional model (2). We define the Fisher matrix in the penalized likelihood (11) as

$$\mathbf{F}(\theta_0, \mathbf{u}, \boldsymbol{\lambda}) = \mathbf{F}(\theta_0, \mathbf{u}) + \text{diag}(0, \lambda_1 I_K, \dots, \lambda_p I_K), \quad (12)$$

where $\text{diag}(\cdot)$ denotes a block diagonal matrix with the arguments as blocks. The part of the Fisher matrix belonging to \mathbf{u} is then

$$\widetilde{\mathbf{F}}(\mathbf{u}, \boldsymbol{\lambda}) = \widetilde{\mathbf{F}}(\mathbf{u}) + \text{diag}(\lambda_1 I_K, \dots, \lambda_p I_K), \quad (13)$$

with $\widetilde{\mathbf{F}}(\mathbf{u})$ denoting the part of the Fisher matrix from the conditional model (2) belonging to \mathbf{u} . Following Schall (1991) we can now estimate λ_l^{-1} (iteratively) through

$$\widehat{\lambda}_l^{-1} = \frac{\mathbf{u}_l^t \mathbf{u}_l}{\text{df}(\lambda_l)}, \quad (14)$$

where

$$\text{df}(\lambda_l) = \text{tr} \left\{ \left[\widetilde{\mathbf{F}}^{-1}(\mathbf{u}, \boldsymbol{\lambda}) \widetilde{\mathbf{F}}(\mathbf{u}, 0) \right]_l \right\},$$

and subscript l means that we take only the submatrix matching to component \mathbf{u}_l . See Kauermann (2005), or Krivobokova and Kauermann (2007) for a derivation of the estimate. Finally, the monotonicity constraint (10) is taken into account by quadratic programming which is available in R using the package `quadprog` (Turlach and Weingessel, 2013). The following algorithm describes the iterative procedure.

Algorithm 1: Fit non-parametric ERGM, i.e. estimate $\boldsymbol{\beta} = (\theta_0, \mathbf{u}^t)^t$ and $\boldsymbol{\lambda}$.

Preparation: Fit Standard GLM for

$$\text{logit} \left[\mathbb{P}(Y_{ij} = 1 | \mathbf{Y} \setminus Y_{ij}; \boldsymbol{\theta}) \right] = \theta_0 + \sum_{l=1}^p \Delta_{ij} s_l(\mathbf{y}) \theta_l$$

to determine effect directions. The smooth effect $m_l(\cdot)$ is constrained to

- (a) a monotonically increasing function if $\widehat{\theta}_l \geq 0$, and
- (b) a monotonically decreasing function if $\widehat{\theta}_l < 0$.

Matrix \mathbf{A} is set up using the resulting monotonicity constraints according to (10).

Instead of maximizing $\ell_p(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \ell_p(\boldsymbol{\beta}, \boldsymbol{\lambda})$ directly under the constraints from (10), we use a Taylor expansion of

$$\ell_p(\boldsymbol{\beta}, \boldsymbol{\lambda}) - \ell_p(\boldsymbol{\beta}^{(t)}, \boldsymbol{\lambda}) \approx \mathbf{s}_p(\boldsymbol{\beta}^{(t)}, \boldsymbol{\lambda})^t (\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}) + \frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)})^t \mathbf{H}_p(\boldsymbol{\beta}^{(t)}, \boldsymbol{\lambda}) (\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}),$$

where $\mathbf{s}_p(\cdot)$ denotes the penalized score function and \mathbf{H}_p the penalized Hessian.

Initiate starting values $\boldsymbol{\beta}^{(0)}, \boldsymbol{\lambda}^{(0)}, t = 0, s = 0$.

Step 1: Use current value $\boldsymbol{\lambda}^{(s)}$ and iterate until convergence or until max. no. of iterations t_{\max} is reached:

- (i) Solve $\min_{\mathbf{b}} \left(-\mathbf{d}^t \mathbf{b} + 1/2 \mathbf{b}^t \mathbf{D} \mathbf{b} \right)$ for $\mathbf{b} = \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)} \right)$ with constraint $\mathbf{A}^t \mathbf{b} \geq \mathbf{b}_0$, where $\mathbf{d} = \mathbf{s}_p(\boldsymbol{\beta}^{(t)}, \boldsymbol{\lambda}^{(s)})$, $\mathbf{D} = -\mathbf{H}_p(\boldsymbol{\beta}^{(t)}, \boldsymbol{\lambda})$ and $\mathbf{b}_0 = -\mathbf{A}^t \boldsymbol{\beta}^{(t)}$.
- (ii) Update $\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \mathbf{b}$.
- (iii) Set $t = t + 1$.

Step 2: As long as maximum no. of iterations s_{\max} or convergence is not reached:

- (i) Use current value $\boldsymbol{\beta}^{(t)} = \left(\theta_0^{(t)}, \left(\mathbf{u}^{(t)} \right)^t \right)$ and update to $\boldsymbol{\lambda}^{(s+1)}$ element-wise according to equation (14):

$$\hat{\lambda}_l^{(s+1)} = \frac{\text{tr} \left\{ \left[\widetilde{\mathbf{F}}^{-1} \left(\mathbf{u}^{(t)}, \boldsymbol{\lambda}^{(s)} \right) \widetilde{\mathbf{F}} \left(\mathbf{u}^{(t)}, 0 \right) \right]_l \right\}}{\left(\mathbf{u}_l^{(t)} \right)^t \mathbf{u}_l^{(t)}}, \quad \text{for } l = 1, \dots, p.$$

- (ii) Set $s = s + 1$.
- (iii) Set $\boldsymbol{\beta}^{(0)} = \boldsymbol{\beta}^{(t)}$ and $t = 0$ and start again with *Step 1*.

Additional Steps:

When during fitting one of the penalty parameter λ_l tends to infinity (in *Step 2* (i)) we set the corresponding smooth effect $m_l(\cdot)$ to zero, i.e. $\widehat{m}_l(\cdot) \equiv 0$, and estimate the remaining components in the model with the above procedure.

If in a later iteration than the first iteration solving *Step 1* (i) fails (e.g., due to numeric problems), we take the current values $\boldsymbol{\beta}^{(t)}$ and check whether all of the estimated effects $\widehat{m}_l(\cdot)^{(t)}$, $l = 1, \dots, p$ exceed a specified threshold in absolute value.

If not, i.e. at least one effect estimate is close to zero, we set the smallest smooth effect to zero and continue.

3.3 Combining the Sample Estimates

Let us now bear in mind that the estimation described in the previous subsection holds for one sample of the network. For each sample we obtain an estimate $\widehat{m}_{l< k >}(x_l) = \mathbf{B}(x_l) \widehat{\mathbf{u}}_{l< k >}$ with obvious definition for $\widehat{\mathbf{u}}_{l< k >}$. We now need to combine the sample estimates which are mutually dependent. One possible approach for combining the sample estimates would be to calculate a mean curve by just averaging the resulting parameter estimates over all samples. We follow a different path here originating in functional data analysis and compute a median curve. The median curve is more robust against outliers than the mean curve. We employ the methods developed by Sun et al. (2012) which are available in the R package `fda` (Ramsay et al., 2014). There is a huge number of options available for computing functional depth, ranking curves accordingly, and determining a

median curve (see, e.g., López-Pintado and Romo, 2009, and Mosler and Polyakova, 2012). We decided to use the approach of Sun et al. (2012) because it is quite fast even for a large number of curves, which seems important in our case (we use the option "Both" from the `fbplot` function `fda`, which first takes two curves for determining a band, and then computes a modified band depth in order to break ties between curves). We compute a joint median curve by sticking all estimated effects together. Computing marginal median curves per effect would be possible as well.

4 Data Example

4.1 Linear Estimation through Subsampling

As data example we use the combined data from ten Facebook ego networks, which has originally been collected by McAuley and Leskovec (2012) and is available from the Stanford Large Network Dataset Collection (Leskovec and Krevl, 2014). Figure 4 shows a plot of the network graph. It is undirected and contains 88,234 edges (Facebook friendships) between 4,039 nodes (actors). This amounts to a network density of roughly 0.01.

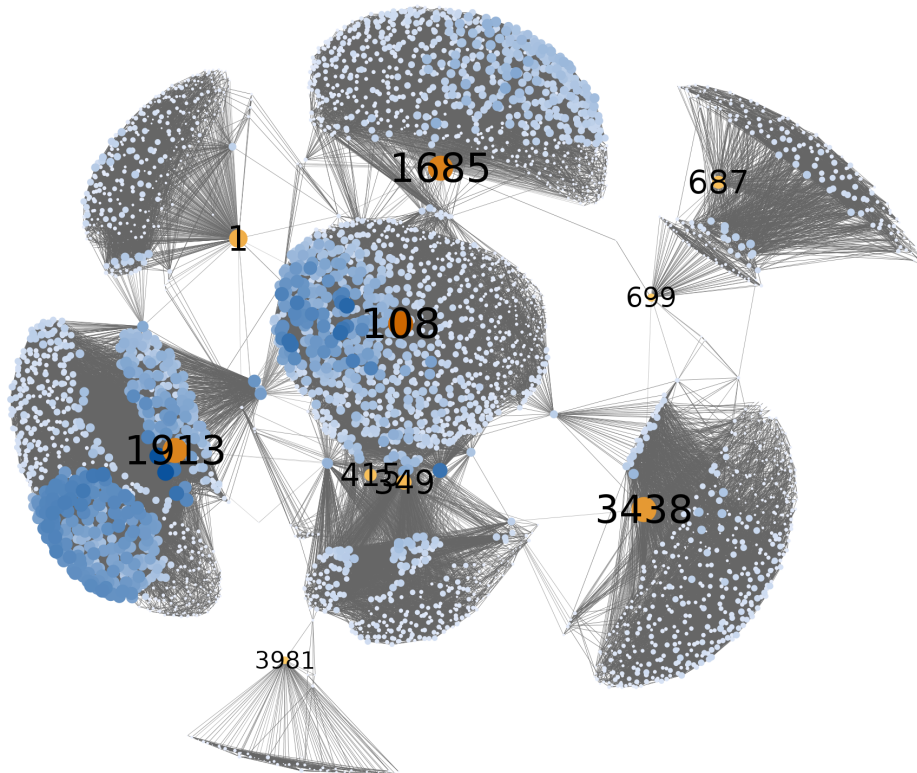


Figure 4 Visualisation of the combined Facebook data. Colouring and size represent nodal degree (darker and bigger corresponds to higher degree). Darkness and thickness of links represents the no. of triangles, the link belongs to. The ten egos are highlighted with a label indicating the node number, and coloured in orange instead of blue. Generated using stress minimization layout in visone (Brandes and Wagner, 2004).

We use data from the first 4,038 rows and columns of the network adjacency matrix² and obtain 4,037 sample subsets $\mathcal{D}_k(n|2)$. To each of these subsets we fit a standard logistic model with edges (as intercept), two-star, and triangle effect. We exclude subsets from the analysis which contain less than three observations with $y_{ij} = 3$ (this affects 56 data subsets). Figure 5 shows pairwise scatterplots of the estimated coefficients. Extreme results with an estimated intercept $\hat{\theta}_{\text{edges}} < -10$ (115 estimates) are excluded. The general impression for the shown results is that the estimated triangle effect $\hat{\theta}_{\text{triangles}}$ is always positive. The estimated two-star effect $\hat{\theta}_{\text{two-stars}}$ is closer to zero with some positive and some negative values. There is some negative correlation between the two parameters. Table 1 displays a numerical summary of the results.

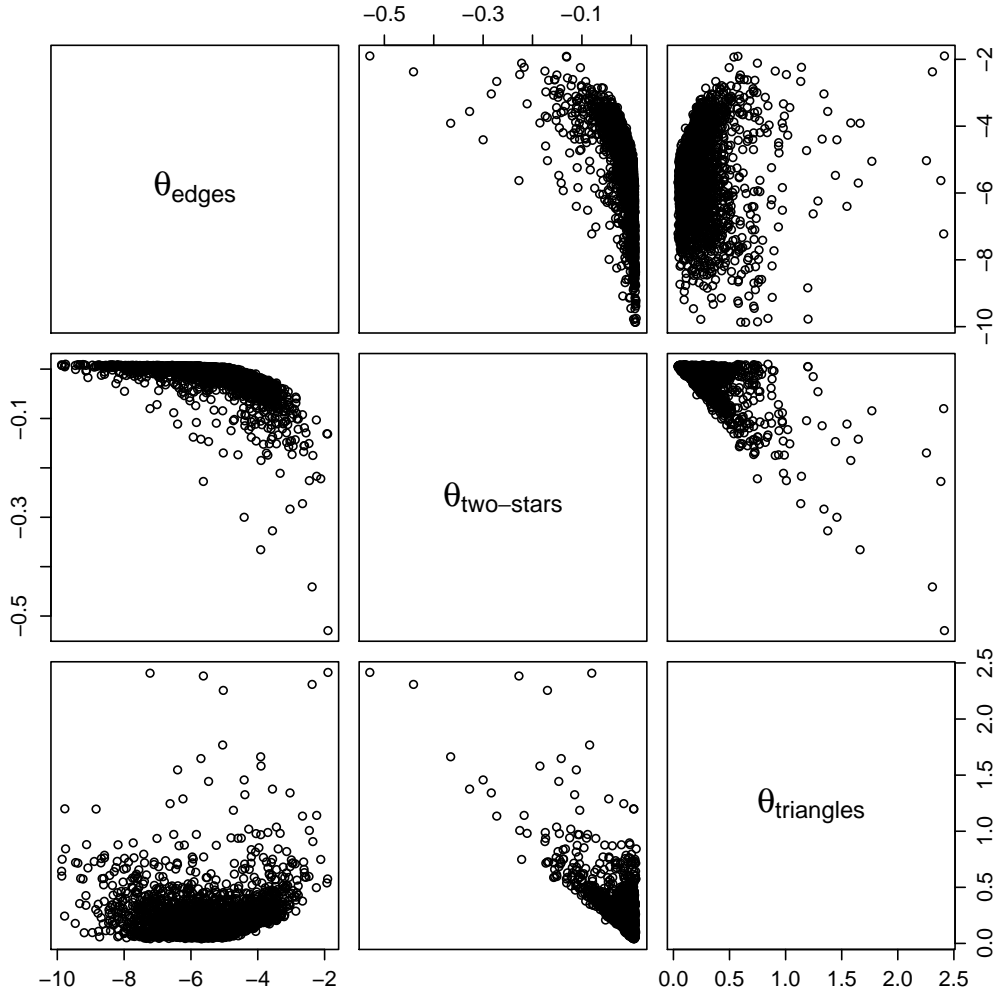


Figure 5 GLM (edges, two-star, and triangle effect) results for the Facebook data. Extreme estimates with an estimated intercept $\hat{\theta}_{\text{edges}} < -10$ (115 estimates) are not shown.

Apparently, in a network of this size we are faced with degeneracy using a two-stars and triangles as model statistics. We therefore do not put too much emphasis in the analysis of the parametric model but go forward to a non-parametric approach in the next subsection.

²As the number of nodes in the network has to be even for construction of the Latin square.

Table 1 GLM (edges, two-star, and triangle effect) results for the Facebook data. Extreme estimates with an estimated intercept $\hat{\theta}_{edges} < -10$ (115 estimates) are not considered.

Parameter	mean est.	median est.	5% quantile	95% quantile
θ_{edges}	-5.436	-5.425	-7.373	-3.687
$\theta_{two-stars}$	-0.012	-0.003	-0.054	0.006
$\theta_{triangles}$	0.207	0.174	0.063	0.483

4.2 Non-parametric Estimation through Subsampling

We stick to the Facebook data example and continue our analysis with a non-parametric Exponential Random Graph Model as described in Section 3. We fit a model containing the edge effect (as intercept), a smooth two-star effect, and a smooth triangle effect.

We exclude subsets from the analysis which contain less than 10 observations with $y_{ij} = 1$, because otherwise we do not have enough information for a stable estimation of the smooth effects. This affects 577 out of 4,037 data subsets. We use 20 exponential distributions as basis functions for each smooth component, with parameters γ_q ranging between values of 0.0005 and 1 as displayed in Figure 3. The maximum number of iterations is 20, which is not reached for any of the fits. The convergence criterion is set to $1e-12$ and for 83 samples the algorithm is aborted, which, e.g., may be caused by separability in these subsets and resulting in non-identifiability. As described in the algorithm in the previous section, the fitted model can simplify if the fitted λ_l goes to infinity. In this case the corresponding functional fit $\hat{m}_l(\cdot)$ equals zero and the model is reduced. This implies that the fitting algorithm itself conducts a model selection. In addition, effects can be set to zero if numerical issues occur when solving the quadratic problem in *Step 1* of the algorithm and the current effect estimate $\hat{m}_l(\cdot)$ is close to zero (we use a value of 0.005 for this criterion here). We therefore record the number of samples where the algorithm converges to a simplified model. Let the different models be labelled as follows:

$$\begin{aligned}
 M_1 &: \text{“two-star”} + \text{“triangle”} \\
 M_2 &: \text{“triangle”} \\
 M_3 &: \text{“two-star”} \\
 M_4 &: \text{intercept only}
 \end{aligned}$$

The notation means that model M_2 , for instance, corresponds to a model where the non-parametric smooth two-star effect is set to zero, while for model M_4 both, two-star and triangle effect are set to zero. Table 2 summarises the results numerically and shows the number of samples for the converged models. There is a clear dominance for model M_2 with intercept and smooth non-parametric triangle effect only.

Figure 6 shows the resulting 3,377 estimates (subsets with convergence, or effect set to zero), containing mean (solid blue lines) and median estimates (dashed orange lines). As general impression we obtain a negative intercept, a positive triangle effect, and a two-star

Table 2 Numeric summary of the model fitting results for the Facebook data. The non-parametric ERGM contains edges, a smooth two-star, and a smooth triangle effect.

Total no. of samples available:	4,037
No. of samples with no fit (less than 10 times $y = 1$ in sample):	577
Model M_1 :	181
Model M_2 :	3,189
Model M_3 :	5
Model M_4 :	2
No. of samples where max. no. iterations was reached:	0
No. of samples with other reason for non-convergence:	83

effect which is set to zero for most samples and fluctuates around zero in the remaining cases. The median curve for the two-star effect is exactly zero.

To explore the validity of the model, we continue our analysis by computing Pearson residuals for all observations

$$e_{ij} = \frac{y_{ij} - \hat{\pi}_{ij}}{\sqrt{\hat{\pi}_{ij}(1 - \hat{\pi}_{ij})}}, \quad \text{for } i = 1, \dots, n, \quad j = 1, \dots, n, \quad i \neq j,$$

where $\hat{\pi}_{ij}$ is a prediction based on the obtained median (curve). As next step we calculate the average Pearson residual for each node through

$$\tilde{e}_i = \frac{1}{n-1} \sum_{j \neq i} e_{ij}, \quad \text{for } i = 1, \dots, n.$$

Figure 7 shows the resulting node-specific average Pearson residuals. Bear in mind that the residuals are not independent, as we are averaging over non-independent samples. They should still have an expected value of zero. Figure 7 shows a clear structure. The nodes with large average residuals \tilde{e}_i are not surprising, as these are the ten egos from the network construction. In the upper plot all ten egos are depicted with a red star. They have more connections as one would expect from the model and therefore stick out. Moreover, some nodes have rather negative Pearson residuals and these nodes can be attributed to specific ego-nets. The lower plot in Figure 7 highlights two ego-nets (for nodes 108 and 1913, these two egos are again depicted with a red star, the residuals belonging to the ego-nets are black, the remaining ones grey) and they account for almost all of these negative residuals. Our conclusion from this residual analysis is that for some parts of the network the overall model seems too simplistic, while for others the outcome appears to be reasonable.

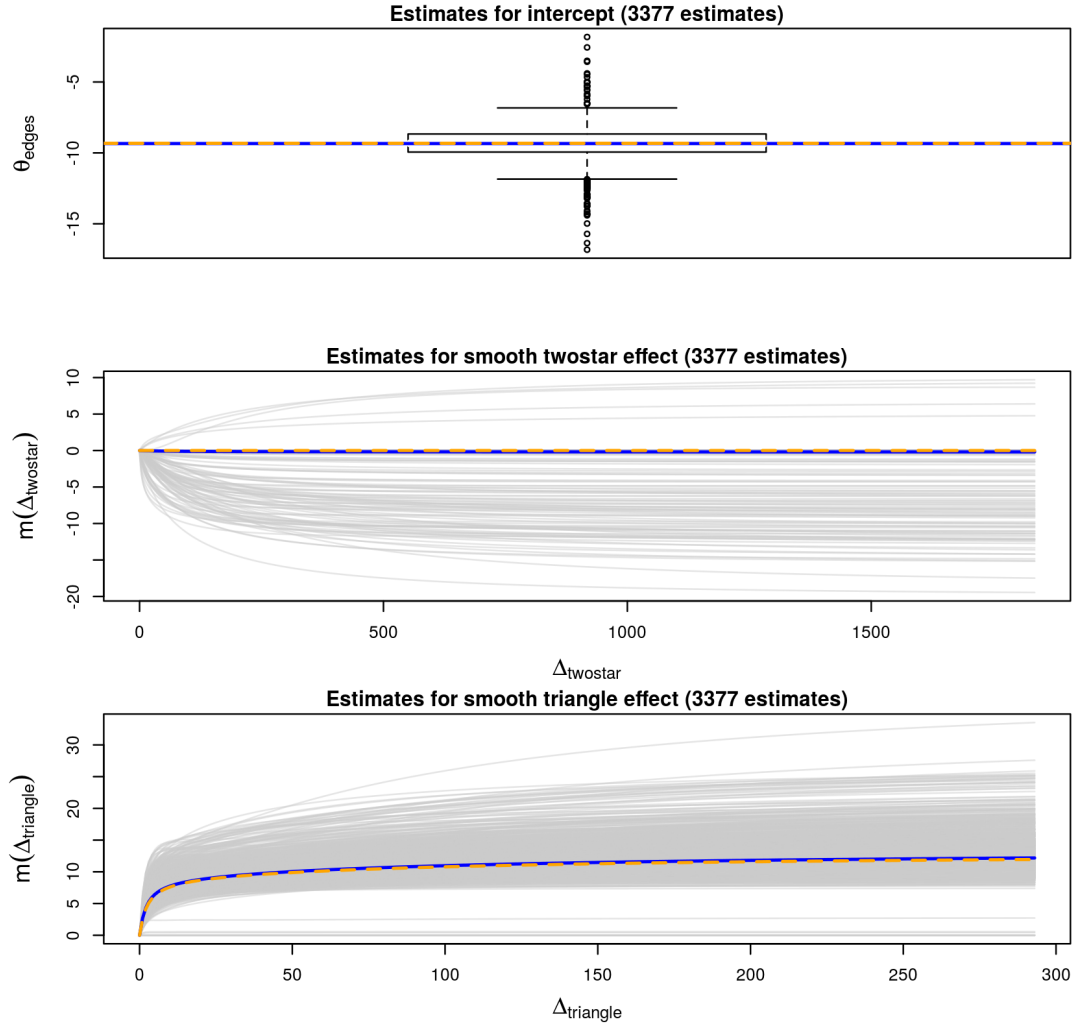


Figure 6 Non-parametric ERGM (edges, smooth two-star, and smooth triangle effect) results for the Facebook data. 3,377 estimates with convergence or effects set to zero are shown. The blue solid lines show the mean estimate; the orange dashed lines depict the median estimate.

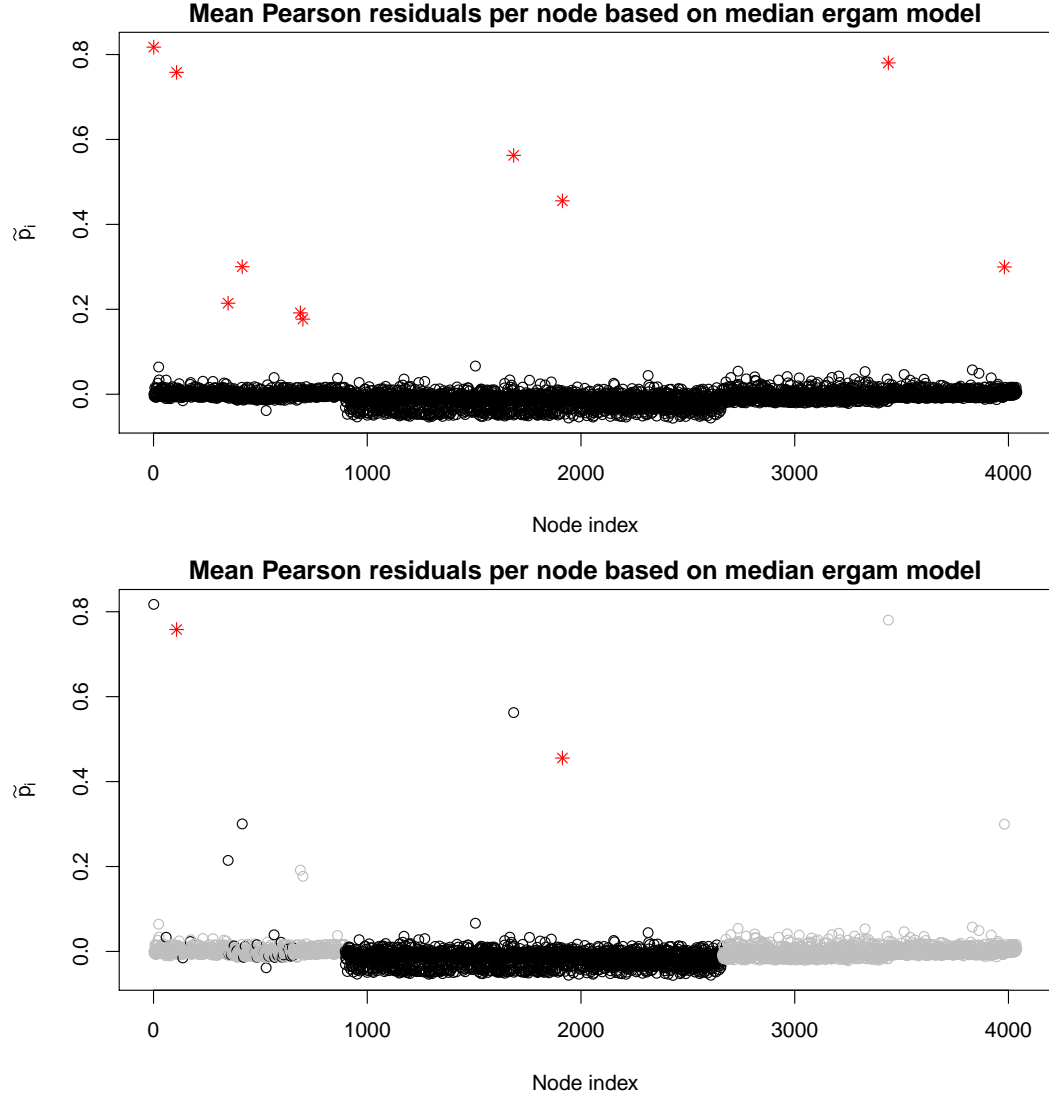


Figure 7 Node-specific average Pearson residuals from non-parametric ERGM for the Facebook data. Prediction for the residuals is based on the overall median model. The ten egos in the network are denoted with a red star in the upper plot. The lower plot highlights two ego-nets (for nodes 108 and 1913). The two egos are denoted with a red star, the corresponding members of their ego-networks are black, the remaining ones of the whole dataset grey.

We continue our analysis and look at the ego-nets of node 108 (contains a majority of nodes with negative average Pearson residual; consists of 1,045 nodes with 26,750 edges), and of node 1,685 (consists of 792 nodes with 14,025 edges) separately. The egos themselves are not part of the subnetworks as they are connected to every other vertex in the corresponding subnetwork (by construction). The setup for the fit is the same as before for the non-parametric ERGM. Figures 8, and 9 show the corresponding estimates. Table 3 summarises the results. When comparing the results to the ones for the whole dataset, the overall impression is similar, with a positive triangle effect, and a two-star effect close to zero (or set to zero for most samples, and a zero median curve). The intercept values are smaller in absolute value, which is not surprising as we are analysing smaller networks (with a higher density).

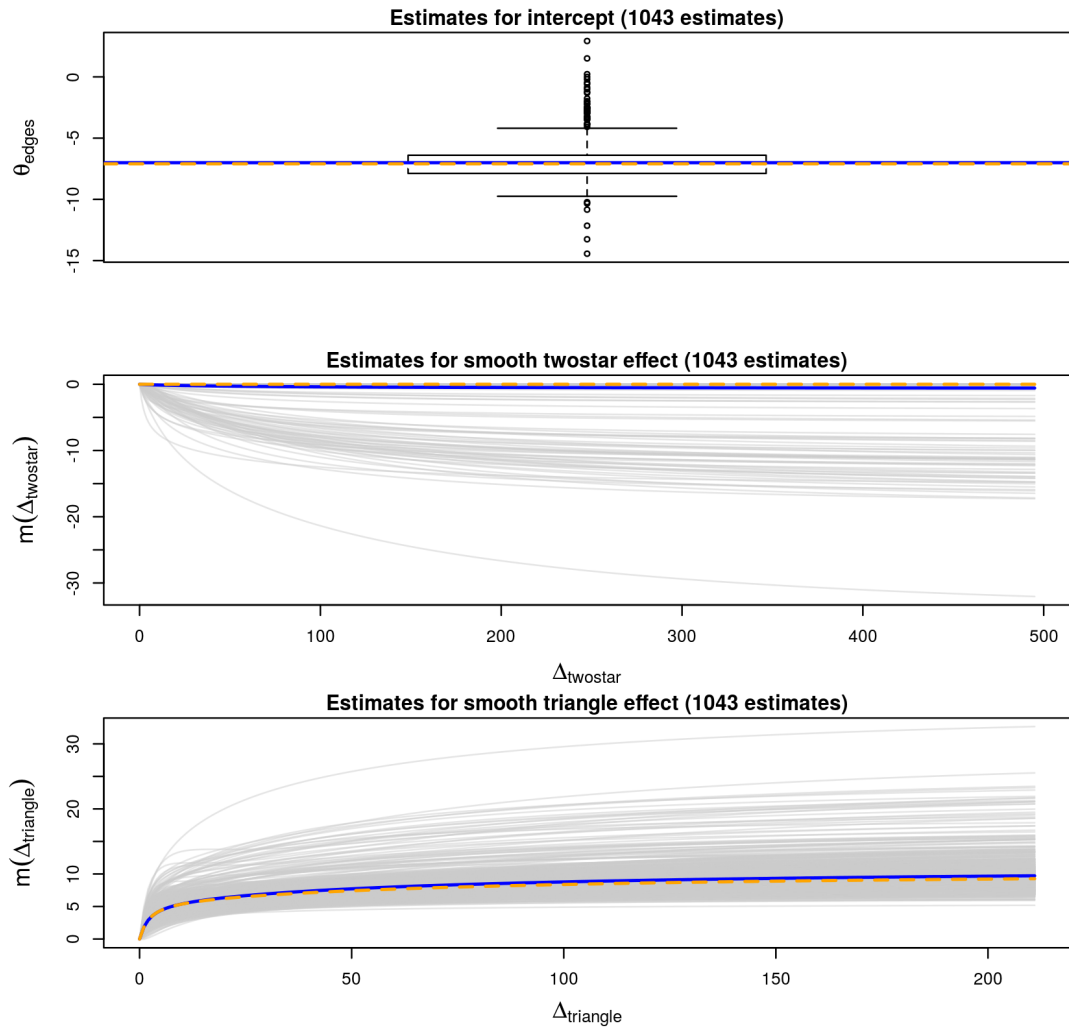


Figure 8 Non-parametric ERGM (edges, smooth two-star, and smooth triangle effect) results for ego-net of node 108 from the Facebook data. 1,043 estimates with convergence or effects set to zero are shown. The blue solid lines show the mean estimate; the orange dashed lines depict the median estimate.

Figure 10 shows the resulting nodes-specific average Pearson residuals for both ego-nets.

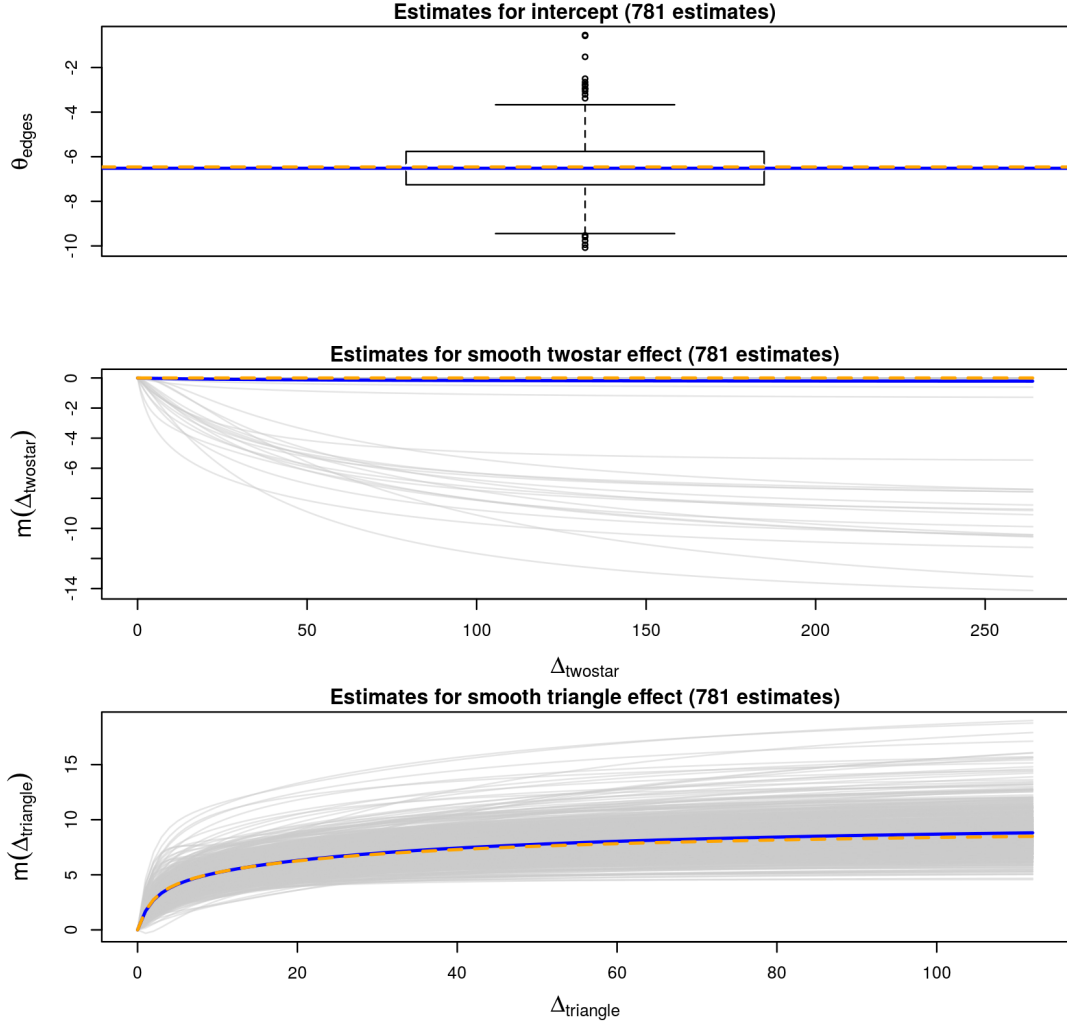


Figure 9 Non-parametric ERGM (edges, smooth two-star, and smooth triangle effect) results for ego-net of node 1,685 from the Facebook data. 781 estimates with convergence, or effects set to zero are shown. The blue solid lines show the mean estimate; the orange dashed lines depict the median estimate.

The result looks more homogeneous than before, but for the ego-net of 108 we still see that there are nodes in the network with a rather negative average Pearson residual, i.e. they have fewer connections than the model would predict. This might be solved by extending the modelling approach and include node-specific or dyadic covariates into the model. This is of course easily possible in combination with the smooth effects but lies beyond the scope of this paper.

Table 3 Numeric summary of the model fitting results for ego-subnets of the Facebook data. The non-parametric ERGM contains edges, a smooth two-star, and a smooth triangle effect.

Ego-net	108	1,685
Total no. of samples available:	1,043	791
No. of samples with no fit (less than 10 times $y = 1$ in sample):	132	65
Model M_1 :	0	10
Model M_2 :	911	716
Model M_3 :	0	0
Model M_4 :	0	0
No. of samples where max. no. iterations was reached:	0	0
No. of samples with other reason for non-convergence:	0	0

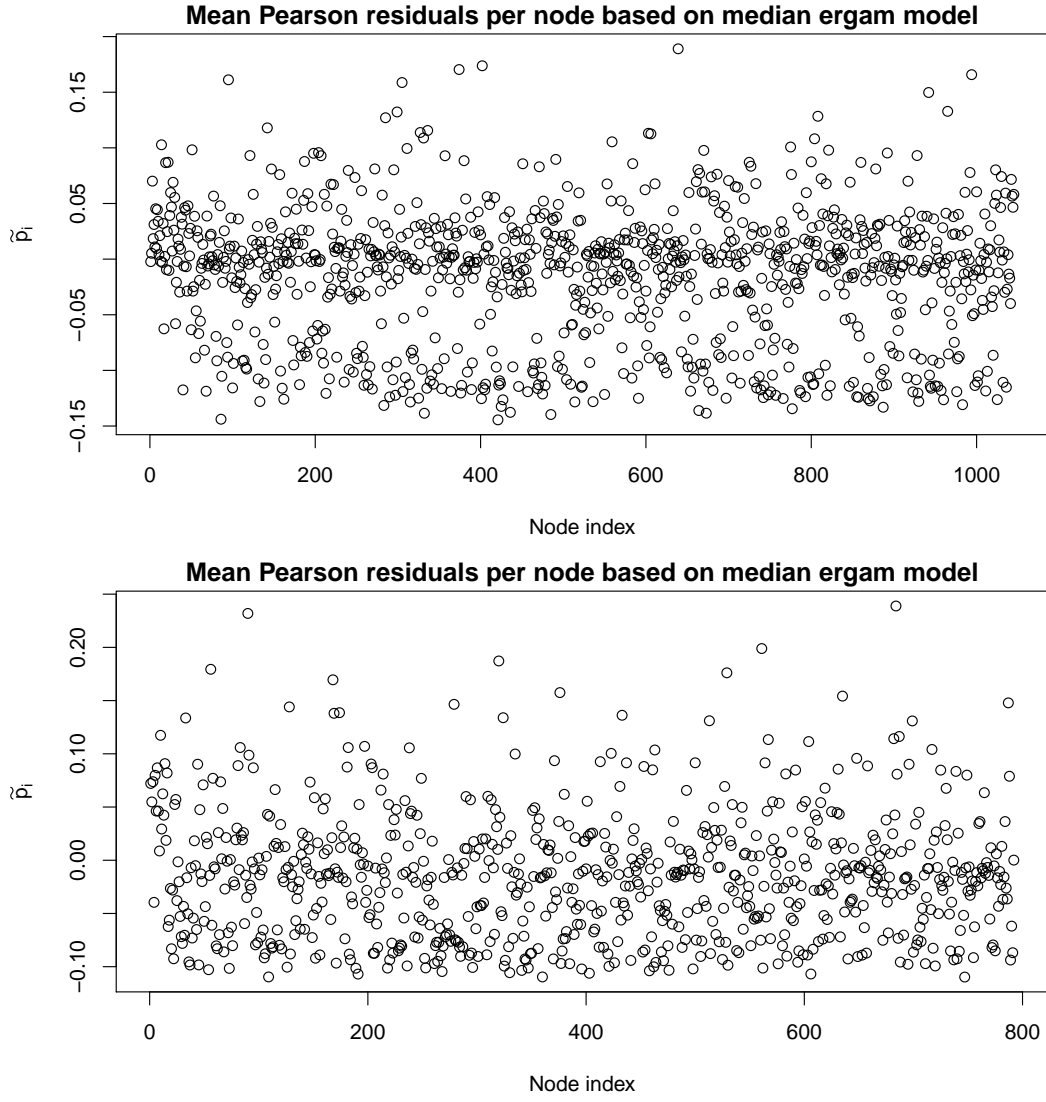


Figure 10 Node-specific average Pearson residuals from non-parametric ERGM for the ego-nets of node 108 (upper plot) and 1,685 (lower plot) from the Facebook data.

5 Discussion

We have shown that it is possible to make use of the Markov independence assumption in the context of Exponential Random Graph Models to obtain samples consisting of independent observations which allow to use standard generalized linear models (GLM) for model fitting. Extending this approach to generalized additive models (GAM) by adding smooth functional components in a non-parametric fashion enables us to gain flexibility while maintaining the simple interpretability of statistics like two-stars and triangles. It circumvents the construction or use of more complex statistics like, e.g., geometrically weighted degree or edgewise shared partners, which also stabilise the model fitting but are very difficult to interpret. In addition, the whole estimation procedure is quite fast (much faster than the standard MCMC based routines available for ERGMs) as we are using well established model fitting routines for GLMs and GAMs, and it can easily be run in parallel as the individual sample fits can be computed independently of each other. The computation for the Facebook data example was run in parallel on 20 cores (with 2.60 GHz) and took less than four minutes (including all data pre-processing and storing the results on disk).

To employ the described models the network needs to be big enough (otherwise the resulting samples are too small), and what can be more problematic, the network has to be dense enough as otherwise we obtain samples consisting only of observations with $y_{ij} = 0$. The latter is a general problem in real-world networks, as it is well-known that with increasing network size n the density tends to become smaller and smaller. The proposed modelling strategy therefore clearly has some caveats. Also, it is difficult to give a general advice on how many actors are needed for our method to work. When using the GLM approach on each subsample of size $\frac{n}{2}$ a smaller number of observations is reasonable than for the non-parametric GAM approach. In the data example we have presented in the previous section, the sample size itself is not an issue with 2,019 observations per subsample for the whole network, and 396 or 522, respectively, for the ego-nets, where we fit models with two smooth functional components plus an intercept term. Still, we had some problems with obtaining samples with enough $y_{ij} = 1$ observations per sample. Of course this issue becomes more severe when the network density (which is 0.01 for the complete Facebook data, and therefore quite high for a network of this size) goes down. Another problem which is apparent from the residual analysis in Figures 7, and 10 is that the residuals are quite low in absolute value. This is a sign of underdispersion in the underlying binomial models and can be explained by zero-inflation, i.e. we have more zeros in the data than we would expect under the model. This result is not surprising, again due to the low density in large networks, where Exponential Random Graph Models tend to be problematic in general. There are approaches going into the direction of assuming local dependence structures, whereas the standard ERGM assumes a global rather strict dependence structure and is therefore probably unrealistic especially in the context of large networks. Schweinberger and Handcock (2015) use hierarchical ERGMs, see also the corresponding R package `hergm` (Schweinberger et al., 2015), where the neighbourhood structure can be taken into account if it is known, or estimated as a latent construct using a Bayesian approach. The latter is computationally very problematic and rather

time consuming or even infeasible for large networks. Another possible solution to handle the zero-inflation using our subsampling approach would be the use of mixture models as available, e.g., in the R package `flexmix` (Leisch, 2004), and employ zero-inflation models (Grün and Leisch, 2008, Section 5.1) for binomial data to each sample. To us this appears to be a promising field for future research.

Acknowledgements

We gratefully acknowledge the help of Ulrik Brandes with the visualisation of the Facebook data example in Figure 4 using `visone` (version 2.16).

References

- Andersen, L. D. and Hilton, A. J. W. (1980). Generalized latin rectangles I: Construction and decomposition. *Discrete Mathematics*, 31:125–152.
- Bogomolny, A. (2016). Latin squares. Simple construction. From Interactive Mathematics Miscellany and Puzzles <http://www.cut-the-knot.org/arithmetic/latin2.shtml>. [Online; accessed: 04 March 2016].
- Brandes, U. and Wagner, D. (2004). visone – analysis and visualization of social networks. In Jünger and Mutzel (2004), pages 321–340.
- Breslow, N. E. and Clayton, D. (1993). Approximate inference in generalized linear models. *Journal of the American Statistical Association*, 88(421):9–25.
- Caimo, A. and Friel, N. (2011). Bayesian inference for exponential random graph models. *Social Networks*, 33(1):41–55.
- Chatterjee, S. and Diaconis, P. (2013). Estimating and understanding exponential random graph models. *The Annals of Statistics*, 41(5):2428–2461.
- Eilers, P. H. C. and Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–121.
- Fienberg, S. E. (2012). A Brief History of Statistical Models for Network Analysis and Open Challenges. *Journal of Computational and Graphical Statistics*, 21(4):825–839.
- Frank, O. and Strauss, D. (1986). Markov graphs. *Journal of the American Statistical Association*, 81(395):832–842.
- Goldenberg, A., Zheng, A. X., Fienberg, S. E., and Airoldi, E. M. (2010). A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233.
- Grün, B. and Leisch, F. (2008). Flexmix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, 28(1):1–35.
- Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. Chapman & Hall, London.
- Hummel, R. M., Hunter, D. R., and Handcock, M. S. (2012). Improving simulation-based algorithms for fitting ERGMs. *Journal of Computational and Graphical Statistics*, 21(4):920–939.
- Hunter, D. R. (2007). Curved exponential family models for social networks. *Social Networks*, 29(2):216–230.
- Hunter, D. R. and Handcock, M. S. (2006). Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*, 15(3):565–583.

- Hunter, D. R., Krivitsky, P. N., and Schweinberger, M. (2012). Computational statistical methods for social network analysis. *Journal of Computational and Graphical Statistics*, 21(4):856–882.
- Jünger, M. and Mutzel, P., editors (2004). *Graph Drawing Software*. Springer Berlin Heidelberg.
- Kauermann, G. (2005). Penalized spline smoothing in multivariable survival models with varying coefficients. *Computational Statistics & Data Analysis*, 49(1):169–186.
- Kauermann, G., Krivobokova, T., and Fahrmeir, L. (2009). Some asymptotic results on generalized penalized spline smoothing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):487–503.
- Kolaczyk, E. D. (2009). *Statistical Analysis of Network Models*. Springer, New York.
- Koskinen, J. and Daraganova, G. (2013). Dependence graphs and sufficient statistics. In Lusher et al. (2013), pages 77–90.
- Krivobokova, T. and Kauermann, G. (2007). A note on penalized spline smoothing with correlated errors. *Journal of the American Statistical Association*, 102(480):1328–1337.
- Leisch, F. (2004). Flexmix: A general framework for finite mixture models and latent class regression in r. *Journal of Statistical Software*, 11(1):1–18.
- Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>.
- López-Pintado, S. and Romo, J. (2009). On the concept of depth for functional data. *Journal of the American Statistical Association*, 104(486):718–734.
- Lusher, D., Koskinen, J., and Robins, G. (2013). *Exponential Random Graph Models for Social Networks*. Cambridge University Press, Cambridge.
- McAuley, J. J. and Leskovec, J. (2012). Learning to discover social circles in ego networks. In *Neural Information Processing Systems*.
- McCulloch, C. E., Searle, S. R., and Neuhaus, J. M. (2008). *Generalized, Linear, and Mixed Models*. Wiley Series in Probability and Statistics. John Wiley & Sons, 2nd edition.
- Mosler, K. and Polyakova, Y. (2012). General notions of depth for functional data. *ArXiv e-prints*.
- Murray, I., Ghahramani, Z., and MacKay, D. (2006). MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Arlington, Virginia. AUAI Press.
- O’Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems. *Statistical Science*, 1(4):502–518.

- R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Ramsay, J. O., Wickham, H., Graves, S., and Hooker, G. (2014). *fda: Functional Data Analysis*. R package version 2.4.4.
- Ruppert, D., Wand, M. P., and Carroll, R. J. (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge.
- Ruppert, D., Wand, M. P., and Carroll, R. J. (2009). Semiparametric regression during 2003–2007. *Electronic Journal of Statistics*, 3:1193–1256.
- Schall, R. (1991). Estimation in generalized linear models with random effects. *Biometrika*, 78(4):719–727.
- Schweinberger, M. (2011). Instability, Sensitivity, and Degeneracy of Discrete Exponential Families. *Journal of the American Statistical Association*, 106(496):1361—1370.
- Schweinberger, M., Handcock, M., and Luna, P. (2015). *hergm: Hierarchical Exponential-Family Random Graph Models with Local Dependence*. R package version 2.2-2.
- Schweinberger, M. and Handcock, M. S. (2015). Local dependence in random graph models: characterization, properties and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(3):647–676.
- Snijders, T. A. B. (2002). Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2):1–40.
- Snijders, T. A. B., Pattison, P. E., Robins, G. L., and Handcock, M. S. (2006). New specifications for exponential random graph models. *Sociological Methodology*, 36(1):99–153.
- Sun, Y., Genton, M. G., and Nychka, D. W. (2012). Exact fast computation of band depth for large functional datasets: How quickly can one million curves be ranked? *Stat*, 1(1):68–74.
- Turlach, B. A. and Weingessel, A. (2013). *quadprog: Functions to solve Quadratic Programming Problems*. R package version 1.5-5.
- Whittaker, J. (2009). *Graphical Models in Applied Multivariate Statistics*. Wiley.
- Wood, S. (2006). *Generalized Additive Models*. Chapman & Hall, London.