Verena Bauer

New Approaches in Network Data Analysis

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Eingereicht am 18. September 2019

Verena Bauer

New Approaches in Network Data Analysis

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Eingereicht am 18. September 2019

Erster Berichterstatter: Prof. Dr. Göran Kauermann
Zweiter Berichterstatter: Prof. Dr. Christoph Stadtfeld
Dritter Berichterstatter: Prof. Dr. Christian Heumann

Tag der Disputation: 06. Dezember 2019

Acknowledgments

I would like to thank Prof. Dr. Göran Kauermann for his support, supervision and positive attitude during the problems we sometimes had submitting a paper. Thank you for giving me the opportunity working in this exciting research field and for encouraging me throughout the years. I appreciate a lot that you gave me the time and support being with my little boy.

I thank Prof. Dr. Christoph Stadtfeld for taking the time to come to Munich and being my second reviewer.

I would like to thank Prof. Dr. Christian Heumann for serving as a reviewer and for the good time during choir practice. Furthermore, thank you for supervising me in my Bachelor thesis and consulting project, which encouraged me to continue.

I would also like to thank

... Prof. Dr. Volker Schmid for serving as chair for my disputation and Prof. Dr. Thomas Augustin for being available as a backup reviewer.

... Prof. Dr. Helmut Küchenhoff for hiring me at the StaBLab and introducing me to the work at the university.

... Eva and Christoph for sharing the office with me in a calm and friendly atmosphere.

... Gunther, Tina, Linda and Andi for the fun lunch and coffee breaks, vacations and good times on mountain tops. Without you I would not have taken this way.

... my colleagues and ex-colleagues from the chair – Ben, Cornelius, Marc, Michael, Michi, Sevag, Schorschi, Ludwig, and Stephi for all the fun meetings, organizations of network workshops, exchanges of stuff and travels to conferences.

... Jacky, Cornelius and Franz for proof-reading my work and giving me valuable input.

Finally and most importantly, I would like to thank my family and friends, especially my parents, husband, sister and parents-in-law for their immense support throughout the years. Without your help – especially for caring for Moritz – I could not have finished my PhD.

Last but not least I thank Moritz, my little sunshine, for waking me up every day with a big smile and making each day to something special.

Summary

This thesis introduces two extensions to statistical approaches improving modeling and estimation in the field of network data analysis. The first contributing publication focuses on cross-sectional networks based on Markov graphs, whereas the second takes the evolution of networks with dynamical structure into account. Analyzing network data is challenging in terms of modeling and computation due to large and dependent data sets. The dissertation starts with an overview of network data in general and gives an introduction to the well-known model framework of exponential random graphs models with its dependence assumptions, estimation routines, challenges, and solution approaches. At the end of the introduction, main ideas of dynamic network models, the profile likelihood approach for multivariate counting processes for network data, and the analogy of the Cox proportional hazards and Poisson model with semiparametric estimation are presented.

The first part of this work proposes an extension for sampling Markov graphs as a subclass of exponential random graph models in parallel to accelerate computation time in simulation-based routines. The estimation of network models, especially of large networks, is demanding and requires Markov chain Monte Carlo simulations. This publication recommends to exploit the conditional independence structure in networks to make use of parallel draws. This idea is applied to a large ego network of Facebook friendships, where an additional log transformation of network statistics accounts for degeneracy problems. This extension is implemented in the open source **R** package **pergm**, available on GitHub and a short introduction to the main functionalities is elaborated on in the thesis.

The second part of this work focuses on dynamic networks. In comparison to cross-sectional networks from the first part, the development and application of longitudinal network data concentrates on modeling changes of relations. Therefore, a profile likelihood approach to model time-stamped event data is combined with a semiparametric approach including covariates built from network history. This flexible semiparametric approach is applicable to large networks because standard software can be used for estimation due to the analogy of the Cox proportional hazards and Poisson model with artificial data structure. This extended method is applied to patent collaboration data of patents submitted jointly by inventors with German residency between 2000 and 2013. Based on penalized smoothing techniques, we include time dependent network statistics and exogenous covariates to capture internal and external effects.

Zusammenfassung

In dieser Arbeit werden zwei Erweiterungen zu statistischen Ansätzen zur Verbesserung der Modellierung und Schätzung im Bereich der Analyse von Netzwerkdaten vorgestellt. Der erste Teil der Arbeit konzentriert sich auf statische Netzwerke, welche auf Markov Graphen basieren, während der zweite Teil dynamische Strukturen von Netzwerken berücksichtigt. Auf Grund der Größe der Datensätze und ihrer abhängigen Struktur ist die Modellierung und die damit verbundenen computationalen Aspekte eine Herausforderung. Die Dissertation beginnt mit einem generellen Überblick über Netzwerkdaten und gibt eine Einführung in die bekannte Modellklasse der Exponential Random Graph Modelle mit ihren Abhängigkeitsannahmen, Schätzroutinen, Herausforderungen und Lösungsansätzen. Abschließend werden Ideen zu dynamischen Netzwerkmodellen, der Profile-Likelihood-Ansatz für multivariate Zählprozesse für Netzwerkdaten und die Analogie zwischen dem Cox-Proportional-Hazards- und Poisson-Modell mit semiparametrischer Schätzung vorgestellt.

Im ersten Teil dieser Arbeit wird eine Erweiterung für das Simulieren von Markov Graphen als Unterklasse der Exponential Random Graph Modelle vorgeschlagen, indem die simulationsbasierten Routinen parallelisiert werden und somit die Rechenzeit verkürzt wird. Die Schätzung von Netzwerkmodellen, insbesondere von großen Netzwerken, ist anspruchsvoll und benötigt Markov-Chain-Monte-Carlo Simulationen. In dieser Arbeit wird empfohlen, die konditionale Unabhängigkeitsstruktur in Netzwerken zur Nutzung paralleler Ziehungen zu verwenden. Diese Idee wird auf ein großes Ego-Netzwerk von Facebook-Freundschaften angewendet und zusätzlich eine Transformation der Netzwerkstatistiken durchgeführt, um das Modell und ihre Schätzung zu stabilisieren. Diese Erweiterung ist mittels der Open-Source Statistiksoftware **R** im Paket **pergm** implementiert, auf GitHub verfügbar und wird hier mit den wichtigsten Funktionalitäten kurz eingeführt.

Im zweiten Teil der Arbeit wird auf dynamische, im Vergleich zu statische Netzwerke aus dem ersten Teil, eingegangen. Die Entwicklung und Anwendung von dynamischen Netzwerkmodellen konzentriert sich auf die Modellierung von Beziehungsänderungen. Um Ereignisdaten, welche einen Zeitstempel haben, modellieren zu können, wird daher ein Profile-Likelihood-Ansatz mit glatt modellierten Kovariablen kombiniert. Diese Kovariablen werden aus der Historie des Netzwerkes gebildet. Dieser flexible, semiparametrische Ansatz kann für die Schätzung von großen Netzwerken angewandt werden, da aufgrund der Analogie zwischen dem Cox-Proportional-Hazards- und Poisson-Modell mit künstlicher Datenstruktur Standard-Software verwendet werden kann. Diese erweiterte Methode wird auf einen Datensatz angewendet, welcher die Zusammenarbeit von Patenterfindern mit deutschem Wohnsitz zwischen den Jahren 2000 und 2013 beinhaltet. In die Modellierung werden zeit-variierende Netzwerkstatistiken und exogene Kovariablen, welche interne und externe Effekte auffangen sollen, mit Hilfe von penalisierten Glättungstechniken aufgenommen.

Contents

Chapter 1

Introduction

1.1 Overview

> *"The oft-repeated statement that "we live in a connected world" perhaps best captures, in its simplicity, why networks have come to hold such interest in recent years."*
>
> – Eric D. Kolaczyk (Kolaczyk, 2009)

The field of 'social network analysis' is spread into many disciplines and roots go back to the beginning of the 20th century, where the term 'social network' is defined as a set of social actors with social interactions. In the literature the term 'network' can have various meanings not least because it is used in a variety of fields such as, e.g., computer science, social science, biology, or political science (Freeman, 2004). Often, the term 'network' describes a system of inter-connected things, but at the same time also a graph representing it (Kolaczyk, 2009). Every day we are exposed to social networks like our friendship network, working relations, or our family relationships. Not surprisingly, this interdisciplinary research area has been gaining importance over the last decades as the quote of Kolaczyk emphasizes.

The focus of statistical analysis of network data – hence the focus of this thesis – is solving the challenges of complex dependence structures of relational data in (often) high-dimensional settings. The classical statistical analysis deals with data derived from independent observations, therefore most methods are based on this assumption. The definition of networks, however, violates this assumption, as the actors are by nature dependent due to social interactions. This crucial point makes network data analysis challenging, but also very interesting. For a comprehensive introduction to statistical analysis of network data, we refer the reader to Kolaczyk (2009).

The following sections give an overview and introduction to the work presented in this thesis and summarize the most important approaches of the contributing articles. Section 1.2 introduces the notation of static and dynamic network data, followed by an overview of data examples (Section 1.3). Section 1.4 constitutes the most important concepts of exponential random graph models with their

dependence assumptions, estimation methods, degeneracy problems as well as further challenges and their solution approaches. In Section 1.5 dynamic network models are discussed and some details about the profile likelihood approach for multivariate counting processes based on the Cox model (Cox, 1972) for network data are given. A brief introduction to semiparametric estimation including covariates more flexible completes this extended approach. For estimating the model the analogy of the Cox proportional hazards and Poisson model with flexible predictor is exploited. Finally, Section 1.6 describes the main functionalities of the contributing **R** package **pergm** (Bauer, 2016).

1.2 Preface

The following notation and terms will be used in this thesis. We focus on modeling network data consisting of N nodes (also called actors or vertices) and edges (also called ties or relationships). Edges are random variables and potential links between a fixed set of nodes. The mathematical graph representation is used to represent a binary network. Let $\boldsymbol{Y} \in \{0,1\}^{N \times N}$ denote the adjacency matrix of a network with N nodes and

$$Y_{ij} = \begin{cases} 1, & \text{if an edge exists between } i \text{ and } j \\ 0, & \text{otherwise,} \end{cases}$$

with $i, j \in \{1, \ldots, N\}$. Note that in most approaches and applications self-loops are excluded, i.e., $Y_{ii} = 0 \ \ \forall\, i$, thus no ties from a node to itself are allowed. In general networks are subdivided in directed and nondirected (also called undirected) graphs, the latter results in symmetric adjacency matrices where $Y_{ij} = Y_{ji}$. Figure 1.1 visualizes an undirected toy network of four nodes and all possible tie connections (left panel). The right panel shows the corresponding adjacency matrix, where $Y_{12} = Y_{21}$ etc., which means that this is an undirected network. Classical examples of undirected

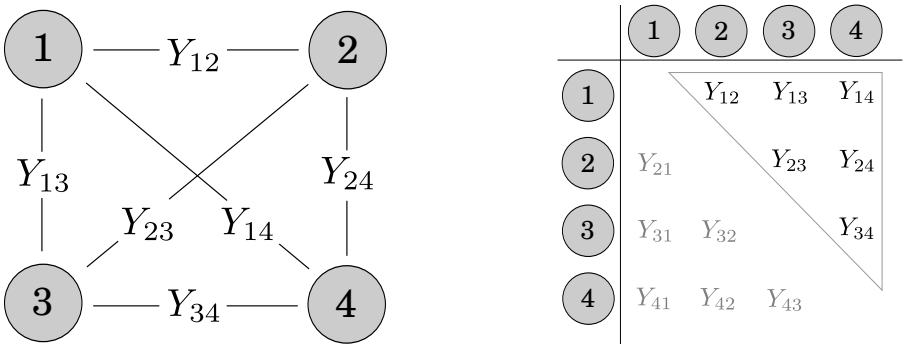


Figure 1.1: Nondirected network with four nodes and possible edges (left panel), and the corresponding adjacency matrix (right panel).

networks are friendships or collaborations where people form relationships, however, objects can also be represented by nodes. Email or traffic flows of computer networks usually result in directed graphs.

Note that the random variable $\boldsymbol{Y}$ is capitalized, while the observed or realized network is denoted by the corresponding lower case latter ($\boldsymbol{y}$). There are also extensions where valued ties are considered, however, in this work we focus on binary ties as is defined above. Table A.1 in the Appendix gives a short overview of the notation used here.
So far, the notation considers static networks that do not evolve over time. In the second contributing article of this thesis, we propose an approach for a smooth dynamic network model. Therefore, we extend the notation to time dependent networks. To be specific, let $Y_{ij}(t)$ be an entry of a matrix valued Poisson process $\boldsymbol{Y}(t) \in \mathbb{R}^{N\times N}$ with cumulated number of events of actor i and j at time t with N actors and $i, j = 1, \ldots, N$. We only observe the process at discrete time points $t_{(1)}, t_{(2)}, \ldots, t_{(m)}$ and define $Y_{ij,d} = Y_{ij}(t_{(d)})$ for the evolving process.

1.3 Data examples

This thesis deals with two different data sets: (1) a snapshot of Facebook friendships and (2) patent collaborations evolving over the timespan of 14 years. This section briefly summarizes the most important attributes of these data examples.

---

(1) Ego network of Facebook friendships
(Leskovec and Mcauley, 2012)
Number of nodes: 1 034
Number of existing edges: 26 749
Network density: 0.05

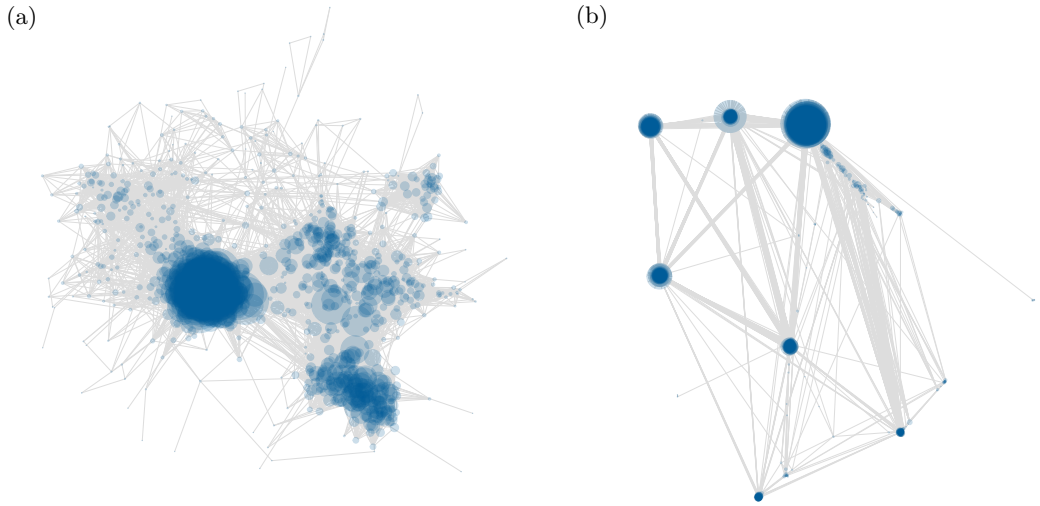


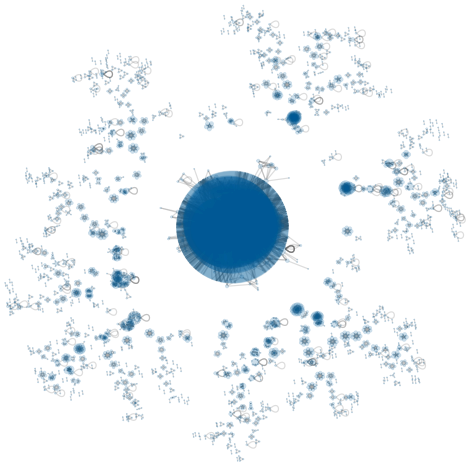
Figure 1.2: Facebook network using the force-directed layout algorithm by Fruchterman and Reingold (1991) (a) and the 'Distributed recursive (Graph) Layout' (b).

In the first article we apply our estimation approach to a subset of Facebook data collected from a survey by Leskovec and Mcauley (2012). We use a subset of one out of ten ego networks available from the Standford Large Dataset Collection (see Leskovec and Krevl, 2014). An ego network consists of a focal node ("ego") and all directly connected nodes including their ties to each other. In the following we exclude the ego node. The remaining network is shown in Figure 1.2 using two different layout algorithms, one is based on the force-directed algorithm by Fruchterman and Reingold (1991), and the other uses the 'Distributed recursive Layout' (DrL) (see Martin et al., 2008), which focuses on large graphs.

In the second article the model approach is applied to a large dynamic network of patent collaborations of inventors with at least one inventor residing in Germany. The patents were submitted to the European Patent Office (EPO) and the German Patent and Trademark Office (Deutsches Patent- und Markenamt, DPMA). The data contain the submission date on a daily basis, an index list of involved inventors, and the technological area in which the patent is submitted. Moreover, we have information about the geographic coordinates of the inventors at the time of submission.

(2) Patent collaboration network of inventors
European Patent Office and German Patent & Trademark Office
Number of inventors: (a) 3 616 and (b) 2 993
Number of patents: (a) 1 707 and (b) 2 078
Network density: (a) 0.0003 and (b) 0.0005

(a) IT-methods



(b) food chemistry

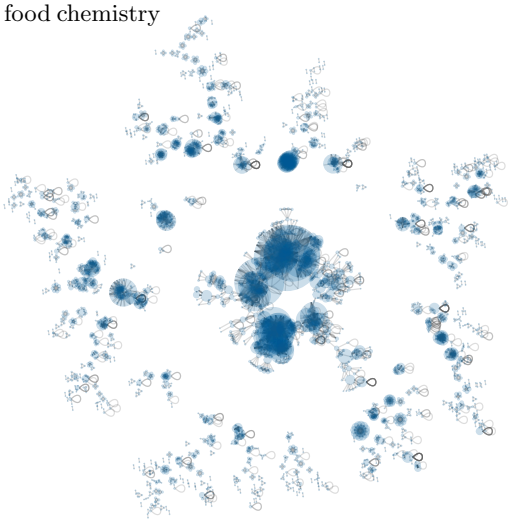


Figure 1.3: Visualization of two inventor networks aggregated over time for two technological areas. Vertex size represents nodal degree.

We focus on four technological areas and include only inventors with at least one joint patent over the time period from 2000 till the end of 2013. Let the inventors be the nodes and the joint patents the edges of our network graph.
Figure 1.3 shows two of these four networks, which are aggregated over time. Note that these networks contain loops, which belong to single invented patents of inventors with at least one joint patent. Moreover, these illustrations contain multiple edges because the same inventor pair is able to submit several joint patents at different times. Furthermore, we see a few clusters referring to patents with a higher number of inventors or inventors with a lot of joint patents. The densities of the two networks per time point vary between 0 and 0.0005.

1.4 Exponential random graph models

1.4.1 Introduction and basic concepts

The reason for finding a statistical model for network data and not just using descriptive techniques like calculating the density or centrality measurements is easy to understand. A statistical model is able to control for both, a repeating pattern in the process of forming and dissolving ties and for unstructured variability. We consider the observed network to be a realization of a set of networks with similar patterns. Our aim is to extract the most important features of the process that generated the observed network. Inference allows us to assess uncertainty, estimate the amount of contribution of multiple mechanisms, and combine network structure with attributes.
The class of Exponential Random Graph Models (ERGMs) is a promising model class for capturing structural tendencies of social networks. These statistical models are able to include complicated dependence structures due to a variety of network statistics. We can think of network statistics as summary measures or equivalent to covariables in regression models. Some of the most important structures in undirected social networks are homophily and transitivity. Homophily expresses the tendency that actors with similar properties are more likely to form a relationship. The well known statement "friends of my friends are my friends" denotes a greater propensity to make friends between two unconnected actors if there are already common friends and leads to transitivity. Frank (1991) as well as Wasserman and Pattison (1996) propose a model of the form suggested by Frank and Strauss (1986), which includes arbitrary statistics for directed and undirected graphs. This leads to the definition of exponential random graph models, where we assume $\boldsymbol{Y}$ to be random with a probability function

$$\mathbb{P}_\theta(\boldsymbol{Y} = \boldsymbol{y}) = \frac{\exp\left(\boldsymbol{\theta}' s(\boldsymbol{y})\right)}{\kappa(\boldsymbol{\theta})} \qquad \boldsymbol{y} \in \mathcal{Y}, \tag{1.1}$$

with $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$ being the parameter vector of interest and $s(\cdot) = (s_1(\cdot), \ldots, s_p(\cdot))$ the corresponding vector of network statistics. We denote the adjacency matrix of a graph with $\boldsymbol{y}$ and introduce the normalizing constant $\kappa(\boldsymbol{\theta})$, which is necessary for a probability distribution. This constant

$$\kappa(\boldsymbol{\theta}) = \sum_{\boldsymbol{y} \in \mathcal{Y}} \exp(\boldsymbol{\theta}' s(\boldsymbol{y})) \tag{1.2}$$

is a sum over all possible networks and therefore unfeasible to calculate for large networks. If we consider for example undirected graphs on N nodes, the sample space consists of $2^{\binom{N}{2}}$ elements. Therefore, estimation requires simulation based methods using numerically demanding routines (Markov chain Monte Carlo) (for more details see Section 1.4.3). In principle, we can include any network configuration in $s(\cdot)$, whereby classical structural network statistics for undirected graphs are counts of edges, 2-stars, or triangles that are visualized in Figure 1.4. An overview of the definitions of these statistics can be found in Table A.2. The 2-star statistic can be written as the sum of the upper

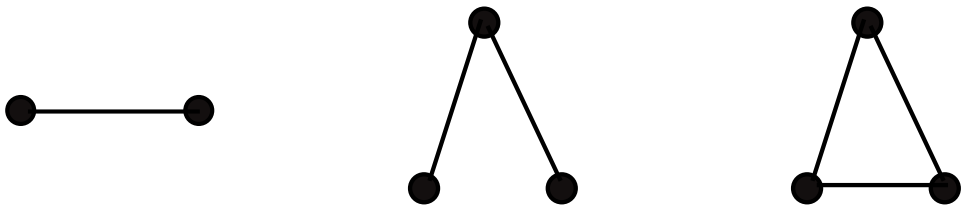


Figure 1.4: Examples of network statistics: edge, 2-star (or 2-path), and triangle.

triangle matrix of the squared adjacency

$$s_2(\boldsymbol{y}) = \sum_{i=1}^{N} \sum_{j>i} \sum_{k=1}^{N} y_{ik} y_{kj}.$$

The $\{i, j\}$-th element of the squared adjacency matrix $\boldsymbol{y}^2$ corresponds to the number of links that node i and node j have in common, in other words, the number of common friends. Another interpretation would be that the edge and 2-star statistic is the mean and variance of the degree distribution:

$$s_1(\boldsymbol{y}) = \frac{1}{2} \sum_{j \geq 1} j \cdot d_j(\boldsymbol{y}) \quad \text{and} \quad s_2(\boldsymbol{y}) = \sum_{j \geq 2} \binom{j}{2} d_j(\boldsymbol{y}),$$

where d_j is the number of nodes of degree j. Therefore, the edge parameter models the average number of edges per node and the 2-star parameter captures the variation in the number of edges of each node. The triangle statistic is defined as

$$s_3(\boldsymbol{y}) = \sum_{i=1}^{N} \sum_{j>i} \sum_{k>j} y_{ik} y_{kj} y_{ji} = \frac{1}{6} \cdot tr(\boldsymbol{y}^3),$$

where $\boldsymbol{y}^3$ corresponds to the number of paths of length three and $tr(\cdot)$ denotes the trace of a matrix. The triangle effect captures network clustering, like the phenomenon of "a friend of a friend is a friend".

In recent years extensions of these simple summary statistics have become popular. New specifications like geometrically weighted degree (GWD), alternating k-star, and alternating k-triangle statistics were developed by Snijders et al. (2006) and are introduced in more detail in Section 1.4.4.

It is also possible to include actor attributes expressing for example the already mentioned homophily effect. We can allow $s(\boldsymbol{y})$ to incorporate information of exogenous covariates on nodes and edges that are independent of the actual state of the edges. Moreover, network statistics are able to combine exogenous and structural covariates. For instance, if we consider a friendship network one could consider adding the number of common friends of the same age range, which is a combination of the graph itself and the nodal covariate age.
Most estimation routines of ERGMs are based on simulations using Markov Chain Monte Carlo (MCMC) methods. Therefore, a short note on the conditional form of the ERGM in (1.1) being expressed through

$$\text{logit}\ \ \mathbb{P}_\theta\left(Y_{ij}=1 \,\middle|\, Y_{hk}=y_{hk},\ \forall\,(h,k)\neq(i,j)\right)=\boldsymbol{\theta}'\Delta_{ij}s(\boldsymbol{y}) \tag{1.3}$$

is helpful. This conditional distribution, given the rest of the network for each dyad (pair of tie variables of two nodes), allows interpretation based on a single edge between i and j. $\Delta_{ij}s(\boldsymbol{y})$ is defined as the change in the network statistic vector, which indicates the difference in network counts (e.g. edges, 2-stars, triangles), where a tie is present at (i,j) or not, given the rest of the network. These so-called change statistics are defined by

$$\Delta_{ij}s(\boldsymbol{y})=s(\boldsymbol{y}\backslash y_{ij}, y_{ij}=y_{ji}=1)-s(\boldsymbol{y}\backslash y_{ij}, y_{ij}=y_{ji}=0),$$

that is, we toggle the edge y_{ij} between nodes i and j from present to absent. We denote the rest of the graph, except for the tie y_{ij} as $\boldsymbol{y}\backslash y_{ij}$. The change statistics of the network configurations in Figure 1.4 are easy to indicate and interpret. For example, we consider a model with these three network statistics, hence the change in the 2-star denoted by $\sum_{k\neq i,j} y_{ik}+\sum_{k\neq i,j} y_{jk}$, expresses the sum of friends of i and j, given the rest of the network. The corresponding parameter indicates the linear effect of one additional friend of i or j on the log-odds assuming all other covariates remain the same. A similar interpretation follows from the triangle change specified through $\sum_{k\neq i,j} y_{ik}y_{jk}$, which indicates the number of friends that i and j have in common. The parameter of the triangle statistic expresses the effect of an additional joint friend, given the rest stays the same.
A second important part, beside the choice of network statistics, is the assumed dependence structure. These two steps go hand in hand as the dependence assumptions imply a particular model class, which corresponds to certain parameter configurations. This is explained in the next Section 1.4.2 in more detail. For a brief introduction and more details to ERGMs we refer to Lusher et al. (2013), Robins et al. (2007b) and Snijders et al. (2006).

1.4.2 Dependence assumptions

A fundamental concept in ERGMs is the assumption of dependence between observations. Different forms of dependence have been proposed over the last decades. In the following, we give a short overview of the most important concepts. The simplest form of dependence leads to the Bernoulli graphs of Erdős and Rényi (1959). Later, Holland and Leinhardt (1981) introduced the p_1 model

class, which is based on the dyadic independence assumption for directed graphs. These simple but restrictive assumptions have been extended, leading to the exponential random graph model, which has also been called p^* model class by Frank and Strauss (1986). These authors introduced Markov dependence, which was further developed by Pattison and Robins (2002) to realization-dependent models, which assume that two tie variables are even conditionally dependent without sharing a node but with a third tie variable being present. In the class of Markov models one can distinguish between nondirected Markov random graphs, Markov models with new specifications of alternating star parameters, and directed Markov graphs. Our work is based on the first subgroup, whereby we describe the nondirected Markov random graph model in more detail below. For more details see Lusher et al. (2013).

Bernoulli assumption
In a Bernoulli graph all tie variables are assumed to be independent. The (log-) probability of a Bernoulli graph is proportional to the density, which is the weighted sum of the number of edges. The logit of the conditional probability of a tie in a Bernoulli model is given by the edge parameter. To model the adjacency matrix $\boldsymbol{Y}$ of a network, only the edge parameter, the corresponding statistic for the number of edges, and the normalizing constant are needed.

Dyad-independent assumption (for directed graphs)
A graph fulfils the dyad-independent assumption if the so-called dyads are independent of each other, which allows the tie from i to j to be independent of the tie from j to i. This assumption allows for tendencies toward reciprocation.

Markov dependence assumption
Two tie variables are assumed to be dependent, given the rest of the graph, if they share a node. The (log-) probability of a Markov random graph can be calculated – except of a constant – from different network statistics like edges, stars, and triangles.

Nondirected Markov random graph model
Nondirected Markov random graph models are based on the Markov dependence assumption, where two tie variables are assumed to be dependent, given the rest of the graph, if they share a node. With this assumption, a full Markov model only depends on the following statistics: the number of edges, the number of k-stars, and the number of triangles. These configurations are nested because higher-order statistics contain lower-order statistics. For example a triangle consists of three 2-stars and three edges. This characteristic allows statistical inference about the necessity of higher-order given lower-order configurations.

For this work, an important nested subset of the full Markov model is the triad model of Frank and Strauss (1986). This model includes the number of edges, 2-stars, and triangles, which captures both, the mean and variance of the degree distribution, and transitivity and clustering.

A limitation of these nondirected Markov random graph models is that these statistics alone do not fit well to social network data, for which social circuit models were developed. The problem with

models based on the Markov dependence assumption is that complex social structures in real data cannot be captured with these simple statistics.

1.4.3 Estimation of exponential random graph models

> *"Far better an approximate answer to the 'right' question, which is often vague, than an 'exact' answer to the wrong question, which can always be made precise."*
>
> – John W. Tukey (Tukey, 1962)

For some years after Frank and Strauss (1986) published their seminal paper on exponential random graph models, the estimation of social networks was restricted to pseudo-likelihood estimation ignoring the dependence structure. Later, Geyer and Thompson (1992) suggested a stochastic algorithm to approximate the maximum likelihood estimate in equation (1.1). Since then a variety of estimation routines were developed. Snijders (2002), for instance, proposed a stochastic approximation algorithm based on an approach of Robbins and Monro (1951), or Bayesian inference based on prior distributions for the unknown parameters were introduced. In the following we focus on the most important concepts and refer the reader to additional literature (e.g. Lusher et al., 2013; Snijders, 2002; Strauss and Ikeda, 1990; Van Duijn et al., 2009) for approaches that are not detailed here.
This section ends with an introduction to network simulation because most of the estimation approaches require simulation procedures. After fitting a model, goodness-of-fit statistics or the examination of graph features are based on making draws from the distribution of graphs. Therefore, one contributing article of this thesis focuses on parallel sampling in Markov graphs to accelerate computation.

Pseudo-likelihood estimation
The first paper of exponential random graph models of Frank and Strauss (1986) already pointed out the difficulties of the parameter estimation in ERGMs (1.1). Strauss and Ikeda (1990) came up with an idea based on a method of Besag (1974) for spatial models that ignores the assumed dependence structure. This pseudo-likelihood approach approximates the maximum likelihood estimation and is computationally equivalent to a logistic regression. For undirected networks, the upper triangle entries of the adjacency matrix correspond to the binary response and the change statistics of equation (1.3) to the predictor vector. The pseudo-log-likelihood for undirected graphs is defined as

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{N} \sum_{j>i} \ln \left(\mathbb{P}_\theta \left(Y_{ij} = y_{ij} \,\middle|\, Y_{hk} = y_{hk},\ \forall\ (h,k) \neq (i,j) \right) \right), \tag{1.4}$$

where for each potential edge in the adjacency matrix the probability, conditioned on the rest of the graph, is summed up. Finding the maximum for equation (1.4) is straightforward, though this pseudo-likelihood estimator (MPLE) is often less efficient and biased (see Van Duijn et al., 2009). Alternatives have been suggested to correct or to avoid this bias in general.

Maximum likelihood estimation: Geyer-Thompson approach

Geyer and Thompson (1992) suggested an algorithm based on the maximum likelihood principle saying that the maximum likelihood estimator (MLE) for a given model and observed data is the value that makes obtaining the observed data most likely. The method uses Markov chain Monte Carlo approximations of the likelihood function (1.1) and was first applied to ERGMs by Handcock (2003a) and extended to curved ERGMs by Hunter and Handcock (2006). This algorithm relies on simulations in order to maximize the ratio of two likelihoods instead of directly evaluating the log-likelihood

$$l(\boldsymbol{\theta}) = \boldsymbol{\theta}' s(\boldsymbol{y}) - \log\left(\kappa(\boldsymbol{\theta})\right), \tag{1.5}$$

with its intractable normalizing constant. Let $\tilde{\boldsymbol{\theta}}$ be an arbitrarily fixed parameter, we can rewrite expression (1.5) as

$$l(\boldsymbol{\theta}) - l(\tilde{\boldsymbol{\theta}}) = (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})' s(\boldsymbol{y}) - \log\left(\frac{\kappa(\boldsymbol{\theta})}{\kappa(\tilde{\boldsymbol{\theta}})}\right), \tag{1.6}$$

where the ratio of normalizing constants can be approximated by the sample mean. To be more specific, we can reformulate the ratio

$$\frac{\kappa(\boldsymbol{\theta})}{\kappa(\tilde{\boldsymbol{\theta}})} = \sum_{y \in \mathcal{Y}} \exp\left[\left(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\right)' s(\boldsymbol{y})\right] \frac{\exp\left(\tilde{\boldsymbol{\theta}}' s(\boldsymbol{y})\right)}{\kappa(\tilde{\boldsymbol{\theta}})} = \mathbb{E}_{\tilde{\theta}}\left(\exp\left[\left(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\right)' s(\boldsymbol{Y})\right]\right), \tag{1.7}$$

with the definition of expectation and formula (1.2). Hence, we approximate the expectation in (1.7) through the sample mean

$$\frac{\kappa(\boldsymbol{\theta})}{\kappa(\tilde{\boldsymbol{\theta}})} \approx \frac{1}{M} \sum_{m=1}^{M} \exp\left[\left(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\right)' s\left(\boldsymbol{Y}^{(m)}\right)\right] \tag{1.8}$$

by exploiting the law of large numbers and replacing the ratio of constants in equation (1.6). To do this, we draw a random sample $\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(M)}$ from the distribution defined by $\tilde{\boldsymbol{\theta}}$, which we obtain by Markov chain Monte Carlo simulations described at the end of this section.

A crucial point in the algorithm is that the parameter $\tilde{\boldsymbol{\theta}}$ has to be close to the true maximum likelihood estimator, in other words, $s(\boldsymbol{y})$ lies in the relative interior of the convex hull on the sample of statistics $s\left(\boldsymbol{Y}^{(1)}\right), \ldots s\left(\boldsymbol{Y}^{(M)}\right)$ (Hummel et al., 2012). A good first choice to start with is the MPLE, but typically the algorithm has to be restarted several times (see e.g. the implementation in Handcock et al., 2017). Hummel et al. (2012) developed the "stepping" algorithm to systematically move closer to the true estimator by defining pseudo-observations, which guarantee to stay in the convex hull of the sample. Combining this and a log-normal approximation maximum likelihood estimation procedures can be carried out, which leads to an improved estimation (see Hummel et al., 2012).

Stochastic approximation: Robbins-Monro algorithm

An alternative approach using maximum likelihood estimation for finding unknown parameters has been proposed by Snijders (2002). This version of the Robbins and Monro (1951) algorithm works without large samples of graphs or good starting parameters. The main aim of this stochastic approximation algorithm is to solve the moment equation. Finding the MLE or solving the moment equation

$$\mathbb{E}_\theta \left(s(\boldsymbol{Y}) \right) = s(\boldsymbol{y}) \tag{1.9}$$

is equivalent because equation (1.9) is satisfied if and only if $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimator of $\boldsymbol{\theta}$. This iterative procedure updates the parameter vector $\boldsymbol{\theta}^{(m)}$ in each iteration m through

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} - a_r \boldsymbol{D}_0^{-1} \left(s\left(\boldsymbol{y}^{(m)} \right) - s\left(\boldsymbol{y} \right) \right), \tag{1.10}$$

a Newton-Raphson-type equation, where $\boldsymbol{D}_0$ denotes a scaling matrix from an initial phase. In each update step, one sampled graph $s\left(\boldsymbol{y}^{(m)}\right)$ from the ERGM with parameter $\boldsymbol{\theta}^{(m)}$ based on MCMC methods is generated. The factor a_r guarantees a decreasing weight of the changes with increasing number of iterations m. For more details or extensions see, e.g., Snijders (2002), Lusher et al. (2013) and Okabayashi et al. (2012).

Simulation methods

A very important part in the inference of exponential random graph models in (1.1) is to sample graphs from a target distribution based on Markov chain Monte Carlo methods. Most of these models are intractable because of their normalizing constant. Nevertheless, evaluating the conditional probability of an edge, assuming the rest remains the same, is straightforward because the constant cancels out. Therefore, making draws from an ERGM is often possible and not computational expensive. The basic idea of a Monte Carlo procedure is to generate a sequence of L graphs $\boldsymbol{Y}^{(0)}, \boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(l)}, \ldots, \boldsymbol{Y}^{(L)}$ by successively updating a tie variable until the L^{th} network, with large enough L, results in a draw from the target distribution of our ERGM. There are two famous algorithms that are mostly used to simulate networks: a Metropolis-Hastings sampling algorithm (Metropolis et al., 1953; Hastings, 1970) or a Gibbs sampler (Geman and Geman, 1987), which is a special case of the first and simulates the edges based on the logit model resulting from (1.1). Note that in most cases Metropolis-Hastings algorithms converge more efficiently because the changes in the adjacency matrix are more frequent. In the following we focus on the Metropolis-Hastings algorithm because the first contributing article of this thesis is based on this method.

We start the Metropolis-Hastings sampler (see also Lusher et al., 2013; Hunter et al., 2008b) with an empty or the observed network $\boldsymbol{Y}^{(0)}$ and sequentially update edges creating L networks. In each iteration l for $l = 1, \ldots, L$, we randomly choose one dyad $y_{ij}^{(l-1)}$ for updating. The proposed network

$\boldsymbol{Y}^*$ is equal to the current graph $\boldsymbol{Y}^{(l-1)}$ but one node pair is toggled from $y_{ij}^{(l-1)}$ to $1 - y_{ij}^{(l-1)}$. We add or remove this tie with a certain acceptance probability

$$\min \left\{ 1, \frac{\mathbb{P}_\theta(\boldsymbol{Y}^*)}{\mathbb{P}_\theta(\boldsymbol{Y}^{(l-1)})} \right\}, \tag{1.11}$$

with $\mathbb{P}_\theta(\boldsymbol{Y})$ being the target distribution. This so-called "Hastings ratio" denotes the ratio of how much more likely the new proposed graph is compared to the old one. If the new proposal has a higher probability, we accept the toggle, if it has a lower probability, we accept the change with a certain probability depending on the difference. If the change is accepted, we set $\boldsymbol{Y}^{(l)} = \boldsymbol{Y}^*$, otherwise, we set $\boldsymbol{Y}^{(l)} = \boldsymbol{Y}^{(l-1)}$. After that we start again by choosing an edge to toggle till our graph is a draw from the target distribution.

Note that no normalizing constant is needed because the Hastings ratio (1.11) can be expressed as

$$\log \left\{ \frac{\mathbb{P}_\theta(\boldsymbol{Y}^*)}{\mathbb{P}_\theta(\boldsymbol{Y}^{(l-1)})} \right\} = \log \left\{ \mathbb{P}(Y_{ij} = 1 - y_{ij}^{(l-1)} | \boldsymbol{Y} \backslash Y_{ij} = y_{ij}^{(l-1)}) \right\} = \boldsymbol{\theta}' \Delta_{ij} s(\boldsymbol{y}), \tag{1.12}$$

where only the differences in the statistics that result in changing $y_{ij}^{(l-1)}$ to $1 - y_{ij}^{(l-1)}$ are needed. We have already encountered this closed form in the section on conditional distributions of ERGMs (1.3), where we called $\Delta_{ij} s(\boldsymbol{y})$ the change statistic for adding an absent tie. Here in equation (1.12), we define the change statistics as the differences in the statistics that result in toggling a tie from zero to one or from one to zero.

The fundamental principle behind Markov chain Monte Carlo simulations is that once the sampler settles into the target distribution – after a certain number of iterations ("burn-in") – the next graphs also derive from the ERGM. Therefore, the simulation chain just has to be started once. After a reasonable number of burn-in iterations the first network is simulated and after further iterations the Markov chain has 'forgotten' the last state and produces additional networks from this given distribution. These so-called "thinning" steps guarantee independent draws. If the burn-in is large enough, the algorithm is independent of the starting network.

Since inference or goodness-of-fit algorithms require us to simulate a large number of networks by generating a sequence of L graphs each, it is important to use an efficient algorithm. A good choice for speeding up the sampling algorithm is the "TNT" (tie-no tie) sampler proposed by Morris et al. (2008). The TNT sampler modifies the Metropolis-Hastings MCMC routine by not selecting the dyads to toggle randomly but with certain probabilities depending on the actual state of y_{ij}. In practice, most networks are sparse and therefore, for random samplers the probability to select an empty dyad and propose a change, which is rejected, is very high. In such cases the Markov chain often stays longer in the same state. To avoid this, the TNT sampler chooses present and absent edges with a probability of 0.5 each, which leads to a faster convergence of the Markov chain.

Further modifications of the simulation procedures to improve convergence of the Markov chains, computation time, and properties of the simulated networks exist in the literature. An idea of Snijders (2002) is to update multiple edges e.g. in form of triples or other natural groups of entries

in the adjacency matrix at once. The decision process follows the same scheme as described above but for multiple edges simultaneously. If a change is accepted, all dyads in the set are updated. The groupwise probabilities are defined analogously but for the set of all possible outcomes for the elements.

Snijders (2002) proposed as well to include "big updates" in the switching process to avoid convergence problems. The idea is to update a bigger set of edges like rows or columns based on the concept of the cluster-flipping algorithms for the Ising model (see Newman and Barkema, 1999). To extend this, the set of edges consists of the complete network graph and results in its inversion. Nevertheless, such an inversion step only occurs with a certain but small probability instead of a 'normal' step.

Referring to the quote of Tukey (1962) at the beginning, it is sometimes better to meet the challenges of using a maximum likelihood approach by approximating it and getting an answer to the 'right' question, than using the simple pseudo-likelihood estimation and getting an 'exact' answer to the wrong question.

1.4.4 Degeneracy problems

The estimation routines of exponential random graph models mentioned above that are used to solve the problem of the intractable normalization constant, all rely on simulation approaches based on Markov chain Monte Carlo methods. Thus, e.g. Snijders (2002), Snijders et al. (2006), Handcock (2003a) or Handcock (2003b) address the problem of model degeneracy, which is related to these procedures. The stationary distribution is termed degenerated if the probability distribution is concentrated on a small subset of sample space. Handcock (2003a) defines the term near-degeneracy as a distribution assigning disproportionate probability mass to a small outcome space, where the parameters lie on the boundary of the convex hull. Especially network models with simple statistics like 2-stars, k-stars, and triangles typically suffer from (near) degeneracy problems because of the so-called avalanche effect of the change statistics, where positive parameters result in a large increase of the change statistics and always get larger (see Snijders et al., 2006). Schweinberger (2011) explores different settings of ERGMs and discovers that these models favour graphs with a transition from low-density to high-density graphs, making inference unstable and leading to convergence problems of the algorithms.

Snijders et al. (2006), Robins et al. (2007a) or Wang et al. (2013) propose ideas avoiding this avalanche effect by using configurations like the alternating k-stars or k-triangles leading to the social circuit models. Accordingly, Snijders et al. (2006) define alternating k-stars as

$$u_\lambda^{(s)}(y) = S_2 - \frac{S_3}{\lambda} + \frac{S_4}{\lambda^2} - \ldots + (-1)^{N-3}\frac{S_{N-1}}{\lambda^{N-3}} = \sum_{k=2}^{N-1}(-1)^k\frac{S_k}{\lambda^{k-2}},$$

where $S_k(y) = \sum_{i=1}^{N}\binom{y_{i+}}{k}$ for $k \geq 2$ is the number of k-stars and y_{i+} is the degree of node i. The alternating sign of the weights balances adjacent k-star counts and decreases problems of degeneracy. Modeling transitivity by using triangle counts and triangles of higher order often results in degenerated

models. Therefore, Snijders et al. (2006) defined – similar to the alternating k-stars – the alternating k-triangles as

$$u_\lambda^{(t)}(y) = 3T_1 - \frac{T_2}{\lambda} + \frac{T_3}{\lambda^2} - \ldots + (-1)^{N-3}\frac{T_{N-2}}{\lambda^{N-3}}, \quad \text{where}$$

$$T_k = \sum_{i=1}^{N}\sum_{j>i}^{N} y_{ij}\binom{\sum_{h\neq i,j} y_{ih}y_{hj}}{k}$$

is the number of k-triangles for $k \geq 2$ and

$$T_1 = \frac{1}{3}\sum_{i=1}^{N}\sum_{j>i}^{N} y_{ij}\sum_{h\neq i,j} y_{ih}y_{hj}$$

is the number of 1-triangles. Note that $\sum_{h\neq i,j} y_{ih}y_{hj}$ is the change statistic of the triangle counts. These ideas are aimed at reducing the weight of the linear effect of the statistics. The same motivation is used in Thiemichen and Kauermann (2017) by proposing a non-parametric ERGM or in the first contributing publication (Bauer et al., 2019) by using a logarithmic transformation of the change statistics that stabilizes the estimation but keeps the interpretability of the parameters.

1.4.5 Challenges and solutions in speeding up computation time

A crucial aspect for estimating large networks is having a reasonable computation time. Several proposals to speed up the time consuming simulation methods have been made by e.g. Morris et al. (2008), who suggested the "TNT" sampler or Snijders (2002), who described a version to update multiple edges simultaneously. These modifications are explained in more detail at the end of Section 1.4.3. The TNT sampler aims to converge more quickly to the target distribution, but is only useful for quite sparse or dense networks, which can be seen in the traceplots in the first contributing publication.

An obvious idea to reduce computation time is to adopt parallel computing, however, parallelization of networks is not straightforward due to the dependent data structure of networks. Handcock et al. (2017) have implemented an option of parallelization in the estimation routines of the **R** package **ergm**, which is able to exploit multiple CPUs, CPU cores or computing clusters. However, this option starts multiple Markov chains simultaneously, leading to an increasing memory and a lower improvement of computation time. Further details and comparisons are evaluated in the first contributing article in Chapter 2, where a different approach of simulating networks in parallel is discussed.

One key rule for parallel computing is that communication between workers (also called threads) costs a lot of time. Therefore, the aim is always to keep the communication to a minimum and to maximize the size of work in each divided step (see Schmidberger et al., 2009). This key rule led us to our first idea, which we call the "block-parallel" algorithm. This algorithm is also based on the Markovian independence assumption to draw independent node pairs in a network, which is described in more detail in Chapter 2. Summing up, we need to construct a way to draw independent node pairs, thus node pairs, which do not share a node. The first way of making use of the Markovian structure in

the simulation step is the one mentioned in the first contributing publication of this thesis where nodes are shuffled randomly, paired and sent to different computing cores as independent tasks. The second possibility is constructing independent node pairs from a symmetric Latin square with a unique diagonal (see e.g. Andersen and Hilton, 1980). Figure 1.5 shows such a Latin square decomposition for a four node example. The adjacency matrix for undirected graphs is symmetric, which means

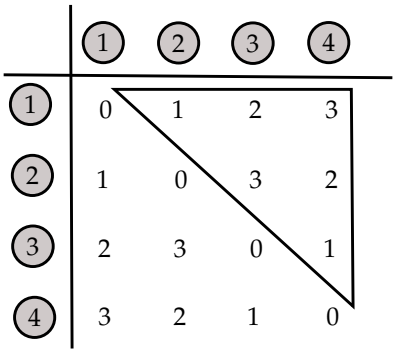


Figure 1.5: Visualization of a symmetric Latin square with a unique diagonal for a four node example.

it is sufficient to only consider the upper triangle. The numbers in the upper triangle correspond to the simulation steps in the parallel Metropolis-Hastings sampler. In the first simulation step, the node pairs $(1, 2)$ and $(3, 4)$ can be simulated simultaneously. In the second step, we simulate the pairs $(1, 3)$ and $(2, 4)$ in parallel, and in the third step $(1, 4)$ and $(2, 3)$. This procedure is scalable to larger networks with an increasing number of computing nodes. Assuming $N/2$ computing cores, we can complete a Metropolis-Hastings loop in $N - 1$ steps. Therefore, the computing time for network simulation increases just linearly with the number of nodes in the network (N), if sufficient computing cores are available. Obviously, when the number of nodes in a network is large, we may not have access to $N/2$ computing cores. It is also likely that the communication task between the cores is getting too demanding and devours the computational gains.

One solution for this kind of problem is a blockwise decomposition of the adjacency matrix by extending the conditional independence ideas from above. In this case, not only single edges are simulated on the cores, but blocks of independent pairs of edges are simulated. This speeds up the computation through the faster pre-processing as well as through saving time because of less communication between the cores. We therefore select submatrices of the adjacency matrix. This approach is also advisable if the number of nodes in the network is large and only a fixed number of cores is available. Figure 1.6 visualizes the idea of grouping a network with $N = 8$ nodes into independent blocks of node pairs. The block number indicates the step number. Blocks with the same number can be simulated simultaneously. In this example, two computing cores complete one Metropolis-Hastings loop in five rounds as the diagonal elements of the Latin square have to be considered at least halfway. In the first simulation step, we consider the edges Y_{13}, Y_{14}, Y_{23}, and Y_{24} sequentially on one core and at the same time the edges Y_{57}, Y_{58}, Y_{67}, and Y_{68} on the other core, in other words, in parallel. These are

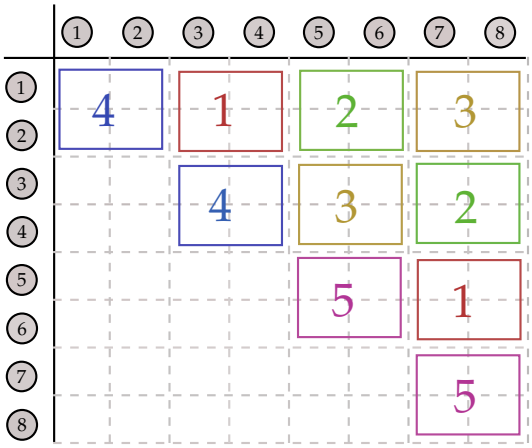


Figure 1.6: Visualization of a blockwise decomposition of an eight node example with the help of a symmetric Latin square.

the submatrices with entry one in the blockwise Latin square in Figure 1.6. The other steps follow the same scheme.

A comparison of the performance between the single entry choice of edges (called "parallel") and blockwise decomposition is shown in Figure 1.7, where three networks are simulated on eight cores for different numbers of nodes. Obviously, a reduction of computing time is achieved for the block-

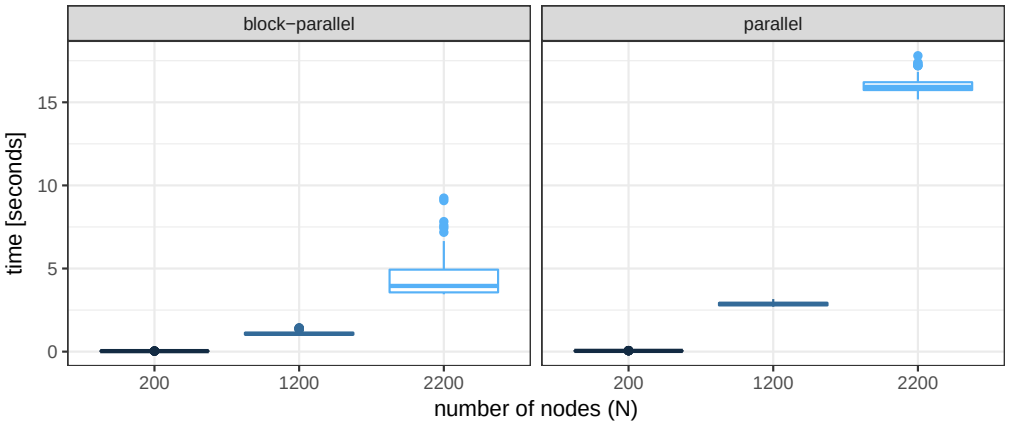


Figure 1.7: Performance of block-parallel and parallel algorithm for different network dimensions. The y-axis shows time in seconds to simulate three networks on eight cores.

parallel algorithm especially for large networks with $N = 2200$ nodes. The main reasons for this are the reduction of communication costs and the load balancing as well as the optimization of data locality issues.

1.4.6 Further restraints and open questions

The field of statistical network analysis is still not completely explored and a lot of research can be done to improve estimation algorithms, or handle different kinds of networks like ones with valued edges (e.g. Krivitsky, 2012; Desmarais and Cranmer, 2012a), missing data (e.g. Handcock and Gile, 2010; Koskinen et al., 2010), and nodal heterogeneity (e.g. Krivitsky et al., 2009; Thiemichen et al., 2016).
Even though estimation methods that try to find the (approximate) maximum likelihood estimator exist, one has to keep in mind that the Markov chains only run a finite time whereas the optimal result would be obtained after an infinite number of steps. Moreover, existing goodness-of-fit routines rely on simulating graphs from the fitted model and compare their statistics graphically to the observed analogues (cf. Hunter et al., 2008a). However, it remains unclear if the simulated networks come from the 'true' distribution or only resemble the network statistics. A surprising fact was found by Handcock (2003a), who fits an ERGM and then compares a large number of simulated networks to the observed one. The simulated networks differ a lot due to the extremely high number of possibilities. The maximum likelihood estimator of a parameter is the value that makes observing a given network most likely, though in most cases the probability in comparison to all possible networks is not high enough.

1.5 Dynamic network models

The second contributing article of this thesis suggests a smooth dynamic network model for patent collaboration data. This model focuses on a profile likelihood approach to model time-stamped event data based on a multivariate counting process. First, an overview of existing dynamic models for network data is given, which is followed by a short introduction of counting processes and the profile likelihood approach based on the Cox model (Cox, 1972). We further explore similarities of the Cox proportional hazards and (additive) Poisson model. We extend this analogy to additive Poisson models because we propose in our article a semiparametric approach including covariates more flexible by penalized smoothing techniques.

1.5.1 Outline

Originally, the analysis for dynamic network data focused on collapsed panel data (e.g. Hanneke et al., 2010) or small networks with only a few observation times and a few hundreds of nodes (Snijders, 2001, 2005). On the one hand, most of the time data collection depends on discrete and manually gathered information, on the other hand, methods for large and time-stamped data are lacking. In general, models for dynamic network data can be divided into three, overlapping and not encompassing, strands: actor-oriented, tie-oriented, and models based on event history (see Figure 1.8 for an overview). The most known models of these strands are the Stochastic Actor-Oriented Model (SAOM) of Snijders (2001), tie-oriented models like the extensions of the Exponential Random Graph Model (ERGM) to longitudinal (LERGM; Koskinen and Snijders, 2013) or temporal (TERGM;

Hanneke et al., 2010) network models, and the Relational Event Model (REM) of Butts (2008). The main differences are different model assumptions that decide about the interpretation of the estimates and the conclusions, which can be drawn. In comparison to SAOMs, which are actor-oriented, ERGMs are global models and focus on the importance of tie structures. Furthermore, the treatment of time differs for these models whereby SAOM and LERGM follow a continuous-time and TERGM follows a discrete-time process. A crucial difference between tie-oriented models and event history models is that the former models a stationary distribution whereas the latter models the changes between two time points and focuses on the search of driving forces for the network evolution and not on predicting future ties.

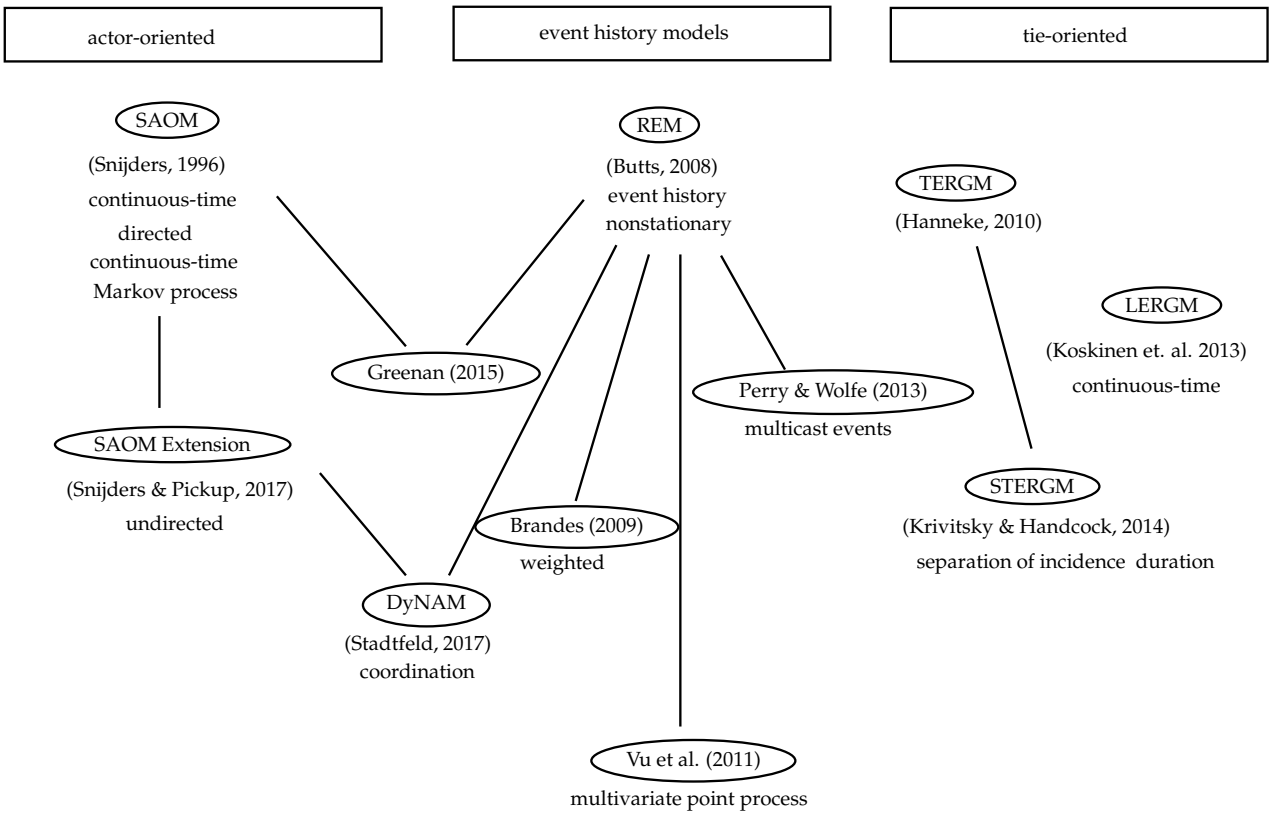


Figure 1.8: Overview of existing dynamic models (not encompassing).

In the following, we introduce the basic concepts of the most important models in dynamic network analysis starting with the first approaches in this field.

Snijders (1996) proposed a model class for longitudinal network data known as stochastic actor-based model (see also Snijders, 2001, 2005; Snijders et al., 2010). SAOMs focus on network dynamics influenced by the change of relations, which are driven by an actor. This actor-oriented perspective has the consequence that tie changes are modeled as results from actions by actors, more precisely, that actors control their outgoing ties. Hence, SAOMs are actor-driven. These decisions are nested

in a Markov process, i.e., for any point of time the present network is not affected by past events, but provides insight into its further development. The models are based on continuous-time Markov chain models although the networks are observed at discrete times with a finite number of observation waves. The network evolution or dissolution process is divided into two parts. The first part determines the waiting time until a change in the network is made by one actor and is modeled by optimizing a 'rate function', which contains information about the general affinity of changing ties of each actor. The second part is called 'objective function' and represents the utility of certain possible tie changes for an arbitrary actor. This function determines the preferred network over the set of all possible ones and is based on a multinomial logit model. Both sub-processes depend on exogenous covariates, endogenous network effects, or effects derived from network positions. The estimation is done using the method of moments approach with stochastic approximation by Robbins and Monro (1951). Snijders and Pickup (2017) extent the latter model for nondirected ties by using a two-step process of opportunity and choice. The two-sided choice resulting from undirected relations is decomposed into a timing or opportunity process with one- or two-sided initiatives and a choice process. The latter process uses one of three opportunities: agreement about a tie between two actors, one actor decides alone about a tie, or the decision is based on a combined objective function. Depending on different assumptions regarding various applications, one of these six combinations is selected as modeling approach.

An early attempt extending the exponential random graph model to dynamic network data settings has been proposed by Robins and Pattison (2001) and was further explored by Hanneke et al. (2010). The idea of Robins and Pattison (2001) is to use a Markov random graph approach for temporal evolution of social networks by generalizing the process to changes of discrete time points. Hanneke et al. (2010) called this extension Temporal Exponential Random Graph Model (TERGM) and added algorithmic and inferential developments like hypothesis tests, more flexible parametrization, and explorations of statistical properties. A major point of these discrete temporal models is the Markov dependence assumption over time. At each time t a network $\boldsymbol{Y}^t$ is assumed to be independent of $\boldsymbol{Y}^1, \ldots, \boldsymbol{Y}^{t-2}$ given $\boldsymbol{Y}^{t-1}$. More generally, a TERGM that incorporates dependencies of K previously observed networks is denoted by

$$\mathbb{P}_\theta(\boldsymbol{Y}^t | \boldsymbol{Y}^{t-K}, \ldots, \boldsymbol{Y}^{t-1}, \boldsymbol{\theta}) = \frac{\exp\{\boldsymbol{\theta}' s(\boldsymbol{Y}^t, \boldsymbol{Y}^{t-1}, \ldots, \boldsymbol{Y}^{t-K})\}}{\kappa(\boldsymbol{\theta}, \boldsymbol{Y}^{t-K}, \ldots, \boldsymbol{Y}^{t-1})}, \tag{1.13}$$

where the choice of $K \in \{0, 1, \ldots, T-1\}$ needs to be in line with the temporal dependence of network $\boldsymbol{Y}^t$. Estimation of TERGMs can be carried out by maximum likelihood methods using MCMC sampling techniques to approximate the intractable normalizing constant. These estimation approaches resemble the ones used in ERGMs, but have small differences. Like in the time-invariant ERGM, the procedures used are challenging and computationally expensive. Therefore, Desmarais and Cranmer (2012b) suggested a pseudo-likelihood approach with bootstrap confidence intervals to correct the biased uncertainty measurement, which was described by Leifeld et al. (2018) for TERGMs. For detailed information we refer the reader to Hanneke et al. (2010), Leifeld et al. (2018) and Desmarais and Cranmer (2012b).

Krivitsky and Handcock (2014) suggested combining the discrete-time temporal exponential random graph model for network dynamics of Hanneke et al. (2010) with the nonlinear parametrization known as curved ERGMs of Hunter and Handcock (2006). These exponential-family random graph models focus on a separable modeling approach of incidence and duration of ties, calling this model class Separable Temporal ERGM (STERGM). This separate parametrization within a time-step allows for individual interpretation and therefore more flexibility. Within this method, a further distinction between tie formation and dissolution can be explored. In many applications this separation is useful. STERGM can be seen as a subclass of TERGM defined in (1.13) with $K = 1$.

A natural procedure for modeling event data, including time stamps, is using survival models like the Cox proportional hazards model (Cox, 1972). Generally, events are independent and influenced by exogenous effects. However, when modeling network data, the focus is primarily on potential endogenous factors, which influence network creation and dissolution.

An approach for modeling social behavior dynamically as events or actions is known as the Relational Event Model (REM) suggested by Butts (2008). This framework allows for likelihood-based inference from event data to detect social dependence patterns, estimate its strength, evaluate competing settings within this patterns, and allow for non-stationary behavior. The relational event model combines network structures with event history models for nondirected and directed ties by exploiting a tie-based approach where potential relations are chosen independently conditioned on the past actions. This framework can include exogenous covariates influencing the future and sufficient statistics capturing the impact of event history. These models aim to combine a theoretical based approach with working inference and estimation for analyzing the process underlying social behavior. The crucial point in the REM is the relational event, or action happening at time $t \in \mathbb{R}$ when an actor (the "sender") is pointing towards one "receiver". It is assumed that the events follow an inhomogeneous Poisson-type process conditioned on the past history and possibly other exogenous covariates.

Brandes et al. (2009) developed a framework for modeling dyadic event data of interactions between actors and apply it to political event networks. Similar model specifications can be found in the relational event model of Butts (2008) but with extension to weighted events. Parameter estimation is based on maximum likelihood techniques and assumes an independence between rate and weight parameters. Network statistics to capture reciprocity, structural balance, or activity and popularity effects can be incorporated to detect influences of the network's past on future events.

Most models in dynamic network analysis focus on directed relations where the actor controls the outgoing ties. In political or social science, however, the question arises how individuals or states in general jointly admit to forming network connections. An example for what Stadtfeld et al. (2017a) call 'coordination' networks, are patent collaborations that arise by inventors, who mutually agree to work together and submit joint patents. Stadtfeld et al. (2017a) introduced the Dynamic Network Actor Model (DyNAM) for modeling relational event data of coordination networks. These models consider dependency between observations, are constructed for undirected ties including a two-sided process of building ties, and allow for time-stamped data. Moreover, it is possible to take different mechanisms like tie formation and dissolution, unequal time gaps, and various types of ties into account. DyNAMs aim to investigate dynamic coordination networks with different facets of ties

(*weighted, windowed, and signed*) and consider internal network structures adjusting for homophily, clustering, or preferential attachment effects. An important question in network science arises when it comes to the decision process of actors formulating their favorable circumstances and preferences given the opportunities. This two-sided process is an agreement of both actors being involved and maybe a more complex dependence structure due to network and temporal dependence (see above two-step process of Snijders and Pickup, 2017). Consider for example that, the data contains the following information:

time	inventors	sign
month 1	$1 \longleftrightarrow 2$	create
month 2	$1 \longleftrightarrow 3$	create
month 6	$2 \longleftrightarrow 3$	create

The decision process can be influenced by the temporal effect of the first two observations and by the network structure that inventor 1 is involved in both patents of month 1 and 2. Another network dependence assumption might play a role in the third observation because inventor 2 and 3 close a triangle with this patent. Dynamic network actor models are designed for such time-stamped coordination data combining parts of Snijders (1996, 2001) stochastic actor-oriented model and the relational event model of Butts (2008). Stadtfeld et al. (2017a) merge an actor-oriented approach where both actors i and j select each other from a set of actors with undirected relations. 'Still' actor-oriented, actor i proposes a new tie to j at any time point, and j has to choose i as favored partner. Inspired by Snijders et al. (2010) micro-model in SAOMs for determining the change, DyNAMs are also developed in a continuous-time framework, modeling mutual choices with multinomial probability models. A linear objective function evaluates changes in the process matrix. In order to model the waiting time between tie proposals and realized changes, further steps are included in the model framework. For optimizing the model parameter in the estimation routine, DyNAMs use a maximum likelihood approach. For more details and discussion about the proposed framework see Stadtfeld et al. (2017a,b), Butts (2017), and Snijders (2017).

Vu et al. (2011a) suggested an event history approach focusing on large networks with nodal statistics, which extends earlier work from Butts (2008). This 'egocentric network model' of Vu et al. (2011a) uses an efficient optimization algorithm derived from the partial likelihood to estimate nodal processes based on statistics from network history. Later, Vu et al. (2011b) generalize and extend their approach for dynamic egocentric models (Vu et al., 2011a) to a general continuous-time regression model for longitudinal networks with time-varying network statistics embedded in an Aalen model (Aalen et al., 2008). Furthermore, they unite techniques for a relational framework including a multivariate counting process for edge formation. The focus of their approach is on large networks and efficient inference thereof. In the general framework, Vu et al. (2011b) formulate $Y_{ij}(t)$ as a counting process, which represents the number of ties from node i to j at time t. In the paper they restrict their model to non-recurrent events and do not take any tie dissolution process into account. The

model considered is based on a multivariate interdependent counting process $\boldsymbol{Y}(t)$ decomposed by the Doob-Meyer theorem (Aalen et al., 2008),

$$\boldsymbol{Y}(t) = \int_0^t \boldsymbol{\lambda}(s) + \boldsymbol{M}(t), \tag{1.14}$$

with $\boldsymbol{\lambda}(t)$ denoting the intensity process or hazard rate and $\boldsymbol{M}(t)$ being the martingale noise. The basis for this theory relies on the fact that counting processes are non-decreasing in time and can therefore be regarded as submartingales. The intensity process is modeled in two different ways, a multiplicative Cox or additive Aalen approach, which both take the past of the networks as network statistics into account. $\boldsymbol{\lambda}(t)$ incorporates linear combinations or time-varying effects of these network statistics. Estimation of the Cox-type model is carried out by exploiting the simplification of just maximizing the so-called partial likelihood instead of the full one. By doing so, the baseline hazard is considered as nuisance parameter. Combining this with the caching method of Vu et al. (2011a) results in an efficient computation for large networks. The estimation of the Aalen model is based on linear regression methods with the possibility to include kernel smoothing techniques for interpretability of time-varying coefficients.

A different approach modeling time-stamped network data for social events is described by De Nooy (2011). Based on a discrete-time event history model, De Nooy (2011) focuses on modeling tie formation, change, or dissolution by combining a multilevel design and time-varying covariates. Network dependencies and endogenous effects are taken into account by applying a multilevel logistic regression analysis approach based on a General Linear (Mixed) Model (GLMM). Therefore, extensions to a non-dichotomous target variable like a competing risk model in survival analysis is possible.

Perry and Wolfe (2013) describe an extension of a multivariate point process approach emphasizing properties of the maximum partial likelihood inference and considering multicast interactions. The authors propose a stratified Cox multiplicative intensity model for directed networks with covariates built from network history. For simplifications, most research excludes simultaneous interactions or uses approximated solutions like the Breslow-Peto correction (Breslow, 1972; Peto, 1972) or the Efron (1977) approximation. Both approaches use a product over the risk terms of the tied events and give a fair approximation of the likelihood function but can in some cases suffer from biased estimations (Scheike and Sun, 2007). Breslow and Peto's suggestion is easy to compute but the approximation of Efron (1977) is closer to the proper likelihood because weighted risks are used. Perry and Wolfe (2013) evaluate the approximation error between the model that includes multiple receivers explicitly and the one that approximates the partial likelihood. They suggest a bias correction procedure based on parametric bootstrap calculation for that error.

Greenan (2015) transfers the stochastic actor oriented model of Snijders (2001) to dynamic social networks with the evolution of diffusion of innovations by combining the SAOM with a proportional hazards model. The network and diffusion process are combined and considered as dependent on each other, while the adoption times of the latter process follow a Cox regression model (Cox, 1972) with

a hazard function depending on covariates. The network dynamics are modeled via rate function like it is known from the SAOM.

1.5.2 Cox's regression model and multivariate counting processes for network data

The most famous model class for fitting time-to-event data is known as the Cox proportional hazards model for survival data. Cox (1972) denotes the hazard or intensity rate (for non-recurrent events)

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{\mathbb{P}\left(t \leq T < t + \Delta t | T \geq t\right)}{\Delta t} \tag{1.15}$$

for the survival time T, describing the risk of having an event at time t, given that no event has occurred until t. The idea is to model the effects of covariates on the hazard rate, rather than the hazard rate itself. For example, does the fact of having joint patents increase or decrease the hazard of submitting new patents? Cox (1972) represents the hazard rate as

$$\lambda(t; x) = \lambda_0(t) \exp\{\beta^T x(t)\}, \quad t \geq 0, \tag{1.16}$$

with covariate vector $x(t)$, which may be time dependent or not and unknown coefficient vector β. The baseline hazard $\lambda_0(t)$ determines the underlying process but is unknown. Due to the individual covariates, the baseline hazard becomes subject-specific. The basic Cox model assumes that all m event times $t_{(1)}, \ldots, t_{(m)}$ are distinct, and that the event for subject d occurred at time $t_{(d)}$. In order to make inference, β and later λ, have to be estimated by maximizing

$$L(\beta) = \prod_{d=1}^{m} \left[\frac{\exp\left(\beta^T x_d(t_{(d)})\right)}{\sum_{d' \in O_d} \exp\left(\beta^T x_{d'}(t_{(d)})\right)} \right]^{\delta_d} \tag{1.17}$$

with respect to β, where $t_{(1)}, \ldots, t_{(m)}$ are the distinct survival times, O_d is the risk set (in our application called 'option set') at time $t_{(d)}$ and δ_d is the event indicator. Later Cox (1975) shows that his suggested likelihood in (1.17) can be derived as the partial likelihood function and that $\hat{\beta}$ is its estimator. For estimating the cumulative hazard $\Lambda(t) = \int_0^t \lambda(s) ds$, Breslow (1972, 1974) proposes the following estimator

$$\hat{\Lambda}(t) = \sum_{t_{(d)} \leq t} \frac{\delta_d}{\sum_{d' \in O_d} \exp\left(\hat{\beta}^T x_{d'}(t_{(d)})\right)}, \tag{1.18}$$

based on linear interpolation between survival times. The baseline hazard $\lambda_0(t)$ is commonly treated as nuisance parameter and considered to be a non-negative function with non-zero values over the event time intervals.

Andersen and Gill (1982) and Johansen (1983) extend the model of Cox (1972) captured in equation (1.16) by allowing recurrent events. Johansen (1983) derives a joint likelihood $L(\beta, \Lambda)$ from an extended Cox model and demonstrates that the partial likelihood of equation (1.17) derived from the

full censored-data likelihood resembles a profile likelihood. To be more specific, $L(\beta) = \max_\Lambda L(\beta, \Lambda)$ where $L(\beta, \Lambda)$ is the joint likelihood for the unknown parameters β and Λ. Note that the maximum of $L(\hat{\beta}, \Lambda)$ is the estimator (1.18) proposed by Breslow (1972).

In the same contribution to the discussion of the paper of Cox (1972), Breslow (1972) and Peto (1972) independently suggested an approximation for handling events that take place at the same time. The Peto-Breslow approximation for ties yields the likelihood

$$L(\beta) = \prod_{d=1}^{m} \frac{\exp\left(\beta^T \sum_{d'_C \in C_d} x_{d'_C}(t_{(d)})\right)}{\left[\sum_{d' \in O_d} \exp\left(\beta^T x_{d'}(t_{(d)})\right)\right]^{|C_d|}} = \prod_{d=1}^{m} \frac{\prod_{d'_C \in C_d} \exp\left(\beta^T x_{d'_C}(t_{(d)})\right)}{\left[\sum_{d' \in O_d} \exp\left(\beta^T x_{d'}(t_{(d)})\right)\right]^{|C_d|}}, \tag{1.19}$$

where $\sum_{d'_C \in C_d} x_{d'_C}(t_{(d)})$ denotes the sum of the covariates over the set C_d, where $|C_d|$ is the number of subjects who have an event at time $t_{(d)}$. This approximation is similar to the original likelihood of Cox (1972) shown in equation (1.17) and results in only small modifications to the score function and fisher matrix. This approach is straightforward to calculate because it uses a product over the risk terms of the tied events.

The Cox model with the hazard rate (1.16) can be formulated as a multivariate counting process considered by Aalen (1978). This formulation allows transferring properties of the estimator and of the distribution theory (cf. Andersen and Gill, 1982), which is advantageous. Furthermore, Andersen and Gill (1982) sketch a generalized version of this counting process formulation for recurrent events, complex censoring structures, and time dependent covariates. We adapt this representation for undirected network data (see Vu et al., 2011a,b). Assume that networks evolve over time by creating edges between two nodes at time t. For a set of N actors (nodes) a counting process $Y_{ij}(t)$ denotes the number of edges between actor i and j up to time t. The counting process is defined for a pair (i, j) of actors $(i, j = 1, \ldots, N)$, resulting in a multivariate counting process

$$\boldsymbol{Y}(t) = (Y_{ij}(t), i, j \in \{1, \ldots, N\}, i \neq j) \tag{1.20}$$

for all pairs of actors where no assumptions about the independence of individual processes are made. We are primarily interested in modeling the intensity process $\boldsymbol{\lambda}(t)$ by decomposing the multivariate counting process $\boldsymbol{Y}(t)$ using the Doob-Meyer theorem (see Aalen et al., 2008):

$$\boldsymbol{Y}(t) = \int_0^t \boldsymbol{\lambda}(s) + \boldsymbol{M}(t), \tag{1.21}$$

where $\boldsymbol{M}(t)$ is a local continuous-time martingale. The intensity process $\boldsymbol{\lambda}(t)$ describes the tendency to experience events. Andersen and Gill (1982) propose modeling this process in Cox tradition (see equation (1.16)) but with an additional indicator variable that is adjusted for recurrent events.

In our contributing article in Chapter 3 we discretize the evolving process $Y_{ij}(t_{(d)}) = Y_{ij,d}$ with $t_{(1)}, \ldots, t_{(m)}$ where $d \in \{1, \ldots, m\}$ and model the intensity process as $\lambda_d \exp(\eta_{ij,d})$, where λ_d is

the baseline hazard of the process and $\eta_{ij,d}$ the covariate process modeled by penalized smoothing techniques. For simplicity we have assumed that the baseline hazard as well as the covariate process are piecewise constant between the observed time points for $t \in (t_{(d-1)}, t_{(d)}]$. The predictor $\eta_{ij,d} = m_{(1)}(x_{(1),ij,d}) + \ldots + m_{(P)}(x_{(P),ij,d})$ contains smooth functions $m(\cdot)$ based on penalized B-splines for P covariates derived from network statistics or external effects. To be more specific, the smooth functions are replaced by

$$m(x) = \sum_k B_k(x)u, \tag{1.22}$$

where $B_k(\cdot)$ is a K dimensional B-spline basis (see De Boor, 1978; Wood, 2017). That means the predictor can be written as $\boldsymbol{\eta}_d = \boldsymbol{B}_d\boldsymbol{u}$ with $\boldsymbol{B}_d = (B_{(1),d}, B_{(2),d}, \ldots)$ and $\boldsymbol{u}^T = (u_{(1)}^T, u_{(2)}^T, \ldots)$. $B_{(1),d}$ is the basis matrix for the first covariate built from the the rows of the spline basis for each observation. In order to make inference, the corresponding coefficient vector $\boldsymbol{u}$, which contains the weights for the basis functions $\boldsymbol{B}_d$, has to be estimated. The contributing article in Chapter 3 gives more details.

Based on a piecewise constant baseline hazard $\lambda_0(t) = \lambda_d$ and covariate process $\eta_{ij}(t) = \eta_{ij,d}$ for $t \in (t_{(d-1)}, t_{(d)}]$, we derive the log-likelihood

$$l(\lambda_1, \ldots, \lambda_m, \boldsymbol{u}) = \sum_{d=1}^{m} \left[\sum_{(i,j)\in C_d} [\log \lambda_d + \eta_{ij,d}] - \lambda_d \cdot \left(\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right) \right) \right], \tag{1.23}$$

where C_d is the index set of events at time $t_{(d)}$ and O_d is the option or risk set. Due to ties, the index set C_d consists of $|C_d|$ elements, which is mostly greater than one. In order to find the estimators, we optimize the score function

$$s(\lambda_1, \ldots, \lambda_m, \boldsymbol{u}) = \frac{\partial l(\lambda_1, \ldots, \lambda_m, \boldsymbol{u})}{\partial \lambda_1, \ldots, \partial \lambda_m} = \sum_{d=1}^{m} \left[\sum_{(i,j)\in C_d} \left[\frac{1}{\lambda_d} \right] - \left(\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right) \right) \right] \tag{1.24}$$

by setting $s(\lambda_1, \ldots, \lambda_m, \boldsymbol{u}) = 0$ and $\sum_{(i,j)\in C_d} \left[\frac{1}{\lambda_d} \right] = \frac{|C_d|}{\lambda_d}$, which yields

$$\hat{\lambda}_d = \frac{|C_d|}{\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right)}. \tag{1.25}$$

Following the ideas of a profile likelihood approach, we insert (1.25) into (1.23) and obtain the profile log-likelihood

$$l(\boldsymbol{u}) \propto \sum_{d=1}^{m} \left[\sum_{(i,j)\in C_d} \eta_{ij,d} - |C_d| \cdot \log \left(\sum_{(i',j')\in O_d} \exp(\eta_{i'j',d}) \right) \right], \tag{1.26}$$

when the constant terms are deleted (for details see equation (A.33) in the Appendix). This profile likelihood (without the constant terms) is equal to the partial likelihood (1.19) with the Peto-Breslow approximation for ties. Thus, inference results in the same estimators.

1.5.3 Analogy of Cox proportional hazards model and Poisson model

In the following we adopt the wording and notation that is used in the data example of the second contributing article where the submission of a patent from inventor i and j is defined as an event. If this event happens at time d, the event indicator $y_{ij,d}$ is equal to one. As our event is defined over a set of pairs of inventors, the notation uses two subscripts (i, j) instead of the well-known term 'subject i'.

Following Whitehead (1980) and Friedman (1982), under certain conditions the likelihood of the Cox proportional hazards model (1.16) is proportional to the one of a Poisson regression model and therefore, can be estimated as such. Taking advantage of this analogy, inference is simplified as standard software, e.g. **mgcv** (Wood, 2017) in **R** (R Core Team, 2017), can be used. Moreover, extending the model with spatial, random, or nonlinearly time-varying effects is straightforward by resorting to implemented algorithms and methods (see Bender et al., 2018). In our contributing article we apply penalized smoothing techniques based on basis functions to a likelihood type approach. The software package **mgcv** offers a great variety of smoothing approaches and estimation methods that could be adopted.

As already mentioned, in our contributing article we model the covariate process using smoothing techniques and in the application we have to handle tied observations. We show below that the partial likelihood with Breslow-Peto correction for ties with a nonlinear predictor, which was introduced in the section above, is equivalent to an additive Poisson regression model.

To see the similarity between these two approaches, we assume at each event time $t_{(d)}$ an artificial response variable $y_{ij,d}$ that indicates for each inventor pair (i, j) whether a patent is submitted at time $t_{(d)}$ ($y_{ij,d} = 1$) or not ($y_{ij,d} = 0$). This artificial data is produced for the whole option set and over the complete observed time period due to the presence of recurrent events, where the option set remains the same until the end. The generalized Poisson regression model is obtained by

$$y_{ij,d} \overset{iid}{\sim} \text{Po}(\mu_{ij,d}) \text{ with } \mu_{ij,d} = \mathbb{E}(y_{ij,d}|\eta_{ij,d}) = \exp(\log(\lambda_d) + \eta_{ij,d} + o_{ij,d}), \tag{1.27}$$

where each artificial response variable $y_{ij,d}$ is linked to its corresponding predictor $\eta_{ij,d}$ including the covariates. In our application, all offsets $o_{ij,d}$ are equal to zero since the event times $t_{(d)} - t_{(d-1)}$ only differ by one for each $d \in \{1, \ldots, m\}$. Hence, $o_{ij,d} = \log(t_{(d)} - t_{(d-1)}) = \log(1) = 0$ and equation (1.27) simplifies to

$$\mu_{ij,d} = \exp(\alpha_d + \eta_{ij,d}), \tag{1.28}$$

where $\log(\lambda_d)$ is replaced by α_d. The intercept term α_d for the d^{th} time interval $(t_{d-1}, t_d]$ can be included in the Poisson model as a factor variable or, as in our application, as a smooth component of the time. Since the artificial data $y_{ij,d}$ is Poisson distributed with density $f(y_{ij,d}) =$

$(\mu_{ij,d})^{y_{ij,d}}/y_{ij,d}! \exp(-\mu_{ij,d})$ and $y_{ij,d} \in \{0,1\}$, the factorial can be ignored. Thus, the contribution to the likelihood for $t_{(d)}$ with index set C_d containing inventor pairs with events at this time point is

$$\begin{aligned} L_d &= \prod_{\{(i,j)|y_{ij,d}=1\}} (\mu_{ij,d})^{y_{ij,d}} \cdot \exp(-\mu_{ij,d}) \cdot \prod_{\{(i',j')|y_{i'j',d}=0\}} (\mu_{i'j',d})^{y_{i'j',d}} \cdot \exp(-\mu_{i'j',d}) \\ &= \prod_{(i,j)\in C_d} \mu_{ij,d} \cdot \exp\left(-\sum_{(i',j')\in O_d} \mu_{i'j',d}\right) \end{aligned} \tag{1.29}$$

(for details see equation (A.34) in the Appendix).
In order to get the profile likelihood, we insert (1.28) into (1.29), which yields

$$\begin{aligned} L_d &= \prod_{(i,j)\in C_d} \exp\left(\alpha_d + \eta_{ij,d}\right) \cdot \exp\left(-\sum_{(i',j')\in O_d} \exp\left(\alpha_d + \eta_{i'j',d}\right)\right) \\ &= \frac{\exp\left(\sum_{(i,j)\in C_d} \left(\alpha_d + \eta_{ij,d}\right)\right)}{\exp\left(\sum_{(i',j')\in O_d} \exp\left(\alpha_d + \eta_{i'j',d}\right)\right)}. \end{aligned} \tag{1.30}$$

This expression is maximized with respect to α_d. This maximum $\hat{\alpha} = \log\left(\frac{|C_d|}{\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right)}\right)$, is inserted in (1.30) to obtain the profile likelihood

$$L(\boldsymbol{u}) = \prod_{d=1}^{m} \frac{\exp\left[|C_d| \cdot \log\left(|C_d|\right)\right] \cdot \exp\left[\sum_{(i,j)\in C_d} \eta_{ij,d}\right]}{\exp\left(|C_d|\right) \cdot \left(\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right)\right)^{|C_d|}} \tag{1.31}$$

with its log-likelihood

$$\begin{aligned} l(\boldsymbol{u}) &= \sum_{d=1}^{m} \left[|C_d| \cdot \log\left(|C_d|\right) + \sum_{(i,j)\in C_d} \eta_{ij,d} - |C_d| - |C_d| \cdot \log\left(\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right)\right)\right] \\ &\propto \sum_{d=1}^{m} \left[\sum_{(i,j)\in C_d} \eta_{ij,d} - |C_d| \cdot \log\left(\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right)\right)\right] \end{aligned} \tag{1.32}$$

(detailed steps can be found in equation (A.35) in the Appendix).
This log-likelihood is proportional to the partial likelihood with Breslow-Peto correction derived in equation (1.26) (cf. Wood, 2017; Whitehead, 1980; Friedman, 1982). Exploiting this analogy makes it possible to estimate such data using flexible (additive) Poisson regression models. One drawback is that the artificial data exceedingly increase the size of the data set. This implies that the memory and computation time increases for the estimation. Nevertheless, our application includes time dependent covariates and recurrent events for which this artificial data representation is useful.

1.6 Software example

In this section we illustrate the usage of the **R** package **pergm** (Bauer, 2016) which implements the simulation algorithm in parallel of the first contributing article. The algorithm is implemented in C++ and the parallelization works in C++ via OpenMP (Open Multi-Processing). As a first step, the package has to be installed and loaded. Use `?simulate_networks` to see the help page.

```
library(devtools)
install_github("VerenaMaier/pergm")
library(pergm)
?simulate_networks
```

The function `simulate_networks` simulates networks following nondirected Markov random graph models. To make use of the suggested algorithm based on parallel computing, a Markovian conditional independent structure is needed. Therefore, network statistics like edges, 2-stars, or triangles are allowed to be included. The function needs as input arguments a starting adjacency matrix (`adjacency`) or the number of nodes (`dim`), a vector of the estimated coefficients of the network statistics (`theta`), and the number of networks that need to be simulated (`n_sim`). If the starting adjacency is missing, an empty matrix with dimension `dim` is generated. Additional arguments like the number of updates in the Markov chain (`n_update`), the burn-in updates (`burnin`), or the number of used computing cores (`n_cores`) can be specified. The argument `return_nw` indicates if complete simulated networks should be returned or just the network statistics. The default is to return just the network statistics of each simulation draw because returning complete adjacency matrices is inefficient for memory handling. The default return value is a matrix containing the number of edges, 2-stars, and triangles of each simulated network.

```
simulate_networks(adjacency = adj, dim = N, theta = c(0, 0, 0), n_sim = 1,
   n_update = 1e+06, burnin = 1000, n_cores = 4, return_nw = FALSE,
   log_change = TRUE)
```

The argument `log_change = TRUE` transforms the change statistics to a monotone and concave shape by using the logarithm. Note that with this transformation the network statistics 2-stars and triangles fulfill the Markov independence only asymptotically. The aim of using the logarithm is to reduce degeneracy problems and to avoid the avalanche effect following Snijders et al. (2006).
The following **R** chunk gives a short example of the usage of this parallel simulation function. Therefore, one needs to specify the dimension of the network to be simulated and a vector containing the estimated coefficients.

```
set.seed(123)
# number of nodes
N <- 100
# the given parameter vector of edges, 2-stars and triangles
theta <- c(-0.1, -0.01, 0.03)
# simulate one network on four computing cores
simulate_networks(dim = N, theta = theta,
   log_change = TRUE, n_sim = 1, n_cores = 4)

     [,1]     [,2]     [,3]
[1,] 2435 5559.362 2556.917
```

The function call from above simulates one network with hundred actors on four computing cores in parallel and returns the corresponding transformed network statistics. Note that a reasonable time saving arises only for large networks with more than 1000 nodes.
Supplementary material is available at `https://github.com/VerenaMaier/pergm`.

References

Aalen, O. (1978). Nonparametric inference for a family of counting processes. *The Annals of Statistics*, pages 701–726.

Aalen, O., Borgan, O., and Gjessing, H. (2008). *Survival and event history analysis: A process point of view*. Springer Science & Business Media.

Andersen, L. D. and Hilton, A. J. W. (1980). Generalized latin rectangles I: Construction and decomposition. *Discrete Mathematis*, 31:125–152.

Andersen, P. K. and Gill, R. D. (1982). Cox's regression model for counting processes: A large sample study. *The Annals of Statistics*, pages 1100–1120.

Bauer, V. (2016). pergm: Parallel exponential random graph model simulation. `https://github.com/VerenaMaier/pergm`. Accessed 5 Dec 2018.

Bauer, V., Fürlinger, K., and Kauermann, G. (2019). A note on parallel sampling in Markov graphs. *Computational Statistics*, 34(3):1087–1107.

Bauer, V., Harhoff, D., and Kauermann, G. (2019). A smooth dynamic network model for patent collaboration data. *arXiv e-prints, arXiv:1909.00736*.

Bender, A., Groll, A., and Scheipl, F. (2018). A generalized additive model approach to time-to-event analysis. *Statistical Modelling*, 18(3-4):299–321.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236.

Brandes, U., Lerner, J., and Snijders, T. A. (2009). Networks evolving step by step: Statistical analysis of dyadic event data. In *2009 International Conference on Advances in Social Network Analysis and Mining*, pages 200–205. IEEE.

Breslow, N. E. (1972). Contribution to the discussion of the paper by D.R. Cox. *Journal of the Royal Statistical Society, Series B (Methodological)*, 34:216–217.

Breslow, N. E. (1974). Covariance analysis of censored survival data. *Biometrics*, 30(1):89–99.

Butts, C. T. (2008). A relational event framework for social action. *Sociological Methodology*, 38(1):155–200.

Butts, C. T. (2017). Comment: Actor orientation and relational event models. *Sociological Methodology*, 47(1):47–56.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220.

Cox, D. R. (1975). Partial likelihood. *Biometrika*, 62(2):269–276.

De Boor, C. (1978). *A Practical Guide to Splines*, volume 27. Springer, New York.

De Nooy, W. (2011). Networks of action and events over time. A multilevel discrete-time event history model for longitudinal network data. *Social Networks*, 33(1):31–40.

Desmarais, B. A. and Cranmer, S. J. (2012a). Statistical inference for valued-edge networks: The generalized exponential random graph model. *PLoS ONE*, 7(1):1–12.

Desmarais, B. A. and Cranmer, S. J. (2012b). Statistical mechanics of networks: Estimation and uncertainty. *Physica A: Statistical Mechanics and its Applications*, 391(4):1865–1876.

Efron, B. (1977). The efficiency of Cox's likelihood function for censored data. *Journal of the American Statistical Association*, 72(359):557–565.

Erdős, P. and Rényi, A. (1959). On random graphs. *Publicationes Mathematicae Debrecen*, 6(290):290–297.

Frank, O. (1991). Statistical analysis of change in networks. *Statistica Neerlandica*, 45(3):283–293.

Frank, O. and Strauss, D. (1986). Markov graphs. *Journal of the American Statistical Association*, 81(395):832–842.

Freeman, L. C. (2004). The development of social network analysis. *A Study in the Sociology of Science*, 1.

Friedman, M. (1982). Piecewise exponential models for survival data with covariates. *The Annals of Statistics*, 10(1):101–113.

Fruchterman, T. M. and Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164.

Geman, S. and Geman, D. (1987). Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images. In *Readings in Computer Vision*, pages 564–584. Elsevier.

Geyer, C. J. and Thompson, E. A. (1992). Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54(3):657–699.

Greenan, C. C. (2015). Diffusion of innovations in dynamic networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 178(1):147–166.

Handcock, M. S. (2003a). Assessing degeneracy in statistical models of social networks. *Technical report, Center for Statistics and Social Sciences, University of Washington*. `http://www.csss.washington.edu/Papers/wp39.pdf`. Accessed 11 Jan 2015.

Handcock, M. S. (2003b). Statistical models for social networks: Inference and degeneracy. In *Dynamic social network modeling and analysis: Workshop summary and papers*, pages 229–252. National Academies Press, Washingtion, DC.

Handcock, M. S. and Gile, K. J. (2010). Modeling social networks from sampled data. *The Annals of Applied Statistics*, 4(1):5–25.

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., Krivitsky, P. N., and Morris, M. (2017). *ergm: fit, simulate and diagnose exponential-family models for networks*. The Statnet Project (`http://www.statnet.org`). R package version 3.6.1. Accessed 11 Feb 2018.

Hanneke, S., Fu, W., Xing, E. P., et al. (2010). Discrete temporal models of social networks. *Electronic Journal of Statistics*, 4:585–605.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

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.

Hummel, R. M., Hunter, D. R., and Handcock, M. S. (2012). Improving simulation-based algorithms for fitting ERGMs. *Journal of Computational and Graphical Statistics*, 21(4):920–939.

Hunter, D. R., Goodreau, S. M., and Handcock, M. S. (2008a). Goodness of fit of social network models. *Journal of the American Statistical Association*, 103(481):248–258.

Hunter, D. R. and Handcock, M. S. (2006). Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*, 15(3):565–583.

Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M., and Morris, M. (2008b). ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, 24(3):1–29.

Johansen, S. (1983). An extension of Cox's regression model. *International Statistical Review/Revue Internationale de Statistique*, pages 165–174.

Kolaczyk, E. D. (2009). *Statistical Analysis of Network Models*. Springer, New York.

Koskinen, J. and Snijders, T. A. (2013). Longitudinal models. In Lusher et al. (2013), pages 130–140.

Koskinen, J. H., Robins, G. L., and Pattison, P. E. (2010). Analysing exponential random graph (p-star) models with missing data using bayesian data augmentation. *Statistical Methodology*, 7(3):366–384.

Krivitsky, P. N. (2012). Exponential-family random graph models for valued networks. *Electronic Journal of Statistics*, 6:1100–1128.

Krivitsky, P. N. and Handcock, M. S. (2014). A separable model for dynamic networks. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):29–46.

Krivitsky, P. N., Handcock, M. S., Raftery, A. E., and Hoff, P. D. (2009). Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 31(3):204–213.

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).

Leskovec, J. and Krevl, A. (2014). SNAP datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data/`. Accessed 22 Jan 2016.

Leskovec, J. and Mcauley, J. J. (2012). Learning to discover social circles in ego networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 539–547. Curran Associates, Inc.

Lusher, D., Koskinen, J., and Robins, G. (2013). *Exponential Random Graph Models for Social Networks*. Cambridge University Press, Cambridge.

Martin, S. B., Brown, W. M., Klavans, R., and Boyack, K. W. (2008). Drl: Distributed recursive (graph) layout. *SAND Reports 2936*, pages 1–10.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Morris, M., Handcock, M. S., and Hunter, D. R. (2008). Specification of exponential-family random graph models: Terms and computational aspects. *Journal of Statistical Software*, 24(4):1548–7660.

Newman, M. and Barkema, G. (1999). *Monte Carlo Methods in Statistical Physics*. Oxford University Press: New York, USA.

Okabayashi, S., Geyer, C. J., et al. (2012). Long range search for maximum likelihood in exponential families. *Electronic Journal of Statistics*, 6:123–147.

Pattison, P. and Robins, G. (2002). Neighborhood–based models for social networks. *Sociological Methodology*, 32(1):301–337.

Perry, P. O. and Wolfe, P. J. (2013). Point process modelling for directed interaction networks. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(5):821–849.

Peto, R. (1972). Contribution to the discussion of the paper by D.R. Cox. *Journal of the Royal Statistical Society, Series B (Methodological)*, 34:205–207.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407.

Robins, G. and Pattison, P. (2001). Random graph models for temporal processes in social networks. *The Journal of Mathematical Sociology*, 25(1):5–41.

Robins, G., Snijders, T., Wang, P., Handcock, M., and Pattison, P. (2007a). Recent developments in exponential random graph (p*) models for social networks. *Social networks*, 29(2):192–215.

Robins, G. L., Pattison, P., Kalish, Y., and Lusher, D. (2007b). An introduction to exponential random graph (p^*) models for social networks. *Social Networks*, 29(2):173–191.

Scheike, T. H. and Sun, Y. (2007). Maximum likelihood estimation for tied survival data under Cox regression model via EM-algorithm. *Lifetime Data Analysis*, 13:399–420.

Schmidberger, M., Morgan, M., Eddelbuettel, D., Yu, H., Tierney, L., and Mansmann, U. (2009). State-of-the-art in parallel computing with R. *Journal of Statistical Software*, 47(1).

Schweinberger, M. (2011). Instability, sensitivity, and degeneracy of discrete exponential families. *Journal of the American Statistical Association*, 106(496):1361–1370.

Snijders, T. A. (1996). Stochastic actor-oriented models for network change. *Journal of Mathematical Sociology*, 21(1-2):149–172.

Snijders, T. A., Van de Bunt, G. G., and Steglich, C. E. (2010). Introduction to stochastic actor-based models for network dynamics. *Social Networks*, 32(1):44–60.

Snijders, T. A. B. (2001). The statistical evaluation of social network dynamics. *Sociological Methodology*, 31(1):361–395.

Snijders, T. A. B. (2002). Markov chain Monte Carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2):1–40.

Snijders, T. A. B. (2005). Models for longitudinal network data. In *Models and Methods in Social Network Analysis*, pages 215–247. Cambridge University Press, Cambridge.

Snijders, T. A. B. (2017). Comment: Modeling of coordination, rate functions, and missing ordering information. *Sociological Methodology*, 47(1):41–47.

Snijders, T. A. B., Pattison, P. E., Robins, G. L., and Handcock, M. S. (2006). New specifications for exponential random graph models. *Sociological Methodology*, 36(1):99–153.

Snijders, T. A. B. and Pickup, M. (2017). Stochastic actor-oriented models for network dynamics. In *The Oxford Handbook of Political Networks*, pages 221–247. Oxford University Press, Oxford.

Stadtfeld, C., Hollway, J., and Block, P. (2017a). Dynamic network actor models: Investigating coordination ties through time. *Sociological Methodology*, 47(1):1–40.

Stadtfeld, C., Hollway, J., and Block, P. (2017b). Rejoinder DyNAMs and the grounds for actor-oriented network event models. *Sociological Methodology*, 47(1):56–67.

Strauss, D. and Ikeda, M. (1990). Pseudolikelihood estimation for social networks. *Journal of the American Statistical Association*, 85(409):204–212.

Thiemichen, S., Friel, N., Caimo, A., and Kauermann, G. (2016). Bayesian exponential random graph models with nodal random effects. *Social Networks*, 46:11–28.

Thiemichen, S. and Kauermann, G. (2017). Stable exponential random graph models with non-parametric components for large dense networks. *Social Networks*, 49:67–80.

Tukey, J. W. (1962). The future of data analysis. *The Annals of Mathematical Statistics*, 33(1):1–67.

Van Duijn, M. A. J., Gile, K. J., and Handcock, M. S. (2009). A framework for the comparison of maximum pseudo likelihood and maximum likelihood estimation of exponential random graph models. *Social Networks*, 1(31):52–62.

Vu, D. Q., Asuncion, A. U., Hunter, D., and Smyth, P. (2011a). Dynamic egocentric models for citation networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 857–864.

Vu, D. Q., Hunter, D., Smyth, P., and Asuncion, A. U. (2011b). Continuous-time regression models for longitudinal networks. In *Advances in Neural Information Processing Systems*, pages 2492–2500.

Wang, P., Pattison, P., and Robins, G. (2013). Exponential random graph model specifications for bipartite networks - a dependence hierarchy. *Social Networks*, 35:211–222.

Wasserman, S. and Pattison, P. (1996). Logit models and logistic regressions for social networks: I. An introduction to markov graphs and p*. *Psychometrika*, 61(3):401–425.

Whitehead, J. (1980). Fitting Cox's regression model to survival data using GLIM. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(3):268–275.

Wood, S. N. (2017). *Generalized Additive Models: An Introduction with R*. Chapman & Hall/Crc Texts in Statistical Science, Boca Raton, 2 rev ed. edition.

Appendix

Network statistics – Notation

Symbol	Definition	Interpretation/Note
$\boldsymbol{Y}$	adjacency matrix	representation of the network as a graph
y_{ij}	realized edges between i and j	$y_{ij} = 1$ if an edge exists between i and j, $y_{ij} = 0$ otherwise
$\boldsymbol{Y} = \boldsymbol{Y}^T$	symmetric adjacency matrix	undirected network
$\boldsymbol{y} \backslash y_{ij}$	graph $\boldsymbol{y}$ except edge between i and j	
N	dimension of $\boldsymbol{Y}$	number of nodes/actors
θ	parameter of interest	corresponding parameter of network statistics
$s(\boldsymbol{y})$	network statistic	e.g. number of edges, 2-stars, triangles
$\Delta_{ij} s(\boldsymbol{y})$	change statistic	difference in network counts when toggling edge y_{ij} from existent to non-existent
d_j	$d_j = \sum_{i=1}^{N} y_{ij}$	degree of node j, number of edges/friends of j
$\boldsymbol{y}^2 =$	$\begin{pmatrix} d_1 & \sum_{k=1}^{N} y_{1k}y_{k2} & \cdots & \cdots \\ & d_2 & \sum_{k=1}^{N} y_{2k}y_{k3} & \cdots \\ & & \vdots & \vdots \end{pmatrix}$	i^{th} and j^{th} element of $\boldsymbol{y}^2$ is the number of links node i has in common with j (friends in common)
$\boldsymbol{y}^3$		diagonal elements of $\boldsymbol{y}^3$ correspond to number of (not unique) triangles

Table A.1: Overview of selected notation. Note that the random variable $\boldsymbol{Y}$ is capitalized, while the observed or realized network is denoted by the corresponding lower case latter.

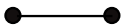
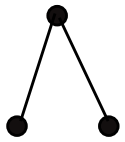
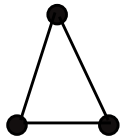
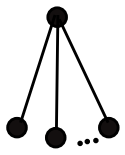
	Network statistic $s(\boldsymbol{y})$		**Change statistic** $\Delta_{ij}s(\boldsymbol{y})$
edge	$\sum_{i=1}^{N}\sum_{j>i} y_{ij} =$ $\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_{ij} =$ $\frac{1}{2}\sum_{j\geq 1} j \cdot d_j(\boldsymbol{y})$		1
2-star	$\sum_{i=1}^{N}\sum_{j>i}\sum_{k=1}^{N} y_{ik}y_{kj} =$ $\frac{1}{2}\sum_{i=1}^{N}\sum_{j>i} y_{ij}\left(\sum_{k\neq i,j} y_{ik} + \sum_{k\neq i,j} y_{jk}\right) =$ $\sum_{j\geq 2}\binom{j}{2} d_j(\boldsymbol{y})$		$\sum_{k\neq i,j} y_{ik} + \sum_{k\neq i,j} y_{jk}$
triangle	$\sum_{i=1}^{N}\sum_{j>i}\sum_{k>j}^{N} y_{ik}y_{kj}y_{ji} =$ $\frac{1}{3}\sum_{i=1}^{N}\sum_{j>i}^{N} y_{ij}\left(\sum_{k\neq i,j}^{N} y_{ik}y_{kj}\right) =$ $\frac{1}{6}\cdot tr(\boldsymbol{y}^3)$		$\sum_{k\neq i,j} y_{ik}y_{jk}$
k-star	$s_k(\boldsymbol{y}) = \sum_{i=1}^{N}\binom{\sum_{l=1}^{N} y_{il}}{k}$		$\binom{\sum_{k\neq i,j} y_{ik}}{k-1} + \binom{\sum_{k\neq i,j} y_{jk}}{k-1}$

Table A.2: Overview and illustration of the network statistic "edge", "2-star", "triangle", and "k-star" with corresponding formula for the network statistic and the change statistic. $tr(\cdot)$ denotes the trace of a matrix.

Dynamic network models – Supplements

Following the ideas of a profile likelihood approach, we insert (1.25) into (1.23) and obtain the profile log-likelihood

$$\begin{aligned} l(\boldsymbol{u}) =& \sum_{d=1}^{m} \left[\sum_{(i,j)\in C_d} \left[\log\left(\frac{|C_d|}{\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right)} \right) + \eta_{ij,d} \right] \right. \\ & \left. - \left(\frac{|C_d|}{\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right)} \right) \cdot \left(\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right) \right) \right] \\ =& \sum_{d=1}^{m} \left[\sum_{(i,j)\in C_d} \left[\log\left(|C_d|\right) - \log\left(\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right) \right) + \eta_{ij,d} \right] - |C_d| \right] \\ =& \sum_{d=1}^{m} \left[|C_d| \cdot \log\left(|C_d|\right) - |C_d| \cdot \log\left(\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right) \right) + \sum_{(i,j)\in C_d} \eta_{ij,d} - |C_d| \right] \\ \propto& \sum_{d=1}^{m} \left[\sum_{(i,j)\in C_d} \eta_{ij,d} - |C_d| \cdot \log\left(\sum_{(i',j')\in O_d} \exp(\eta_{i'j',d}) \right) \right], \end{aligned} \tag{A.33}$$

when the constant terms are deleted.

The contribution to the likelihood for time point $t_{(d)}$ for a Poisson model with tied events can be formulated as following:

$$\begin{aligned} L_d =& \prod_{\{(i,j)|y_{ij,d}=1\}} (\mu_{ij,d})^{y_{ij,d}} \cdot \exp(-\mu_{ij,d}) \cdot \prod_{\{(i',j')|y_{i'j',d}=0\}} (\mu_{i'j',d})^{y_{i'j',d}} \cdot \exp(-\mu_{i'j',d}) \\ =& \prod_{\{(i,j)|y_{ij,d}=1\}} \mu_{ij,d} \cdot \exp(-\mu_{ij,d}) \cdot \prod_{\{(i',j')|y_{i'j',d}=0\}} \exp(-\mu_{i'j',d}) \\ =& \prod_{(i,j)\in C_d} \mu_{ij,d} \prod_{(i',j')\in O_d} \exp(-\mu_{i'j',d}) \\ =& \prod_{(i,j)\in C_d} \mu_{ij,d} \cdot \exp\left(- \sum_{(i',j')\in O_d} \mu_{i'j',d} \right). \end{aligned} \tag{A.34}$$

In the following, detailed steps of deriving the profile likelihood by inserting the corresponding maximum

$$\hat{\alpha} = \log\left(\frac{|C_d|}{\sum_{(i',j')\in O_d} \exp\left(\eta_{i'j',d}\right)} \right)$$

in the likelihood of the generalized Poisson regression model with artificial data (1.30) is given:

$$
\begin{aligned}
L(\boldsymbol{u}) &= \prod_{d=1}^{m} \frac{\exp\left[\sum_{(i,j)\in C_d}\left(\log\left(\frac{|C_d|}{\sum_{(i',j')\in O_d}\exp\left(\eta_{i'j',d}\right)}\right)+\eta_{ij,d}\right)\right]}{\exp\left[\sum_{(i',j')\in O_d}\exp\left(\log\left(\frac{|C_d|}{\sum_{(i'',j'')\in O_d}\exp\left(\eta_{i''j'',d}\right)}\right)+\eta_{i'j',d}\right)\right]} \\
&= \prod_{d=1}^{m} \frac{\exp\left[\sum_{(i,j)\in C_d}\left(\log\left(|C_d|\right)-\log\left(\sum_{(i',j')\in O_d}\exp\left(\eta_{i'j',d}\right)\right)+\eta_{ij,d}\right)\right]}{\exp\left[\frac{|C_d|}{\sum_{(i'',j'')\in O_d}\exp\left(\eta_{i''j'',d}\right)}\cdot\sum_{(i',j')\in O_d}\exp\left(\eta_{i'j',d}\right)\right]} \\
&= \prod_{d=1}^{m} \frac{\exp\left[|C_d|\cdot\log\left(|C_d|\right)\right]\cdot\exp\left[-|C_d|\cdot\log\left(\sum_{(i',j')\in O_d}\exp\left(\eta_{i'j',d}\right)\right)\right]\cdot\exp\left[\sum_{(i,j)\in C_d}\eta_{ij,d}\right]}{\exp\left(|C_d|\right)} \\
&= \prod_{d=1}^{m} \frac{\exp\left[|C_d|\cdot\log\left(|C_d|\right)\right]\cdot\exp\left[\sum_{(i,j)\in C_d}\eta_{ij,d}\right]}{\exp\left(|C_d|\right)\cdot\left(\sum_{(i',j')\in O_d}\exp\left(\eta_{i'j',d}\right)\right)^{|C_d|}}
\end{aligned}
\tag{A.35}
$$

ERRATA

During the review process for the paper *A smooth dynamic network model for patent collaboration data* a discrepancy in the implementation and the notation in the paper of the covariate vector was found. We improved the notation to

$$x_{(3),ij,d} = \sum_{\substack{k \neq i \\ k \neq j}} \mathbb{1}_{\{Y_{ik,d-1}>0\}} + \sum_{\substack{k \neq j \\ k \neq i}} \mathbb{1}_{\{Y_{jk,d-1}>0\}}$$

and

$$x_{(4),ij,d} = \sum_{\substack{k \neq i \\ k \neq j}} \mathbb{1}_{\{Y_{ik,d-1}>0\}} \cdot \mathbb{1}_{\{Y_{jk,d-1}>0\}}.$$

An updated version can be found on *arXiv e-prints, arXiv:1909.00736v2.*

Chapter 2

A note on parallel sampling in Markov graphs

Chapter 2 introduces a framework for parallel sampling in Markov graphs, which provides faster computation. Networks with statistics inducing a conditional independence structure allow independent draws of edges and therefore, parallel computing can be exploited. The simulation study and application can be reproduced using the **R** package **pergm** available on GitHub.

Contributing article:

Bauer, V., Fürlinger, K., and Kauermann, G. (2019). A note on parallel sampling in Markov graphs. *Computational Statistics*, 34(3):1087–1107

Author contributions:

Verena Bauer and Göran Kauermann developed the general framework for parallel sampling in Markov graphs. The basic idea for using Markovian conditional independence structure to simulate networks in parallel and to proof the approximation for log statistics came from Göran Kauermann. Verena Bauer wrote the algorithmic implementation, performed the simulation studies and the example. Karl Fürlinger assisted with some technical questions and performing ideas. All authors contributed and proofread the manuscript.

Supplementary material available at:

`https://github.com/VerenaMaier/pergm`

Chapter 3

A smooth dynamic network model for patent collaboration data

Chapter 3 introduces a framework for modeling dynamic network data based on a profile likelihood approach estimated as an additive Poisson model. The model uses time-stamped data of patent collaborations and includes semiparametric covariates built from network history.

Contributing article:

Bauer, V., Harhoff, D., and Kauermann, G. (2019). A smooth dynamic network model for patent collaboration data. *arXiv e-prints, arXiv:1909.00736*

Author contributions:

Verena Bauer and Göran Kauermann were involved in developing the methodology for estimating a dynamic network for patent collaboration data. Both authors performed the statistical modeling, inference and application to the data. Verena Bauer wrote the implementation for the estimation. Most of the manuscript was written by Verena Bauer and Göran Kauermann. Dietmar Harhoff provided the patent data set and helpful input. All authors were involved in proof-reading.

A smooth dynamic network model for patent collaboration data

Verena Bauer[1], Dietmar Harhoff[2], and Göran Kauermann[1]

[1]*Department of Statistics, Ludwig-Maximilians-Universität München, Germany.*

[2]*Max Planck Institute for Innovation and Competition, Germany.*

September 4, 2019

Abstract

The development and application of models, which take the evolution of networks with a dynamical structure into account are receiving increasing attention. Our research focuses on a profile likelihood approach to model time-stamped event data for a large-scale network applied on patent collaborations. As event we consider the submission of a joint patent and we investigate the driving forces for collaboration between inventors. We propose a flexible semiparametric model, which allows to include covariates built from the network (i.e. collaboration) history.

Keywords: *profile likelihood, network data, event data, patent data, penalized spline smoothing, social network analysis*

1 Introduction

The analysis of network data has seen increasing interest in the recent years. Many network data thereby contain a dynamic structure, be it the development of network ties over time or observations of the network at different time points. Such data structures have led to numerous extensions of classical

network models. A first paper in this direction is from Robins and Pattison (2001) who propose temporal dependence in an Exponential Random Graph Model (ERGM). The idea was generalized in Hanneke et al. (2010) towards temporal Exponential Random Graph Models (tERGM). The principle idea behind the models is to include the network history as covariates in the model. This in turn forms a Markov Chain of networks. The model class has been extended and generalized in various ways. Leifeld et al. (2018) focus on the implementation and added bootstrap methods for evaluating uncertainty. Krivitsky and Handcock (2014) decomposed the network dynamics into the formation of new edges and the dissolution of existing edges leading to the separable temporal Exponential Random Graph Model (stERGM).

A different strand of dynamic network models arise if time is considered as continuous. Holland and Leinhardt (1977) develop a dynamic model for social networks based on a continuous-time Markov process. Snijders (2005) and Snijders et al. (2010) extended this towards so-called stochastic actor-oriented models. This model is based on the assumption that the evolution of the network occurs as the consequence of small changes induced by the actors. It is further assumed that the observed network derives from a Markov process evolving in continuous time, though the network is observed only at discrete time points. Greenan (2015) extends this approach by combining it with hazard function estimation and Cox regression models for duration time models (Cox, 1972). Our approach is in line with Greenan (2015) but we extend the model class further by including non-linear time dynamics.

Our paper focuses on a profile likelihood approach modeling time-stamped event data for a large-scale network. The data describe the collaboration of inventors based on joint patents. The successful submission of a new patent is thereby considered as the event of interest and the number of joint patents of two inventors provides network based count data. The analysis of event data, in special cases also referred to as survival analysis, is a well developed field in statistics. In this paper we extend this field towards dynamic network data (see also Brandes et al., 2009; Block et al., 2018).

The idea of using a partial likelihood approach for network data is not new. Perry and Wolfe (2013) estimate a Cox multiplicative intensity model for a directed e-mail network. The model can incorporate covariates that depend on the history of the process modeling homophily, network effects and multiplicity. Vu et al. (2011) propose a continuous-time regression model for time-stamped network data. Estimation routines use an efficient partial likelihood approach focusing on large networks. The authors include time-

variant and time-independent network statistics based on the history. This approach is applied in Vu et al. (2011) to a dynamic egocentric model for citation networks based on a multivariate counting process. The models are based on the Relational Event Model (REM) from Butts (2008) who introduced a flexible likelihood-based framework for social actions (also called relational event). This approach includes the event history in modeling the behavior and its complex dependence structure. While Butts (2008) focuses on modeling the rate of relational events, Brandes et al. (2009) extend this framework to weighted events where the influence of the quality of actions is in the center of interest.

In the cited papers above, all covariate effects are included linearly in the model. We propose a semiparametric approach for modeling the covariates in a more flexible way. We follow the idea of penalized spline smoothing as proposed in Ruppert et al. (2003) (see also Eilers and Marx, 1996; Ruppert et al., 2009). The basic idea is to replace linear functions by spline based functions and to achieve smoothness, a penalty is imposed on the spline coefficient. Penalized spline smoothing can be considered as the state-of-the-art smoothing technique where we refer to Wood (2017) for a general discussion in the framework of (generalized) regression models.

The paper is organized as follows. First, in Section 2 we introduce the patent data set of the application with some basic ideas and descriptive statistics. Then, in Section 3, we outline the general framework of the structure of the underlying patent data set, give a short introduction into the notation and motivate the construction of the covariates from the network history. Then we take a closer look on inference and how the model based on a profile likelihood approach can be estimated with linear covariates and its extensions to penalized spline smoothing. We give a brief outlook on computational issues, before we apply in Section 4 these techniques to the example data set. Finally, we summarize the most important issues.

2 Patent data

We will first introduce the patent data in detail before describing the model in the next section. We consider all patent applications submitted to the European Patent Office (EPO) and the German Patent and Trademark Office (Deutsches Patent- und Markenamt, DPMA), which listed at least one inventor with an address on German territory between 2000 and 2013. This

selection should yield a comprehensive database of all inventions filed in patent applications by German inventors. It is possible that some inventors may have submitted applications directly to patent offices of other countries, but in practice such cases are extremely rare, since the invention would not enjoy patent protection in the inventors home country. The data were extracted from the PATSTAT database of the European Patent Office (version October 2018). The data consist of patented inventions from different technological areas. For each patent we have information about the submission day and for most inventors geographic coordinates of their registered home address at the time of submission is given. We assume that the inventor location stays the same until new information due to new patent submissions is given.

We focus on four technological areas – *basic communication processes* (105), *IT-methods* (107), *analysis of biological material* (111) and *food chemistry* (118) – with different numbers of inventors, patents and therefore network densities. Some exploratory information is provided in Table 1. The

area		basic communication processes (105)	IT-methods (107)	analysis of biological material (111)	food chemistry (118)
number of ...					
inventors		4089	3616	4923	2993
patents		3664	1707	2731	2078
single ownership patents		1104	194	342	427
inventor pairs		5825	12713	9604	5412
patents per inventor	mean	2.02	1.37	1.72	1.86
	max	52	16	83	36
inventors per patent	mean	2.25	2.9	3.11	2.68
	max	14	93	13	17

Table 1: Summary statistics of different technological areas for the time period of 14 years. The statistics are summarized and averaged over time.

table summarizes the selected inventor networks for the whole time period and Figure 8 in the Appendix explicitly visualizes two of them. The *basic communication processes* technological area has the highest number of

patents, but quite less inventor pairs applying for a patent during the observed time period of 14 years. In this area the number of single ownership patents is three till six times higher than for the other fields. On average over all technological areas the inventors applied for 1.7 patents, whereas one person in area 111 (*analysis of biological material*) is involved in 83 patents. The number of involved inventors per patent varies between one and 93, with an overall average of 2.7 inventors.

As time stamp we choose the earliest filing date, which is aggregated on a monthly basis. To adjust for incomplete data, we select only patents from the full years 2000 till the end of 2013, resulting in 168 months. We are interested in inventors that jointly apply for patents. Therefore, we only include inventors with at least one joint patent. Note, that there are of course single ownership patents in the data sets if the inventor has other joint patents.

Beside the number of inventors (about three to five thousand), also the number of patents in total and patents with single ownerships vary in the four technologies. Noticeable is that the number of observed inventor pairs applying for a patent during the observed time period of 14 years is quite small in comparison to the possible number of pairs ($N(N-1)/2$). In other words the density of the networks is small. Furthermore, in all areas the mean number of patents per inventor is quite low. Therefore and due to content-related reasons, we restrict the actor sets to sets of active inventors in a period of three years. To do so, we split the data sets from above into four periods of three years starting from the beginning of 2002. We will analyse each time interval separately. The first two years of data from 2000 to the end of 2001 are used as "burn-in" for the covariates. We include only active inventors in the option set. An active inventor is defined as a person with at least one patent within the observed time period of three years (e.g. inventor 4 or 7 in Figure 1), or at least one patent within and one beyond the time period (e.g. inventor 6 or 8 in Figure 1), or at least one patent before and one after the time period (e.g. inventor 5 in Figure 1). We want to point out, that the covariates are based on a five years retrospective interval, meaning that the inventors' history beyond the five years is ignored in the calculation of the covariates.

In the application we focus on a model with four network specific covariates: the overall sum of patents of each inventor pair within the considered inventors' history ("*patents_ij*"), the number of joint patents per pair ("*joint_patent*"), the number of inventors that hold a joint patent with inven-

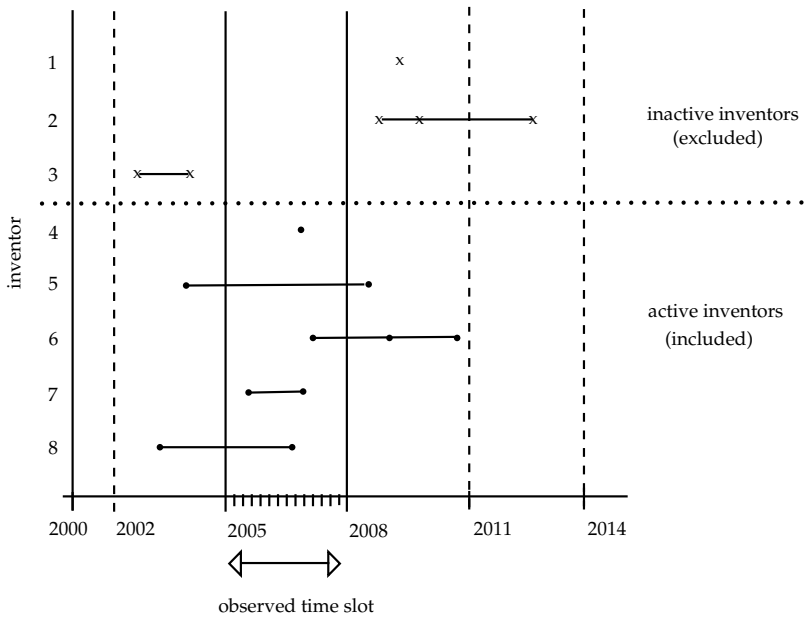


Figure 1: Outline of data management. The time period from 2000 till the end of 2013 is divided in four periods ($2002-2004$, $2005-2007$, $2008-2010$ and $2011-2013$) of three years each. The data is aggregated on a monthly grid. The years 2000 and 2001 are used as a burn-in time for the covariates. If e.g. the observed time period from the beginning 2005 till the end of 2007 is selected, only active inventors, like inventor $4-8$ are included in the option set. Inventors with a patent structure visualized with 'x' in this figure, are excluded.

tor i or j ("*2-star*") and the number of inventors that jointly hold a patent with i and j ("*triangle*"). These numbers differ for the technological areas and time periods. Table 2 gives an overview for the covariate ranges due to the variation between the different time periods. Furthermore, we see that the number of inventors varies between 753 and 1466 depending on the technological area and selected time period. The realized number of edges in the considered networks at the end of the observation period ranges between 1188 and 6306. Figure 2 visualizes two networks of the *food chemistry* area (118) for two selected time periods.

area		basic communication processes (105)	IT-methods (107)	analysis of biological material (111)	food chemistry (118)
no. of inventors		1018 - 1396	790 - 959	1095 - 1466	753 - 949
no. of edges		1285 - 2065	1234 - 6306	2247 - 2853	1188 - 1711
density		0.002 - 0.003	0.003 - 0.016	0.003 - 0.004	0.004 - 0.004
"*patents_ij*"	mean	2.35 - 3.16	1.4 - 1.91	2.17 - 2.62	2.11 - 2.64
	max	49 - 58	16 - 26	111 - 118	27 - 45
"*joint_patent*"	mean	0	0 - 0.01	0	0
	max	6 - 13	3 - 7	20 - 30	7 - 13
"*2-star*"	mean	3.69 - 5.13	3.28 - 23.29	5.63 - 6.46	4.98 - 5.99
	max	62 - 113	39 - 410	111 - 156	67 - 133
"*triangle*"	mean	0.01 - 0.01	0.01 - 0.9	0.01 - 0.04	0.02 - 0.02
	max	53 - 210	23 - 283	496 - 855	58 - 162

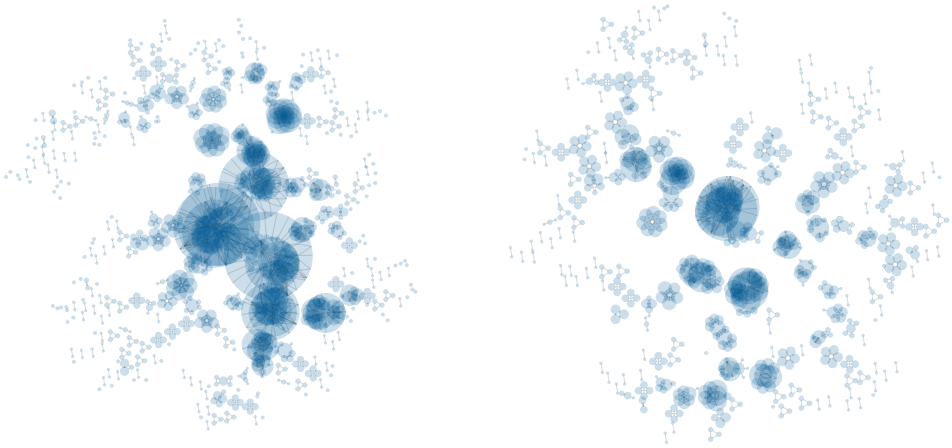
Table 2: Summary statistics for different technological areas and time periods. There are shown the ranges for the different time periods per area of number of nodes, edges and covariates. The covariates are the number of patents of i and j ("*patents_ij*"), the number of joint patents of i and j ("*joint_patent*"), the number of inventors that hold a joint patent with i or j ("*2-star*") and the number of inventors that jointly hold a patent with i and j ("*triangle*").

3 Poisson process network model for count data

3.1 Model description

We motivate the model by directly referring to our data example. Let Z_r be a patent indexed with a running number $r = 1, ..., R$. Each patent can be defined through the following attributes:

- t_r = time point at which patent r was successfully submitted
- I_r = index list of inventors on patent r
- B_r = technological area in which patent r is submitted
- z_r = additional covariates like geocoordinates of registered addresses of all inventors



(a) Time period 2: years 2005 - 2007 (b) Time period 4: years 2011 - 2013

Figure 2: Visualization of two time periods of the inventor network for *food chemistry* (118). Vertex size represent nodal degree. Colouring is transparent to better examine the clusters. The layout uses maximal connected components and applies the layout separately.

For a set of actors (inventors) $A = \{1, ..., N\}$ we define with $\boldsymbol{Y}(t) \in \mathbb{R}^{N \times N}$ the matrix valued Poisson process counting the number of (joint) patents. To be specific, let

$$\begin{aligned} Y_{ij}(t) &= \text{cumulated number of joint patents of inventor } i \text{ and } j \\ &\qquad \text{up to five years at time } t \\ &= \#\{r : (i,j) \in I_r, t_r \leq t, r = 1, ..., R\} \end{aligned}$$

for $i, j = 1, ..., N$, where $Y_{ii}(t)$ defines the number of patents of inventor i including single ownership patents. We are primarily interested in the number of joint patents of a retrospective history of five years and observe the process at the time points where one or more joint patents have been successfully submitted. We define with $Y_{ij,d} = Y_{ij}(t_{(d)})$ the evolving process, where $t_{(1)}, t_{(2)}, \ldots, t_{(m)}$ is the discretized version of time at which patents have been submitted. We model the intensity of the above process as

$$\lambda_0(t) \exp\left(x_{ij}(t)\beta\right) \tag{1}$$

where $\lambda_0(t)$ is the baseline hazard of the process and $x_{ij}(t)$ is the covariate process, which will be defined in the following section. We assume for sim-

plicity that both, the baseline hazard as well as the covariate process are piecewise constant between the observed time points, that is

$$\begin{aligned} \lambda_0(t) = &\quad \lambda_d \quad \text{for } \ t \in (t_{(d-1)}, t_{(d)}] \\ x_{ij}(t) = &\quad x_{ij,d} \quad \text{for } \ t \in (t_{(d-1)}, t_{(d)}]. \end{aligned}$$

This leads to the log-likelihood function

$$l(\lambda_1, \ldots, \lambda_m, \beta) = \sum_{d=1}^{m} \left[\sum_{(i,j) \in C_d} (\log \lambda_d + x_{ij,d}\beta) - \lambda_d \cdot \left(\sum_{(i',j') \in O_d} \exp\left(x_{i'j',d}\beta\right) \right) \right] \quad (2)$$

where C_d is the index set of events at time point $t_{(d)}$,

$$C_d = \{(i,j) : j > i; Y_{ij,d} > Y_{ij,d-1}\}$$

and O_d is the "option" set, that is the set of inventor pairs that could submit a joint patent. This option set can be regarded as the set of inventors who are able to work together. In our application this restriction occurs from being in the same technological area and being an active inventor like defined above in the description of the data. Maximizing the above likelihood with respect to $\lambda_1, \ldots, \lambda_m$ yields

$$\hat{\lambda}_d = \frac{|C_d|}{\sum_{(i',j') \in O_d} \exp\left(x_{i'j',d}\beta\right)}$$

and inserting this in (2) provides the profile log-likelihood

$$l(\beta) = \sum_{d=1}^{m} \left[\sum_{(i,j) \in C_d} x_{ij,d}\beta - |C_d| \log \left(\sum_{(i',j') \in O_d} \exp(x_{i'j',d}\beta) \right) \right], \quad (3)$$

omitting all constant terms. In principle and based on the Poisson process we observe at each time point a single patent submission only, resulting in maybe more edges at t if more than two inventors are involved in a patent submission. In practice, however, the time points are discretized so that at each discrete valued time point $t_{(d)}$ we may observe more than just one submitted patent. Note that this may be caused by one (or more) patent submitted at the same discrete time point but with more than two inventors

as patent holders or by more than one patent submitted at time point $t_{(d)}$ by different inventor pairs.

Let $\boldsymbol{Y}_d = (Y_{ij,d})$ be the process network matrix. The profile log-likelihood (3) is also obtained if we assume that the probability for a single change $Y_{ij,d} = y_{ij,d-1} + 1$ is proportional to

$$P(\boldsymbol{Y}_d = \boldsymbol{Y}_{d-1} + 1_{ij}) \propto \exp(x_{ij,d}\beta)$$

where 1_{ij} refers to an increment of 1 in entry $Y_{ij,d}$ and $x_{ij,d}$ is a vector of covariates calculated from the previous process matrix $\boldsymbol{Y_{d-1}}$. If $|C_d| = 1$, i.e. only a single patent was submitted by i and j at time point $t_{(d)}$, we obtain

$$P(\boldsymbol{Y}_d|\boldsymbol{Y}_{d-1}) = \frac{\exp(x_{ij,d}\beta)}{\sum_{(i',j')\in O_d}\exp(x_{i'j',d}\beta)}, \tag{4}$$

where O_d being an inventor tuple from the "option" set. If $|C_d| > 1$ we approximate (4) with

$$P(\boldsymbol{Y}_d|\boldsymbol{Y}_{d-1}) = \frac{\prod_{(i,j)\in C_d}\exp(x_{ij,d}\beta)}{\left[\sum_{(i',j')\in O_d}\exp(x_{i'j',d}\beta)\right]^{|C_d|}}. \tag{5}$$

We can now easily derive the log-likelihood from equation (3) and obtain the score function

$$s(\beta) = \sum_{d=1}^{m}\left[\sum_{(i,j)\in C_d} x_{ij,d}^T - |C_d| \quad \frac{\sum_{(i',j')\in O_d} x_{i'j',d}^T \exp(x_{i'j',d}\beta)}{\sum_{(i',j')\in O_d}\exp(x_{i'j',d}\beta)}\right].$$

Defining

$$\pi_{i'j',d} = \frac{\exp(x_{i'j',d}\beta)}{\sum_{(k',l')\in O_d}\exp(x_{k'l',d}\beta)}$$

allows to write the second order derivative

$$J(\beta) = -\sum_{d=1}^{m}|C_d|\left[\sum_{(i',j')\in O_d} x_{i'j',d}^T x_{i'j',d}\pi_{i'j',d} - \left(\sum_{(i',j')\in O_d} x_{i'j',d}^T\pi_{i'j',d}\right)^T\left(\sum_{(i',j')\in O_d} x_{i'j',d}^T\pi_{i'j',d}\right)\right]. \tag{6}$$

In the survival model context formula (5) is also known as Breslow approximation (see Breslow, 1974).

3.2 Covariates

The covariate vector $x_{ij,d}$ is built from the network history and exogenous covariates. We describe the network related covariates first. First, we take the total number of patents of inventor i and j at time point $t_{(d-1)}$. That is

$$x_{(1),ij,d} = Y_{ii,d-1} + Y_{jj,d-1}.$$

In the application we refer to it as "*patents_ij*". Moreover, the number of previous "*joint_patents*" of inventor i and j is included as covariate, which is calculated through

$$x_{(2),ij,d} = Y_{ij,d-1}.$$

Furthermore, a so-called 2-star statistic ("*2-star*") is included, which expresses the number of inventors that hold a joint patent with inventor i or j. This is obtained through

$$x_{(3),ij,d} = \sum_{\substack{k \neq i \\ k \neq j}} Y_{ik,d-1} + \sum_{\substack{k \neq j \\ k \neq i}} Y_{jk,d-1}.$$

A common choice in network analysis are also "*triangle*" statistics. This counts the number of inventors that jointly hold a patent with i and j:

$$x_{(4),ij,d} = \sum_{\substack{k \neq i \\ k \neq j}} Y_{ik,d-1} \cdot Y_{jk,d-1}.$$

We restrict our analysis to these four structural covariates, which are visualized in Figure 3.

As exogenous quantity in our application we include the inventor-pair-specific distance in kilometers, that is

$$x_{(5),ij,d} = ||s_{i,d} - s_{i,d}||$$

where $s_{i,d}$ are the geocoordinates of the address of inventor i and $s_{j,d}$ accordingly and $||\cdot||$ denotes the Euclidean distance. We assume that the inventors do not move until new location information on the basis of submitting a new patent becomes available. Due to only few data points, distances over 1000 kilometers are set to 1000 kilometers.

Toy network graph at time $t_{(d-1)}$

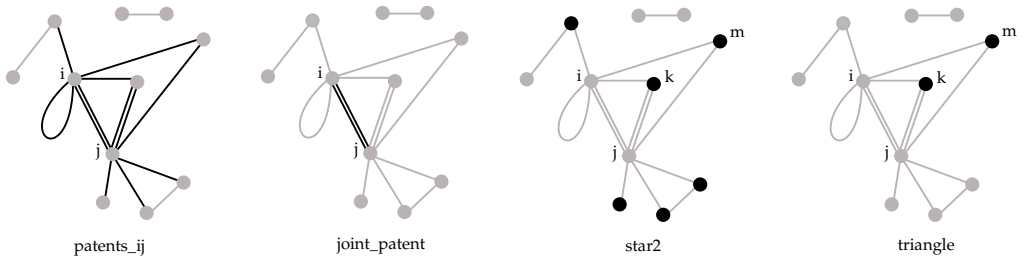


Figure 3: Visualization of covariates from network history of a toy network graph: Number of patents of inventor i and j with $x_{(1),ij,d} = 6 + 8$ (black edges), including self-loops (single ownership patents) and multiple patents (first panel). Number of joint patents of inventor i and j with $x_{(2),ij,d} = 2$ (black edges), counting the number of edges of i and j (second panel). Number of inventors that hold a joint patent with inventor i or j with $x_{(3),ij,d} = 3 + 6$ (black nodes), counting the joint inventors k and m twice and counting k twice because of two previous joint patents between k and j (third panel). Number of inventors that jointly hold a patent with i and j with $x_{(4),ij,d} = 1 \cdot 2 + 1 \cdot 1$ (black nodes), counting k twice because of a multi-patent (fourth panel).

3.3 Semiparametric Estimation

We now extend the model towards penalized smoothing techniques to obtain more flexibility. We therefore replace the linear predictor $\eta_{ij,d} = x_{ij,d}\beta$ in (3) through the additive nonparametric setting

$$\eta_{ij,d} \quad = \quad m_{(1)}(x_{(1),ij,d}) + m_{(2)}(x_{(2),ij,d}) + \dots .$$

and $\lambda_0(t) = \exp(m_0(t))$ being a smooth term of the time. Here $m(\cdot)$ are smooth but otherwise unspecified functions, which extend the linear effects. To achieve identifiability of the model we postulate $m_{(p)}(0) = 0$ for $p = 1, 2, \dots, P$. To estimate the unknown functions we employ B-splines and replace $m(x)$ by

$$m(x) = \sum_k B_k(x)u,$$

where $B_k(\cdot)$ is a K dimensional B-spline basis (see de Boor, 1978; Wood, 2017).

For simplicity of notation we now replace the index pair (i, j) by a single index l running from 1 to $n = \frac{N \cdot (N-1)}{2}$. Consequently, we can rewrite

$$\eta_{l,d} \quad = \quad m_{(1)}(x_{(1),l,d}) + m_{(2)}(x_{(2),l,d}) + \dots$$

With this in matrix notation we get

$$\begin{aligned}\boldsymbol{\eta}_d &= B_{(1),d}u_{(1)} + B_{(2),d}u_{(2)} + \ldots \\ &= \boldsymbol{B}_d\boldsymbol{u}\end{aligned}$$

where $B_{(1),d}$ is the basis matrix for the first covariate built from rows $B_{(1)}(x_{(1),l,d})$ for $l = 1, \ldots, N(N-1)/2$ and analogous definitions for $B_{(2),d}$ etc. Setting $\boldsymbol{B}_d = (B_{(1),d}, B_{(2),d}, \ldots)$ and $\boldsymbol{u}^T = (u_{(1)}^T, u_{(2)}^T, \ldots)$ provides the final notation.

In the estimation we need to take the constraint $m_{(p)}(0) = 0$ for the covariates built from network history into account. This constraint means that there is no effect to estimate if the independent variables are zero. We make use of this point constraint, which is an alternative to the sum-to-zero identifiability constraints for smooth terms, which is used for the exogenous covariate.

With this notation we can reformulate the profile likelihood in (3) as following:

$$\sum_{d=1}^{m} \left[(\boldsymbol{B}_d\boldsymbol{u})^T \cdot \mathbb{1}_{C_d} - |C_d| \cdot \log \left[\exp(\boldsymbol{B}_d\boldsymbol{u})^T \cdot \mathbb{1}_{[n\times 1]} \right] \right], \tag{7}$$

where $\mathbb{1}_{C_d}$ is a vector defined as

$$\mathbb{1}_{C_d} = \begin{cases} 1, & \text{if } l = (i,j) \in C_d \\ 0, & \text{otherwise,} \end{cases}$$

$\mathbb{1}_{[n\times 1]}$ is a vector of ones of length n and $\boldsymbol{B}_d$ is a matrix of $n \times P \cdot K$.

Following Eilers and Marx (1996) we use a large number of knots but regularize the estimation by introducing a roughness penalty (see also Ruppert et al., 2003, 2009). This leads to the penalized smooth log-likelihood

$$l^{pen}(\boldsymbol{u}) = l(\boldsymbol{u}) - \frac{1}{2} \cdot \boldsymbol{u}^T \boldsymbol{K}(\boldsymbol{\lambda})\boldsymbol{u}, \tag{8}$$

where $\boldsymbol{K}(\boldsymbol{\lambda})$ is a second-order penalty matrix. The smoothing parameter vector $\boldsymbol{\lambda}$ penalizes large differences in adjacent basis coefficients. Details are provided in the Appendix C.

3.4 Computational issues

In principle, computation is straight forward, because we can derive the corresponding likelihood function and its derivatives. Nevertheless, we have

a huge option set of inventors pairs for each time point. A data set with N inventors results in $N(N-1)/2$ times T time points and therefore in about 18 million data points for e.g. $N = 1000$ inventors and $T = 38$ months.

For estimating the parameters, we need to maximize the profile log-likelihood of equation (3) for the linear framework and the penalized smooth log-likelihood (8) for semiparametric estimation. To do so, we can make use of the flexible toolbox available in the package `mgcv` (see Wood, 2011, for further information) in the software `R` (R Core Team, 2017). This becomes possible since a Cox proportional hazards model can be estimated as a linear or additive Poisson model in case of smooth predictors (see Whitehead, 1980). To do so, an ordinary time-to-event data representation is necessary (see Tutz et al., 2016). At each event time $t_{(d)}$ an artificial response variable $y_{ij,d}$ for every inventor pair from the option set is included with $y_{ij,d} = 1$ if a patent was submitted at time $t_{(d)}$ or $y_{ij,d} = 0$ if not. The generalized Poisson model can than be fitted to that artificial data with the time variable as intercept and including offsets $o_{ij,d}$. Friedman et al. (1982) showed that with this data structure the log-likelihood resembles the log-likelihood of the Poisson regression model, which we utilize here.

With this simple data transformation we can take advantage of the whole machinery of `mgcv` with the methods and algorithms for model extensions like including spatial, random, or nonlinearly time-varying effects (see Bender et al., 2018). Furthermore, a variety of automated smoothing approaches including constraints can be used. The estimation of the smoothing parameters are solved by using the Un-Biased Risk Estimator (UBRE) criterion.

4 Data analysis

4.1 Linear estimation

We first estimate a model with linear effects for the four technology areas for the different time periods. All models include the above mentioned structural covariates "*patents_ij*", "*joint_patent*", "*2-star*", and "*triangle*", and the exogenous covariate "*distance [100 km]*". The quite small standard errors can be explained with the huge data set. Figure 4 compares the estimates for the four considered time periods. The different technology areas show more or less the same behaviour. The biggest effect can be seen for the variable *joint_patent*. The more joint patents two inventors have, the more likely they

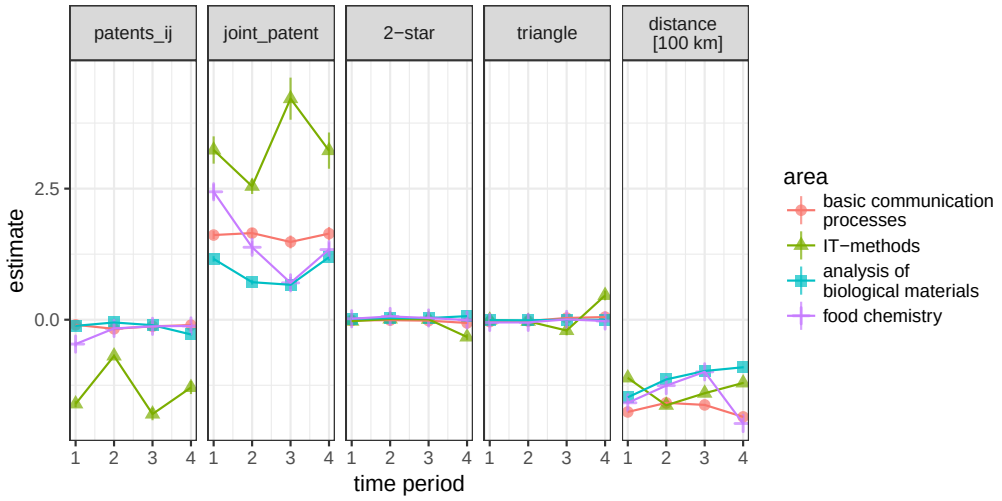


Figure 4: Estimates for different covariates, technological areas and time periods. For each of the four areas and four covariates we have four estimates for the time periods with the corresponding errorbars (standard error $\times$ 2).

collaborate in the future. The estimates for 2-star and triangle are quite small. A negative or quite small effect results for the number of patents of inventor i and j. This means that a lot of own patents (with other inventors or even single inventor patents) reduce the potential of new joint patents. The distance in 100 kilometers has a negative effect on the patents meaning that inventors with regional proximity are collaborating more likely.

4.2 Semiparametric estimation

This subsection visualizes the estimated smooth effects for our model with linear effects replaced by spline based fits for "*patents_ij*", "*joint_patent*", "*2-star*", "*triangle*", and "*distance*". In Figure 5 the model is estimated exemplary for the technological area *food chemistry* (118) and the second time period. We see that the sum of patents of inventor i and j has a negative effect, whereas the number of joint patents has a positive and stronger effect. This means that if the inventors have already submitted several own patents (with other inventors or even single inventor patents) their affinity of being involved in new patents decrease. On the other hand, if the inventor pair has already joint patents in the past, they are more likely to work together in future. Both effects are bounded. The effects of the structural statistics like

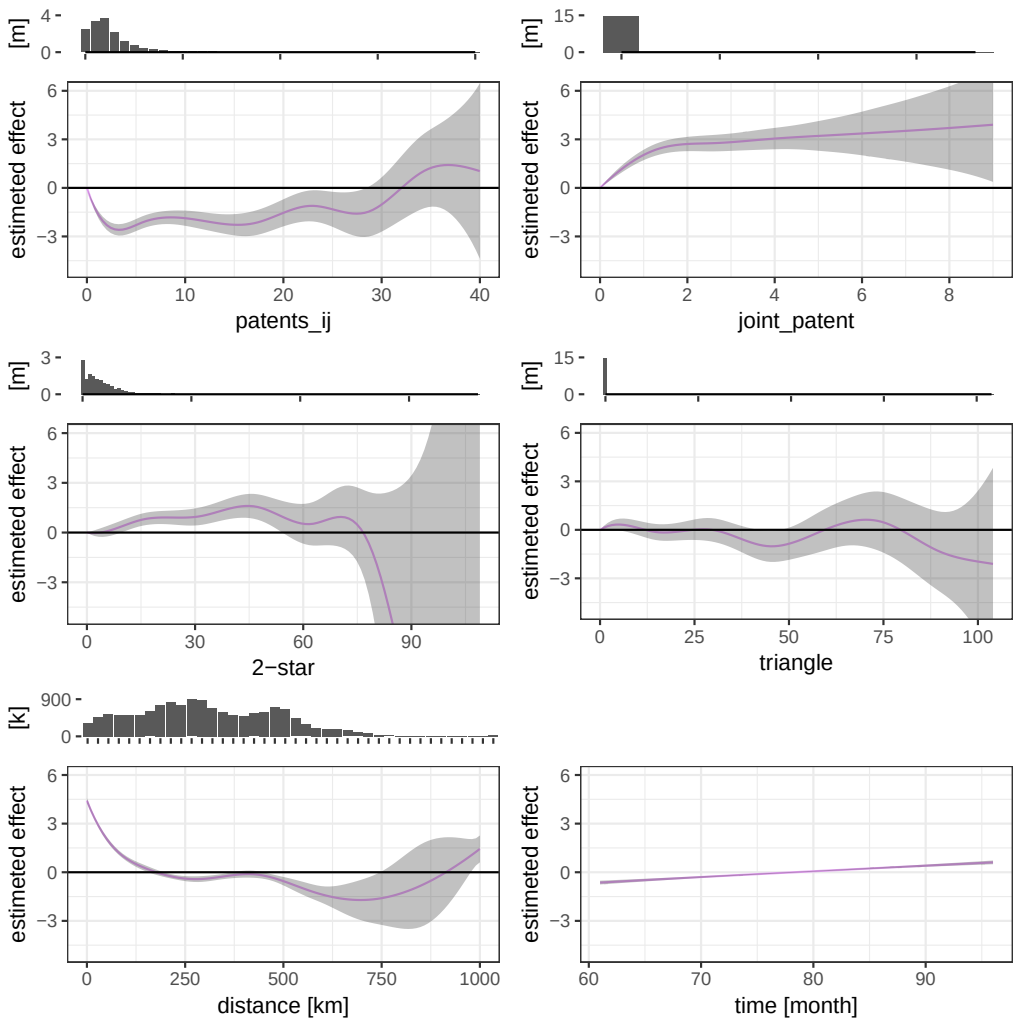


Figure 5: Estimated smooth effects for *food chemistry* (118) area and second time period.

the number of inventors that hold a joint patent with inventor i or j or the number of inventors that jointly hold a patent with i and j do not show a significant tendency. Moreover, the geodesic distance of two inventors plays an important rule. There is a larger positive effect for small distances, which decreases with increasing distance. For distances larger than 250 kilometres the effect is almost zero or negative. This means that if there is a certain distance between the inventors, it does not matter how many kilometers exactly.

Figure 6 visualizes the positive effects of *"joint_patent"* for the four time periods. Each time period lasts 36 months. The tendency of the effects is the

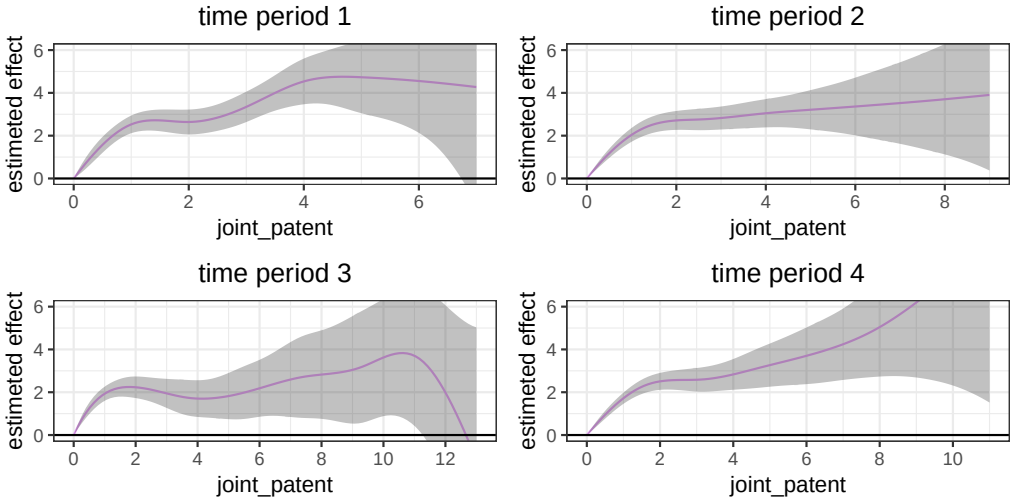


Figure 6: Estimated smooth effects for *"joint_patent"* of *food chemistry* (118) area and different time periods.

same for all periods, while there is a steep increase at the beginning, which then becomes bounded. In period three and four the effect decreases and increases, respectively, at the end of the observation period. This should not be interpreted too strictly as the frequency of more than 10 joint patents is quite low. We can see similar behaviours for the other areas (see Appendix).

4.3 Model validation

To evaluate the model performance, we select the third time period of *food chemistry* area (118). This data set is divided into a training set, which contains the first two years and a test data based on the last year of that time period. We estimate our model on the training set and predict the probability of present ties in the test data. The predicted probabilities are most of the time almost zero and have a mean of 0.0000069. The maximal value is 0.05. This result is not surprising as the network is very sparse, only a proportion of 0.0001 patents are present. This is visualized in Figure 7 in the left panel, where the predicted values are plotted separately by the true observed ties. The (red) dashed added line shows the optimal cut point for

the classification into present and absent ties for the test year and is based on sensitivity and specificity. The right panel shows the precision recall curve for the test set. Recall, or also called true positive rate (sensitivity), is defined as the fraction of present patents that are correctly predicted by the model. Precision measures the fraction of predicted present patents that actually occur. This measurement is useful in model validation, like in this case, where the number of existent ties are quite small in comparison to all possible ties (see Maalouf and Trafalis, 2011). The visualized precision recall curve in Figure 7 has a maximal value around 0.13 and drops down to zero quite fast. This indicates that an optimal cut point for the predicted probabilities needs to be quite low.

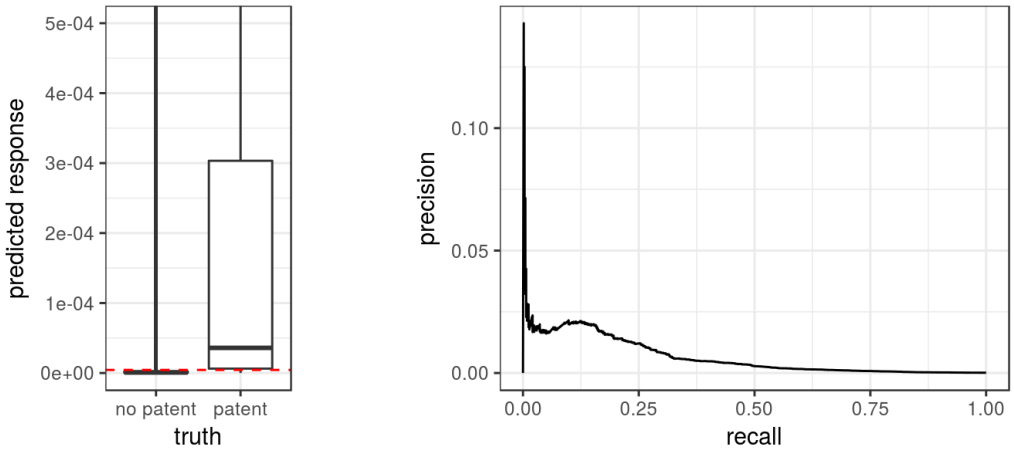


Figure 7: Predicted versus true present and absent ties (patent or no patent) where the (red) dashed line shows the optimal cut point for the classification (left panel). Precision and recall curve (right panel).

5 Conclusion

In this paper we propose a flexible approach to model large-scale dynamic network data with structural and exogenous covariates. Our approach is based on a profile likelihood method exploiting well-established estimation routines. We applied this idea to a large data set of patents submitted jointly by inventors from Germany between 2000 and 2013. We showed advantages of including covariates in a semiparametric and therefore flexible way. The

results show the driving forces in collaboration of inventors and demonstrate their behaviour over time. The models could be fitted with standard software and therefore invite to be used in other data constellations as well.

References

Bender, A., A. Groll, and F. Scheipl (2018). A generalized additive model approach to time-to-event analysis. *Statistical Modelling 3*, 299–321.

Block, P., J. Koskinen, J. Hollway, C. Steglich, and C. Stadtfeld (2018). Change we can believe in: Comparing longitudinal network models on consistency, interpretability and predictive power. *Social Networks 52*, 180–191.

Brandes, U., J. Lerner, and T. A. Snijders (2009). Networks Evolving Step by Step: Statistical Analysis of Dyadic Event Data. In *Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in*, pp. 200–205. IEEE.

Breslow, N. (1974). Covariance Analysis of Censored Survival Data. *Biometrics 30*(1), 89–99.

Butts, C. T. (2008). A relational event framework for social action. *Sociological Methodology 38*(1), 155–200.

Cox, D. R. (1972). Regression Models and Life-Tables. *Journal of the Royal Statistical Society. Series B (Methodological) 34*(2), 187–220.

de Boor, C. (1978). *A Practical Guide to Splines*, Volume 27. New York: Springer.

Eilers, P. H. and B. D. Marx (1996). Flexible Smoothing with B-splines and Penalties. *Statistical science 11*(2), 89–102.

Friedman, M. et al. (1982). Piecewise exponential models for survival data with covariates. *The Annals of Statistics 10*(1), 101–113.

Greenan, C. C. (2015). Diffusion of Innovations in Dynamic Networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society) 178*(1), 147–166.

Hanneke, S., W. Fu, E. P. Xing, et al. (2010). Discrete Temporal Models of Social Networks. *Electronic Journal of Statistics 4*, 585–605.

Holland, P. W. and S. Leinhardt (1977). A Dynamic Model for Social Networks. *The Journal of Mathematical Sociology 5*(1), 5–20.

Krivitsky, P. N. and M. S. Handcock (2014). A Separable Model for Dynamic Networks. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*(1), 29–46.

Leifeld, P., S. J. Cranmer, and B. A. Desmarais (2018). Temporal Exponential Random Graph Models with btergm: Estimation and Bootstrap Confidence Intervals. *Journal of Statistical Software 83*(1), 1–36.

Maalouf, M. and T. B. Trafalis (2011). Rare Events and Imbalanced Datasets: An Overview. *International Journal of Data Mining, Modelling and Management 3*(4), 375–388.

Perry, P. O. and P. J. Wolfe (2013). Point Process Modelling for Directed Interaction Networks. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 75*(5), 821–849.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Robins, G. and P. Pattison (2001). Random Graph Models for Temporal Processes in Social Networks. *The Journal of Mathematical Sociology 25*(1), 5–41.

Ruppert, D., M. P. Wand, and R. J. Carroll (2003). *Semiparametric Regression*. Cambridge: Cambridge University Press.

Ruppert, D., M. P. Wand, and R. J. Carroll (2009). Semiparametric regression during 20032007. *Electronic Journal of Statistics 3*, 1193–1256.

Snijders, T. A. (2005). Models for Longitudinal Network Data. *Models and methods in social network analysis 1*, 215–247.

Snijders, T. A., G. G. Van de Bunt, and C. E. Steglich (2010). Introduction to Stochastic Actor-Based Models for Network Dynamics. *Social networks 32*(1), 44–60.

Tutz, G., M. Schmid, et al. (2016). *Modeling discrete time-to-event data*. Springer International Publishing.

Vu, D. Q., A. U. Asuncion, D. Hunter, and P. Smyth (2011). Dynamic Egocentric Models for Citation Networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 857–864.

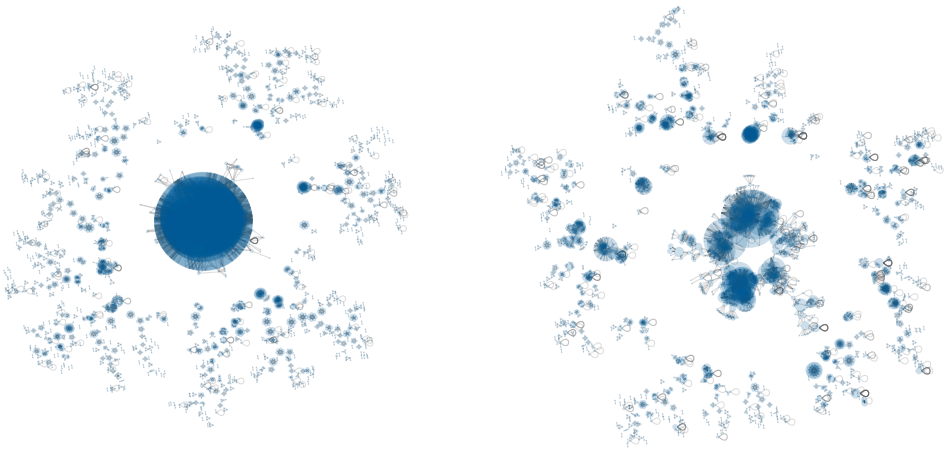
Vu, D. Q., D. Hunter, P. Smyth, and A. U. Asuncion (2011). Continuous-Time Regression Models for Longitudinal Networks. In *Advances in Neural Information Processing Systems*, pp. 2492–2500.

Whitehead, J. (1980). Fitting Cox's regression model to survival data using GLIM. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 29*(3), 268–275.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B) 73*(1), 3–36.

Wood, S. N. (2017). *Generalized Additive Models: An Introduction with R* (2 Rev ed. ed.). Boca Raton: Chapman & Hall/Crc Texts in Statistical Science.

Appendix A: Further Descriptive Analysis



(a) IT-methods (107) (b) food chemistry (118)

Figure 8: Visualization of two inventor networks aggregated over time. Vertex size represent nodal degree. Colouring is transparent to better examine the clusters. The layout uses maximal connected components and applies the layout separately. Noticeable are the loops, which belongs to single ownership patents. Furthermore, we see a few clusters referring to patents with a higher number of inventors. Especially for the area 107 we see the patent, which was submitted with 94 inventors. A lot of nodes look like single isolates, but consist at least of two inventors by construction.

area		basic communication processes (105)	IT-methods (107)	analysis of biological material (111)	food chemistry (118)
"*patents_ij*"	min	0	0	0	2
	mean	4.00	2.73	3.41	3.72
	max	100	30	161	70
"*joint_patent*"	min	0	0	0	0
	mean	0	0	0	0
	max	13	7	40	15
"*2-star*"	min	0	0	0	0
	mean	7.31	14.82	9.47	8.96
	max	168	410	260	205
"*triangle*"	min	0	0	0	0
	mean	0.01	0.12	0.01	0.02
	max	234	283	1375	283

Table 3: Summary statistics of covariates for different areas for the time period of 14 years. The covariates are the number of patents of i and j ("*patents_ij*"), the number of joint patents of i and j ("*joint_patent*"), the number of inventors that hold a joint patent with i or j ("*2-star*"), and the number of inventors that jointly hold a patent with i and j ("*triangle*").

Appendix B: Further Results

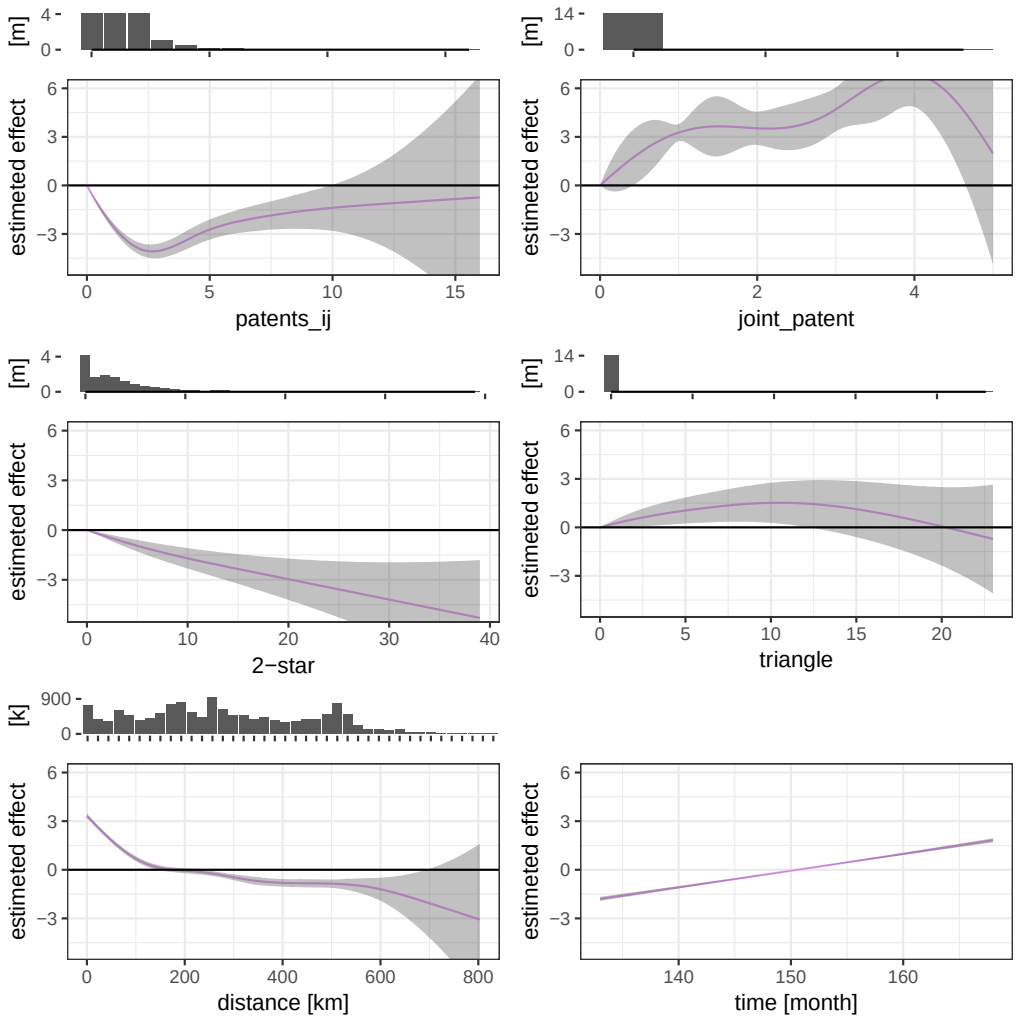


Figure 9: Estimated smooth effects for *IT-methods* (107) area and fourth time period.

Appendix C: Technical Details

The second-order difference penalty matrix can be defined as

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_{(1)} & 0 & 0 & 0 \\ 0 & \boldsymbol{K}_{(2)} & 0 & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \boldsymbol{K}_{(P)} \end{bmatrix} \text{ and } \boldsymbol{K}_{(p)} = \begin{bmatrix} 1 & -2 & 1 & & & & \\ -2 & 5 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 5 & -2 \\ & & & & 1 & -2 & 1 \end{bmatrix}$$

with dimension $[P \cdot K \times P \cdot K]$ and $[K \times K]$, respectively. P is the number of covariates. The second-order penalty matrix $\boldsymbol{K}$ can be derived from $\boldsymbol{K}_{(p)} = \boldsymbol{D}_2^T \boldsymbol{D}_2$ where $\boldsymbol{D}_2 = \boldsymbol{D}_1 \boldsymbol{D}_{2-1}$ is a recursively obtained difference matrix with

$$\boldsymbol{D}_1 = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}$$

with dimension $[(K-1) \times K]$. The corresponding derivatives to apply the Newton-Raphson algorithm are straight forward:

$$s^{pen}(\boldsymbol{u}) = s(\boldsymbol{u}) - (\boldsymbol{K}(\boldsymbol{\lambda}))\,\boldsymbol{u}$$

$$J^{pen}(\boldsymbol{u}) = J(\boldsymbol{u}) - \boldsymbol{K}(\boldsymbol{\lambda})$$

Acknowledgement
The project was partially supported by the European Cooperation in Science and Technology [COST Action CA15109 (COSTNET)].

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, §8 Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, den 18.09.2019 Verena Bauer