

Ludwig-Maximilians University Munich
Department of Statistics

Modeling of dynamic networks considering heterogeneous actors

Master's Thesis
to obtain the academic degree
Master of Science

Author:

Thomas Eder

Submission Date: March 17, 2022

Supervisor: Sevag Kevork

Supervisor: Prof. Dr. Göran Kauermann

Abstract

One of the most widely used network model, the Exponential Random Graph Model (ERGM), is based on the assumption of homogeneous network nodes. This assumption is not very realistic, and a big concern with respect to both model validity and estimation instability. Kevork and Kauermann proposed a way to account for the presence of unobserved node-specific heterogeneity by including node-specific random effects in the ERGM. The estimation is then achieved by iterating between approximate pseudolikelihood estimation for the random effects, and maximum likelihood estimation for the ERGM parameters. The goal of this work is to extend their approach to dynamic networks, using the Temporal ERGM (TERGM).

Contents

1	Introduction	1
2	Network Analysis	2
3	Network Modeling	3
3.1	Exponential Random Graph Models	6
3.2	Temporal Exponential Random Graph Models	9
3.2.1	Markov Chains	9
3.2.2	TERGM Statistics	11
3.2.3	Markov Chain Monte Carlo Estimation	13
3.3	Alternative Models	14
3.3.1	Seperated TERGM (STERGM)	14
3.3.2	Longitudinal ERGM	17
3.3.3	Stochastic Actor Oriented Models (SAOM)	17
4	Heterogeneity	19
4.1	Definition	19
4.2	Modeling Random Effects	21
4.2.1	Linear Mixed Models	21
4.2.2	Logistic Regression	22
4.2.3	Generalized Linear Mixed Models	25
4.3	Nodal Random Effects in Networks	26
4.4	Iterative Estimation using TERGM	29
4.4.1	Initial Values	29
4.4.2	Estimate TERGM	29
4.4.3	Estimate Random Effects	29

5	TERGM Implementations in R	30
5.1	Change Statistics	30
5.2	TERGM Package in statnet	31
5.2.1	CM(P)LE	31
5.2.2	EGMME	33
5.3	TERGM Package btergm	33
5.3.1	Bootstrapped Pseudo-Likelihood	34
5.3.2	MCMC-MLE	35
5.3.3	Bayesian Estimation	35
5.4	GLMM Package mgcv	36
6	Evaluation	37
6.1	Samples of Heterogeneous Networks	37
6.2	Parameter Estimation	39
6.3	Comparison with traditional Approach	39
6.4	Results	40
7	Conclusion	52
A	List of Figures	54
B	List of Tables	56
C	Bibliography	57

1 Introduction

At first glance, network science may seem like a niche within the statistical domain. But in a world which becomes more and more connected and globalized, the structures of the emerging new networks become ubiquitous, and so does their relevance. This makes network science an interdisciplinary field in high demand. In the last two decades, more than 50,000 scientists published around 30,000 papers in this field (cf. [Molontay and Nagy, 2019]). Of course, not every one of them is a game changer, but the amount of citations of [Granovetter, 1973] ($> 64,000$) or [Watts and Strogatz, 1998] ($> 48,000$) speaks volumes.

This work aims to contribute by expanding on [Kevork and Kauermann, 2021], which introduces a method to account for *heterogeneity* in the popular *Exponential Random Graph Models (ERGM)*, which assumes homogeneity of actors in networks. We will transfer this method to the TERGM, which is simply an ERGM with added temporal dimension.

The structure of this work is as follows: In chapters 2 and 3 we will establish a framework for modeling networks. In chapter 4 we will lay out a way to part with the homogeneity assumption. Chapter 5 will give a quick overview on available implementations of estimation procedures for network models. And in chapter 6 we will put our findings to the test by applying them on simulated networks.

2 Network Analysis

When it comes to networks, we find the same divide between descriptive and inferential statistics as in many other statistical domains. Although the graph-like nature of networks makes the descriptive approach dominated by visualizations rather than plain 'statistics' (metrics describing the data at hand with aggregated numbers). This natural inclination towards qualitative visuals, however, comes at the cost of more difficulties when building a framework for statistical modeling, which requires quantitative metrics.

If we consider a network as a set of nodes ('actors') and edges (connections/relationships) between a set¹ of two nodes (a 'dyad'), we can represent it as a *graph*. This approach can be used for directed, undirected, valued (weighted), unvalued, and bipartite networks. For simplicity, we will focus on undirected, unweighted networks. So we only distinguish *tie* (connection between nodes) and *no tie*. If there is a tie between two nodes, we call them *adjacent*.

The network (with n nodes) can then be represented by an $n \times n$ dimensional adjacency matrix \mathbf{Y} , where the matrix element $Y_{ij} = 1$ when there is a tie between the nodes i and j ($i, j \in \{1, \dots, n\}$ and $i \neq j$), and $Y_{ij} = 0$ otherwise. This matrix is symmetric. A realization of the random variable $\mathbf{Y} \in \mathcal{Y}$ is denoted as \mathbf{y} . As we can see from the definition of the adjacency matrix, we are modeling cases where the nodes are known and fixed, and the edges between them are the subject of interest. It is also possible to include covariates for both nodes and dyads to this model.

When the network is evolving over time, we represent this dynamic in the form of a discrete time-series, where we have one adjacency matrix \mathbf{Y}^t for each time step t .

¹or a 2-tuple, if we consider the relationships to have a direction

3 Network Modeling

Statisticians are usually less interested in *describing* network y in front of them, than making *inferences* on the process that created it, as this allows them to generalize their findings. Accordingly, they are interested in the parameters of the process itself, rather than the parameters of this particular network sample the process created. So, following the lead of every other statistical domain, we consider the given network y as realization of the stochastic variable Y representing all networks that could have been. For statistical inference, we now need to conceive a *statistical model* with parameter θ that assigns every considered probability distribution $\mathcal{P}_\theta(Y)$ over Y a unique value for θ , allowing us to estimate the value based on the given network y . [Kolaczyk and Csardi, 2020] identifies three main classes of such statistical network models.

The class of *Exponential Random Graph Models (ERGM)*, which we will be using in this work, resembles standard regression models. The *Stochastic Block Models*, in their most basic form a mixture of classical random graph models. And *Latent Network Models*, using both observed and unobserved variables in modeling.

When we imagine a network y with N nodes, one of the simplest models we could come up with would be to consider the potential existence of each individual edge as (independent) Bernoulli trial with probability p . p would then be the parameter of our simple model giving us the probability distribution $\mathcal{P}_p(Y)$ over all possible networks Y with N nodes. With all dyads (a set of two nodes, regardless if there exists an edge between them) being assumed independent and therefore indistinguishable, all networks with the same number of edges will be assigned the same probability. The number of edges $s(Y)$ is therefore a *sufficient statistic*, as it contains all information about the network necessary for this model.

We can formalize it as

$$s(Y) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n Y_{ij}, \quad Y_{ij} = \begin{cases} 1, & \text{if edge between } i \text{ and } j \\ 0, & \text{otherwise.} \end{cases}$$

for undirected, unweighted edges. The model can then be explicitly written as

$$\mathcal{P}_p(Y = y) = \left(\frac{p}{1-p} \right)^{s(y)} (1-p)^{N(N-1)/2} \quad (3.1)$$

(cf. [Kauermann et al., 2019]) This type of random graph is often referred to as an “Erdős-Rényi” graph. As the simplest model of its kind, they often serve as “null” hypothesis for more elaborate models. This model is historically also referred to as p_0 . From here, the search for more sophisticated models forks into two directions. On the one side we find models referred to as p_1 and p_2 , while the other path leads us to the p^* model, better known as ERGM.

The first path (as succinctly summarized by [Thiemichen, 2016]) leads us to [Holland and Leinhardt, 1981]’s p_1 model, by adding node-specific covariates represented by z_{ij} :

$$\log \left\{ \frac{\mathcal{P}(Y_{ij} = 1)}{1 - \mathcal{P}(Y_{ij} = 1)} \right\} = \alpha_i + \alpha_j + z'_{ij}\beta$$

where α_i and α_j are fixed nodal effects. Going one step further, the p_2 model presented by [Duijn et al., 2004] treats the node-specific effects as latent, non-observable random effects ϕ :

$$\log \left\{ \frac{\mathcal{P}(Y_{ij} = 1|\phi)}{1 - \mathcal{P}(Y_{ij} = 1|\phi)} \right\} = \phi_i + \phi_j + z'_{ij}\beta \quad (3.2)$$

where

$$\phi = (\phi_1, \phi_2, \dots, \phi_n)' \sim \mathcal{N}(0, \sigma_\phi^2 \mathcal{I}_n)$$

with \mathcal{I}_n being the n dimensional unit matrix. Both models build upon the notion that the probability of an edge in the network depends on which nodes this tie would connect.

There is, however, as mentioned before, another direction one can go from the p_0 model. As [Kauermann et al., 2019] laid out, if we transform p into a new parameter $\theta \equiv \log(\frac{p}{1-p})$ and define a normalization factor $Z(\theta) \equiv (1-p)^{-N(N-1)/2}$, (3.1) can be rewritten as

$$\mathcal{P}_\theta(Y = y) = \exp\{\theta s(y)\} q^{N(N-1)/2} = \frac{\exp\{\theta s(y)\}}{Z(\theta)} \quad (3.3)$$

which fulfills the formal requirements for the multidimensional random variable Y to belong to an *exponential family*. This allows us to take advantage of the plethora of well-established statistical tools designed for classical *generalized linear models (GLM)*, including methods for construction, fitting and comparison of models. A natural extension of this model would be to make use of the GLMs freedom and add more dimensions to θ , allowing to add more statistics like $s(Y)$, resulting in what we call *Exponential Random Graph Models (ERGM)*.

A more exhaustive overview on the landscape of network models which trace back to the Erdős-Rényi-Gilbert graph model is presented in 3.1 from [Fritz et al., 2019], where *DyNAM* = *dynamical actor oriented model*, *REM* = *relational event model*.

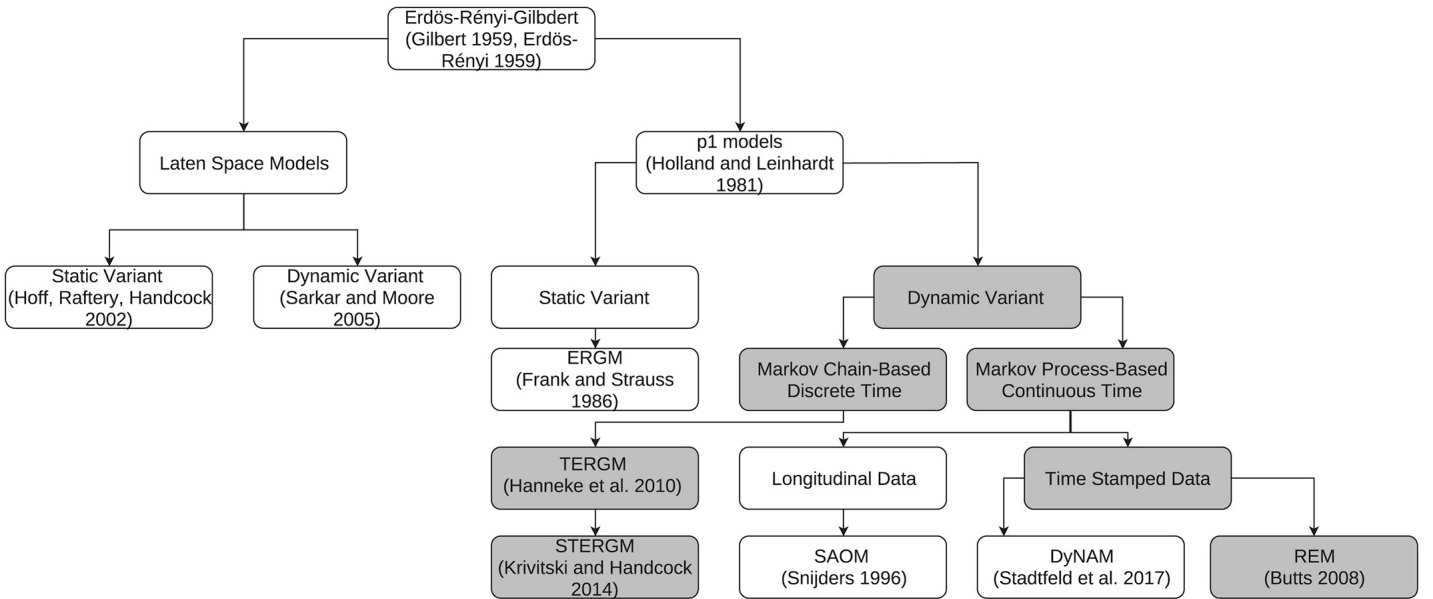


Figure 3.1: Dependencies between network models by [Goldenberg et al., 2010], expanded by [Fritz et al., 2019]

3.1 Exponential Random Graph Models

Exponential random graph models (ERGM) describe networks as multivariate observations, where relations in the network may depend on exogenous data (covariates) as well as endogenous processes. They “*describe parsimoniously the local selection forces that shape the global structure of a network*”, as [Hunter et al., 2008] put it. Although it might seem more appropriate to refer to these modeled *networks* as *snapshots of networks*, to emphasise they are not static, as we will put our focus on the dynamic changes of networks over time. Mathematically, we basically only need to add more dimensions to the parameter of the (3.3) representation of p_0 :

$$\mathcal{P}(Y = y|\boldsymbol{\theta}) = \frac{\exp\{\boldsymbol{\theta}'\boldsymbol{\Psi}(y)\}}{Z(\boldsymbol{\theta})}$$

If $\boldsymbol{\theta} \in \mathbb{R}^k$, we also need a vector of suitable model statistics $\boldsymbol{\Psi} : \mathcal{Y} \rightarrow \mathbb{R}^k$. This extension is one of the appeals of the ERGM. It is very flexible in capturing a wide range of sufficient statistics describing connectivity patterns in the networks. Hence, there finally is a way to account for probably the most important factor of all, the actual shape of the network itself, finally acknowledging that edges are *not* independent, which has been ignored by previous models. And it does so in a simple, ‘*plug and play*’ fashion. We just need to define sufficient statistics representing certain characteristics of networks (see [Handcock et al., 2014] for an extensive list of examples).

Usually, these characteristics follow a Markovian assumption, called a *Markovian neighborhood*, as described in detail by [van der Pol, 2019]. This assumption derives from the understanding, that the odds of a tie between nodes are mostly impacted by local, node-level dependencies, i.e. the relation with the nearest neighbors of the dyad. Like the number of shared partners. This not only appears like a reasonable assumption, but also avoids severe computational issues with statistics that span the whole network, like the *longest distance between nodes*. Local characteristics of dyads can be calculated independently and then summed up to get a sufficient network statistic.

Incorporating exogene covariates is just as straight forward. On top of that, ERGMs can be used to simulate networks.

But naturally, this models has its pet peeves as well. Although we can draw on most on the standard inferential infrastructure of GLMs, as pointed out by [Kolaczyk and Csardi, 2020], their asymptotic approximation to χ^2 distributions is not formally justified for ERGMs yet. Plus, the normalization constant $Z(\boldsymbol{\theta}) = \sum_{Y \in \mathcal{Y}} \exp\{\boldsymbol{\theta}'\boldsymbol{\Psi}(Y)\}$ may be intractable to compute. (A comparably small, undirected network of 10 nodes for example would be accompanied by $2^{\frac{1}{2}(n^2-n)} = 35 \cdot 10^{12}$ possible graphs.)

In addition to the normalization constant, the chosen statistics themselves may be impractical to compute. Many ERGM statistics are what we call *unstable*, meaning the statistic grows *faster than linearly* with increasing number of potential links, i.e. $\psi(y) > \mathcal{O}(N^2)$, N being the number of nodes in the network (cf. [Kauermann, 2020] and [Thiemichen, 2016]). The very intuitive (undirected) *two-star* (2 nodes connected through a third node) and *triangle* (3 nodes connected with each other) counts, for example, fall into this category. Thus, especially in large networks, linear statistics may have to be replaced by smooth functional components [Thiemichen and Kauermann, 2016]. This is not just a computational burden, though, as unstable network statistics are known to lead to *degenerated* networks (e.g. most of the probability distribution’s mass concentrated on the empty and/or complete network). This proves a severe obstacle in the estimation process. As the normalization constant is computationally intractable, Markov Chain Monte Carlo simulations are employed. When the model is not a good representation of the observed network, however, these simulations may not just be far enough off to affect the estimation process, but in the worst case, make the algorithm fail altogether. Either by not being able to converge at all, or by producing samples that would not have been produced under the Maximum Likelihood Estimator (cf [Handcock et al., 2021]). How hard hitting the space of non-degenerate parameter values can be is visualized in 3.2 by [Schweinberger, 2011]. The plots show the MCMC sample proportions of the respective parameters plottet against their corresponding natural parameters (in the third plot, the two-star parameter is fixed to 1). The shaded areas are supposed to indicate unstable regions, visualizing his conclusion that the two-star and triangle statistics are unstable for all values $\neq 0$. The discovery in the third plot, that the linear combination $\theta_{twostar} = -\theta_{triangle}/3$ leads to stable networks, is visualized in 3.3 by [Thiemichen, 2016], demonstrating how delicate non-degenerate settings of unstable parameters can be.

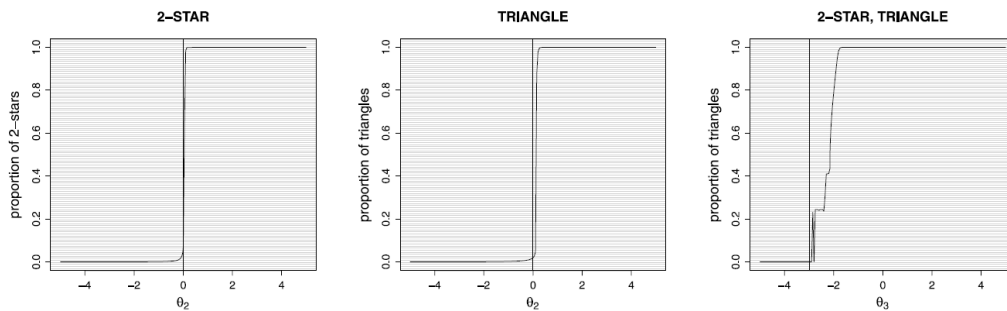


Figure 3.2: Effect of given parameters to resulting network structure

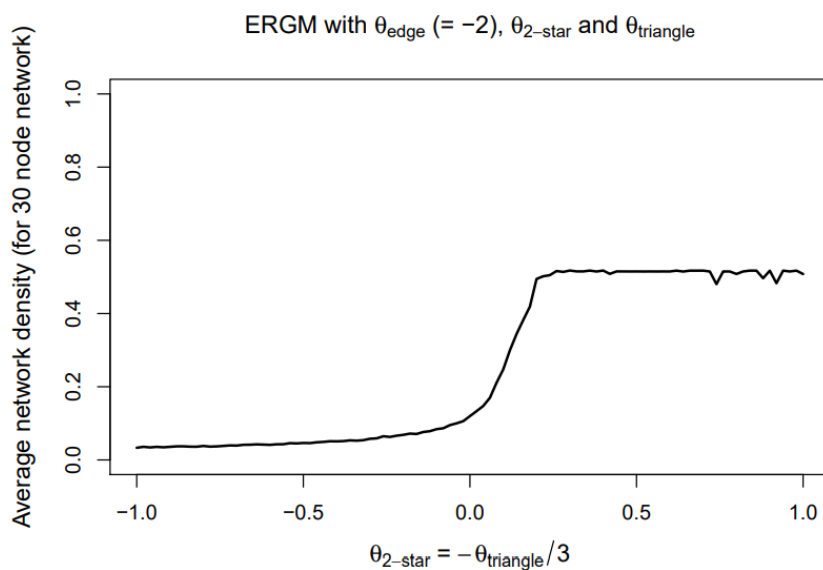


Figure 3.3: Effect of the given linear combination of resulting network structure

Lastly, these models do not account for (potential, but very realistic) nodal heterogeneity. Thus, the choice between the two paths (p_1, p_2 vs p^*) partly comes down to the question of modeling network structure vs modeling heterogeneity. To overcome this problem, [Kevork and Kauermann, 2021] and [Thiemichen, 2016] have proposed two different approaches for amending ERGMs. The work of latter one, building upon the Bayesian framework for ERGMs by [Caimo and Friel, 2011], derives an approximation of the Bayes factor for model selection. The most basic benefit being the ability to identify if the model generating processes contain structural effects (captured by a classic ERGM), nodal random effects (captured by the p_2 model),

or both. In this work, we will translate the work of [Kevork and Kauermann, 2021], focused on estimation of ERGMs taking into account heterogeneity, to an extension of the ERGM, which also has a *temporal* component, referred to as TERGM.

3.2 Temporal Exponential Random Graph Models

This chapter is based on [Hanneke et al., 2010]. Expanding the ERGM for multiple time steps, we now have to consider multiple snapshots of the same network over time, again represented by the respective weight matrices $Y^1, Y^2, Y^3, \dots, Y^t$. To account for the dependence of every state on the preceding ones, we will need a new set of statistics making use of this additional information.

Thus we define a new function $\Psi : \mathbb{R}_{n \times n} \times \mathbb{R}_{n \times n} \times \dots \times \mathbb{R}_{n \times n} \rightarrow \mathbb{R}^k$ representing a sufficient statistic on $Y^t | Y^{t-1}, \dots, Y^1$ allowing us to model each step t in our dynamic network series as an ERGM:

$$\mathcal{P}(Y^t = y^t | Y^{t-1} = y^{t-1}, Y^{t-2} = y^{t-2}, \dots, Y^1 = y^1, \theta) = \frac{\exp\{\theta' \Psi(y^t, y^{t-1}, \dots, y^1)\}}{Z(\theta, y^{t-1}, \dots, y^1)}$$

This model is referred to as *temporal* ERGM, in short TERGM.

3.2.1 Markov Chains

The temporal dimension, which is meant to capture the dynamic behaviour of networks is modeled as a discrete-time stochastic process. A stochastic process is essentially a sequence of random variables. It is said to be a *Markov Chain*, if fulfills the following requirement:

The *Markov Property* states, that for any discrete-time stochastic process \mathbf{Y} (with at least n time steps) having this property, any realization y^1, y^2, \dots, y^t of this process fulfills the following equation:

$$\mathcal{P}(Y^t = y^t | Y^1 = y^1, \dots, Y^{t-1} = y^{t-1}) = \mathcal{P}(Y^t = y^t | Y^{t-1} = y^{t-1})$$

for every $t \leq n$.

This means every time step Y^t , although depending on every previous time step y^1, \dots, y^{t-1} , is *conditionally* independent of every time step except the one right before y^{t-1} . In other words, every information about the history of the process we need to predict Y^t is contained in 'the present', y^{t-1} . What matters is *where we are* at $t-1$, not *how we got there*. This assumption is feasible for many real-world processes, and will simplify our model drastically. Generally, Y^t depends on t . In this work, we only consider *homogeneous* Markov chains, where Y^t does not depend on t . If needed, however, t can simply be included as covariate.

When we now break the probability distribution of the network series down into its factors, the *Markov assumption* gives us:

$$\mathcal{P}(Y^2, Y^3, \dots, Y^t | Y^1) = \mathcal{P}(Y^t | Y^{t-1}) \cdot \mathcal{P}(Y^{t-1} | Y^{t-2}) \cdot \dots \cdot \mathcal{P}(Y^2 | Y^1)$$

This also simplifies our function $\Psi : \mathbb{R}_{n \times n} \times \mathbb{R}_{n \times n} \rightarrow \mathbb{R}^k$ to represent a sufficient statistic on $Y^t | Y^{t-1}$, reducing our model to

$$\mathcal{P}(Y^t = y^t | Y^{t-1} = y^{t-1}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta}, y^{t-1})} \exp\{\boldsymbol{\theta}' \Psi(y^t, y^{t-1})\} \quad (3.4)$$

for each individual time step Y^t . As this model requires information about preceding states of the dynamic network, however, the joint distribution requires a distribution over the initial network Y^1 to be specified. This can be done with a classic ERGM in a previous step.

The one-step Markovian dependence assumption can be relaxed and generalized by including more than one time-step in the 'recent history', allowing investigation of phenomena stretching across multiple time-steps. These are called *higher order* Markov chains. Mathematically, this extension is straight forward:

$$\mathcal{P}(Y^t = y^t | Y^{t-1} = y^{t-1}, Y^{t-2} = y^{t-2}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta}, y^{t-1}, y^{t-2})} \exp\{\boldsymbol{\theta}' \Psi(y^t, y^{t-1}, y^{t-2})\}$$

As in many applications (3.4) suffices, we will focus on this case for the remainder of this work. But everything explained here, including the following chapter about example statistics, can easily be adapted to cover additional time steps, if needed. Nevertheless, there are some statistics, which require full access to historical information, for example *age* (of extant ties) based terms like the ones found in the `tergm`

package of R (cf [Krivitsky, v3]) So would the tergm-term `mean.age()` need the full history in order to compute the mean age of each tie, i.e. the number of time steps between its formation and its dissolution. The fact that these terms rejecting the Markov assumption are, at the time of this writing (package version 4.0.2), still not available during estimation in the current implementation of the package, only as target variables or summary statistics, may be a strong hint for the merits of making the Markov assumption from a pragmatic perspective. In addition to the Markov assumption, there is another property of statistics in the succeeding section, which is highly recommended to strive for when choosing TERGM statistics.

3.2.2 TERGM Statistics

When it comes to constructing network statistics with a temporal component, functions satisfying the following form are of special interest:

$$\Psi(Y^t, Y^{t-1}) = \sum_{ij} \Psi_{ij}(Y_{ij}^t, Y^{t-1}) \quad (3.5)$$

In these cases, the conditional probability distribution $\mathcal{P}(Y^t|Y^{t-1})$ factors over the individual edges of Y^t . If all k statistics incorporated in the model fulfill this property, then the model can be rewritten as

$$\begin{aligned} \mathcal{P}(Y^t = y^t | Y^{t-1} = y^{t-1}, \boldsymbol{\theta}) &= \frac{1}{Z(\boldsymbol{\theta}, y^{t-1})} \exp \left\{ \sum_k \theta_k \Psi_k(y^t, y^{t-1}) \right\} \\ &= \frac{1}{Z(\boldsymbol{\theta}, y^{t-1})} \exp \left\{ \sum_k \theta_k \left(\sum_{ij} \Psi_{ijk}(y_{ij}^t, y^{t-1}) \right) \right\} \end{aligned} \quad (3.6)$$

As explained by [Hanneke et al., 2010], this representation of the model reaps two very appealing benefits:

1. For one, when it comes to estimation, this property makes it possible to “tractably perform exact updates in Newton’s method”, instead of having to resort to approximation through sampling-based estimation procedures.
2. Additionally, it alleviates the *degeneracy* issues mentioned in the previous chapter, ERGMs often suffer from. With TERGM statistics being calculated

based on more than one snapshot of the network, finding ways to avoid this pitfall is of particular importance. As [Handcock, 2003] concludes, the *effective* parameter space of ERGMs is limited to a small, bounded subset of the theoretical parameter space. Limiting inferences to this region can be achieved with a Bayesian framework. But degenerate distributions reside very close to non-degenerate distributions, causing slight variations of parameters to cause degeneracy. This imposes a severe burden on the commonly used MCMC methods for maximum likelihood estimation as well as inference, causing it to require an impractically large number of samples, assuming them to converge at all.

Thankfully, it can be shown that for (3.6), as long as $\sum_k |\theta_k|$ is not too large, the entropy of the fitted model will be reasonably large, preventing degenerate probability distributions, resulting in consistently empty or complete graphs. In fact, [Hanneke et al., 2010] show that for (3.6), where each Ψ_{ijk} has a range contained in $[-\beta, \beta]$ for some $\beta > 0$, then the expected number of ties in Y^t is in the range

$$\left[\frac{n(n-1)}{\exp\{2\beta \sum_k |\theta_k|\} + 1}, \quad n(n-1) \left(1 - \frac{\exp\{2\beta \sum_k |\theta_k|\}}{\exp\{2\beta \sum_k |\theta_k|\} + 1} \right) \right]$$

The reason being that the edges of Y^t are conditionally independent given Y^{t-1} , as long as the dependence of Y^t on Y^{t-1} is not too strong. So the degeneracy issues innate to ERGMs are not obstructing their temporal extension.

To show this restriction does not impose a strong hindrance to common applications [Hanneke et al., 2010] presents three examples for such statistics:

1. *stability*, capturing the tendency of a tie's existence (or lack thereof) in $t - 1$ to continue at time t

$$\Psi_S(Y^t, Y^{t-1}) = \frac{1}{1-n} \sum_{ij} [Y_{ij}^t Y_{ij}^{t-1} + (1 - Y_{ij}^t)(1 - Y_{ij}^{t-1})]$$

2. *reciprocity*, describing the tendency of a tie from i and j at $t-1$ to be amended with a tie from j to i at time t (in directed networks)

$$\Psi_R(Y^t, Y^{t-1}) = n \cdot \frac{\sum_{ij} Y_{ji}^t Y_{ij}^{t-1}}{\sum_{ij} Y_{ij}^{t-1}}, \quad \text{setting } \frac{0}{0} \equiv 0$$

3. *transitivity*, representing the tendency of two ties from i to j and from j to k respectively in $t-1$ to result in a tie from i to k at time t (in directed networks)

$$\Psi_T(Y^t, Y^{t-1}) = n \cdot \frac{\sum_{ijk} Y_{ik}^t Y_{ij}^{t-1} Y_{jk}^{t-1}}{\sum_{ijk} Y_{ij}^{t-1} Y_{jk}^{t-1}}, \quad \text{setting } \frac{0}{0} \equiv 0$$

3.2.3 Markov Chain Monte Carlo Estimation

When it comes to the estimation of θ , the normalizing constant Z turns out to be a problem, as it may very well be computationally intractable in many cases. This problem, however, is not unique to dynamic networks, but, as already mentioned, immanent to ERGMs in general. So there are solutions based on *Markov Chain Monte Carlo (MCMC)* ([Gilks et al., 1996]) sampling for approximate maximum likelihood estimation, which have been well studied and tested. This approach is useful when the probability density function is known except for the normalizing constant, especially when dealing with very high dimensional densities. Instead of drawing from the density itself, MCMC methods generate a Markov chain, which has this density as stationary distribution. [Hanneke et al., 2010] expanded these algorithms to account for TERGMs:

To calculate the log-Likelihood

$$\mathcal{L}(\theta; Y^T, \dots, Y^2, Y^1) = \log \mathcal{P}(Y^T, \dots, Y^3, Y^2 | Y^1, \theta) \quad (3.7)$$

he defines

$$\mathbf{M}(t, \theta) = \mathbb{E}_\theta[\Psi(Y^t, Y^{t-1}) | Y^{t-1}] \quad (3.8)$$

and

$$\mathbf{C}(t, \theta) = \mathbb{E}_\theta[\Psi(Y^t, Y^{t-1}) \Psi(Y^t, Y^{t-1})' | Y^{t-1}] \quad (3.9)$$

leading to

$$\nabla \mathcal{L}(\boldsymbol{\theta}; Y^T, \dots, Y^2, Y^1) = \sum_{t=2}^T (\Psi(Y^t, Y^{t-1}) - M(t, \boldsymbol{\theta}))$$

and

$$\nabla^2 \mathcal{L}(\boldsymbol{\theta}; Y^T, \dots, Y^2, Y^1) = \sum_{t=2}^T (M(t, \boldsymbol{\theta})M(t, \boldsymbol{\theta})' - C(t, \boldsymbol{\theta}))$$

We can see the Markov assumption paying off in (3.8) and (3.9). The expectation values then need to be approximated by Gibbs sampling (or Metropolis-Hastings algorithms, cf. [Hunter et al., 2008]) from conditional distributions and used to update the parameter values in (3.7). This two-step process is then repeated until convergence. This is where statistics satisfying (3.5) come in handy, as they allow for performing exact updates using Newton’s method instead. Not forgetting to mention that the behaviour of MCMC algorithm is highly dependent on the chosen statistics ([Snijders, 2002]). In this regard, [Hunter et al., 2008] cites studies having reported difficulties obtaining convergence to non-degenerate networks at all when trying to simulate social networks.

3.3 Alternative Models

3.3.1 Separated TERGM (STERGM)

As [Krivitsky and Handcock, 2014] points out, there is a big issue with interpretation of TERGMs. By splitting a dynamic network into a discrete series of snapshots, the resulting instances only provide information about prevalence of ties, convoluting two distinct processes: incidence (the rate at which new ties are formed) and duration (how long ties tend to last). This blending of processes is not inherent to the discrete time series, but due to the way it is parametrized in TERGM models, which means it can be accounted for by adapting the model. The authors provide a few examples how the TERGM loses this kind of information.

First, the most basic statistic of all: the count of *edges* in a network. A high coefficient assigns a higher probability to networks with more ties. But in a dynamic network, this can be accomplished in two ways: Increasing the rate of formation, or making existing ties last longer.

The same happens with a TERGM-specific term, *stability* (ratio of the number of ties whose state did not change between two time steps, to the maximum number of possible ties). Here again, a higher coefficient (indicating a slower rate of change in the network) is the result of both fewer ties formed and longer lasting ties.

This not only adds additional complexity to the problematic of correlating parameters in (T)ERGMs, but presents a serious obstacle to model specification and interpretation.

A *seperable* TERGM (STERGM) solves this issue by assuming the formation and dissolution of ties occurring independently from each other within a time step. Each of these independent processes can be modeled as an ERGM.

If we define the relation \supseteq as: $\mathbf{y} \supseteq \mathbf{y}' \iff$ any tie present in \mathbf{y}' is also present in \mathbf{y} , for two networks $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$, and define $\mathcal{Y}^+(\mathbf{y}) = \{\mathbf{y}' \in \mathcal{Y} : \mathbf{y}' \supseteq \mathbf{y}\}$ (the set of possible networks constructed by forming ties in \mathbf{y}), as well as its counterpart $\mathcal{Y}^-(\mathbf{y}) = \{\mathbf{y}' \in \mathcal{Y} : \mathbf{y}' \subseteq \mathbf{y}\}$ (all networks which can be constructed by dissolution of ties in \mathbf{y}), we can model the *formation network* \mathbf{Y}^+ and respective *dissolution network* \mathbf{Y}^- as

$$\mathcal{P}(\mathbf{Y}^+ = \mathbf{y}^+ | \mathbf{Y}^t = \mathbf{y}^t; \boldsymbol{\theta}^+) = \frac{\exp\{\eta(\boldsymbol{\theta}^+) \cdot \Psi^+(\mathbf{y}^+)\}}{Z(\boldsymbol{\theta}^+, \mathcal{Y}^+(\mathbf{Y}^t))}, \quad \mathbf{y}^+ \in \mathcal{Y}^+(\mathbf{y}^t)$$

and

$$\mathcal{P}(\mathbf{Y}^- = \mathbf{y}^- | \mathbf{Y}^t = \mathbf{y}^t; \boldsymbol{\theta}^-) = \frac{\exp\{\eta(\boldsymbol{\theta}^-) \cdot \Psi^-(\mathbf{y}^-)\}}{Z(\boldsymbol{\theta}^-, \mathcal{Y}^-(\mathbf{Y}^t))}, \quad \mathbf{y}^- \in \mathcal{Y}^-(\mathbf{y}^t)$$

where the formation statistics $\Psi^+(\mathbf{y}^+)$, with respective formation parameter $\boldsymbol{\theta}^+$, is conditional only on added ties, and $\Psi^-(\mathbf{y}^-)$ only on dissolving ties. The subsequent time step \mathbf{Y}^{t+1} can then be constructed by applying \mathbf{Y}^+ and \mathbf{Y}^- to \mathbf{y}^t :

$$\mathbf{Y}^{t+1} = \mathbf{Y}^+ - (\mathbf{Y}^t - \mathbf{Y}^-)$$

This process is visualized in 3.4 by [Krivitsky and Goodreau, 2016]. Now we can formulate the previous assumption more precisely: We call the model of a dynamic network separable if \mathcal{Y}^+ is conditionally independent of \mathcal{Y}^- given \mathcal{Y}^{t-1} , and the parameter space of $\boldsymbol{\theta}$ is the product of the individual parameter spaces of $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$.

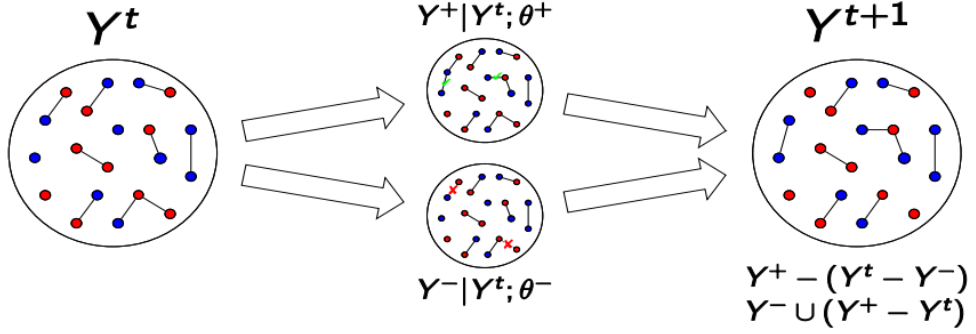


Figure 3.4: Visualization of the STERGM concept

As we can see, the step from TERGM to STERGM is not far, and the proposed methods to account for heterogeneous networks in this work can be easily extended to STERGM. Formally speaking, the stated independence assumption allows us to factorise the probability:

$$\begin{aligned}
\mathcal{P}(\mathbf{Y}^t = \mathbf{y}^t | \mathbf{Y}^{t-1} = \mathbf{y}^{t-1}; \theta) &= \mathcal{P}(\mathbf{Y}^+ = \mathbf{y}^+ | \mathbf{Y}^{t-1} = \mathbf{y}^{t-1}; \theta) \times \\
&= \mathcal{P}(\mathbf{Y}^- = \mathbf{y}^- | \mathbf{Y}^{t-1} = \mathbf{y}^{t-1}; \theta) \\
&= \frac{\exp\{\eta(\boldsymbol{\theta}^+) \cdot \Psi^+(\mathbf{y}^+, \mathbf{y}^{t-1})\}}{Z(\boldsymbol{\theta}^+, \mathcal{Y}^+(\mathbf{y}^{t-1}))} \times \frac{\exp\{\eta(\boldsymbol{\theta}^-) \cdot \Psi^-(\mathbf{y}^-, \mathbf{y}^{t-1})\}}{Z(\boldsymbol{\theta}^-, \mathcal{Y}^-(\mathbf{y}^{t-1}))} \\
&= \frac{\exp\{\eta(\boldsymbol{\theta}) \cdot \Psi(\mathbf{y}^t, \mathbf{y}^{t-1})\}}{Z(\boldsymbol{\theta}^+, \mathcal{Y}^+(\mathbf{y}^{t-1})) \cdot Z(\boldsymbol{\theta}^-, \mathcal{Y}^-(\mathbf{y}^{t-1}))},
\end{aligned}$$

resulting in a TERGM, proving the STERGM is a subclass of a (Markov) TERGM.

3.3.2 Longitudinal ERGM

[Koskinen et al., 2015] introduced a longitudinal extension of ERGM. In contrast to the TERGM, the *LERGM* is based on a continuous time model, where changes are restricted to sequences of changes of single tie variables. Similar to the basic ERGM, a logistic regression is used to model opportunities for change of dyads, conditional on the rest of the graph. Due to its continuous nature, within the ERGM family, this model is the closest one to *Stochastic Actor Oriented Models*, one of the most influential non-ERGM-based approaches.

3.3.3 Stochastic Actor Oriented Models (SAOM)

The *Stochastic Actor Oriented Model (SAOM)* introduced by [Snijders et al., 2010], sometimes also called *Stochastic Actor Based Model (SABM)*, is based on the notion that a network evolves as a stochastic process *driven by its actors*. The probabilities of tie changes are partly endogenous (as a function of the current network structure), partly exogenous (as a function of the actors themselves, as well as covariates of the respective dyad). Due to this actor-focus, the SAOM is designed with directed networks in mind. Its assumptions include

- a continuous time parameter, with networks representing observations, also referred to as *network panel waves* in analogy to panel surveys, at discrete points in time
- the changing network being the outcome of a Markov process
- actors controlling their outgoing ties (while still being subject to structural constraints)
- at any given moment (in the continuous process), on (probabilistically selected) actor may get the opportunity to change one outgoing tie

As a consequence, the actor-based change process consists of two (stochastic) sub-processes:

1. a *change opportunity process*, modeling the frequency of tie changes by actors
2. a *change determination process*, modeling the actual tie changes made (after actors got the opportunity)

As exact calculations for this model are infeasible, simulation based procedures are used. The estimation method used in practice is a method of moments procedure, but Bayesian and frequentist likelihood-based estimators have been developed as well (cf [Snijders, 2017]).

The SAOM's focus on the actors of the network makes it a great choice for representing theories describing how actors change their outgoing ties, where interpretation on a micro- (/actor-) level is important (as opposed to interpretations limited to network-level). This does not mean microlevel interpretations of TERGMs are not possible, though, as demonstrated in [Desmarais and Cranmer, 2011].

An in-depth theoretical and empirical comparison of the TERGM and the SAOM can be found in [Leifeld and Cranmer, 2019].

4 Heterogeneity

4.1 Definition

While the p_2 model of [Duijn et al., 2004], an alternative to ERGMs, accounts for *heterogeneous* nodes, one of the limitations of ERGMs is the obligatory assumption of *homogeneous* nodes, except for differences captured in exogenous nodal covariates [Thiemichen, 2016]. This limitation carries over to TERGMs. The homogeneity assumption, however, does not seem very realistic, as we intuitively expect nodes to display different levels of *affinity* for forming ties. And in many cases we may not be able to fully attribute this affinity to a set of exogenous covariates. Instead, we should consider this phenomenon to be caused by a latent variable we need to approximate through an indicator variable. This indicator variable is endogenous, meaning we need to extract it from the network structure itself.

The official **ERGM** package for R provides a term called *sociality*, which is defined as follows: “*This term adds one network statistic for each node equal to the number of ties of that node.*” (cf. [Handcock et al., 2014])

We cannot ‘override’ the ERGM’s assumption of homogeneity by adding this term to every model we fit, as the number of parameters to be estimated becomes way too large. We will need a more elaborate approach. So we will start by formulating the issue at hand mathematically. Fortunately, [Kevork and Kauermann, 2021] and [Kevork and Kauermann, 2022] already worked out a solution for ERGMs, so we only need to make a few modifications to expand his concept to include a temporal component. For the remainder of this work, we will refer to the latent variable as *affinity* (*of a given node towards forming ties*) to avoid confusion with the *sociality* term implemented in the **ERGM** package. We will use the *degree* (the number of ties) of a node as indicator variable for the node’s *affinity* towards forming nodes.

Replicating their approach, we extend the linear predictor $\boldsymbol{\theta}'\mathbf{s}(y^t, y^{t-1})$ of our TERGM

$$\mathcal{P}(Y^t = y^t | Y^{t-1} = y^{t-1}, \boldsymbol{\theta}) = \frac{\exp\{\boldsymbol{\theta}'\mathbf{s}(y^t, y^{t-1})\}}{Z(y^{t-1}, \boldsymbol{\theta})}$$

to include a metric \mathbf{u} capturing the individual *affinity* of each node:

$$\mathcal{P}(Y^t = y^t | Y^{t-1} = y^{t-1}, \boldsymbol{\theta}, \mathbf{u}) = \frac{\exp\{\boldsymbol{\theta}'\mathbf{s}(y^t, y^{t-1}) + \mathbf{u}'\mathbf{t}(y^t)\}}{Z(y^{t-1}, \boldsymbol{\theta}, \mathbf{u})}$$

with $\mathbf{t}(y^t) = \left(\sum_{j \neq 1} y_{1j}^t, \sum_{j \neq 2} y_{2j}^t, \dots\right)$ being the n dimensional vector of node degrees (where n is the number of nodes in the network).

This also requires the normalization $Z(\cdot)$ to be updated:

$$Z(y^{t-1}, \boldsymbol{\theta}, \mathbf{u}) = \log \sum_{y^t \in Y^t} \exp(\boldsymbol{\theta}'\mathbf{s}(y^t, y^{t-1}) + \mathbf{u}'\mathbf{t}(y^t))$$

As already mentioned before, this creates a plethora of parameters exceeding the number of nodes in the network. To resolve this, [Box-Steffensmeier et al., 2018] proposed a pseudo-likelihood approach. As [Duijn et al., 2009] pointed out, by assuming each edge in the network to be independent of every other edge, the MLE and MPLE approach are identical. If this condition is not met, however, estimates and standard errors from the MPLE approach are biased. Recognizing that this assumption is unfeasibly strict, they suggest applying a *penalized likelihood* method originally proposed by [Firth, 1993] to reduce the asymptotic bias of estimates. [Kevork and Kauermann, 2021], whose method we will use in this work, on the other hand propose to iterate between separate estimations of $\boldsymbol{\theta}$ and \mathbf{u} respectively, allowing to estimate $\boldsymbol{\theta}$ via MLE, avoiding the assumption of independent edges, while keeping the MPLE solution for \mathbf{u} .

4.2 Modeling Random Effects

Formally speaking, the way we integrate the *sociality* vector \mathbf{u} into our model is as *random effects*, expanding the linear predictor $\boldsymbol{\theta}'\mathbf{s}(y^t, y^{t-1})$ into a *linear mixed model*.

4.2.1 Linear Mixed Models

As [Wood, 2006] describes, a linear mixed model extends the linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}\sigma^2),$$

with fixed population effects \mathbf{b} to

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon}, \quad \mathbf{b} \sim N(\mathbf{0}, \boldsymbol{\psi}_\vartheta), \quad \boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Lambda}_\vartheta)$$

with all elements of \mathbf{b} and $\boldsymbol{\epsilon}$ mutually independent. The random vector \mathbf{b} contains the individual, not necessarily independent, *random effects*. The (positiv definite) covariance matrix $\boldsymbol{\psi}_\vartheta$, however, depends on some parameters ϑ . By introducing a new error term $\mathbf{u} = \mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon}$, we get a linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}, \quad \mathbf{u} \sim N(\mathbf{0}, \mathbf{Z}\boldsymbol{\psi}_\vartheta\mathbf{Z}'/\sigma^2 + \mathbf{I})$$

(cf [Kevork, 2017]), where the required assumptions, i.e. the identical and independent distribution of the error terms, is fulfilled, allowing to estimate $\boldsymbol{\beta}$ using the *least squares criterion*.

Now we can model the existence of a tie Y_{ij}^t between node i and j in the network Y^t taking into account random effects as follows:

$$"Y_{ij}^t" = \boldsymbol{\theta}'\boldsymbol{\Delta}_{ij}\mathbf{s}(y^t, y^{t-1}) + u_i + u_j, \quad 1 \leq i < j \leq n$$

As you can see, there is a small modification to $\mathbf{s}(y^t, y^{t-1})$. In order to consider each edge in the network individually, we define the *change statistics*¹ $\boldsymbol{\Delta}_{ij}\mathbf{s}(y^t, y^{t-1}) = \mathbf{s}(y_{ij}^t = 1, y_{-ij}^t, y^{t-1}) - \mathbf{s}(y_{ij}^t = 0, y_{-ij}^t, y^{t-1})$. The notation Y_{ij}^t for the ties is in quotation marks, because in our case we only consider unweighted and undirected ties,

¹Let y_{-ij}^t be the network y^t excluding information about the dyad ij

reducing the possible values for Y_{ij}^t to 1 (*tie between i and j*) and 0 (*no tie between i and j*), requiring the use of a link-function, which will be covered in the succeeding chapter.

4.2.2 Logistic Regression

As our target variable Y_{ij}^t only has two values, *edge* and *no edge*, represented as 1 and 0, we essentially have a classification problem. And as [Molnar, 2019] points out, “*The linear regression model can work well for regression, but fails for classification.*”

There are various reasons for this. One of the more obvious ones being the fact that the linear model extrapolates, meaning the response variable can yield values beyond 1 and below 0, which not presents a challenge for interpretation, but can also jeopardize subsequent analysis depending on well-defined values.

But, as [Kevork, 2017] pointed out, there are additional, more subtle issues with this approach. For once, the homoscedasticity assumption of the error terms, stating that the variance of the response variable shall not depend on the independent variables, is violated:

Let ϵ_{ij} be the error term in our model, then we can easily see that in our binary case, $\text{var}(\epsilon_{ij}) = \mathcal{P}(Y_{ij}^t = 1) \cdot (1 - \mathcal{P}(Y_{ij}^t = 1))$. And as $\mathcal{P}(Y_{ij}^t = 1)$ depends on our linear predictor, the error terms depend on our independent variables.

Another violated assumption of *classical regression models* is the error term not following a normal distribution, as the response variable only takes on two possible values.

But probably the most severe issue with this approach is that a linear regression is incredibly unreliable when misused for classification. In order to obtain a binary result, we will need to define a threshold in the continuous spectrum between 0 and 1. In graphic 4.1, [Molnar, 2019] provides an illustrative example, why this is not going to work. The data for this example classifies tumors as malignant (1) or benign (0). A linear model fit to predict the classification of the tumor depending on its size.

0.5 is used as classification threshold. The left and right picture show the results for different samples, with the right one having a few additional 'obvious' cases (i.e. very big and malignant tumors). As we can see, the regression line is shifted and no longer useful after adding these obvious cases, which should not happen in a classification model.

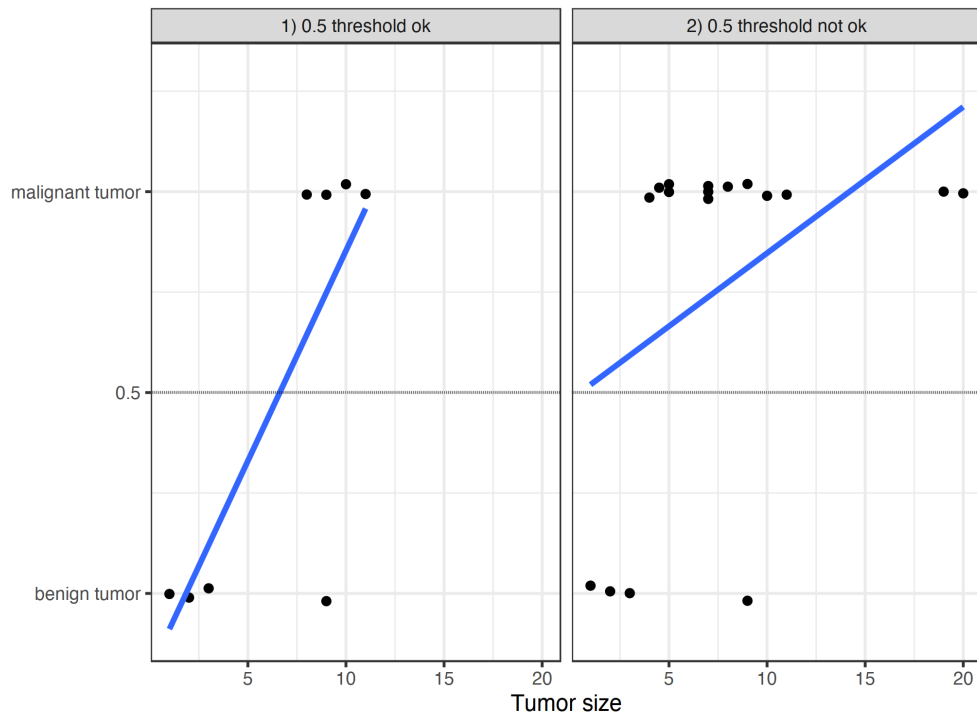


Figure 4.1: Attempts of classification with a linear model

A linear model classifies tumors as malignant (1) or benign (0) given their size. The lines show the prediction of the linear model. For the data on the left, we can use 0.5 as classification threshold. After introducing a few more malignant tumor cases, the regression line shifts and a threshold of 0.5 no longer separates the classes, rendering the model useless.

Fortunately, all these issues can be resolved quite easily by transforming the linear predictor into a form that emits monotonically increasing values between 0 and 1, allowing us to interpret them as probabilities for a tie between the respective actors i and j to exist:

$$\mathcal{P}(Y_{ij}^t = 1 | \boldsymbol{\theta}, \mathbf{u}, y_{-ij}^t, y^{t-1}) = \frac{\exp(\boldsymbol{\theta}' \boldsymbol{\Delta}_{ij} \mathbf{s}(y^t, y^{t-1}) + u_i + u_j)}{1 + \exp(\boldsymbol{\theta}' \boldsymbol{\Delta}_{ij} \mathbf{s}(y^t, y^{t-1}) + u_i + u_j)}$$

Now the model produces a curve as displayed in graphic 4.2. This result satisfies the needs for a binary classification. Conveniently, its steepness represents how fuzzy or clear the separation of both categories turned out in the sample.

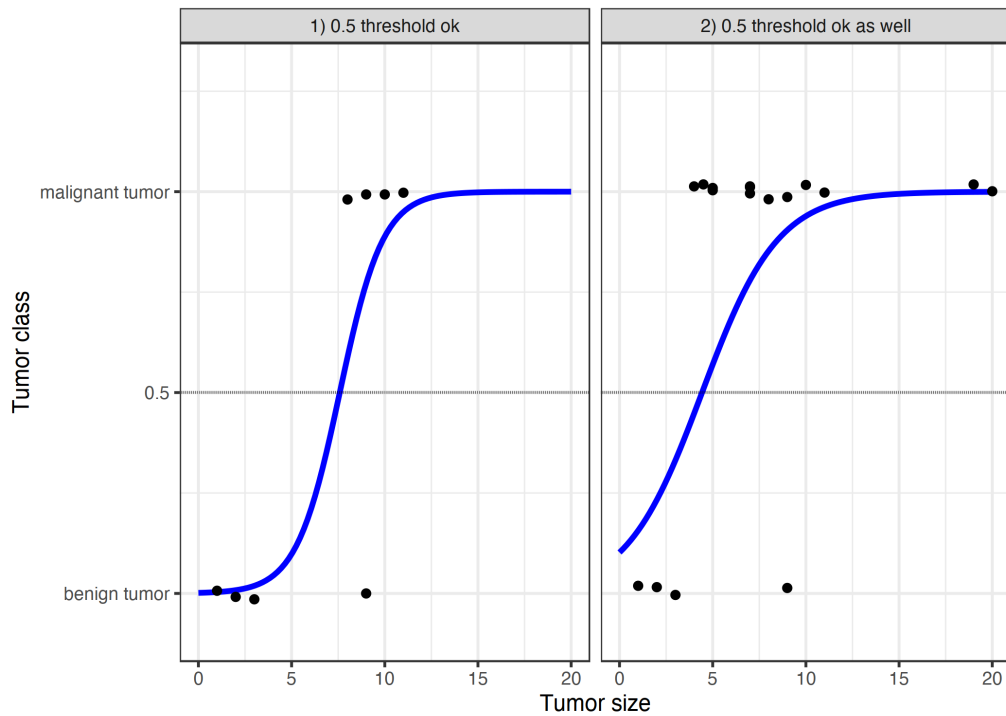


Figure 4.2: Classification with a logistic regression model

When it comes to estimation, however, linear models have very appealing properties we do not want to miss out on. There is a simple mathematical 'trick', which allows us to combine the advantages of both the logistic model and the linear mixed model we had earlier. As this happens to be the standard approach to estimating logistic models, there is a plethora of well-established research and literature on this topic, known as *generalized linear (mixed) models*.

4.2.3 Generalized Linear Mixed Models

When trying to model non-linear response variables, other than constructing non-linear predictors, there is often the possibility to instead transform the response variable into a linear quantity. Doing this gives us what we call a *generalized linear (mixed) model* (GLMM). Let $\mu^b \equiv \mathbb{E}(y|b)$, then a GLMM has the form

$$g(\mu^b) = X_i\beta + Z_ib, \quad b \sim N(0, \psi_\vartheta)$$

where $y_i|b$ follow an exponential family distribution and are independent, and g is a monotonic link function. (cf [Wood, 2006], [Fahrmeir et al., 2013])

To get to this form, we define a *response function* $h(\cdot)$,

$$h(\eta_{ij}) = \frac{\exp(\eta_{ij})}{1 + \exp(\eta_{ij})},$$

to link the probability \mathcal{P} to the linear predictor:

$$\mathcal{P}(Y_{ij}^t = 1 | \boldsymbol{\theta}, \mathbf{u}, y_{-ij}^t, y^{t-1}) = \mathbf{h}\left(\boldsymbol{\theta}' \boldsymbol{\Delta}_{ij} \mathbf{s}(y^t, y^{t-1}) + u_i + u_j\right)$$

$h(\cdot)$ has a distribution from the exponential families, the *binomial family* in our case, and $h(\eta_{ij}) \in [0, 1]$.

As $h(\cdot)$ is strictly monotonically increasing, we can define its inverse function $g(\cdot) = h^{-1}(\cdot)$, also called *link function*. Let $\pi_{ij} = h(\eta_{ij})$, then

$$g(\pi_{ij}) = \log\left(\frac{\pi_{ij}}{1 - \pi_{ij}}\right) = \eta_{ij},$$

which gives us the GLMM

$$\mathbf{g}\left(\mathcal{P}(Y_{ij}^t = 1 | \boldsymbol{\theta}, \mathbf{u}, y_{-ij}^t, y^{t-1})\right) = \boldsymbol{\theta}' \boldsymbol{\Delta}_{ij} \mathbf{s}(y^t, y^{t-1}) + u_i + u_j$$

or, in explicit form

$$\log\left\{\frac{\mathcal{P}(Y_{ij}^t = 1 | y_{-ij}^t, y^{t-1}, \boldsymbol{\theta}, \mathbf{u})}{1 - \mathcal{P}(Y_{ij}^t = 1 | y_{-ij}^t, y^{t-1}, \boldsymbol{\theta}, \mathbf{u})}\right\} = \boldsymbol{\theta}' \boldsymbol{\Delta}_{ij} \mathbf{s}(y^t, y^{t-1}) + u_i + u_j$$

If we replace $1 - \mathcal{P}(Y_{ij}^t = 1|\dots)$ with $\mathcal{P}(Y_{ij}^t = 0|\dots)$, we get

$$\log \left\{ \frac{\mathcal{P}(Y_{ij}^t = 1|y_{-ij}^t, y^{t-1}, \boldsymbol{\theta}, \mathbf{u})}{\mathcal{P}(Y_{ij}^t = 0|y_{-ij}^t, y^{t-1}, \boldsymbol{\theta}, \mathbf{u})} \right\} = \boldsymbol{\theta}' \boldsymbol{\Delta}_{ij} \mathbf{s}(y^t, y^{t-1}) + u_i + u_j$$

This form reminds us of (3.2) of model p_2 , meaning we have successfully integrated the features of p_2 into TERGM. This *logit*-model-typical form also provides an easy means of interpretation: the *log odds ratio*.

Although inference in GLMMs is based on the same likelihood or Bayes principles as in LMMs, the commonly non-linear relationship between mean and predictor often requires numerical methods or approximations, e.g. approximate penalized likelihood or MCMC techniques (cf [Fahrmeir et al., 2013]). The `mgcv` library we are using employs penalized regression terms.

4.3 Nodal Random Effects in Networks

Now we can formalize our full model. We start off with a TERGM and account for the heterogeneity of the nodes by extending the linear predictor to include random effects u for each node:

$$\mathcal{P}(Y^t = y^t | Y^{t-1} = y^{t-1}, \boldsymbol{\theta}, \mathbf{u}) = \frac{\exp\{\boldsymbol{\theta}' \mathbf{s}(y^t, y^{t-1}) + \mathbf{u}' \mathbf{t}(y^t)\}}{Z(y^{t-1}, \boldsymbol{\theta}, \mathbf{u})} \quad (4.1)$$

with

$$\mathbf{u} \sim \mathcal{N}(0, \sigma_u^2 \mathcal{I}_n), \quad (4.2)$$

where \mathcal{I}_n is the n dimensional identity matrix. We derive the log-likelihood

$$\begin{aligned} l(\boldsymbol{\theta}, \sigma_u^2) &= \log \int \frac{\exp\{\boldsymbol{\theta}' \mathbf{s}(y^t, y^{t-1}) + \mathbf{u}' \mathbf{t}(y^t)\}}{Z(y^{t-1}, \boldsymbol{\theta}, \mathbf{u})} \cdot \frac{1}{(2\pi\sigma_u^2)^{\frac{n}{2}}} \cdot \exp\left(-\frac{1}{2} \frac{\mathbf{u}' \mathbf{u}}{\sigma_u^2}\right) \prod_{i=1}^n du_i \\ &= \boldsymbol{\theta}' \mathbf{s}(y^t, y^{t-1}) - \frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma_u^2) + \\ &\quad + \log \int \exp\left(\mathbf{u}' \mathbf{t}(y^t) - \log(Z(y^{t-1}, \boldsymbol{\theta}, \mathbf{u})) - \frac{1}{2} \cdot \frac{\mathbf{u}' \mathbf{u}}{\sigma_u^2}\right) \prod_{i=1}^n du_i \end{aligned} \quad (4.3)$$

Replacing the integral with a Laplace approximation, as proposed by [Breslow and Clayton, 1993], yields

$$l(\boldsymbol{\theta}, \sigma_u^2) \approx \boldsymbol{\theta}' \mathbf{s}(y^t, y^{t-1}) + \hat{\mathbf{u}}' \mathbf{t}(y^t) - \log(Z(y^{t-1}, \boldsymbol{\theta}, \hat{\mathbf{u}})) - \frac{1}{2} \frac{\hat{\mathbf{u}}' \hat{\mathbf{u}}}{\sigma_u^2} - \frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma_u^2) - \frac{1}{2} \log \left| \frac{\partial^2 (\hat{\mathbf{u}}' \mathbf{t}(y^t) - \log(Z(y^{t-1}, \boldsymbol{\theta}, \hat{\mathbf{u}})) - \frac{1}{2} \cdot \frac{\hat{\mathbf{u}}' \hat{\mathbf{u}}}{\sigma_u^2})}{\partial \hat{\mathbf{u}} \partial \hat{\mathbf{u}}} \right| \quad (4.4)$$

We can consider this as (approximated) profile likelihood where we can find our estimator $\hat{\mathbf{u}}$ by maximizing the last term in (4.4), using

$$\frac{\partial (\mathbf{u}' \mathbf{t}(y^t) - \log(Z(y^{t-1}, \boldsymbol{\theta}, \mathbf{u})) - \frac{1}{2} \cdot \frac{\mathbf{u}' \mathbf{u}}{\sigma_u^2})}{\partial \mathbf{u}} = 0$$

and

$$\frac{\partial \log(Z(y^{t-1}, \boldsymbol{\theta}, \mathbf{u}))}{\partial \mathbf{u}} = \mathbb{E}(\mathbf{t}(y^t) | \mathbf{u}) \quad (4.5)$$

Then we can focus solely on $\boldsymbol{\theta}$:

$$\mathcal{P}(Y^t = y^t | Y^{t-1} = y^{t-1}, \boldsymbol{\theta}, \hat{\mathbf{u}}) = \frac{\exp\{\boldsymbol{\theta}' \mathbf{s}(y^t, y^{t-1}) + \hat{\mathbf{u}}' \mathbf{t}(y^t)\}}{Z(y^{t-1}, \boldsymbol{\theta}, \hat{\mathbf{u}})} \quad (4.6)$$

for which we can use any TERGM estimation method, like the common MCMC-MLE. There is still an issue with (4.5), though, as the high dimensionality of \mathbf{u} proves numerically challenging. This is where we need the GLMM we prepared in previous chapters:

$$\log \left\{ \frac{\mathcal{P}(Y_{ij}^t = 1 | y_{-ij}^t, y^{t-1}, \boldsymbol{\theta}, \mathbf{u})}{\mathcal{P}(Y_{ij}^t = 0 | y_{-ij}^t, y^{t-1}, \boldsymbol{\theta}, \mathbf{u})} \right\} = \boldsymbol{\theta}' \Delta_{ij} \mathbf{s}(y^t, y^{t-1}) + u_i + u_j \quad (4.7)$$

Following [Duijn et al., 2009] we can now derive the pseudo-log-likelihoods

$$l_{pseudo}(\mathbf{u}) = \sum_i \sum_{j>i} y_{ij}^t \{o_{ij} + u_i + u_j\} - \log\{1 + \exp(o_{ij} + u_i + u_j)\} \quad (4.8)$$

and

$$l_{pseudo}(\sigma_u^2) \approx -\frac{1}{2} \cdot \frac{\check{u}'\check{u}}{\sigma_u^2} - \frac{n}{2} \log(\sigma_u^2), \quad (4.9)$$

where \check{u} is the maximizer of

$$\mathbf{u}'\mathbf{t}(y^t) - \sum_i \sum_j \log\{1 + \exp(o_{ij} + u_i + u_j)\} - \frac{1}{2} \cdot \frac{\mathbf{u}'\mathbf{u}}{\sigma_u^2} \quad (4.10)$$

In combination with (4.2), we receive a (pseudo) generalized linear mixed model and can use [Breslow and Clayton, 1993] for estimation. The attentive reader may have spotted a new symbol in (4.8) and (4.10). We integrated $\boldsymbol{\theta}'\boldsymbol{\Delta}_{ij}\mathbf{s}(y^t, y^{t-1})$ as *given offset* o_{ij} .

Here we can see one final problem with our approach: \mathbf{u} depends on $\boldsymbol{\theta}$, meaning we need $\boldsymbol{\theta}$ to estimate \mathbf{u} as much as we need \mathbf{u} to estimate $\boldsymbol{\theta}$. The solution is an iterative process alternately updating the respective values of \mathbf{u} and $\boldsymbol{\theta}$.

4.4 Iterative Estimation using TERGM

Putting it all together, we will use our TERGM and GLMM to iteratively estimate \mathbf{u} and $\boldsymbol{\theta}$. We estimate the first $\boldsymbol{\theta}_{(0)}$ as intercept alongside $\mathbf{u}_{(0)}$, which we can then use as offset for the first TERGM estimation of $\boldsymbol{\theta}_{(1)}$, which we then again use to estimate an updated $\mathbf{u}_{(1)}$ with the GLMM, where we set $\boldsymbol{\theta}'\boldsymbol{\Delta}_{ij}\mathbf{s}(y^t, y^{t-1})$ as fixed offset. We continue this process until θ converges.

4.4.1 Initial Values

First we need to obtain an initial prediction for \mathbf{u} and estimate σ_u^2 . As we do not yet have an estimation for θ , we fit the first GLMM with an intercept $\theta_{(0)}$:

$$\log \left\{ \frac{\mathcal{P}(Y_{ij}^t = 1 | y_{-ij}^t, y^{t-1}\boldsymbol{\theta}, \mathbf{u})}{\mathcal{P}(Y_{ij}^t = 0 | y_{-ij}^t, y^{t-1}, \boldsymbol{\theta}, \mathbf{u})} \right\} = \theta_{(0)} + u_i + u_j$$

(4.8) and (4.9) give us $\hat{\mathbf{u}}$ and $\hat{\boldsymbol{\sigma}}^2$.

4.4.2 Estimate TERGM

Now we can estimate (4.6):

$$\mathcal{P}(Y^t = y^t | Y^{t-1} = y^{t-1}, \boldsymbol{\theta}, \hat{\mathbf{u}}) = \frac{\exp\{\boldsymbol{\theta}'\mathbf{s}(y^t, y^{t-1}) + \hat{\mathbf{u}}'\mathbf{t}(y^t)\}}{Z(y^{t-1}, \boldsymbol{\theta}, \hat{\mathbf{u}})}$$

where $\hat{\mathbf{u}}'\mathbf{t}(y^t)$ is fixed as given offset.

4.4.3 Estimate Random Effects

We can now use the resulting $\boldsymbol{\theta}'\boldsymbol{\Delta}_{ij}\mathbf{s}(y^t, y^{t-1})$ as offset to replace the intercept, and fit the GLMM (4.7):

$$\log \left\{ \frac{\mathcal{P}(Y_{ij}^t = 1 | y_{-ij}^t, y^{t-1}\boldsymbol{\theta}, \mathbf{u})}{\mathcal{P}(Y_{ij}^t = 0 | y_{-ij}^t, y^{t-1}, \boldsymbol{\theta}, \mathbf{u})} \right\} = \boldsymbol{\theta}'\boldsymbol{\Delta}_{ij}\mathbf{s}(y^t, y^{t-1}) + u_i + u_j$$

using (4.8) and (4.9) again and update $\hat{\mathbf{u}}$ and $\hat{\boldsymbol{\sigma}}^2$ for the next iteration cycle.

5 TERGM Implementations in R

An overview over relevant software implementations of network modeling can be found in [Kauermann et al., 2019]. Most software for statistical inference is written for the R environment, with a notable exception being the PNet ¹ software for statistical network models, developed at the University of Melbourne. Especially for implementations of models residing right at the frontier of academic research, R's open source model seems like the perfect fit. A good starting point is R's `statnet` suite of packages ², offering a plethora of statistical network models. This is also the one used for the code behind this work. For *stochastic actor oriented models (SAOM)* on the other hand, the `RSiena` ³ package is worth a look. An approach which can be considered analogous to the SAOM is the *temporal network autocorrelation model (TNAM)* ⁴. There is also a generalized ERGM for networks with weighted ties under development in the `GERGM` package⁵. A library focused on simulation of networks (based on the SAOM) is `NetSim`⁶.

5.1 Change Statistics

For the initial estimation of the GLMM, the network's change statistics are needed. The ERGM package makes their extraction easy by providing this functionality out of the box. With TERGM libraries, however, we are not as lucky. So, for the first GLMM fit, manual calculation of the change statistics of specific TERGM terms may be necessary.

¹<https://www.melnet.org.au/pnet/>, last visited on 10.2.2022

²<https://statnet.org/>, last visited on 10.2.2022

³<https://www.stats.ox.ac.uk/~snijders/siena/>, last visited on 10.2.2022

⁴<https://CRAN.R-project.org/package=tnam>, the package is archived and currently not available through the CRAN repository

⁵<https://CRAN.R-project.org/package=GERGM>, last visited 15.3.2022

⁶<https://rdr.io/rforge/NetSim/>, last accessed 16.3.2022

5.2 TERGM Package in statnet

The specifications supported in the `tergm` package of the `statnet` suite are based on a separation of edge loss and edge creation functions, as proposed by [Krivitsky and Handcock, 2014], allowing to smoothly transition between TERGM and STERGM by choosing the respective formula. Unfortunately, at the time of this writing (version 4.0.2), the library does not have implemented estimation for TERGM-terms yet. They can only be used as targets, monitors or summary statistics, as stated in the documentation (cf [Krivitsky, v3]).

The available `tergm`-terms are all following a durational pattern, e.g. `mean.age`, `edge.ages`, `degree.mean.age`, et cetera.

The estimation comes with two options (currently only supporting ERGM-terms): Conditional-Maximum-Likelihood-Estimation (CMLE) and a simulated method of moments procedure, named Equilibrium-Generalized-Method-of-Moments (EGMME).

5.2.1 CM(P)LE

The *CMPL*E version (using pseudo-likelihood) is not recommended, as it is still very inaccurate at this time [Krivitsky, v3]. The CMLE version on the other hand, which is the one used in this work, is very accurate, assuming it converges, which is not guaranteed unfortunately, as it is prone to degeneracy when the model specification does not capture the true data-generating process. The CMLE algorithm is based on [Krivitsky and Handcock, 2014]. It is an extension of [Hunter and Handcock, 2006], which uses a Method-of-Moments estimator

$$\mathbb{E}_{\boldsymbol{\theta}} \boldsymbol{\Psi}(Y) = \boldsymbol{\Psi}(y_{obs})$$

with a simulated Newton-Raphson zero-finding algorithm. The extension includes cases where $\boldsymbol{\eta}(\boldsymbol{\theta}) \neq \boldsymbol{\theta}$. The conditional MLE is obtained by maximizing the log-likelihood

$$l(\boldsymbol{\theta}) = \boldsymbol{\eta}(\boldsymbol{\theta}) \cdot \left(\sum_{t=1}^T \boldsymbol{\Psi}(\mathbf{y}^t, \mathbf{y}^{t-1}) \right) - \log \left(\prod_{t=1}^T Z_{\boldsymbol{\eta}, \boldsymbol{\Psi}}(\boldsymbol{\theta}, \mathbf{y}^{t-1}) \right)$$

with the log-likelihood-ratio

$$l(\boldsymbol{\theta}) - l(\boldsymbol{\theta}^0) = (\boldsymbol{\eta}(\boldsymbol{\theta}) - \boldsymbol{\eta}(\boldsymbol{\theta}^0)) \cdot \left(\sum_{t=1}^T \boldsymbol{\Psi}(\mathbf{y}^t, \mathbf{y}^{t-1}) \right) - \log \left(\prod_{t=1}^T \frac{Z_{\eta, \Psi}(\boldsymbol{\theta}, \mathbf{y}^{t-1})}{Z_{\eta, \Psi}(\boldsymbol{\theta}^0, \mathbf{y}^{t-1})} \right)$$

The problematic normalizing constant can be expressed as

$$\begin{aligned} \prod_{t=1}^T \frac{Z_{\eta, \Psi}(\boldsymbol{\theta}, \mathbf{y}^{t-1})}{Z_{\eta, \Psi}(\boldsymbol{\theta}^0, \mathbf{y}^{t-1})} &= \prod_{t=1}^T \sum_{\mathbf{y} \in Y} \exp((\boldsymbol{\eta}(\boldsymbol{\theta}) - \boldsymbol{\eta}(\boldsymbol{\theta}^0)) \cdot \boldsymbol{\Psi}(\mathbf{y}^t, \mathbf{y}^{t-1})) \times \frac{\exp(\boldsymbol{\eta}(\boldsymbol{\theta}^0) \cdot \boldsymbol{\Psi}(\mathbf{y}^t, \mathbf{y}^{t-1}))}{Z_{\eta, \Psi}(\boldsymbol{\theta}^0, \mathbf{y}^{t-1})} \\ &= \prod_{t=1}^T \sum_{\mathbf{y} \in Y} \exp((\boldsymbol{\eta}(\boldsymbol{\theta}) - \boldsymbol{\eta}(\boldsymbol{\theta}^0)) \cdot \boldsymbol{\Psi}(\mathbf{y}^t, \mathbf{y}^{t-1})) \times \mathcal{P}(\mathbf{Y}^t = \mathbf{y} | \mathbf{Y}^{t-1} = \mathbf{y}^{t-1}; \boldsymbol{\theta}^0) \quad (5.1) \end{aligned}$$

(5.1) is the product of expectations over the conditional distribution of \mathbf{Y}^t given \mathbf{Y}^{t-1} at $\boldsymbol{\theta}^0$. These can be estimated by MCMC simulation, using the algorithm proposed in [Hunter and Handcock, 2006]:

1. Select initial value for $\boldsymbol{\theta}^0$
2. Generate MCMC sample $\mathbf{Z}(Y_1), \dots, \mathbf{Z}(Y_m)$ based on $\boldsymbol{\theta}^0$
3. Iterative estimation until convergence, obtaining a maximizer $\tilde{\boldsymbol{\theta}}$ for the log-likelihood ratio $l(\boldsymbol{\theta}) - l(\boldsymbol{\theta}^0)$.
4. If the estimated variance of the log-likelihood-ratio is too large compared to $\hat{l}(\boldsymbol{\eta}(\tilde{\boldsymbol{\theta}}))$, then set $\boldsymbol{\theta}^0 = \tilde{\boldsymbol{\theta}}$ and repeat from step 2.
5. $\tilde{\boldsymbol{\theta}}$ is the MCMCMLE

The reason for the use of a *conditional* MLE in the first place is, that a new sample would have to be generated each time the value of $\boldsymbol{\theta}$ changed, which is computationally highly expensive. Thus, a single sample is generated based on a fixed parameter $\boldsymbol{\theta}^0$ instead.

5.2.2 EGMME

The Equilibrium Generalized Method of Moments Estimation is based on a single cross-sectional network with retrospective temporal information (collected on the duration of ties), assuming that it is a product of a TERGM process running to its stationary (equilibrium) distribution.

We could imagine a real world example where this kind of data is collected in a one-time survey. The EGMME serves as method to better account for these scenarios. It assumes that this process has reached an equilibrium and uses the relationship

$$prevalence = incidence \times duration$$

of ties and use the given data to solve for *incidence* in the formation model (cf. [Krivitsky et al.,] for more details). The implementation is based on [Krivitsky, 2012]. The distinction between incidence (formation) and duration (time until dissolution) aligns with the idea behind the STERGM.

5.3 TERGM Package `btergm`

Because the TERGM library from the `statnet` suite is computationally demanding, which makes estimation for big networks unfeasible, [Leifeld et al., 2018] created the `btergm` library to fill that gap.

The `btergm` package comes packed with three implemented temporal network statistics:

- `delrecip`: *delayed reciprocity or mutuality*
- `timecov`: *time trend or interaction with time*
- `memory`: *'memory'-statistic of one of the three types* positive autoregression, dyadic stability, edge innovation or loss

The package provides three different methods of estimation. It's main option, *bootstrapped pseudo-likelihood*, as well as an *MCMC-MLE* (`mtergm`) option for smaller networks, and the recently added *Bayesian* (`tbergm`) counterpart.

5.3.1 Bootstrapped Pseudo-Likelihood

As laid out in [Leifeld et al., 2018], *maximum pseudo-likelihood estimation (MPLE)* replaces the joint likelihood with the product over dyadic tie probabilities π_{ij} conditional on the network, where

$$\pi_{ij}(\boldsymbol{\theta}) = \mathcal{P}(Y_{ij} = 1 | Y_{-ij}, \boldsymbol{\theta})$$

The estimation is then, considering π_{ij} to be a function of the past $t - K$ networks, done by maximizing

$$\operatorname{argmax}_{\boldsymbol{\theta}} \sum_{t=K+1}^T \sum_{\langle ij \rangle} \ln \left[(\pi_{ij}^t(\boldsymbol{\theta}))^{Y_{ij}^t} (1 - \pi_{ij}^t(\boldsymbol{\theta}))^{1 - Y_{ij}^t} \right],$$

where T indicates the networks in the time series, and $\langle ij \rangle$ indicates the set of dyads in the network. The optimization is achieved with a hill-climbing algorithm which does not involve simulation. This is where the speed behind this approach originates, keeping estimations computationally feasible for big volumes of data. As the MPLE asymptotically converges to the MLE, this is where this approach shows its strength. When the amount of data, i.e. the number of nodes or the number of timesteps, goes to infinity, the estimation is consistent, while at the same time other, computationally more expensive, approaches become impractical.

Furthermore, taking into account the fact that

$$\sum_{t=K+1}^T \frac{\partial \sum_{\langle ij \rangle} \ln \left[(\pi_{ij}^t(\hat{\boldsymbol{\theta}}))^{Y_{ij}^t} (1 - \pi_{ij}^t(\hat{\boldsymbol{\theta}}))^{1 - Y_{ij}^t} \right]}{\partial \hat{\boldsymbol{\theta}}} = 0$$

when taking n estimates of $\boldsymbol{\theta}$ by drawing n samples of $T - K$ networks, shows this bootstrapping approach (originally described by [Desmarais and Cranmer, 2012]) mitigates the problem of substantially underestimating the MPLE's variance, like it would be the case for single network ERGMs. With the sample data used in this work being limited to only two time steps, however, this approach will not work for our purposes.

5.3.2 MCMC-MLE

Even though the focus of the `btergm` package lies on the fast method of MPLE with bootstrap confidence intervals, it also offers the classic Markov Chain Monte Carlo Maximum Likelihood Estimation (MCMC-MLE) for TERGMs. In this approach, all networks are combined into a big block-diagonal matrix (with structural zeros prohibiting cross-network edges), which is then used to estimate an ERGM using the procedure proposed by [Hunter et al., 2008]. This implementation, however, is extremely slow and not recommended for anything but small networks with very few observations. As our iterative approach would only magnify this speed issue, this was not a feasible option for our objective. Additionally, although it may yield more accurate results than the bootstrapped counterpart, it is prone to degeneracy when the model is misspecified.

5.3.3 Bayesian Estimation

Recently, there was added another option to this package, `tbergm`⁷, acting as a wrapper for the `bergm` function in the `Bergm` package. It computes TERGMs through Bayesian estimation (cf. [Caimo and Friel, 2014]) using block-diagonal matrices and structural zeros, similar to the MCMC-MLE approach. For a mathematical formulation of our model (TERGM including nodal random effects, (4.1)) in a Bayesian framework, we can extend the formulation for ERGMs in [Thiemichen, 2016] by imposing a prior distribution $p(\boldsymbol{\theta})$ on $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)'$, resulting in the posterior

$$p(\boldsymbol{\theta}, \mathbf{u}, \sigma_u^2 | \mathbf{y}^t, \mathbf{y}^{t-1}) = \frac{f(\mathbf{y}^t, | \mathbf{y}^{t-1}, \boldsymbol{\theta}, \mathbf{u}) p(\boldsymbol{\theta}) p(\mathbf{u} | \sigma_u^2) p(\sigma_u^2)}{p(\mathbf{y})},$$

where $p(\mathbf{u} | \sigma_u^2)$ is the prior distribution of the random nodal effects \mathbf{u} and

$$f(\mathbf{y}^t, | \mathbf{y}^{t-1}, \boldsymbol{\theta}, \mathbf{u}) = \frac{\exp\{\boldsymbol{\theta}' \mathbf{s}(\mathbf{y}^t, \mathbf{y}^{t-1}) + \mathbf{u}' \mathbf{t}(\mathbf{y}^t)\}}{Z(\mathbf{y}^{t-1}, \boldsymbol{\theta}, \mathbf{u})}$$

The respective distributions are $\mathbf{u} \sim \mathcal{N}(0, \sigma_u^2 \mathbf{I}_n)$, $\boldsymbol{\theta} \sim \mathcal{N}(0, \rho^2 \mathbf{I}_p)$ (ρ^2 chosen such that the prior distribution is flat), and the assumption of the inverse gamma distribution $\sigma_u^2 \sim \mathcal{IG}(a, b)$ (a, b again being chosen so that the hyper prior distribution is flat). It shall be noted, however, that at the time of this writing this library is

⁷<https://rdrr.io/cran/btergm/man/tbergm.html>, last accessed 14.2.2022

rather new and there is not much documentation about the implementation to be found.

5.4 GLMM Package `mgcv`

When it comes to the random effects, we will use the same library as original version of this approach in [Kevork and Kauermann, 2021]. While the `mgcv` package provides the `gamm` function for fitting GLMMs, the `gam` function will be used instead. The reason being that, for estimation purposes, the smooth components of GAMs can be viewed as random effects. In this approach, the random effects are represented as penalized regression terms.

Other than classic GLMM libraries, however, `gam`, needs the full penalty matrix for each random effect, and doesn't use sparse matrices for efficient computation, like `gamm`. This can lead to high computation times for bigger networks. Because computation time is a crucial factor in our iterative approach, we will use the `bam` function instead. It fits based on derivatives of the *Reduced Maximum Likelihood* score, reducing the expensive computations to one parallel block Cholesky decomposition per `bam`-iteration. [Wood, 2021]

6 Evaluation

In order to properly evaluate our algorithm, we will run it on simulated networks, so we both have knowledge over the true parameters of the network, as well as have the ability to see how changes of the parameters affect the result. We will focus on dynamic networks consisting of exactly *two time steps*. Meaning we have one initial network and one for the TERGM estimation. Our networks will be undirected and unweighted. As we do not use bootstrap procedures (like in `btergm`), we will also have to limit ourselves to small network sizes, as calculation time is a crucial bottleneck in our iterative approach.

Once we have the simulated network series, we will estimate the parameters of the chosen model with both our iterative approach and the traditional TERGM estimation assuming homogeneity. We will then compare the two results with the true parameters behind the data generation process.

6.1 Samples of Heterogeneous Networks

To simulate our networks, we will need to choose network statistics and their parameters. In addition, we need to pick a network size n (n being the number of nodes) and decide how *heterogeneous* we want our networks to be. Incorporating the heterogeneity will be straight forward, as the implementations of the `statnet` package we are using for simulation allows us to specify a `sociality` parameter for each node. We will randomly generate those values based on a normal distribution with $\mu = 0$ and $\sigma = \sqrt{h}$, with h being our heterogeneity-parameter.

The simulation of the initial network will be through an ERGM. This network will then be passed on as starting point for the TERGM simulation of the second network. The TERGM simulation will use the same parameters as the ERGM simulation. As ERGM also requires a starting point to simulate the equilibrium

state we are looking for, we will hand it an empty network. Unfortunately, the TERGM estimation algorithm does not provide a step-wise option like the ERGM package, so each iteration *all* sub-iterations within the TERGM estimation will be executed until convergence. With no option shortcut this process, we are limited to small network sizes to keep calculation times feasible. On top of that, with long execution time, it would be hard to identify degeneracy issues (which happen a lot through poor choice of parameters) preventing any convergence early on, meaning we would not be able to see if the computation is making progress or already stuck until a long amount of time has passed. Table 6.1 shows the parameter settings chosen for each dynamic network series. Finding a good combination of statistics is not trivial, as they may be correlated, causing degeneracy. Furthermore, as already mentioned, the statistics may also face stability issues. A very simple, although admittedly not very sophisticated, remedy to mitigate these issues is the use of very few parameters per model. In this case, we are using the count of all *edges (ties)* present in the network, and the Geometrically-Weighted Edgewise Shared Partnerships (*gwesp*) statistic, which models triad closure and is calculated as

$$w = e^\alpha \sum_{i=1}^{n-2} \{1 - (1 - e^{-\alpha})^i\} p_i,$$

where α is the *decay* parameter (in our case $\alpha = 0.6$), and p_i is the number of actor pairs with exactly i shared (edgewise) partners. n is the number of nodes in the network. This statistic was chosen due to its stabilizing effect.

Parameter	Sample 1	Sample 2	Sample 3	Sample 4
<i>network size (n)</i>	25	25	25	25
<i>heterogeneity (h)</i>	0.6	0.6	0.01	0.6
<i>edges</i>	-2.5	-2.5	-2.5	-2.5
<i>gwesp</i>	0.2	0.2	0.2	-0.5
<i>mean age</i> ¹	0.2	-0.01	-0.01	-0.01

Table 6.1: Parameters

Even though the network statistics with a temporal dimension of the used library cannot be used for estimation yet, we used `mean age` for the simulation of the

¹This statistic equals the average amount of time elapsed since formation of all ties present in the network.

sample networks, to ensure we receive two consecutive snapshots of a single time series, rather than two samples of unrelated networks who simply share the same parameters.

6.2 Parameter Estimation

The actual estimation is where our iterative algorithm comes in. We start by extracting the change statistics for each time step, and feeding the data to our GLMM to estimate the random effects with an intercept using the `mgvc` package. Getting the change statistics of ERGM-terms is straight forward, as the `statnet` package takes care of that. As for the TERGM-terms, however, change statistics have to be calculated manually from the network matrices, as this feature appears to be missing at the time of this writing.

Once we have the random effects, we loop between...

1. ..using them as offset for the `sociality` parameter to fit a TERGM, and
2. ..feeding the estimated network parameters back to the GLMM to update the random effects

..either until it converges or until it reaches a set maximum number of iterations.

6.3 Comparison with traditional Approach

As the goal of this work is not to find a new, more convoluted approach to a problem that has already been solved, but to provide a remedy for situations not having been accounted for, we need to compare its performance to the traditional approach. Thus, we will estimate the selected parameters of the simulated network series with a plain TERGM estimation assuming homogeneity. For networks constructed with small values for h , we would expect the results of our iterative estimation and a normal TERGM fit to be more or less identical. While for high values for h we would expect our approach to yield more accurate results, justifying the additional effort when fitting the model.

6.4 Results

Plots

In addition to comparing the estimated parameters with the *true* values used for the simulation, we will run a batch of simulations based on the estimated parameters, and compare their features to the original network. With our time series consisting of only two time steps, we will only simulate the second step, conditional on the same initial network as used for the original data.

1. We will extract various network statistics (both the ones used in the models for generation and estimation, and statistics not captured in our models) from the simulations and display the results in boxplots. The plotted statistics are
 - **edges** equal to the number of edges (i.e. nonzero values) in the network (for undirected networks, equal to $kstar(1)$)
 - **kstar(2)** number of 2-stars, where a k-star is defined to be a node n_0 and a set of k different nodes $\{O[1], \dots, O[k]\}$ such that the ties $\{n_0, O[i]\}$ exist for each $i = 1, \dots, k$
 - **gwesp** equal to the geometrically weighted edgewise (not dyadwise) shared partner distribution with decay parameter $\alpha = 0.6$
 - **nsp(2)** equals the number of dyads that do not have an edge in the network with exactly 2 shared partners
 - **kstar(3)** number of 3-stars
 - **dsp(3)** equals the number of dyads in the network with exactly 3 shared partners

The respective values for the original network (blue dot) are added for reference (if they are missing, this means they are beyond the scope of the boxplots).

2. Since the focus is on heterogeneous networks, we will also plot the affinities (true parameters of the latent variable are sky blue) of each node, in order to compare their actual, realized number of degrees (normalized). Again, the values of the original network series are added for reference, as this is the only data available for estimation (The *violet* marker represents the initial

network of the time series used for both original data generation and the simulations based on estimated parameters represented by the boxplots, and the *grey* marker represents the second network of the original time series.). This plot will be generated twice, once for a model estimated with our iterative approach, and once for a model estimated with a classic TERGM estimation assuming homogeneity.

Non-negative Mean Age

As our time series only consists of two networks, both the formation of a new tie and the dissolution of an existing tie would shrink the **mean age** of all ties. Therefore, non-negative parameter values discourage any changes in the network. Nevertheless, the strength of its effect on the network structure proved admittedly unexpected. Even though their illustrations are optically slightly distorted, they are completely identical. We decided to include the full analysis results of these 'twins' anyway as reference, to see how our algorithm behaves, even though this kind of analysis would not be conducted on a time series including only two time steps in real world scenarios, if these are identical.

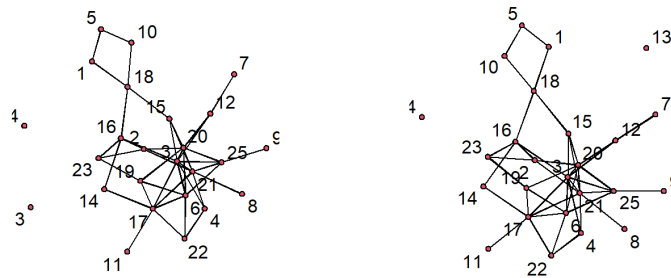


Figure 6.1: Networks simulated with positive mean age

Both the parameters and the variance (h) of the estimated random effects converged in less than 5 iterations. The results can be seen in table 6.2. Even though

this scenario is a distorted edge case, the estimation is not too far off, and slightly better than a classic TERGM estimation. The heterogeneity of the network, however, was not recognized by our algorithm.

Parameter	True Value	Iterative Estimation	Traditional Estimation
<i>heterogeneity (h)</i>	0.6	0.04	"0"
<i>edges</i>	-2.5	-2.5	-2.5
<i>gwesp</i>	0.2	0.4	0.5

Table 6.2: Estimated Parameters

As expected, the *edges* and *gwesp* of the simulations based on the estimated values reflect the original values (see figure 6.2). The other network statistics seem reasonably replicated as well, with only *dsp(3)* being bigger than in the original data.

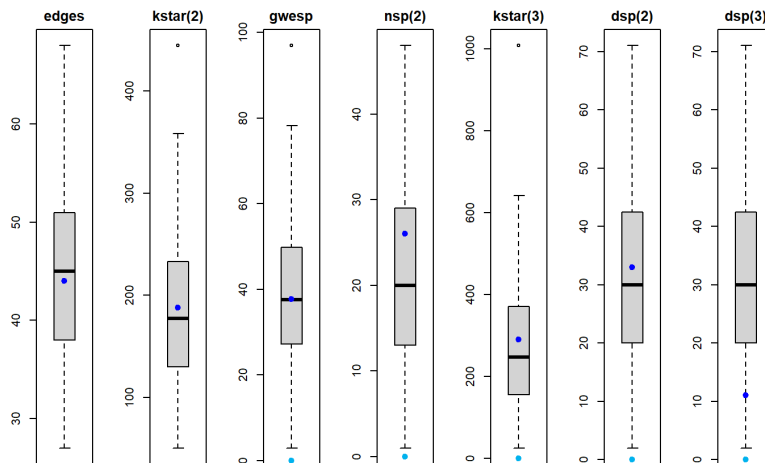


Figure 6.2: Results of simulations based on estimated parameters (blue dots represent original data)

A look at figure 6.3 shows that the individual *affinity* of each node (measured through the *degree*, the number of edges containing the node) was replicated quite well in the simulations, despite the severely underestimated heterogeneity (sky-blue dots: true affinity. rhombus: realized affinity (actual degrees) in original data).

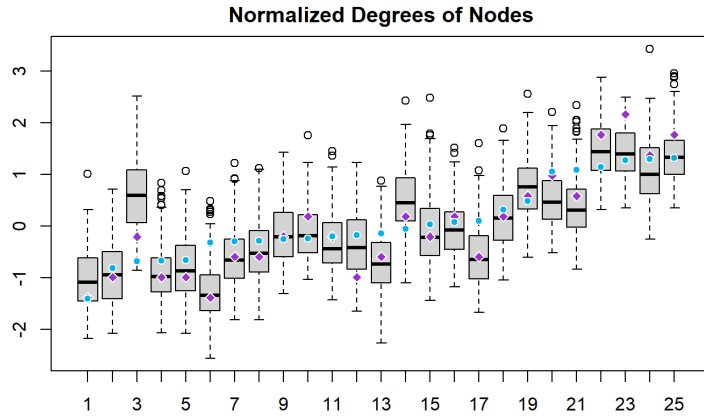


Figure 6.3: Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated with random effects

A look at 6.4, however, shows that the classic TERGM estimation, assuming homogeneity, yielded a very similar result. This is not surprising, as we used the same initial network for the second time step. So, the heterogeneity captured is basically the heterogeneity already contained in the initial network.

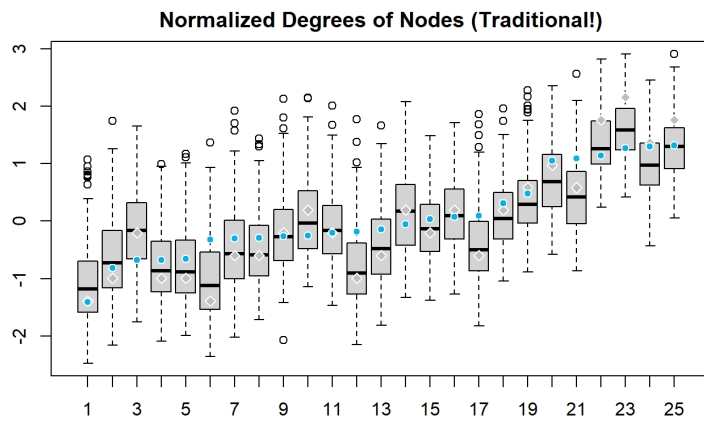


Figure 6.4: Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated without random effects (traditional approach)

Negative Mean Age

Interestingly, as soon as the **mean age** parameter is set below zero, the two time-steps look quite different (see figure 6.5). And the actual value of the parameter does not seem to make much of a difference here (i.e. if it's set to -0.00001 or -2.0). Furthermore, no ties are dissolved. Only ties added. As a consequence, both networks can indeed be considered as two consecutive snapshots from a time series, even though they visually look like two unrelated networks at first glance. A χ^2 -test formally corroborates a connection between the networks (far beyond identical *affinities* of their individual nodes towards forming ties).

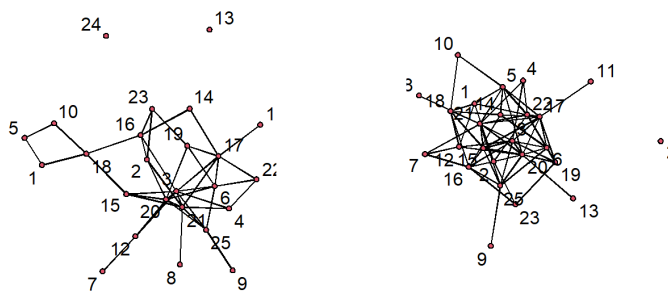


Figure 6.5: Simulated network series

Parameter	True Value	Iterative Estimation	Traditional Estimation
<i>heterogeneity (h)</i>	0.6	0.27	"0"
<i>edges</i>	-2.5	-1.9	-2.4
<i>gwesp</i>	0.2	0.3	0.6

Table 6.3: Estimated Parameters

Again, both the estimations for the parameters and for the variance h converged in less than 5 iterations. The estimated parameters now look quite different between the two approaches (see table 6.3). While the iterative approach made a more accurate prediction of *gwesp*, it is quite off when it comes to *edges*.

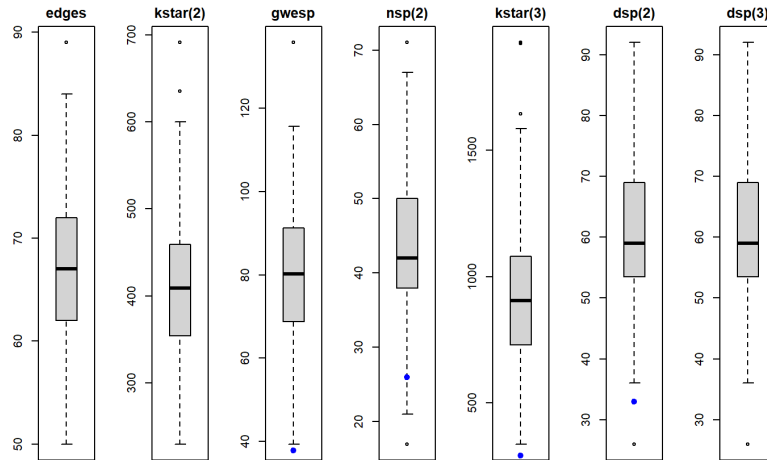


Figure 6.6: Results of simulations based on estimated parameters (blue dots represent original data)

An assessment of the estimated values can be seen in figure 6.6. Most of the original values are not even within the displayed scope. A juxtaposition of figure 6.7 and 6.8 reveals that the affinities (/random effects) of the nodes have been severely misjudged. Maybe the initial network's structure was a bit atypical for the respective affinities, leading to strong bias in the iterative approach. While the classic approach, ignoring the affinity-distribution completely, was probably 'saved' by regression to the mean.

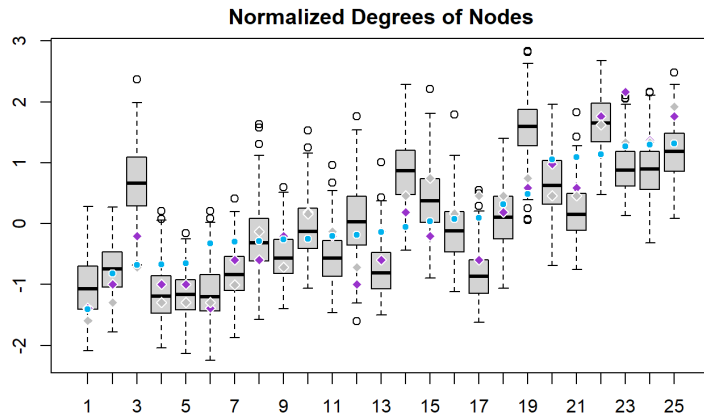


Figure 6.7: Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated with random effects

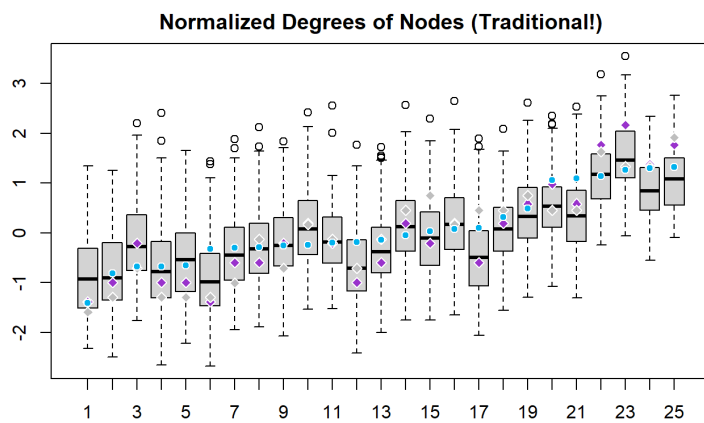


Figure 6.8: Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated without random effects (traditional approach)

Low Heterogeneity

One criterion for success of this approach would be results similar to the traditional TERGM estimation when heterogeneity is very low, meaning the homogeneity assumption inherent in ERGMs and TERGMs is fulfilled. To test this special case, we set $h = 0.01$ (figure 6.9).

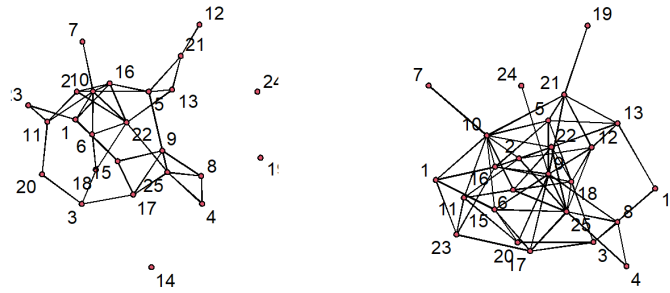


Figure 6.9: Simulated network series (homogeneous nodes)

Parameter	True Value	Iterative Estimation	Traditional Estimation
<i>heterogeneity (h)</i>	0.01	0.04	"0"
<i>edges</i>	-2.5	-1.9	-2.0
<i>gwesp</i>	0.2	0.3	0.4

Table 6.4: Estimated Parameters

Table 6.4 shows that the iterative approach recognized the low heterogeneity. Nevertheless, the estimation for the *edges* parameter was again quite off. This time, however, for the classic TERGM estimation as well.

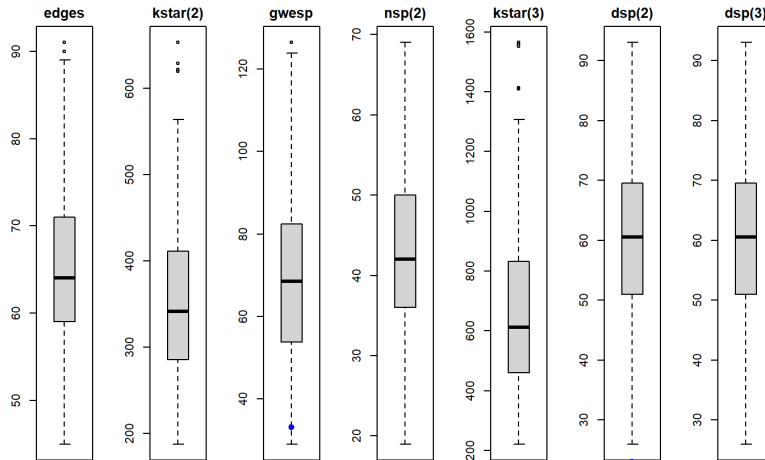


Figure 6.10: Results of simulations based on estimated parameters (blue dots represent original data)

The resulting simulations don't have much in common with the original networks (see 6.10). But, as expected, figures 6.11 and 6.12 show only very little difference between both approaches. In conclusion, the results of this parameter-combination suggest, that for low heterogeneity, the new approach, which models potential heterogeneity, does not create artifacts if used in homogeneous networks.

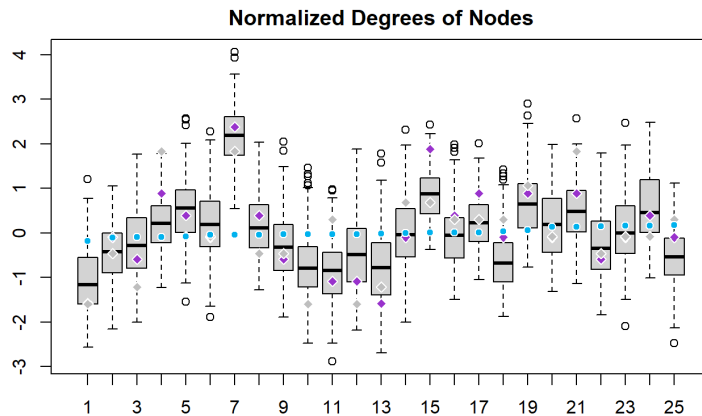


Figure 6.11: Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated with random effects

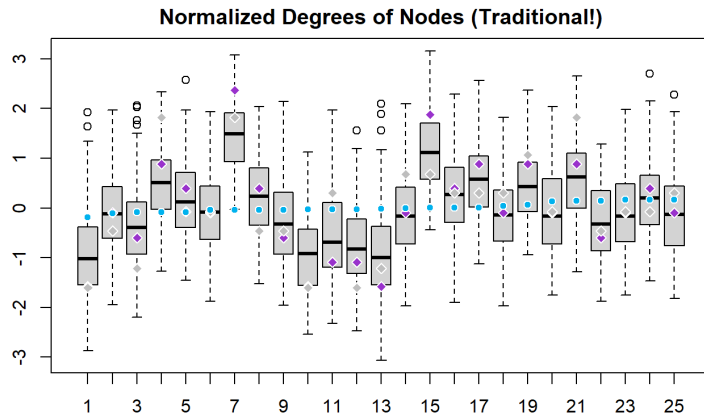


Figure 6.12: Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated without random effects (traditional approach)

Negative GWESP

In another attempt, the *gwesp* parameter was changed to a negative value. In this scenario, the second time step, again, contains all the edges present in the previous time step. Convergence was again achieved within 5 iterations.

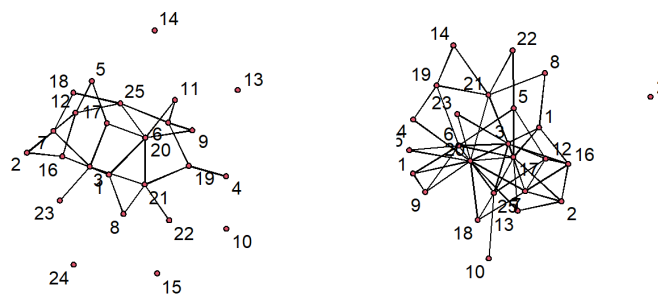


Figure 6.13: Simulated network series (negative gwesp)

Parameter	True Value	Iterative Estimation	Traditional Estimation
<i>heterogeneity (h)</i>	0.6	0.08	"0"
<i>edges</i>	-2.5	-2.0	-2.2
<i>gwap</i>	-0.5	0.27	0.37

Table 6.5: Estimated Parameters

Table 6.5 shows that the pattern continues. The estimation for *gwap* is closer, while the *edges* estimation is more off than in the classic TERGM estimation. None of the estimations, however, reached a negative value for *gwap*.

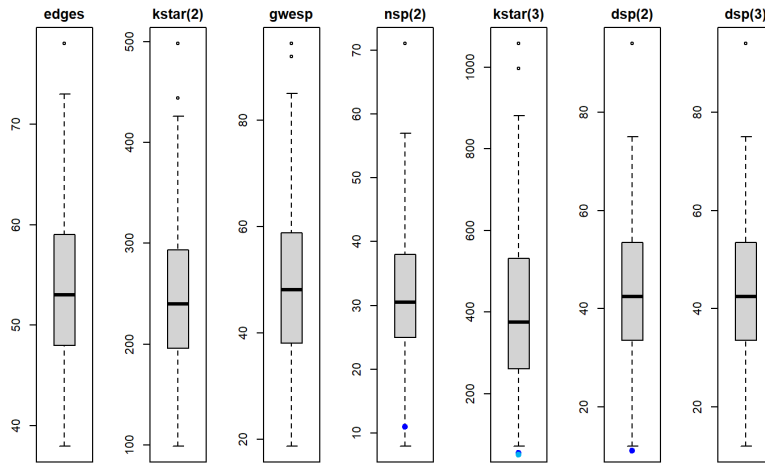


Figure 6.14: Results of simulations based on estimated parameters (blue dots represent original data)

The resulting simulations, again, do not have much in common with the original networks (see 6.14). And the estimated *affinities* of the nodes seem more chaotic and capricious in 6.15, than in its traditional counterpart 6.16.

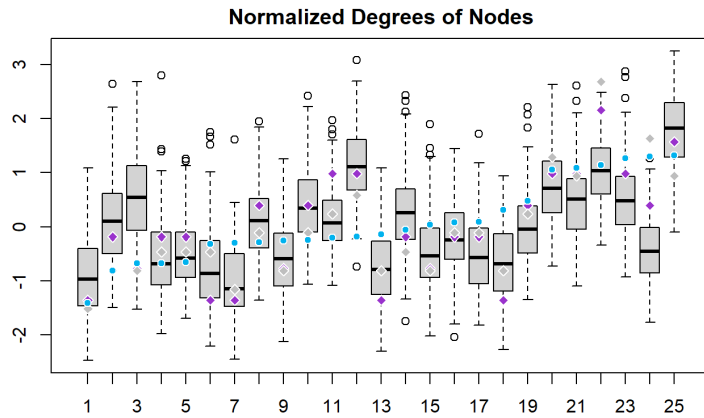


Figure 6.15: Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated with random effects

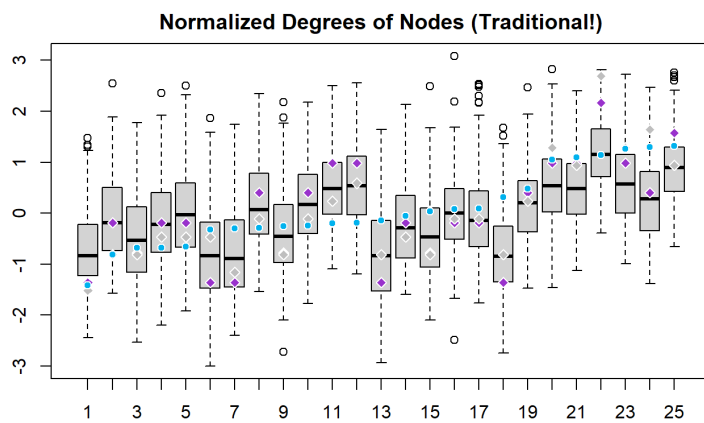


Figure 6.16: Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated without random effects (traditional approach)

7 Conclusion

We have shown that we can account for unobserved node-specific heterogeneity by including node-specific random effects in the TERGM. In our test-cases, however, this did not result in a significant improvement compared to a classic TERGM estimation. Maybe this changes for sufficiently large networks (or a network series with sufficiently many time steps). As this will become computationally very expensive, a shift to the bootstrap implementation may be necessary. This, on the other hand, may leave a gap with moderately sized networks, as the bootstrap implementation may require big network sizes, and more than two time steps. Another question that remains open is the effectiveness of the presented approach for networks with a heterogeneity parameter greater than one. So far, testing bigger values was not possible due to computational and degeneracy problems, which will require further studies.

Declaration of Authorship

I hereby confirm that I have written the present thesis independently and without illicit assistance from third parties and using solely the aids mentioned.

Munich, March 17, 2022

(Thomas Eder)

A List of Figures

3.1	Dependencies between network models by [Goldenberg et al., 2010], expanded by [Fritz et al., 2019]	5
3.2	Effect of given parameters to resulting network structure	8
3.3	Effect of the given linear combination of resulting network structure	8
3.4	Visualization of the STERGM concept	16
4.1	Attempts of classification with a linear model	23
4.2	Classification with a logistic regression model	24
6.1	Networks simulated with positive mean age	41
6.2	Results of simulations based on estimated parameters (blue dots represent original data)	42
6.3	Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated with random effects	43
6.4	Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated without random effects (traditional approach)	43
6.5	Simulated network series	44
6.6	Results of simulations based on estimated parameters (blue dots represent original data)	45
6.7	Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated with random effects	46
6.8	Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated without random effects (traditional approach)	46
6.9	Simulated network series (homogeneous nodes)	47

6.10	Results of simulations based on estimated parameters (blue dots represent original data)	48
6.11	Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated with random effects	48
6.12	Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated without random effects (traditional approach)	49
6.13	Simulated network series (negative gwesp)	49
6.14	Results of simulations based on estimated parameters (blue dots represent original data)	50
6.15	Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated with random effects	51
6.16	Degrees (normalized) of individual nodes, across multiple simulated networks - Based on parameters estimated without random effects (traditional approach)	51

B List of Tables

6.1	Parameters	38
6.2	Estimated Parameters	42
6.3	Estimated Parameters	44
6.4	Estimated Parameters	47
6.5	Estimated Parameters	50

C Bibliography

- [Box-Steffensmeier et al., 2018] Box-Steffensmeier, J. M., Christenson, D. P., and Morgan, J. W. (2018). Modeling unobserved heterogeneity in social networks with the frailty exponential random graph model. *Political Analysis*, 26(1):3–19.
- [Breslow and Clayton, 1993] Breslow, N. E. and Clayton, D. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88:9–25.
- [Caimo and Friel, 2011] Caimo, A. and Friel, N. (2011). Bayesian inference for exponential random graph models. *Social Networks*, pages 41–55.
- [Caimo and Friel, 2014] Caimo, A. and Friel, N. (2014). Bergm: Bayesian exponential random graphs in r. *Journal of Statistical Software*, 61(2):1–25.
- [Desmarais and Cranmer, 2012] Desmarais, B. and Cranmer, S. (2012). Statistical mechanics of networks: Estimation and uncertainty. *Physica A: Statistical Mechanics and its Applications*, 391(4):1865–1876.
- [Desmarais and Cranmer, 2011] Desmarais, B. A. and Cranmer, S. J. (2011). Micro-level interpretation of exponential random graph models with application to estuary networks. *Political Methods: Quantitative Methods eJournal*.
- [Duijn et al., 2009] Duijn, M., Gile, K., and Handcock, M. (2009). A framework for the comparison of maximum pseudo likelihood and maximum likelihood estimation of exponential family random graph models. *Social networks*, 31:52–62.
- [Duijn et al., 2004] Duijn, M., Snijders, T., and Zijlstra, B. (2004). p2: A random effects model with covariates for directed graphs. *Statistica Neerlandica*, 58.
- [Fahrmeir et al., 2013] Fahrmeir, L., Kneib, T., Lang, S., and Marx, B. (2013). *Regression: Models, Methods and Applications*.

- [Firth, 1993] Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1):27–38. <https://www2.stat.duke.edu/~scs/Courses/Stat376/Papers/GibbsFieldEst/BiasReductionMLE.pdf>, last accessed 13 February 2022.
- [Fritz et al., 2019] Fritz, C., Lebacher, M., and Kauermann, G. (2019). Tempus volat, hora fugit: A survey of tie-oriented dynamic network models in discrete and continuous time. *Statistica Neerlandica*, 74.
- [Gilks et al., 1996] Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). Introducing markov chain monte carlo. *Markov Chain Monte Carlo in Practice*.
- [Goldenberg et al., 2010] 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.
- [Granovetter, 1973] Granovetter, M. S. (1973). The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380.
- [Handcock, 2003] Handcock, M. (2003). Assessing degeneracy in statistical models of social networks. *Journal of the American Statistical Association*, 76:33–50.
- [Handcock et al., 2014] Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., Krivitsky, P. N., Morris, M., Wang, L., Li, K., Bender-deMoll, S., Klumb, C., Bojanowski, M., Bolker, B., and Schmid, C. (2014). Ergm terms cross-reference. <https://cran.r-project.org/web/packages/ergm/vignettes/ergm-term-crossRef.html>, accessed on 14.2.2022.
- [Handcock et al., 2021] Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., Krivitsky, P. N., Morris, M., Wang, L., Li, K., Bender-deMoll, S., Klumb, C., Bojanowski, M., Bolker, B., and Schmid, C. (2021). Introduction to exponential-family random graph models with ergm. <https://cran.r-project.org/web/packages/ergm/vignettes/ergm.pdf>, accessed on 14.2.2022.
- [Hanneke et al., 2010] Hanneke, S., Fu, W., and Xing, E. P. (2010). Discrete temporal models of social networks. *Electronic Journal of Statistics*, 4(none):585 – 605.

- [Holland and Leinhardt, 1981] Holland, P. W. and Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50.
- [Hunter and Handcock, 2006] 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 et al., 2008] Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M., and Morris, M. (2008). ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, 24(3):1–29.
- [Kauermann, 2020] Kauermann, G. (2020). Exponential random graph models and beyond. Guest Lecture at Ludwig-Maximilians-Universität München.
- [Kauermann et al., 2019] Kauermann, G., Biagini, F., and Meyer-Brandis, T. (2019). *Network Science*. Springer International Publishing, 1 edition.
- [Kevork, 2017] Kevork, S. (2017). Separable temporal exponential random graph models. Master’s thesis, Ludwig-Maximilians-Universität München. https://epub.ub.uni-muenchen.de/77445/1/MA_Kevork.pdf, last accessed 13 February 2022.
- [Kevork and Kauermann, 2021] Kevork, S. and Kauermann, G. (2021). Iterative estimation of mixed exponential random graph models with nodal random effects. *Network Science*, 9(4):478–498.
- [Kevork and Kauermann, 2022] Kevork, S. and Kauermann, G. (2022). Bipartite exponential random graph models with nodal random effects. *Social Networks*, 70:90–99.
- [Kolaczyk and Csardi, 2020] Kolaczyk, E. C. and Csardi, G. (2020). *Statistical Analysis of Network Data with R*. Springer Nature Switzerland AG, 2 edition.
- [Koskinen et al., 2015] Koskinen, J., Caimo, A., and Lomi, A. (2015). Simultaneous modeling of initial conditions and time heterogeneity in dynamic networks: An application to foreign direct investments. *Network Science*, 3:58–77.

- [Krivitsky and Handcock, 2014] Krivitsky, P. and Handcock, M. (2014). A separable model for dynamic networks. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 76:29–46.
- [Krivitsky, 2012] Krivitsky, P. N. (2012). Modeling tie duration in ergm-based dynamic network models. *Faculty of Engineering and Information Sciences - Papers: Part A.*, 2798. <https://ro.uow.edu.au/cgi/viewcontent.cgi?referer=&httpsredir=1&article=3807&context=eispapers>, accessed on 14.2.2022.
- [Krivitsky, v3] Krivitsky, P. N. (v3). Documentation of tergm package (r). <https://www.rdocumentation.org/packages/tergm/versions/3.7.0/topics/ergm-terms>, accessed on 14.2.2022.
- [Krivitsky and Goodreau, 2016] Krivitsky, P. N. and Goodreau, S. M. (2016). Stergm - separable temporal ergms for modeling discrete relational dynamics with statnet. <https://mran.microsoft.com/snapshot/2016-10-05/web/packages/tergm/vignettes/STERGM.pdf>, accessed on 14.2.2022.
- [Krivitsky et al.,] Krivitsky, P. N., Morris, M., Handcock, M. S., Butts, C. T., Hunter, D. R., Goodreau, S. M., Klumb, C., and de Moll and, S. B. Temporal exponential random graph models (tergms) for dynamic network modeling in statnet. https://statnet.org/Workshops/tergm/tergm_tutorial.html#EGMME, accessed on 14.2.2022.
- [Leifeld and Cranmer, 2019] Leifeld, P. and Cranmer, S. J. (2019). A theoretical and empirical comparison of the temporal exponential random graph model and the stochastic actor-oriented model. *Network Science*, 7(1):20–51.
- [Leifeld et al., 2018] Leifeld, P., Cranmer, S. J., and Desmarais, B. A. (2018). Temporal exponential random graph models with btergm: Estimation and bootstrap confidence intervals. *Journal of Statistical Software*, 83(6):1–36.
- [Molnar, 2019] Molnar, C. (2019). *Interpretable Machine Learning*. <https://christophm.github.io/interpretable-ml-book/>, last accessed 13 February 2022.
- [Molontay and Nagy, 2019] Molontay, R. and Nagy, M. (2019). Two decades of network science: As seen through the co-authorship network of network scientists. In

Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '19, page 578–583, New York, NY, USA. Association for Computing Machinery. <https://arxiv.org/pdf/1908.08478.pdf>, last accessed 15.3.22.

- [Schweinberger, 2011] Schweinberger, M. (2011). Instability, sensitivity, and degeneracy of discrete exponential families. *Journal of the American Statistical Association*, 106(496):1361–1370. PMID: 22844170.
- [Snijders, 2002] Snijders, T. (2002). Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3.
- [Snijders, 2017] Snijders, T. (2017). Stochastic actor-oriented models for network dynamics. *Annual Review of Statistics and Its Application*, 4.
- [Snijders et al., 2010] Snijders, T., Bunt, G. G., and Steglich, C. (2010). Introduction to actor-based models for network dynamics. *Social Networks*, 32:44–60.
- [Thiemichen, 2016] Thiemichen, S. (2016). *Extensions of Exponential Random Graph Models for Network Data Analysis*. PhD thesis, Ludwig-Maximilians-Universität München. https://edoc.ub.uni-muenchen.de/19638/1/Thiemichen_Stephanie.pdf, last accessed 13 February 2022.
- [Thiemichen and Kauermann, 2016] Thiemichen, S. and Kauermann, G. (2016). Stable exponential random graph models with non-parametric components for large dense networks. *Social Networks*, 49.
- [van der Pol, 2019] van der Pol, J. (2019). Introduction to network modeling using exponential random graph models (ergm): Theory and an application using r-project. *Computational Economics*, 54.
- [Watts and Strogatz, 1998] Watts, D. and Strogatz, S. (1998). Collective dynamics of small world networks. *Nature*, 393:440–2.
- [Wood, 2006] Wood, S. (2006). *Generalized Additive Models: An Introduction With R*, volume 66.

[Wood, 2021] Wood, S. (2021). Mixed gam computation vehicle with automatic smoothness estimation. <https://cran.r-project.org/web/packages/mgcv/mgcv.pdf>, accessed on 14.2.2022.