# Latent drivers for dynamic networks

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera Italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Igor Artico

under the supervision of
Ernst Wit

05/ 2023

## Dissertation Committee

| | |
|---|---|
| **Ernst J.C.Wit** | Università della Svizzera italiana, Switzerland |
| **Michael Multerer** | Università della Svizzera italiana, Switzerland |
| **Stefan Wolf** | Università della Svizzera italiana, Switzerland |
| **Alessandro Lomi** | Università della Svizzera italiana, Switzerland |
| **Veronica Vinciotti** | Università di Trento, Italy |

Dissertation accepted on 25/ 05/ 2023

| | |
|---|---|
| Research Advisor | PhD Program Director |
| **Ernst Wit** | **Stefan Wolf** |

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Igor Artico
Lugano, 25/ 05/ 2023

# Abstract

Over the past few decades, network analysis has gained popularity in various fields, and understanding the dynamics of networks has become crucial. This thesis explores the dynamics of networks through a statistical approach, focusing on latent drivers that underlie network evolution. The thesis builds upon various key projects, each of which explores different aspects of network dynamics.

The first project proposes a statistical testing procedure to determine whether the degree distribution of a given network follows a preferential attachment process, i.e., a power-law marginal distribution. The second project focuses on dynamic networks where the relational events constitute time-stamped edges and proposes a dynamic latent space relational event model, leveraging a Kalman filter EM algorithm. The third project extends it and addresses the challenge of dealing with huge relational event networks using machine learning optimization tools.

The three projects investigate the complex phenomenon of network growth and transformation, shedding light on the role of latent drivers that shape the structure of observed networks. By studying the underlying drivers, analysts can better understand how networks impact various domains.

# Contents

# Chapter 1

# Introduction

Over the past few decades, network analysis has become an increasingly popular field of study across a range of disciplines, from physics to sociology to computer science. Networks are used to represent complex systems, such as social networks, transportation systems, and the Internet. These networks are not static, but rather they evolve and change over time. As such, understanding the dynamics of networks has become a crucial research topic. In this thesis, we explore the dynamics of networks through a statistical approach, with a focus on latent drivers that underlie the evolution of networks. We build upon three key projects, each of which explores a different aspect of network dynamics. In the first project, we test the power-law marginal distributions of growing networks and propose a statistical testing procedure that considers the complex issues in testing degree distributions in networks. The second and third projects both explore latent dynamics in networks, with the second article developing a dynamic latent space relational event model and the third article proposing an extension for dealing with huge networks. Through these articles, we demonstrate the power of statistical techniques for exploring the latent drivers that underlie the evolution of networks and contribute to the growing body of literature on network analysis.

In this chapter, we introduce the necessary scientific and methodological background to each of the three projects. For each project, we present a separate section with the essential techniques used in that chapter.

## 1.1 Power-law marginal distributions of networks

This is the companion section to the first project, studying the power-law nature of real-life networks. It first introduces random graph models, which describe network objects from a probabilistic point of view. The power-law distribution is a special type of random graph model. As we want to test whether real-life networks conform to the power-law distribution, we then discuss the theory of distributional testing. In particular, we focus on Kolmogorov-Smirnov testing.

### 1.1.1 Random graph models

Random graphs are a type of mathematical model used to describe the properties of networks. They are defined as a probability measure on a graph space. In particular, let $\mathscr{G}(V)$ be the set of all undirected graphs on a finite vertex set $V$. A random graph model $P$ is a probability measure on $\mathscr{G}(V)$, such that

$$P(G) \geq 0, \quad \text{and} \sum_{G \in \mathscr{G}(V)} P(G) = 1.$$

Random graphs are mathematical structures that model the behavior of complex networks in a probabilistic way. These networks can represent anything from social networks to transportation systems to biological systems. Random graphs are constructed by a random process according to some probability distribution.

Random graphs can exhibit a wide variety of properties, depending on the specific model used to generate them. For example, some random graphs may be highly connected, meaning that there are many edges between vertices. Other random graphs may be relatively sparse, meaning that there are relatively few edges between vertices.

The most basic model for generating random graphs is the Erdős-Rényi model. In this model, a graph is generated by starting with a certain number of nodes and connecting each pair of nodes with a fixed probability $\pi$. Thus nodes create random patterns of connectivity resulting from a Bernoulli independent process, i.e.,

$$P(G) = \prod_{i>j} \pi^{G_{ij}} (1-\pi)^{1-G_{ij}}.$$

Another model, the Watts-Strogatz, is a model of small-world networks, which are networks characterized by a high degree of clustering, where nodes tend to be connected forming groups, and a short average path length, where the distance between two nodes is relatively small. The Watts-Strogatz model has been used to study a wide range of phenomena, including social networks, the spread of information, and diseases.

Another commonly used model is the Barabási-Albert model. In this model, a graph is generated via a preferential attachment process where newly added nodes are more likely to connect to existing nodes with a high degree, where the degree of a node is its number of connections. The preferential attachment process, which follows the “rich get richer” rule, is commonly present in social attraction between individuals. It also has ethical implications, such as the meritocracy of research publications. The preferential attachment process produces a scale-free network.

### 1.1.2 Scale-free networks

Scale-free networks are a type of network that is characterized by a power-law distribution of node degrees, meaning that there are a few nodes with many connections, known as "hubs", while most nodes have only a few connections.

In a power-law distribution of node degrees, the probability that a node has k connections is proportional to $k^{-\alpha}$, where $\alpha$ is a parameter that determines the degree discrepancy between the hubs and the least connected nodes. Scale-free networks relate their name to Mandelbrot fractals theory. Mandelbrot fractals are a type of fractal that exhibits self-similarity at different scales, which means that the same patterns are repeated at different scales. Scale-free networks, on the other hand, exhibit self-similarity in their degree distribution, which means that the same power-law pattern is observed across different scales. Hence the network macro structure has the same probability law as the micro.

Scale-free networks are very common in many fields and have been used to study the spread of infectious diseases, the behavior of social networks, and the structure of the Internet.

In this project, we aim to test whether a degree distribution is power-law. However, many observed networks present a power-law behavior only in the tail of the degree distribution. Thus instead of testing for a Barabási-Albert model, we instead test for a de Solla Price model, a power-law generalization that allows fitting a different shape in the lower degrees. The focus is indeed testing tails, and we cannot risk rejecting the test because of a bad fit in the lower degrees.

### 1.1.3 Kolmogorov-Smirnov test

The Kolmogorov-Smirnov test is used for testing the deviation of a degree distribution from a power-law. The test works by comparing the empirical cumulative distribution function (ECDF) of the observed network degrees to the theoretical cumulative distribution function (CDF) of a de Solla Price degree distribution. The ECDF is a step function that assigns a probability of 1/n to each data point,

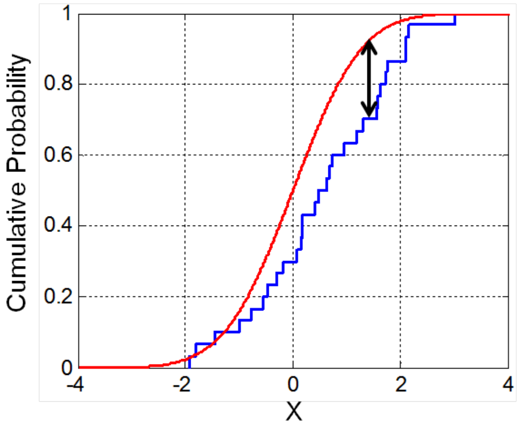


Figure 1.1. Illustration of the Kolmogorov–Smirnov statistic. The red line is a model CDF, the blue line is an ECDF, and the black arrow is the KS statistic. Image borrowed from Wikipedia [Bscan, 2013].

where n is the sample size, and steps up by 1/n at each data point

$$F_n(x) = \frac{\text{number of (nodes whom degree is} < \text{x)}}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{[-\infty,x]}(X_i).$$

The indicator function $\mathbb{I}_{[-\infty,x]}(X_i)$ is equal to 1 in case the $i^{th}$ node degree $X_i < x$. The CDF, on the other hand, gives the probability that a random variable takes on a value less than or equal to a given value

$$F(x) = \int_{-\infty}^{x} f(x)dx.$$

In the case of a discrete distribution, the integral translates into a sum. An example of both these functions is shown in Figure 1.1. The test statistic for the Kolmogorov-Smirnov test is the maximum distance between the ECDF and the CDF

$$D_n = \sup_x \mid F_n(x) - F(x) \mid .$$

If the data is constituted by i.i.d. samples (independent and identically distributed), then the test follows an asymptotic distribution, and the sample can be refused if the Kolmogorov-Smirnov statistic $D_n$ is larger than a significance threshold.

Unfortunately, the node degrees of a network are dependent as a result of coming from a preferential attachment process. This produces a smaller variance of the empirical $F_n(x)$ and smaller $D_n$ as a consequence. Thus asymptotic distribution for the i.i.d. case is no longer valid. Moreover, the ECDF variance is not constant over the degrees, and the largest deviations occur more frequently for lower degrees, see Figure 2.1. In order to achieve the same sensitivity along the tail, we rescale the distances with the variance. Both the variance and the test significance threshold are calculated over parametric Bootstrap samples, which consists of generating Monte Carlo samples from the de Solla Price model that we are testing for. Thus the tail variance is the empirical variance of the simulated samples while the significance threshold is the empirical quantile 95%, corresponding to a 5% type I error.

### 1.1.4 Test power

Test power is the probability of correctly rejecting a false null hypothesis. It is the probability that a statistical test will correctly detect an effect when one exists. A high test power means that the test is able to detect even small effects, while a low test power indicates that the test may fail to detect effects that are present. In our case, the deviation from power-law.

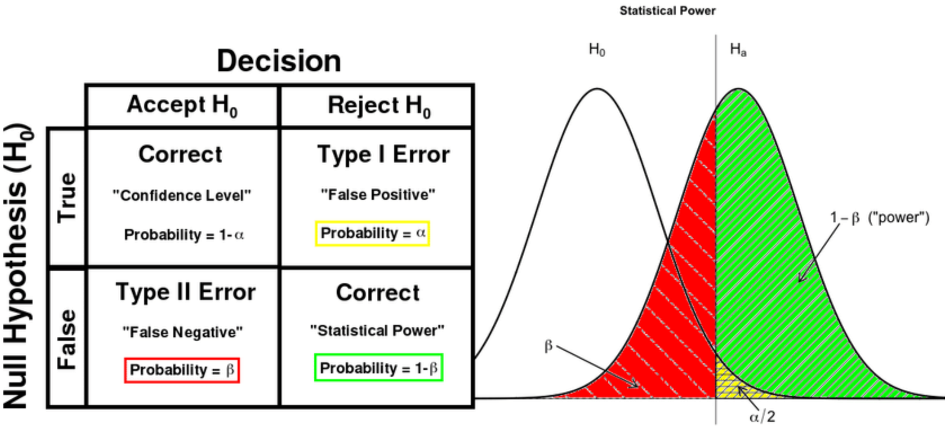
In contrast to the test power, a Type I error occurs when a statistical test rejects a null hypothesis that is actually true, see Figure 1.2. It is commonly set at 5%, like in our case.

Test power and type I error are two related concepts in hypothesis testing. Test power measures the ability of a test to detect an effect when one exists, while type I error measures the probability of erroneously rejecting the null hypothesis when it is true. A good statistical test should have high test power and a low type I error rate.

Test power depends on several factors, including the sample size, the level of significance chosen for the test, the effect size, and the variability of the data. These factors are clearly related to how long the empirical tail is, thus, if the network is sufficiently grown. This is because a short tail does have not enough power and, therefore cannot be safely tested. This project differentiates from the previous literature by claiming that the majority of networks cannot be tested because otherwise, you would get misleading results.

Test power is a less commonly used concept in hypothesis testing because it is difficult to determine. In practice, it is often more feasible to specify the Type I error rate instead. This is because it can be challenging to construct an alternative distribution that represents all possible deviations from the null hypothesis, as there are infinitely many potential distributions that could be used.

However, for power-law testing, this problem can be feasibly handled because it is necessary to just specify as an alternative distribution a power-law tail that



| Null Hypothesis ($H_0$) | Decision: Accept $H_0$ | Decision: Reject $H_0$ |
|---|---|---|
| True | Correct<br>"Confidence Level"<br>Probability = 1-α | Type I Error<br>"False Positive"<br>Probability = α |
| False | Type II Error<br>"False Negative"<br>Probability = β | Correct<br>"Statistical Power"<br>Probability = 1-β |

Figure 1.2. Relationship between test power and type I Error. Test power depends on several factors, including the sample size, the level of significance chosen for the test, the effect size, and the variability of the data. Image borrowed from Wheeler et al. [2014].

drops at the end, as presented in Figure 2.3. This, the partial tail scenario, is the most common and dangerous scenario that we want our test to be able to detect.

## 1.2 Latent Space dynamic Relational Event Model

This is the companion section to the second project. We focus on Relational Event Modeling, which aims to study the factors that influence the links exchange between the network nodes. We assume a node's specific latent factors, like the positioning of a node into a latent space where the frequency of links between two nodes depends on their distance. The latent space is also known as the "Social space". Nodes are allowed to move as time passes and change their connectivity patterns. A Kalman filter methodology is developed to assess node dynamics.

### 1.2.1 Relational Event Modeling

Relational Event Modeling (REM) is a statistical modeling technique that is used to analyze social networks and the interactions between individuals. It is commonly used to study social interactions between individuals or groups, such as the spread of information, the formation of alliances, or the occurrence of conflicts. The basic idea behind REM is that social interactions can be thought of as events that occur between actors in a network, and these events can be analyzed to gain insight into the relationships between the actors and the structure of the network.

REM is a type of regression analysis that models the occurrence of events across dyads over time. The framework is based on the assumption that the occurrence of links between actors in a network is influenced by the network structure and the characteristics of the actors involved. For example, the occurrence of an event between two actors may depend on the strength of their relationship.

REM has been applied in a variety of fields, including sociology, political science, and epidemiology, to study complex social phenomena such as social influence, network formation, and disease transmission. It is a powerful tool for understanding the dynamics of social interactions and can help researchers identify key drivers of social processes.

The REM mathematical background lies in a point process. A point process is a stochastic process whose realizations are points in the timeline, $t_1 < t_2 < ... < t_n$. In case of REM the events $e_k$ are links between nodes $E = \{e_k = (i_k, j_k, t_k) | t_k \in [0, T], k = 1, \dots, n\}$, where $i$ and $j$ are sender and receiver respectively. Each pair of nodes share a point process that describes their interaction history, as shown in Figure 1.3.

In REM literature a point process is assumed to follow Exponential waiting times: the time between links is distributed Exponential($\lambda_{ij}(t)$) and the frequency of connection between two actors $\lambda_{ij}(t)$ varies in time according to the actors' actions. The probability density, i.e. the likelihood, of the process is

$$\begin{aligned} L(E;\lambda) &= \prod_{k=1}^{n} f(e_k | e_{k-1}) \\ &= \prod_{i,j} \prod_{t \in E(i,j)} f(t,i,j | \tau < t) \\ &= \prod_{i,j} \left( \prod_{t \in E(i,j)} e^{\lambda_{i,j}(t)} \right) e^{-\int_0^T \lambda_{i,j}(t) dt} \end{aligned}$$

where $E(i,j)$ is the restricted set of interactions between $i$ and $j$. In the REM

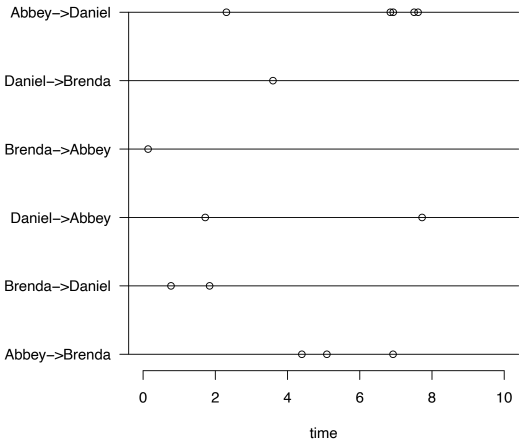


Figure 1.3. Point processes associated with actor pairs.

literature is more common to find an approximation of it, assuming that rates $\lambda_{ij}(t)$ are step functions that are allowed to change only when a link happens. This assumption simplifies the likelihood to the one presented in Butts [2008]

$$\begin{aligned}
L(E;\Lambda) &= \prod_{k=1}^{n} f(e_k|e_{k-1}) \\
&= \prod_{k=1}^{n} f(t_k \cap i_k \to j_k|e_{k-1}) \\
&= \prod_{k=1}^{n} f(t_k|e_{k-1}) f(i_k \to j_k|e_{k-1}) \\
&= \prod_{k=1}^{n} \left(\sum_{ab} \lambda_{ab}(t_k)\right) e^{-\sum_{ab}\lambda_{ab}(t_k)(t_k-t_{k-1})} \frac{\lambda_{i_k j_k}(t_k)}{\sum_{ab}\lambda_{ab}(t_k)}
\end{aligned}$$

where the last factor

$$PL = \prod_{k=1}^{n} \frac{\lambda_{i_k j_k}(t_k)}{\sum_{ab}\lambda_{ab}(t_k)}$$

is a Multinomial distribution and is called Partial Likelihood (PL). In REM literature, and more popularly in medicine, maximizing the PL is the main objective for parameter estimation. The remaining part of the density can be excluded since is not very informative to the parameters. This procedure is known as Cox regression modeling [Cox, 1972].

$\lambda_{ij}(t)$ can be parametrized in multiple ways. It can be affected by both fixed and random effects, as well as time-varying covariates, to account for the effects of different factors on the occurrence of links. Two common effects are reciprocation (if an actor receives a link then increase the chance of a reply) and triadic closure (if A contact B and B contact C, then increase the chances that A and C make a contact). In this project, we focus on making

$$\lambda_{ij}(x_t) = e^{-\|x_i(t)-x_j(t)\|^2}$$

dependant on the latent mapping of actors, where actors take a position in a latent space and they interact more frequently with closer actors. This mapping is, in social science, referred to as the “social space”, a space that reflects actors’ social vicinity. As time passes, the latent space reflects the dynamics behind the interactions and if actors change their interaction preferences they then move to different locations.

### 1.2.2 Kalman filter and Smoother

We assume that the actors’ latent locations follow a Gaussian process, meaning that actors make a sequence of Gaussian jumps that move them in the latent space. Making inferences on these processes involves the calculation of their conditional distribution, in Bayesian theory known as the posterior. In the Bayesian framework, the state of the system is represented by a probability distribution. The prior distribution represents the prior knowledge about the state of the system, while the posterior distribution represents the updated knowledge about the state of the system after new observations are taken into account. The Kalman Filter and Smoother use Bayesian inference to update the prior distribution to the posterior distribution. They were developed by Rudolf Kalman in the 1960s, and are widely used in control systems, signal processing, and robotics. The Kalman filter leverages the basic concepts of Bayesian inference. Given a prior distribution of actor locations at a fixed time point, the links observed at that time update the prior to a posterior. This posterior becomes the prior for the next time point. The update process is repeated until the time sequence reaches the end. The Smoother does a similar procedure but backward.

Since the entire calculation of the posterior is too complex, the Kalman filter and Smoother estimate only the mean and the variance of the actors’ locations, conditioned to the observed links. This simplification allows estimating a posterior distribution by means of a linear regression only, which translates the problem into the prediction of the latent mean, which depends linearly on the observed links.

From a practical perspective, the Kalman filter and Smoother as well are often interpreted as error correction models. It uses a two-step process to estimate the state of the system. In the prediction step, the prior distribution of the locations is predicted based on the system dynamics function

$$x_t = \mathbb{E}[\text{forward}(x_{t-1}) + \eta_t].$$

In the case of our Gaussian process, forward$(\cdot)$ is the identity function because we do not put any constraint on the nodes' direction. $\eta_t \sim N(0, Q)$ is the Gaussian jump that moves locations in the current time step. The predicted locations are used to calculate the expected number of links in the observed state

$$\hat{y}_t = \mathbb{E}[\lambda(x_t) + \epsilon_t].$$

$\epsilon_t$ is the noise associated with the observed data. In the update step, the links are observed and the prediction error is propagated to the locations state in order to make a correction on their position

$$x_t \leftarrow x_t + K_t(y_t - \hat{y}_t).$$

Practitioners commonly assume that the observed state is continuous and Gaussian, obtaining a filtering matrix $K_t$ that minimizes the Gaussian prediction error

$$K_t = \mathbb{E}[(y_t - \hat{y}_t)^2].$$

In our case, the links come from a Poisson distribution, and the Kalman filtering matrix is obtained by maximizing the likelihood of the process. The filtering sequence is presented in Algorithm 1 and 2.

In conclusion, the Kalman Filter and Smoother are Bayesian tools for estimating the state of a dynamic system based on noisy observations. They are particularly useful in situations where there is uncertainty in the measurement or modeling of the system and where the accuracy of the estimate is critical to the performance of the system. They have been recently successful in Computer Vision tasks and self-driving car systems.

## 1.3 Fast inference for large REM networks

This is the companion section to the third project. We aim to scale the latent space Relational Event Model to very large networks, in the order of millions of nodes. We make inferences to the model by leveraging machine learning optimization techniques, such as the mini-batch stochastic gradient descent. Moreover, the model is equipped with a clustering penalty that facilitates the interpretation of a large number of results by grouping together nodes with similar trajectories.

### 1.3.1 Mini-batch stochastic gradient descent

Mini-batch stochastic gradient descent (SGD) is a popular optimization algorithm used to train machine learning models, particularly deep neural networks. It is a variant of gradient descent and works by randomly selecting a subset $B$ or "mini-batch" of the data, in our case, a subset of links. In this subset, the likelihood and its gradient are computed, updating parameters with a gradient step as

$$\alpha \leftarrow \alpha + \psi \nabla f(\alpha)_B.$$

The gradient step is repeated, taking different subsets of the data, making the estimated gradient a stochastic quantity. It is an unbiased estimator of the full data gradient. Moreover, we use *Adam* [Kingma and Ba, 2014], an extension of the mini-batch SGD, which takes an average of the past history of gradients in order to retrieve a direction with higher precision.

The advantage of mini-batch SGD is faster convergence because the memory and computational cost are contained by the mini-batch size. The introduction of randomness into the optimization process can help the algorithm escape local minima and find better solutions. Mini-batch SGD can also lead to better generalization performance because of the randomness introduction into the training process.

We take the Latent Space REM framework and we use mini-batch SGD to make inferences of the latent locations. However, the algorithm is designed and optimally works for training deep neural networks. We thus focus this project on the adaptation of this algorithm to network data. In particular, we deal with a sparse gradient update problem where the mini-batch, created via network subsampling, might not contain enough information for a reliable gradient estimate. The gradient is sparse when a network subsample presents high sparsity in the links or sparsity in the nodes. We hence propose oversampling links techniques and network-specific mini-batch sizes in order to assure sufficiently dense gradients.

# Chapter 2

# How rare are power-law networks really?

I declare that the content of this chapter comes from the original paper [Artico et al., 2020] which is published in the Proceedings of the Royal Society A in collaboration with I. Smolyarenko, V. Vinciotti, E.C. Wit.

## 2.1 Summary

The putative scale-free nature of real-world networks has generated a lot of interest in the past 20 years: if networks from many different fields share a common structure, then perhaps this suggests some underlying “network law”. Testing the degree distribution of networks for power-law tails has been a topic of considerable discussion. *Ad hoc* statistical methodology has been used both to discredit power-laws as well as to support them.

This paper proposes a statistical testing procedure that considers the complex issues in testing degree distributions in networks that result from observing a finite network, having dependent degree sequences, and suffering from insufficient power. We focus on testing whether the tail of the empirical degrees behaves like the tail of a de Solla Price model, a two-parameter power-law distribution. We modify the well-known Kolmogorov-Smirnov test to achieve even sensitivity along the tail, considering the dependence between the empirical degrees under the null distribution, while guaranteeing sufficient power of the test. We apply the method to many empirical degree distributions. Our results show that power-law network degree distributions are not rare, classifying almost 65% of the tested networks as having a power-law tail with at least 80% power.

## 2.2 Introduction

Networks play an important role in many fields, from epidemiology and ecology to engineering and sociology. They are a powerful way to represent and study the interaction structure of complex systems. An important measure of the network topology is the distribution of the number of connections per node: the *connectivity distribution* [Barabási and Oltvai, 2004], also known as the *degree distribution*. Many empirical networks have been reported to exhibit *scale-free* behavior based on the distribution of the connectivities of the network nodes [Newman, 2003; Mitzenmacher, 2004]. Describing networks can be justified in two distinct ways: either phenomenologically based on network data or from first principles.

Power-law networks have been proposed as a "universal" model, as they possess a number of important properties, such as the presence of hubs and large numbers of nodes with few connections [Jeong et al., 2001] as well as a typical small-world behavior [Amaral et al., 2000]. The latter allows fast communication between nodes even for huge networks, given the small diameter characteristic of small-world networks. The definition of a power-law network varies across the literature, but one often cited definition is its degree distribution $P$ satisfies $P(d) \propto d^{-\gamma}$, where $\gamma > 1$ [Clauset et al., 2009]. Some versions make additional requirements, e.g., requiring that node degrees evolve via a preferential attachment mechanism [Albert and Barabási, 2002], and specify, mathematically more correctly, that the power-law only should hold asymptotically in the upper tail of the degree distribution [Mitzenmacher, 2004; Voitalov et al., 2019].

However, from a phenomenological point of view, observed networks are (almost) always finite, hence a power-law network is indistinguishable from a network with a sufficiently distant exponential cut-off of a power-law degree distribution. If our sole purpose is fitting an observed degree sequence, then a large class of models will do an equally good job for the types of networks we tend to encounter in practice. Nevertheless, ever since De Solla Price started to experiment with potential generative network models in the 1960s, it has become clear that a small number of substantively plausible and generative principles are capable of generating network structures that correspond to empirical networks. Particularly, various forms of preferential attachment rules have been shown to result in network structures whereby the degree sequences are generally described by ratios of gamma functions [Krapivsky and Redner, 2001], i.e., power-laws. This putative universality of the power-law degree distributions sets it up as a natural paradigm for falsification [Popper, 1962], i.e., as a natural null hypothesis. It is from this epistemological point of view that we approach the question of power-law networks in this paper. On top of this, others have also argued that it is practically important to know whether networks are power-law, as such networks are, for example, more susceptible to epidemics and other viral events [Newman, 2002].

A long-standing issue in network science is how prevalent the power-law property is in empirical networks. A spate of early analyses, often using a fairly crude methodology, resulting in a widespread acceptance of the belief that power-law degree distributions, viewed as a proxy for a network being scale-free, are quite ubiquitous [Redner, 1998; Laherrere and Sornette, 1998; Faloutsos et al., 1999; Albert et al., 1999]. This coincided with intensive theoretical efforts to explain the putative universality of power-law degree distributions. More recently, more sophisticated statistical techniques have cast doubt on the extent of the scale-free universality. Starting with the work of Khanin and Wit [2006], biological networks were shown to fit better with a truncated power-law model, i.e., a power-law regime followed by a sharp drop-off, $P(d) \propto d^{-\gamma}e^{-d/k_c}$. The authors found that the number of connections in biological networks significantly differs from the power-law distribution and that these networks are not scale-free. Another critique was levied in a recent paper by Broido and Clauset [2019], who use a likelihood ratio test within a nested testing procedure, suggesting that the evidence for power-law distribution is often weak. A drawback of these critiques is the emphasis on identifying “pure” power-law tails as this leads to two conflicting requirements: a cut-off far into the distribution tail to ensure, in some sense, sufficient closeness to the asymptotic power-law, and the availability of a sufficiently large number of data points for meaningful statistical testing. The same issue has recently been highlighted by Voitalov et al. [2019], who devise consistent estimation procedures for the exponent $\gamma$ taking into account the asymptotic nature of power-laws, but who reject the possibility of a formal testing procedure.

Even though a number of studies have considered testing for power-law degree distributions in empirical networks, the final verdict is still open. This current paper takes a complementary view to Voitalov et al. [2019]: we make stronger parametric assumptions about the asymptotic form of the tail of the degree distribution, avoiding the impossibility arguments [Voitalov et al., 2019, Section V], in order to get a lower bound on the fraction of empirical networks that exhibit power-law behavior. This parametric assumption consists of assuming that the tail of the degrees comes from a de Solla Price network process, a two-parameter preferential attachment model. This does not mean that a de Solla Price is a sensible model for real-world networks, but being a subset of the power-law distributions, not being able to reject with sufficient power a de Solla Price model would mean that we have positive evidence for a power-law tail.

In Section 2.3 we present the landscape of the main methodological issues encountered in testing degree distributions in empirical networks. In Section 2.4 we present the proposed testing framework. We present (i) a specific parametric asymptotic power-law model that will be used to test the goodness-of-fit of the empirical degree distribution, (ii) a modification of the classic Kolmogorov-Smirnov statistic to deal with dependent degree samples as well as heterogeneous variances and (iii) a way to calculate the power of the test-statistic. In Section 2.5 we apply the testing framework to 4,482 empirical networks. Our aim is to decide whether in a large body of networks the power-law property holds or should be seen as too simplistic. In Section 2.6 we present our conclusions.

## 2.3 Issues in testing empirical degree distributions

In this section, we present an overview of the main issues encountered in testing whether empirical degree distributions are power-law. In particular, (i) we will introduce the exact asymptotic definition of a power-law degree distribution and relate this to the problem of observing only finite networks; (ii) we explain how the dependency of a single empirical degree sample affects the distribution of a Kolmogorov-Smirnov test statistic and (iii) we show how asymptotic tests must balance the delicate equilibrium between the power of the test and the asymptotic power-law property. The issues introduced in this section will be resolved in Section 2.4.

### 2.3.1 What is a degree distribution?

A simple random graph on the vertex set $V = \{1, \ldots, N\}$ is defined by its graph distribution $H : E \to [0, 1]$, which associates with any graph $G$ a probability $H(G)$. For directed graphs with possible self-loops $E = \{0, 1\}^{N\times N}$, whereas for directed graphs without self-loops or undirected graphs, $E$ is a strict subset of $\{0, 1\}^{N\times N}$. For any vertex $i$ in the graph $G$, we define its degree $d_G(i)$ as the number of edges in $G$ that involve vertex $i$. In the case of directed networks, one could focus on the in-degree or out-degree instead, but this will not change the exposition below. Given a particular degree definition, we define the *marginal degree distribution* $P(\cdot|i) : \{0, \ldots, N\} \to [0, 1]$ *for vertex $i$* as the probability over all graphs $G$ for which vertex $i$ has a particular degree,

$$P(d|i) = \sum\nolimits_{d_G(i)=d} H(G).$$

Two important points to notice are that the measures $P(\cdot|i)$ and $P(\cdot|j)$ for $i \neq j$ are generally *dependent* and *not identical*. Only if the measure $H$ is exchangeable, then the marginal degrees are identically distributed. Only in very special cases, such as for certain types of Erdős-Rènyi graphs, these marginal degrees are both independent and identically distributed.

The *average degree distribution* $P : \{0, \dots, N\} \to [0, 1]$ is defined as the marginal degree distribution of a randomly selected vertex,

$$P(d) = \frac{1}{N} \sum_{i=1}^{N} P(d|i).$$

We will refer to this distribution simply as *the degree distribution*. In fact, it is this distribution that one commonly considers in practice, for example, by plotting the histogram of degrees of all the vertices in a particular graph.

For graphs with infinitely countable vertex sets, the same definition for the marginal degree distribution can be given, whereas the (average) degree distribution is defined as a limit,

$$P_{\text{inf}}(d) = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} P(d|i).$$

For the Barabási-Albert preferential attachment model it can be shown that $P_{\text{inf}}(d) = \frac{4}{(d+1)(d+2)(d+3)}$ for the in-degree $d \in \mathbb{N}_0$ [Albert and Barabási, 2002].

We define *power-law degree distributions* as those degree distributions for infinite graphs that possess a particular asymptotic property in their tail. In particular, an infinite graph degree distribution $P_{\text{inf}}$ is considered *power-law* if there exists a $\gamma > 1$ such that

$$\lim_{d \to \infty} d^{\gamma} P_{\text{inf}}(d) = c, \tag{2.1}$$

where $c > 0$ is an arbitrary positive constant, e.g., for the Barabási-Albert preferential attachment model $\lim_{d \to \infty} d^3 P_{\text{inf}}(d) = 4$. This definition of a power-law is more restrictive than the regular variation definition in Voitalov et al. [2019], but this is sufficient for our purposes.

### 2.3.2 Finitely observed network

As any empirically sampled network is finite, in what sense can this finite network be related to the power-law? Since a vertex in a simple graph without self-loops cannot have more connections than the total number of vertices excluding itself, the degree distribution has a support that is bounded above by $N - 1$. This means that it is impossible to detect scale-free networks, whose power-law regime "starts" above $N - 1$. *Every finite network degree distribution could potentially behave like a power-law on the unseen degrees*. That is why, strictly speaking, talking about power-law degree distributions for finite networks is meaningless.

However, if the finite network is, in a certain sense, a "random sample" from an infinite network, then under certain conditions it might be possible to relate the finite sample degree distribution to the infinite population distribution. Sampling subnetworks is more complicated than sampling ordinary populations, as specific choices have to be made: whether to sample primarily vertices or edges and how to sequence the sampling. Most state-of-the-art network sampling schemes, i.e., link tracing, star, snowball, induced, and incident sampling, have drawbacks that lead to certain biases in the estimation of the degree frequencies [Kolaczyk and Csárdi, 2014, Ch 5.6]. We will show how certain generative sampling assumptions will allow us to sample finite networks that asymptotically form a subclass of the power-law degree distribution networks.

### 2.3.3 Dependent vs. independent degree samples

Essentially all existing work on empirical degree distributions [Newman, 2005; Mislove et al., 2007; Lima-Mendez and van Helden, 2009; Broido and Clauset, 2019; Barabási, 2018; Voitalov et al., 2019, e.g.] treats the observed degree sequence of an empirical network as an *independent* random sample. However, depending on the underlying random graph distribution, observing a degree for a particular node may well be positively or negatively correlated to the degree of another node. A sample of degrees coming from a single realization of a network should, therefore, be considered as a dependent sample. The impact of this dependence on test statistics that involve the empirical degree distribution has not been studied in any detail until now.

Smolyarenko [2019] shows that tests based on the empirical degree distribution can have markedly different behavior from what would be expected under independence. In particular, the scaled empirical cumulative distribution function for degree distributions in standard synthetic networks does not converge to a Brownian bridge [Mansuy and Yor, 2008] — see Appendix A for details. We will show that under certain network distributions, the variance of the empirical degree distribution is lower than expected under independence, invalidating traditional Kolmogorov-Smirnov tests.

### 2.3.4 Power of goodness-of-fit test

As we want to test the null hypothesis that an empirical degree distribution comes from a power-law network, it is important to be able to control the power of the goodness-of-fit test. Regardless of the test choice, not rejecting "$H_0$ : *network is power-law*" is not necessarily proof of the validity of $H_0$ without additional control of the power of the test. The power of a test controls the probability of rejecting $H_0$ when it is false. Although one clearly desires a high level of power in order to correctly detect power-law networks, this does not come for free: it involves determining the level and type of departure of power-law that is practically insignificant. We will make recommendations on how to set sensible values for this allowable deviation.

Furthermore, since a power-law is a tail property, the test statistic will focus on the tail of the degree distribution. This leads to two, possibly conflicting requirements, since the further along in the tail of the degree distribution we check, (i) the more likely our parametric power-law distribution is able to fit a power-law tail if it is present, but (ii) the less power the goodness-of-fit has to detect it. We have to find a balance between, on the one hand, testing the tail and, on the other hand, having sufficient tail observations to guarantee a certain power of the test.

## 2.4 Testing framework

In this section, we present an integrated testing framework that addresses the issues that were described in Section 2.3. Our aim is to describe a comprehensive procedure that is based on a non i.i.d. degree sequence from a finite network is able to test the null hypothesis

$$H_0 : \textit{The degree distribution } P_{\text{inf}} \textit{ is power-law},$$

where the finite network is assumed to be a particular type of sample of $P_{\text{inf}}$ as described in Section 2.4.1. Then in Section 2.4.2 we operationalize the concept of a power-law degree distribution by means of a flexible, generative family of degree distributions. In Section 2.4.3 we introduce a modified Kolmogorov-Smirnov test statistic that deals with all the difficulties we identified above and in Section 2.4.4 we show how we can control the power of this test.

### 2.4.1 Sampling finite networks

Empirical finite networks can occur in many different ways [Crane, 2018]. It could be that the vertex set is fixed and the edges are drawn from some distribution. These networks are not of interest to us in this manuscript. Clearly, such non-growing networks have no relationship with any underlying, infinite network distribution that might or might not exhibit power-law behavior. Instead, in this manuscript we assume that $P_{\text{inf}}$ is the resulting degree distribution from a generative and additive network sampling scheme that at each moment can be stopped to obtain a finite network.

For example, the Barabási-Albert preferential attachment model is a generative network sampling scheme that at each step adds a vertex to the network that it connects to one of the other vertices already in the network with a probability proportional to their degrees. This procedure can be stopped for any finite size $N$ network, leading to a degree distribution $P_N(d)$. Whereas the finite Barabási-Albert preferential attachment model converges to a network with a power-law degree distribution, other iterative sampling schemes might not.

### 2.4.2 A finite de Solla Price power-law

As the power-law property is a mere asymptotic characteristic of a network, the class of power-law networks is vast. On purpose, we will restrict ourselves in this manuscript to a subfamily of power-law networks. As our main assumption in Section 2.4.1 is that the finite network is in a generative way associated with the infinite network measure, we will focus on a generative class of power-law distributions, namely preferential attachment models. These models iteratively extend the network, both in terms of vertices and edges, in such a way that networks of any particular size can be achieved.

Krapivsky and Redner [2001] describe a rich class network models constructed by means of a general generative preferential attachment procedure with arbitrary connection kernels. They show that these kinds of models result in degree distributions that can be described by ratios of gamma functions. Ratios of gamma functions are the discrete analogs of power-laws. Using finite gamma ratios as a model for power-law degree distributions has the crucial advantage of treating some of the "midsection” of the degree distribution as signal rather than noise. Broido and Clauset [2019], Khanin and Wit [2006] and others have been unnecessarily restrictive in trying to find pure power-laws rather than accept that some aspects of curving in log-log plots are informative, starving typical power-law tests of data.

We focus on a particular two-parameter gamma ratio model, known as the de Solla Price model introduced in 1965 for modeling growing citation networks [De Solla Price, 1965]. In the context of a growing network, $m$ is the number of new edges added to the network at each iteration of the growing algorithm and $d + w$ is proportional to the preferential attachment probability for the vertices with $d$ incoming links. Van der Hofstad [2016] and Newman and Girvan [2004] shows that the infinite degree distribution is given by

$$P^{sp}_{\text{inf}}(d; w, m) = c_{m,w} \frac{\Gamma(d + w)}{\Gamma(d + 2 + w + w/m)}$$

where $0 < w < \infty$, $m \in \mathbb{N}$ and the normalizing constant $c_{m,w} = (1+\frac{w}{m})\frac{\Gamma(1+w+w/m)}{\Gamma(w)}$. The model is a generalization of the Barabási-Albert model, which is the special case when $m = w = 1$ and $d \in \mathbb{N}_0$ is the in-degree. Combinations of the parameters $(w, m)$ allow for more flexibility and the model is, therefore, better able to capture empirical distributions at lower degrees. As $d \to \infty$ the model shows a power-law behavior proportional to $d^{-\gamma}$, i.e,

$$P^{sp}_{\text{inf}}(d; w, m) = c_{m,w} d^{-\gamma}(1 + O(1/d))$$

where $\gamma = 2 + w/m$ [Van der Hofstad, 2016].

The finite de Solla Price degree distribution of size $N$ is indicated as $P^{sp}_N(\cdot; w, m)$. We will use $F^{sp}_N(d; w, m) = \sum_{i=0}^{d} P^{sp}_N(i; w, m)$ as notation for the cumulative distribution function of the finite de Solla Price model. Although the de Solla Price model is flexible and can fit a wide range of empirical power-law degree distributions, the model is still not flexible enough for our purposes. In order to address this issue, we define a model that behaves as de Solla Price on the degrees above a specified cutoff $c$ and is free to take any other shape for the degrees below, in particular,

$$P^{sp}_{c,N}(d; w, m) = \begin{cases} p_k & d = 0, \ldots, c-1 \\ P^{sp}_N(d; w, m) & d = c, \ldots, N-1 \end{cases}$$

with its associated cumulative degree distribution function $F^{sp}_{c,N}(\cdot; w, m)$. Barabási [2018] suggested that power-law networks often have such low degree deviations, which should be ignored. We refer to this network model as the extended de Solla Price network model, which is generated by arbitrarily rewiring edges between low-degree vertices.

### 2.4.3 A weighted Kolmogorov-Smirnov testing procedure

Given the de Solla Price subclass of power-law networks, our aim is to test the more stringent null hypothesis

$H_0$ : *The network is drawn from an extended de Solla Price network model,*

based on a single finite empirical network sample. The idea is that the number of non-rejected tests, each with sufficient power, will give us an idea of the lower bound on the ubiquity of empirical power-law networks.

#### Traditional Kolmogorov-Smirnov test statistic

Traditionally the Kolmogorov-Smirnov (KS) test statistic is one of the common statistics used to test for the goodness-of-fit of a particular presumed distribution of the data. It is defined as the largest distance between the empirical cdf and the hypothesized one,

$$D_{KS} = \sqrt{N} \sup_{d \geq 0} \left| \hat{F}_N(d) - F^{sp}_{c,N}(d; w, m) \right|, \tag{2.2}$$

where $d$ stands for the degree, $F^{sp}_{c,N}$ and $\hat{F}_N$ are respectively the true (under $H_0$) and the empirically observed degree distributions, $N$ is the overall number of observations, i.e., the number of vertices in the empirical network. The empirical degree distribution is defined as $\hat{F}_N(d) = \frac{1}{N} \sum_{v \in V} \mathbb{1}_{\{d_v \leq d\}}$ where $d_v$ is the observed degree of vertex $v$. Under the independent sampling assumption, the $D_{KS}$ statistic converges in distribution to the Kolmogorov limit distribution [Kolmogorov, 1933]. The convergence of $D_{KS}$ to the Kolmogorov limit distribution is based on the assumption of continuous data and independent observations, both of which are violated in the case of an empirical degree distribution from a single network. As shown by Smolyarenko [2019], the KS test statistic for empirical degree distributions in evolving networks does not converge to the usual Kolmogorov limit distribution.

#### Variance of the empirical degree distribution

As pointed out by Anderson and Darling [1954], the KS statistic does not achieve uniform sensitivity over all quantiles. Under the independent sampling assumption, for a fixed degree $d$ we have that

$$N\hat{F}_N(d) \sim \mathrm{Bin}(N, F^{sp}_{c,N}(d; w, m)) \tag{2.3}$$

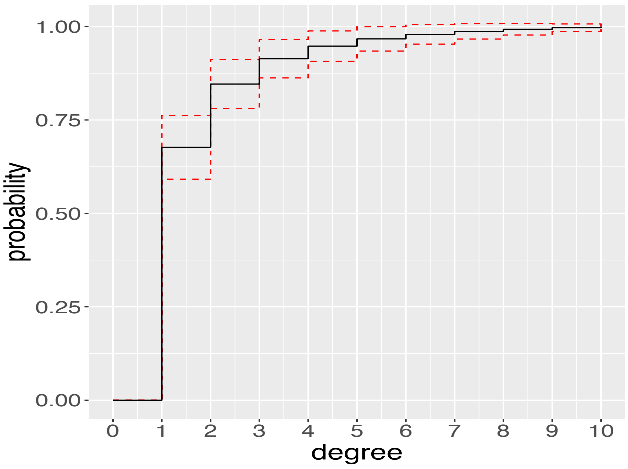


Figure 2.1. An example of a de Solla Price cumulative degree distribution; dashed lines indicate the standard deviation of the empirical degree distribution considering a network of size 30.

with variance $NF^{sp}_{c,N}(d;w,m)(1-F^{sp}_{c,N}(d;w,m))$. Although independence is a rather unrealistic assumption, it can give an insight into the variance behavior in empirical cumulative degree distributions. In particular, $\hat{F}_N(d)$ achieves its highest variance at $d = F^{sp-1}_{c}(0.5)$ and decreases to zero in the tails — i.e., in the right tail in case of degree distribution. The distances $\left|\hat{F}_N(d) - F^{sp}_{c,N}(d;w,m)\right|$ are not identically distributed over $d$ and, more importantly, the decrease of the variance leads to a decrease of the sensitivity in the tail of the degree distribution. In typical network scenarios, this means that the KS statistic is mainly influenced by low degrees, whereas one mainly wants to detect deviations for high degrees. For example, Figure 2.1 shows an empirical degree distribution, whose first degree $d = 1$ takes 66% of the overall probability and therefore is the main contributor to the KS statistic.

Our aim is to modify the Kolmogorov-Smirnov statistic in such a way that it achieves even sensitivity across the empirical degrees. Beyond the uneven variance addressed by the Darling-Anderson modification [Anderson and Darling, 1954] described above, there are three additional considerations that affect the behavior of the empirical degree distribution. In particular, we show how (i) the estimation of the parameters $(w,m)$ and (ii) the dependence among the empirical degrees lead to a reduction of variance, whereas (iii) the randomness of the observed degrees inflates the variance as compared to the independently sampled binomial case in (2.3) that we consider as our baseline.

(i) Variance reduction due to parameters estimation. In order to be able to calculate the KS statistic, one needs to estimate the parameters of the de Solla Price model. We use maximum likelihood to estimate its parameters. In particular, given a fixed value for $c$, we estimate the lower degree probabilities by their empirical counterparts. As the empirical distribution function and the MLE of the flexible de Solla Price coincide for low degrees, we have $\left|\hat{F}_N(d) - F^{sp}_{c,N}(d; w, m)\right| = 0$ for $d < c$. In general, estimation of the parameters reduces the variance of the KS statistic [Feigelson and Babu, 2013].

(ii) Variance reduction due to dependent observed degrees. As described in Section 2.3, the empirical degree distribution is a dependent sample of degrees. We will show that this affects the distribution of KS statistic $D_{KS}$. Chicheportiche and Bouchaud [2012] show that the behavior of the KS statistic can be studied by analyzing the random function $Y(u) = \sqrt{N}\left(\hat{F}(F^{\text{sp}-1}_{c,N}(u)) - u\right)$, $u \in [0, 1]$ is the $u^{th}$ theoretical quantile, since $D_{KS} = \sup_u y(u)$. If $\hat{F}$ was estimated by independent observations, then (2.3) would imply that $V(Y(u)) = u(1-u)$. This is shown as the red line in Figure 2.2.

Although the correlations between the empirical degrees are only of order $1/N$, the fact that there are $\binom{N}{2}$ of them, has a dramatic impact on the overall variance of $Y(u)$ and therefore on the KS statistic $D_{KS}$ [Smolyarenko, 2019]. We simulated from the de Solla Price preferential attachment model, using different values of $w$, the preferential attachment probability of the nodes with no incoming links, and $m$, the number of new links that each new node makes with the remaining nodes at each iteration of the growing process. Figure 2.2 shows that in all the scenarios the observed variance of $Y(u)$ and therefore $D_{KS}$, was lower than expected under independence. The negative correlations between the empirical degrees results in a significantly lower variance. This clearly casts doubt on a large scale of methodologies and past results which were based on the independence assumption [Broido and Clauset, 2019; Clauset et al., 2009, e.g.].

(iii) Variance inflation due to randomly observed degrees. The baseline case, as described in (2.3), holds only for *fixed* degrees $d$ under the independent sampling assumption. However, the supremum taken in (2.2) will occur at an observed, i.e., *random* degree. As Goldman and Kaplan [2016] showed for continuous distributions, the empirical degree $\hat{F}_N(d_{(i)})$ has beta distribution, i.e., $\hat{F}_N(d_{(i)}) \sim \beta(i, N+1-i)$, which holds approximately for high degrees due to the near continuous behavior of $\hat{F}_N$ in the degree tail for large networks. This

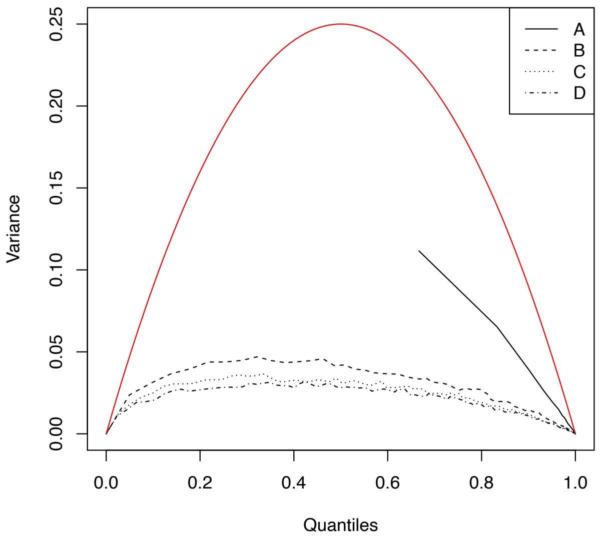


Figure 2.2. Brownian Bridge's empirical variance with A:(w=1, m=1), B:(w=134, m=23), C:(w=267, m=44), D:(w=400, m=51). The red line is the variance under independent degree sampling (see Appendix A). Line A is complete, but starts from the first rescaled degree $F^{\text{sp}}(0;1,1)=0.66$.

results in a higher variance of the KS statistic than the binomial one. Clearly this is true under the independent sampling assumption. For empirical degree distributions, it is challenging to quantify the overall variance inflation due to the degree of randomness since we also have to consider the possible variance deflation due to the previous points.

### A modified Kolmogorov-Smirnov test statistic

Here we will describe a test statistic that resolves the uneven variance, the reduced variance, and the inflated variance that the KS statistic experiences for empirical degree distributions. As it is impossible to calculate analytically the effect of the various complicating factors, we resort to bootstrapping in order to define a uniformly sensitive, KS-like test statistic for testing the null hypothesis of a de Solla Price power-law degree distribution. This is possible because the de Solla Price is a generative network model, which can be sampled efficiently.

In particular, we consider an empirical network, for which we want to test whether it might have appeared from a finite de Solla Price network, $F_{c,N}(\cdot;w,m)$.

We will assume that the cut-off $c$ is given — its value involves power considerations, described in Section 2.4.4.

First, we estimate the parameters of the model $(w, m)$ from the data. A number of methods are proposed in the literature for power-law estimation, such as the Hill estimator for the tail coefficient of Wang and Resnick [2020] and the maximum likelihood approach on the network evolution data of Gao and van der Vaart [2017], whereas a comparison between different estimators is provided in Clauset et al. [2009]. In our framework, we estimate the unknown parameters $(w, m)$ by numerically maximizing the pseudolikelihood

$$L(d; w, m) = \prod_{i=1}^{N} P^{sp}_{c,N}(d_i; w, m)$$

via an iterative algorithm [Gay, 1990]. Crowder [1976] showed that these estimates are consistent. For fixed discrete values of $m$, we maximize the likelihood according to $w$. We repeat the maximization procedure for a reasonable range of $m$ values. Finally, we select the $(m, w)$ values with the highest likelihood. This procedure is known as *profile* pseudolikelihood maximization. Further generalizations might be possible by specifying a random $m$ parameter [Deijfen et al., 2009] that can be sampled among the most likely values.

Then we define the test statistic $T$ as

$$T = \sqrt{N} \max_{v: d_v \geq c} \left[ \frac{\left|\hat{F}_N(d_v) - F^{sp}_{c,N}(d_v; \hat{w}, \hat{m})\right|}{\sqrt{\hat{z}(d_v, \hat{w}, \hat{m})}}, \lim_{a \to d_v^-} \frac{\left|\hat{F}_N(a) - F^{sp}_{c,N}(a; \hat{w}, \hat{m})\right|}{\sqrt{\hat{z}(a, \hat{w}, \hat{m})}} \right] \tag{2.4}$$

where $\{d_v\}$ are the observed degrees on the vertex set $V$ of size $N$ and $\hat{z}$ are the Monte Carlo estimated variances of the empirical degree distribution at the observed degrees for simulated de Solla Price networks with parameters $(\hat{w}, \hat{m})$. The distribution of the test statistic $T$ under the null hypothesis is obtained via a parametric bootstrap [Efron, 1992]. The parametric bootstrap consists of sampling degree distributions from the null hypothesis, i.e., a de Solla Price network generating process. The unknown parameters $(w, m)$ are substituted with the maximum likelihood estimates, meaning sampling from the most likely de Solla Price distribution according to the observed data. We calculate the test statistics $T$ on each of them, and obtain $T_1, \ldots, T_B$ bootstrap realizations of the test statistics distribution under $H_0$. We reject the hypothesis that the data come from a de Solla Price network if the test statistic $T^{obs}$ calculated on the observed network is greater than the 95% empirical percentile of the bootstrap distribution.

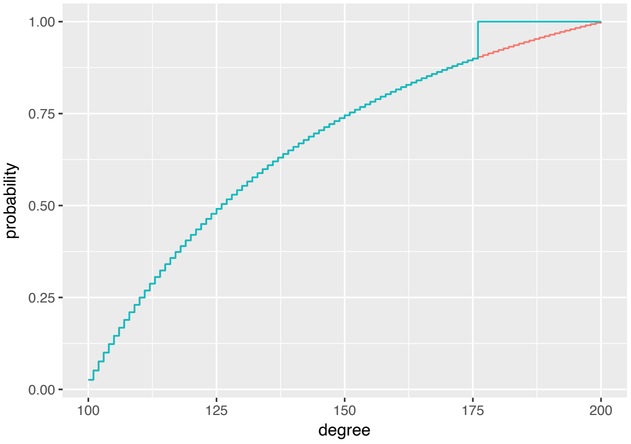


Figure 2.3. Under $H_0$ degree distribution is the red conditional de Solla Price power-law, whereas under $H_1$ the degree distribution is taken to be the blue step function with $h_c = 0.1$ and $c = 100$.

### 2.4.4 Cutoff choice via power analysis

This section selects the cutoff point $c$, by considering how many observations are left in the tail of the empirical degree distribution in order to guarantee the required power level. Although power is loosely defined as P(reject $H_0 \mid H_1$ is true), for continuous alternatives one needs to select a required minimum detectable effect size [Bloom, 1995], which we define as the maximal distance $h$ between the true distribution and the null distribution $F^{\text{sp}}_{c,N}(\cdot; \hat{w}, \hat{m})$.

Power-law are universally known for decreasing to zero slower than any other function. Thus we choose an alternative distribution that decreases faster in the tail. Among all the possible degree distributions with at least $h$ maximum distance, the one that minimizes the power is the degree distribution that is exactly the same as the null $F^{\text{sp}}_{c,N}(\cdot; \hat{w}, \hat{m})$, but with a step of size $h$ placed at the end of the tail, as shown in Figure 2.3. For values of $h$ that are sufficiently small, the distribution can be even closer to the power-law than the log-normal degree distribution. This assures that, once we fix the power for this type of function, all the other degree distributions that are $h$ removed from the de Solla Price power-law will have greater power, i.e., will be detected more easily.

In the practical analyses in Section 2.5 we take a very stringent choice for the cutoff. In particular, we decided to calibrate $h = h_c(1 - F^{sp}_N(c; w, m))$ with $h_c \in [0.01, 0.1]$. This means that we aim to be able to detect degree distributions that have tail behavior that decays faster on roughly the last 0.001 of the degree distribution. We choose a power of 80%, which means that if the true distribution differs from a power-law by only $h$ or more in the tail, then 80% of the time our method will detect it and reject the null hypothesis.

The power calculations are done straightforwardly by simulating $B = 200$ degree samples from the de Solla Price model, with maximum likelihood estimated parameters. Then each sample is censored in correspondence with the degree to which the step occurs, obtaining samples from $H_1$. The statistical test is applied to each of them and the power is finally computed as the rate of rejected tests.

### 2.4.5 Overview of testing procedure

This section provides an overview of all the elements that go into testing whether a single empirical network comes from some de Solla Price power-law model. In five steps, the proposed testing procedure takes into account the power, degree dependency cut off, and even sensitivity over the tail of the test statistic.

1. Step 1: calculate the maximum likelihood estimate on the original sample.

   (a) Fix the cutoff $c$ (for different values of $c$).

   (b) Given an observed degree sequence of size $N$, estimate $\hat{F}_N(\cdot)$ and $F^{sp}_{c,N}(\cdot;\hat{w},\hat{m})$, where $\hat{w}$ and $\hat{m}$ are the maximum likelihood estimates of the de Solla Price model.

2. Step 2: test distribution and variance computation.

   (a) Select number of bootstrap samples $B = 200$.

   (b) Generate $d_1,\ldots,d_B \sim P^{sp}_N(\cdot;\hat{w},\hat{m})$ degree sequences with the de Solla Price preferential attachment algorithm up to a network with $N$ nodes.

   (c) Estimate the empirical degree distribution $\hat{F}^b_N(d)$ and the best fitting de Solla Price model $F^{sp}_{c,N}(d;\hat{w}^b,\hat{m}^b)$ for each of the bootstrap samples $b = 1,\ldots,B$.

   (d) Estimate the bootstrap variance $\hat{z}(\cdot;\hat{w},\hat{m})$ of the difference $|\hat{F}_N(d) - F^{sp}_{c,N}(d;\hat{w},\hat{m})|$.

   (e) For each bootstrap replication, calculate the test statistic, $T^1,\ldots,T^B$ using Equation (2.4).

3. Step 3: test distribution under the alternative hypothesis with tail jump $h_c$ as shown in Figure 2.3:

   (a) Fix the step size $h_c \in [0.01, 0.1]$.
   (b) Truncate $d_1, \dots, d_B$ according to $h_c$, obtaining $d_1^{H_1}, \dots, d_B^{H_1}$.
   (c) Estimate $\hat{F}_N^b(\cdot)$ and $F_{c,N}^{sp}(\cdot; \hat{w}_{H_1}^b, \hat{m}_{H_1}^b)$ on the basis of $d_b^{H_1}$, with $b = 1, \dots, B$.
   (d) Calculate the test statistics, $T_{H_1}^1, \dots, T_{H_1}^B$.

4. Step 4: calculate p-value and power

   (a) Calculate the test statistic on the original data $T^{obs}$.
   (b) Calculate the p-value as the rate of bootstrap statistics that exceed the original statistic p-value= $\frac{\sum_{b=1}^{B} \mathbb{1}\left(T^b > T\right)}{B}$, where $\mathbb{1}(\cdot)$ is the indicator function.
   (c) Obtain $T_{0.95}$ as the 95% quantile of the bootstrap distribution.
   (d) Calculate the power as the rate of $H_1$ statistics that are rejected by the test power= $\frac{\sum_{b=1}^{B} \mathbb{1}\left(T_{H_1}^b > T_{0.95}\right)}{B}$.
   (e) Select the largest $c$ for which the power is at least 80%.

## 2.5 Testing 4482 network for power-law degree distributions

We applied our testing framework to the datasets reported in Broido and Clauset [2019], which consists of a large corpus of nearly 1000 network data sets drawn from social, biological, technological, and informational sources. From these networks, the authors derived 4482 observed degree sequences. The corpus of real-world networks includes both simple graphs and networks with various combinations of directed, weighted, bipartite, multigraph, temporal, and multiplex networks.

Similar to the authors in the original paper we are interested in testing whether the networks exhibit power-law degree distributions. For each degree distribution, we applied our testing framework for several values for the tail sensitivity $h_c = [0.01, 0.015, 0.02, 0.03, 0.05, 0.1]$, fixing a cutoff $c$ at degree 10. For lower values of cutoff, the test tends to reject most of the networks as de Solla Price, because of the other regimes present in the lower degrees that are irrelevant for power-law tail behavior.

By fixing $c$ and $h_c$, it may occur that various networks do not achieve the required power of 80%. Those networks are excluded. Figure 2.4 shows the

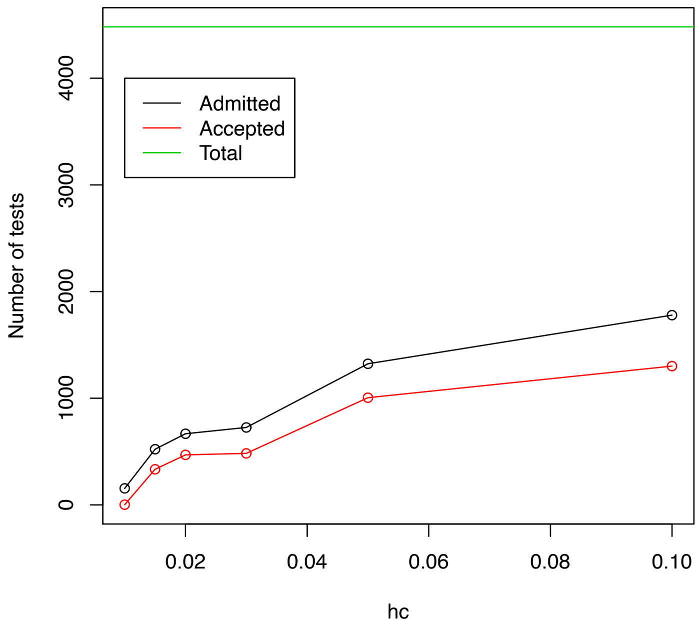


Figure 2.4. The green line shows the overall number of 4482-degree distributions that are possible to test. The black line shows the number of admissible tests that have a power greater than 80% with respect to tail sensitivity $h_c$. The red line illustrates the number of tests for which the de Solla Price power-law seems to be a sensible model.

absolute number of degree distributions that are admissible to being tested, i.e., with power higher than 80%, as well as the absolute number of accepted tests, i.e., tests for which the power-law null distribution could not be rejected.

Figure 2.5 shows the $H_0$ acceptance rate over different $h_c$ values, as the rate of the non-rejected power-laws over the total number of tested networks. The lower $h_c$, the lower the number of admissible networks to be tested. Nevertheless, the rate of networks for which the de Solla Price power-law cannot be rejected is almost constant for $h_c > 0.01$. Using the common elbow rule [Thorndike, 1953], a common practice among engineers, we select a very strong tail sensitivity $h_c = 0.015$ for which 64% of the tested networks exhibit power-law behavior. For each of the non-rejected networks, we calculate the power-law exponent, $\hat{\gamma} = 2 + \hat{w}/\hat{m}$, with estimated parameters shown in Figure 2.6. We find that for the more restrictive tests ($h_c = 0.015$), all the exponents are between 2 and 3,

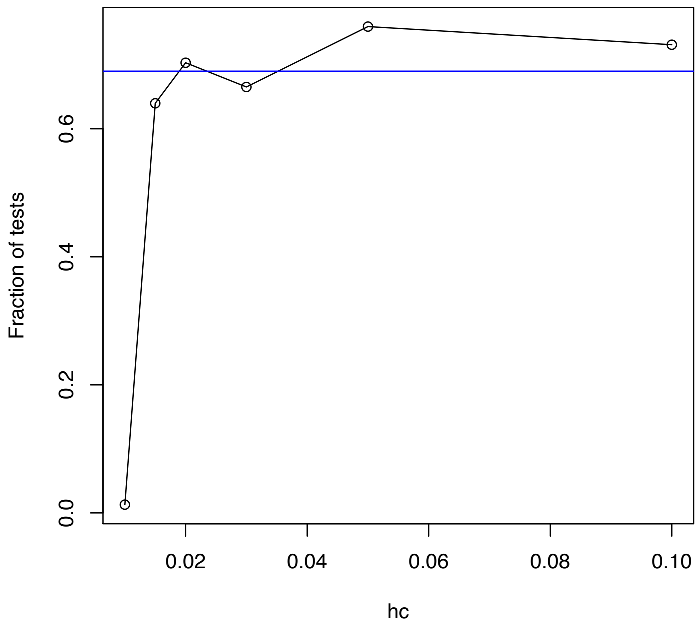


Figure 2.5. Fraction of accepted tests, i.e., rate of detected power-law networks, for $h_c = [0.01, 0.015, 0.02, 0.03, 0.05, 0.1]$. Note that the rate is stable for $h_c > 0.01$. This suggests that roughly 2/3 of all considered real-world networks seem to exhibit power-law tail behavior.

whereas for the most liberal tests ($h_c = 0.1$), 99.1% of all exponents are associated with what is normally called scale-free power-laws. As this acceptance rate stays constant for increasing values of $h_c$ and of the number of admissible networks and as the power-law exponent is between 2 and 3 for almost all accepted degree distributions, we speculate that approximately 2/3 of all empirical, large-scale networks, which can reasonably be considered to have been drawn from some underlying infinite network, are scale-free power-law networks.

Although we have obtained positive evidence that power-law networks are not rare among larger recorded networks that have sufficient observations in the tail, for the most stringent testing scenario with $h_c = 0.015$ we tested only 500 out of the 4482 networks, whereas for the most liberal value $h_c = 0.1$ we could test slightly less than half of all networks. If the tail is not big enough, parameter estimation and testing could be misleading, generating inconclusive results about the nature of the underlying degree distribution.

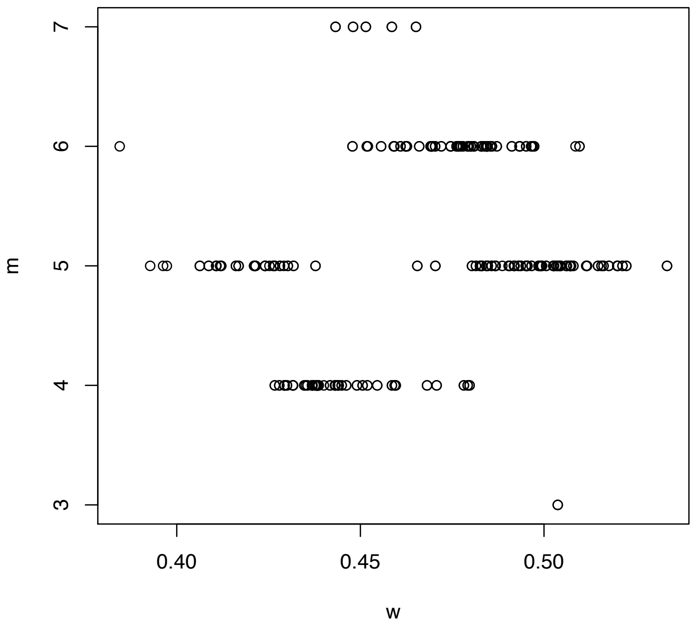


Figure 2.6. Estimates of $(w, m)$ for accepted tests at $h_c = 0.015$

## 2.6 Conclusions

Are power-law degree distributions rare or everywhere [Holme, 2019]? It is perhaps surprising that after 20 years of network science, this issue still has not been resolved and has suddenly flared up again in the scientific debate. As the question has important philosophical and conceptual consequences, however, it is perhaps more surprising that it has taken 20 years before careful technical reviews, such as by Voitalov et al. [2019], have considered this question methodologically. With this current paper, we hope to have contributed to this recent methodological progress.

In this paper we have developed a tail testing procedure, taking into account a host of issues related to testing degree distributions of a single empirical network. We have presented the behavior of the Kolmogorov-Smirnov statistic for the discrete degree distributions, making corrections in order to achieve an even sensitivity on the observed degrees. We have presented an alternative power-law degree distribution that can be tuned to specify the size of the deviation from the power-law, and then use it to calculate the power for the test. The degree of dependency and other issues have been solved by bootstrapping the test distribution via de Solla Price growing network process. The aim of this work is to propose a rigorous approach to test with sufficient power whether sequences of dependent node degrees can be distinguished from a specific power law distribution in the tail. What we mean with rigorous is that given the definition of the modified KS test statistic, our testing procedure is exact, i.e., with exact coverage and power, up to the precision of the bootstrap sampling. Although a power-law is a property that has sometimes been explicitly associated with the in-degree distributions [Mitzenmacher, 2004], our testing framework can be applied to any arbitrary degree sequence, whether in-degree, out-degree, or full-degree distribution, both for directed and undirected simple networks.

Our aim was to re-evaluate the conclusion from Broido and Clauset [2019] by applying our testing framework to 4482 empirical degree distributions. However, in contrast to their claim that power-law distributions are rare, we classified approximately 64% of the networks, for which we have sufficient power, as power-law — and most of those as scale-free. Our conclusion is that power-law networks are not rare at all. Furthermore, we note that in this framework we just tested for power-law networks using the de Solla Price model, which is a small subclass of power-law degree networks. This suggests that an even larger number of real-world networks could be classified as power-law had we used a larger power-law class as the null. Clearly, power-law networks seem empirically ubiquitous.

## 2.7 Supplementary Material

### 2.7.1 Brownian Bridge

For completeness, we reproduce here the standard derivation of the Brownian bridge variance for independent samples [Chicheportiche and Bouchaud, 2011]. Let $X$ be a random vector of $n$ independent and identically distributed variables with marginal cdf $F$, with realization $x_1, \ldots, x_n$. For a given number $x$ in the support of $F$, we define $Y(x)$ the random vector in which $Y_i(x) = \mathbb{1}_{\{X_i < x\}}$ is a Bernoulli variable. Then

$$\mathbb{E}\left[Y_i(x)\right] = F(x)$$

$$\mathbb{E}\left[Y_i(x)Y_j(x')\right] = \begin{cases} F(\min(x, x')) & , i = j \\ F(x)F(x') & i \neq j \end{cases}$$

The centered sample mean of $Y(x)$ is:

$$\bar{Y}(x) = \frac{1}{n}\sum_{i=1}^{n} Y_i(x) - F(x)$$

Denoting $u = F(x)$ and $v = F(x')$, the covariance function of $\bar{Y}$ is:

$$\mathrm{Cov}(\bar{Y}(u), \bar{Y}(v)) = \frac{1}{n}(\min(u,v) - uv)$$

and the sample mean can be rewritten as

$$\bar{Y}(u) = \frac{1}{n}\sum_{i=1}^{n} Y_i(F^{-1}(u)) - u$$

We define the process $y(u)$ as the limit of $\sqrt{n}\bar{Y}(u)$ when $n \to \infty$. According to the Central Limit Theorem, it is Gaussian, and its covariance function is given by:

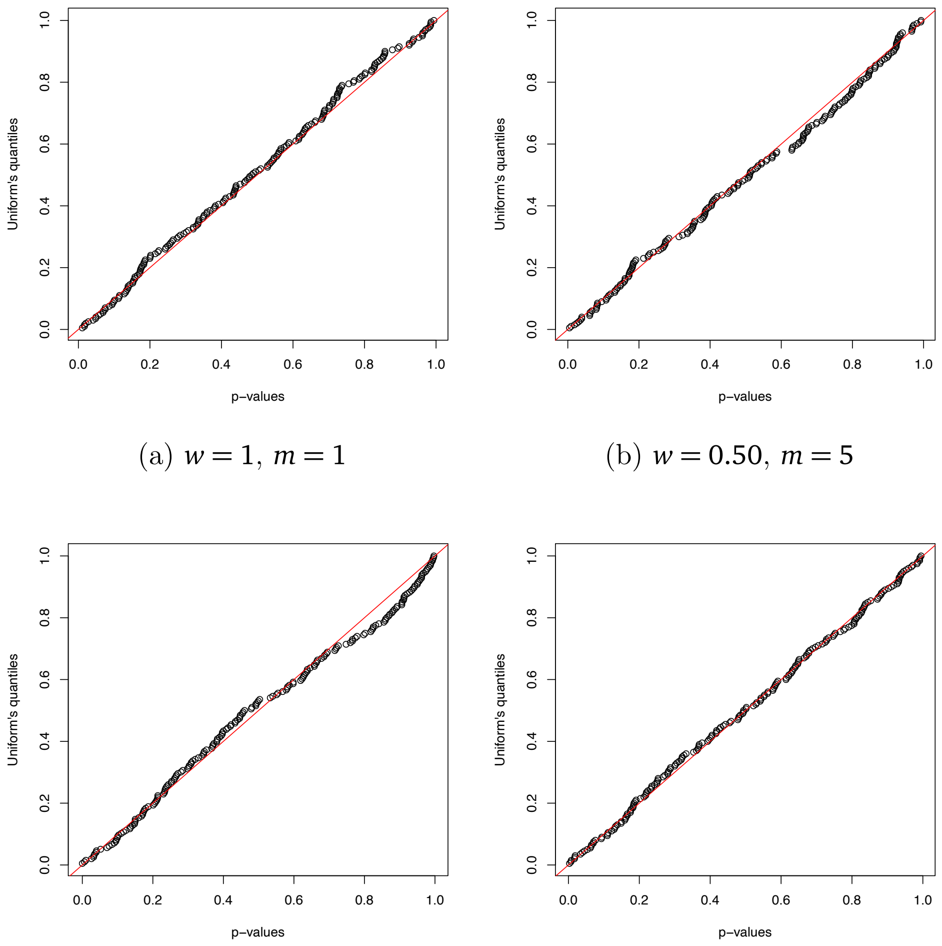
$$I(u,v) = \min(u,v) - uv$$

and thus variance

$$I(u,u) = u - u^2 = u(1-u).$$

### 2.7.2 Simulation study: testing the test

A common practice when dealing with novel statistical methodologies is to run a simulation study. The aim is to check the validity of the procedure in a controlled environment. In the case of a testing procedure, this means checking the Type I Error [Sahoo, 2013] or equivalently the uniformity of p-values. If the procedure is correct, we expect that the p-values have Uniform distribution under the null hypothesis. The simulation study is articulated as follows: for an arbitrarily fixed $(w, m)$ we simulate $B = 200$ realizations of de Solla Price degree distributions, on each of them we apply the testing procedure retrieving a p-value. We verify through qqplot their uniformity. Finally, we repeat the simulation study for different values of $(w, m)$. Figure 2.7 reports some of these cases, showing that the p-values fit quite well with the Uniform distribution, leading to trust in our results on the real datasets.



(a) $w = 1$, $m = 1$

(b) $w = 0.50$, $m = 5$

(c) $w = 0.45$, $m = 4$

(d) $w = 0.48$, $m = 6$

Figure 2.7. We present some qqplots of pvalues versus the quantiles of a Uniform distribution, simulations performed using different parameter settings.

# Chapter 3

# Dynamic latent space relational event model

I declare that the content of this chapter comes from the original paper [Artico and Wit, 2023b] which is published in the Journal of the Royal Statistical Society Series A: Statistics in Society in collaboration with E.C. Wit.

## 3.1 Summary

Dynamic relational processes, such as e-mail exchanges, bank loans, and scientific citations, are important examples of dynamic networks, in which the relational events consist of time-stamped edges. There are contexts where the network might be considered a reflection of underlying dynamics in some latent space, whereby nodes are associated with dynamic locations and their relative distances drive their interaction tendencies. As time passes nodes can change their locations assuming new configurations, with different interaction patterns.

The aim of this paper is to define a dynamic latent space relational event model. We then develop a computationally efficient method for inferring the locations of the nodes. We make use of the Expectation Maximization algorithm which embeds an extension of the universal Kalman filter. Kalman filters are known for being effective tools in the context of tracking objects in space, with successful applications in fields such as geolocalization. We extend its application to dynamic networks by filtering the signal from a sequence of adjacency matrices and recovering the hidden movements. Besides the latent space, our formulation includes also more traditional fixed and random effects, achieving a general model that can suit a large variety of applications.

## 3.2 Introduction

Networks appear in many contexts. Examples include gene regulatory networks [Signorelli et al., 2016], financial networks [Cook and Soramaki, 2014], psychopathological symptom networks [De Vos et al., 2017], political collaboration networks [Signorelli and Wit, 2018], and contagion networks [Užupytė and Wit, 2020]. Studying networks is important for understanding complex relationships and interactions between the components of the system. The analysis can be difficult due to the many endogenous and exogenous factors that may play a role in the constitution of a network. The aim of statistical modeling in this context is to describe the underlying generative process in order to assist in identifying the drivers of these complex interactions. These models can assist in learning certain features of the process, filtering noise from the data, thereby making interpretation possible.

In this manuscript, we are considering temporal random networks, whereby nodes make instantaneous time-stamped directed or undirected connections. Examples are email exchanges, bank loans, phone calls, and article citations. A common approach to these networks has been flattening the time variable and studying the resulting static network. Although this method simplifies the complexity of the calculations, clearly there is a loss of information about the temporal structure of the process. Most networks are inherently dynamic. Subjects repeatedly create ties through time. Since the adjustment of ties is influenced by the existence and non-existence of other ties, the network is both the dependent and the explanatory variable in this process [Brandes et al., 2009]. Thus rather than viewing this as a static network, we consider the generative process as a network structure in which the actors interact with each other through time. Edges are defined as instantaneous events. This quantitative framework is known as *relational event modeling*.

The basic form of a relational event model as an event history model can be found in Butts [2008] with an application to communications during the World Trade Center disaster. The model has been extended by Brandes et al. [2009] to weighted networks: nodes involved in these events are actors, such as countries, international organizations, or ethnic groups. An event is assigned a positive or negative weight depending on a cooperative or hostile type of interaction, respectively. Other examples of relational event modeling include the work by Vu et al. [2017] on interhospital patient transfers within a regional community of health care organizations or the analysis of social interaction between animals [Tranmer et al., 2015].

In a relational event model, the connectivity may depend on the past evolution of the network. Keeping track of the past is challenging for dynamic networks because of the high number of possible configurations (k-stars, k-triangles, etc.) that could be taken into account, as well as their closure time and the time they keep affecting future configurations. We thus propose to take some kind of summary of the past configurations. A solution that can both summarize the process and approximate effectively the past information is the idea of a dynamic latent space. To describe the latent structure of a network one can think of placing the vertices in a space where the distance between two points describes the tendency or lack of tendency to connect. Among social scientists, this is typically called a *social space* where actors with more interactions are close together and vice versa [Bourdieu, 1989]. The locations are allowed to change in time. At each time point, new connections are formed and the subjects develop attraction/repulsion that forces them to change their social space configuration. The new configuration is the one that best reflects the new connectivity behavior. As a result, one location at a certain time reflects past information, within the limits of the latent space formulation. This evolution describes the social history of the subjects, their preferences, and the groups they might join or leave.

There are other temporal network models. The stochastic actor-oriented model [Snijders and Pickup, 2017] defines relationships between social actors that can be created and destroyed. This model is very useful to model interactions that extend in time but are less suitable to model instantaneous interactions, such as communication, patent citations, or financial transactions. The temporal exponential random graph models [Hanneke et al., 2010] models sequences of networks. This approach is agnostic about the underlying generative process, but typically would also focus on persistent network relations. Here we focus on instantaneous interactions, which makes the use of relational event models the method of choice.

### 3.2.1 Related work and novelty of the proposed method

The problem of tracking latent locations has been studied by many authors, specifically for the static case, i.e., tracking locations under the assumption that they are fixed over time. For static binary random graphs Hoff et al. [2002] provide a framework for inference. Some extensions of that model have been developed to overcome the limitations of the latent space formulation [Hoff, 2005, 2008, 2009]. The well-known stochastic block model describes the similarity between the actors by grouping them together, which is similar to latent space formulation. An extension of stochastic block modeling to relational event data is provided by DuBois et al. [2013].

An approach for modeling latent space dynamic binary networks was proposed by Sarkar and Moore [2005]. The method is based on an initial preprocessing phase where rough location guesses are found through generalized multidimensional scaling, followed by an estimation phase in which the dynamic locations are treated as fixed parameters and optimized via a conjugate gradient method. The distances between nodes are approximated by thresholding larger ones and including an additional penalty for forcing distant nodes to be closer. In this work, we avoid making ad hoc inference assumptions.

Sewell and Chen [2015] propose a Bayesian latent space model for temporal binary networks where its radius interpretation of the linear predictor reduces to a Hoff et al. [2002] model with the addition of node-specific random effects. The method employs a Metropolis-within-Gibbs approach, whose computational burden of MCMC integration increases exponentially with the latent dimension $d$, the number of nodes $p$ and the number of time points $n$. Although case-control sampling [Raftery et al., 2012] reduces the likelihood computation from $O(np^2)$ to $O(np)$, its accuracy depends on extensive stratification. By considering one control stratum, Sewell and Chen [2015] weigh heterogeneous distances in the same way, producing a bias. This leads to the paradoxical overlapping of unconnected nodes. Durante and Dunson [2016] developed a Bayesian approach using Polya-Gamma data augmentation for binary links and Gaussian processes for parameter dynamics combined with a non-Euclidean dissimilarity measure. In contrast to the previous two Bayesian approaches, we tackle the problem from a frequentist perspective, which does not require data augmentation. Our Expectation-Maximization algorithm combined with a Kalman filter is deterministic and does not suffer from Bayesian convergence issues. It scales linearly with the number of time periods and achieves a good latent representation after a few iterations. It can scale to several hundred nodes without case-control subsampling. Moreover, whereas Durante and Dunson [2016] assume a discrete time sequence of binary adjacency matrices, we embed our discrete-time observation process into an often more realistic continuous time relational event process. Furthermore, we explicitly consider the availability of covariates, which allow for further disentanglement of known drivers of the interaction dynamics from the unknown factors. Although non-Euclidean alternatives can easily be added, our implementation focuses on an easily interpretable Euclidean latent space.

### 3.2.2 The methodology presented

A dynamic latent space model is particularly useful in an exploratory stage of the analysis. It allows for an interactive investigation of the data to generate hypotheses about the drivers of the generative process by seeing which nodes are close and which nodes are far apart, as well as the way they develop through time. The most obvious example of this approach is simply by visualizing the development of the latent node locations in two dimensions. However, simple multivariate analysis tools, such as PCA, can also explore latent spaces with higher dimensions. If the aim of the analysis is entirely predictive, then the latent space model itself may be of interest as it can be used to generate predictions without knowing the underlying drivers of the process.

The aim of this manuscript is to develop an efficient inference scheme for a relational event process embedded in a latent Gaussian process. The framework is very general and can be extended to networks with weighted edges of any exponential family distribution. There are two dual representations of the process, either as a continuous time exponential or as discrete Poisson counts. Depending on the sparsity of the observed process, one or the other can be selected in the inference procedure. Furthermore, the theoretical burden of the Expectation Maximization framework in the model has been reduced to two analytical steps: for the E-step a Kalman filter and smoother is used, whereas for the M-step a generalized linear model framework is derived. Both are provided by modern packages. Our latent space relational event framework provides an accurate, simple, and computationally efficient way of inferring a wide general class of dynamic social network models.

Section 3.3 describes a motivating patent citation network example. In section 3.4 several formulations of the latent space relational event model are presented. In section 3.5 we propose an efficient inference method that is based on combing the state-space formulation of the model with the EM algorithm. In section 3.6 we check the performance and limitations of our method via a simulation study. In section 3.7 we analyze the latent structure of technological innovations, by studying over 23 million patent citations from 1967 until 2006.

## 3.3 Patent citation networks

Patents are legal documents of intellectual property that testify of some technological innovation. Innovation itself is a complicated process and involves both true novelties as well as the adaptation of existing ideas in a new context. Within the patenting process, this borrowing of existing ideas is referred to as *patent citations*: each inventor that submits a patent to a patent office is required by law to include the current state-of-the-art on which the current patent is based by citing those patents in which those ideas have been deposited.

By tracing which patents cite which other patents, it is possible to establish a dynamic network in which patents accumulate over time citations from other patents. Alternatively, it is possible to group patents together into clusters and track how these clusters cite and are cited by other clusters. Either way, the process of citation shows how certain patents at certain times are particularly important in the technological innovation process. As innovation is important for economic progress and prosperity, it is little surprise that the analysis of the patent citation network has become an important field of study. It is of particular interest to find out what drives technological innovation [Lafond and Kim, 2019]. Furthermore, economists are eager to find out whether or not the innovation process is changing over time.

The International Patent Classification (IPC) scheme is a hierarchical clustering scheme for patents. It assigns each patent to eight main classes, to wit,

A : Human necessities: agriculture, foods, tobacco, personal or domestic articles, health, life-saving, amusement.

B : Performing operations and Transporting: separating, mixing, shaping, printing, transporting, nanotechnology.

C : Chemistry and Metallurgy

D : Textiles; Papers.

E : Fixed constructions: building, earth drilling.

F : Mechanical Engineering; Lightning; Heating; Weapons; Blasting.

G : Physics: instruments, nuclear.

H : Electricity.

Within each main class, there are a large number of subclasses, resulting in overall roughly 500 subclasses. Each subclass has again a number of groups and subgroups, which for the purposes of the analysis here we will ignore. Also, other grouping schemes are possible [Younge and Kuhn, 2016].

The National Bureau of Economic Research in the U.S. released in 2010 patent citation data, consisting of 3.1 million patents, 23.6 million citations over the period 1967-2006, with collection intervals of 1-year length. By studying how citing behavior and cited tendency of the classes and the subclasses change over time, we aim to answer some of the questions we posed above. The latent representation allows for a straightforward similarity assessment, showing which fields are becoming more heterogeneous in their citation patterns. The aim is to develop a methodological framework for inferring dynamic latent space tracking of the technology classes and to show how this changes the nature of patent citations.

## 3.4 Latent space relational event models

In this section, we introduce a general version of a latent space relational event model. We consider a set of actors, defined as a finite vertex set $V = \{1, \ldots, p\}$, that can exchange links or edges in time. In principle, we will consider the exchange of relational events, such as discrete interaction, e.g., sending an email or citing a patent, but we will also consider extensions to the quantitative exchanges, such as import and export. As drivers of the exchange process, we consider both endogenous, such as reciprocity, and exogenous variables, such as vertex characteristics. One particular exogenous variable is the relative location of the vertices in some Euclidean latent space, which itself is defined as a dynamic process.

We consider a non-homogeneous multivariate Poisson counting process $N = \{N_{ij}(t) \mid i, j \in V, t \in [0, T]\}$ and a state-space process $X = \{X_i(t) \in \mathbb{R}^d \mid t \in [0, T], i = 1, \ldots, p\}$ relative to some standard filtration $\mathscr{F}$. In particular, we consider $\mathscr{F}$-measurable rate functions $\lambda_{ij}(t)$ that drive the components of the counting process. In particular, we assume that the rates $\lambda_{ij}(t)$ are functions of the underlying positions $X_i(t)$ and $X_j(t)$, besides possible other exogenous characteristics $B_{ij}(t)$ and endogenous features $N(t)$,

$$\lambda_{ij}(t) = g(d(X_i(t), X_j(t)), B_{ij}(t), N(t)),$$

for some measurable function $g$. Two common choices for the way that the rate depends on the locations are either as a function of the squared distance,

$$d(X_i(t), X_j(t)) = ||X_i(t) - X_j(t)||^2$$

or the relative activity dissimilarity $d(X_i(t), X_j(t)) = \frac{<X_i(t), X_j(t)>}{||X_i(t)||}$ between $i$ and $j$ [Hoff et al., 2002]. The former induces a symmetric interpretation, whereas the latter allows for a more complex asymmetric interpretation of the state-space. In this manuscript, we mainly focus on the Euclidian distance, as we prioritize visual interpretation of the results. However, it is important to mention that switching to another dissimilarity measure requires very little effort. The interaction dynamics $\lambda_{ij}(t)$ can be highly structured and parametrized, i.e., $g = g_\theta$, whereas the state-space dynamics is assumed to be a random walk at equally spaced time points $t_k^x$ in $[0, T]$,

$$X_{t_k^x} = X_{t_k^x} + v_k, \qquad k = 1, \dots, n_x \tag{3.1}$$

with $v_k \sim N(0, \Sigma)$ and $t_0^x = 0$. In this manuscript, we use sometimes the more compact notation $x_k = X(t_k^x)$ or $X(k)$ when we find it more convenient. The covariance matrix $\Sigma$ regulates the evolution of the latent process: a large variance allows longer jumps. Given the joint formulation $(X, N)$ of the state-space and interaction process, we will assume that only the interaction process $N$ is observed and the main aim of this paper is to infer the structure of the state-space $X$ and the rate functions $\lambda$, or more specifically, the parameter $\beta$ associated with functional form $\lambda = g_\beta$.

Next, we will consider two particular special cases of the latent space formulation of the interacting point process defined above. First, we consider the general case, in which the relational events are observed in continuous time. This is the traditional setting for relational events. We will also define a relational event model where the interactions can only happen at specific times. For example, bibliometric citations or patent citations only happen at prespecified publication dates. Furthermore, this model allows a generalization to non-binary relational events, such as export between countries, that can be dealt with in the same inferential framework.

### 3.4.1 Continuous time relational event process $N$

We consider a sequence of $n_e$ relational events, $\{(i_1, j_1, t_1^e), \dots, (i_{n_e}, j_{n_e}, t_{n_e}^e) \mid t_i^e \in [0, T],\ i, j \in V\}$ observed according to the above defined relational counting process $N$. In a latent space relational event model, the rate is defined as

$$\log \lambda_{ij}(t, x, \beta) = -d(x_i(t), x_j(t)) + \beta_G^t B_{ij}(t) + \beta_D^t s(\{N(\tau) | \tau < t\}). \tag{3.2}$$

where the latent space effect $d(X_i(t), X_j(t))$ that captures the “vicinity” of the actors. The drivers of the network dynamics can be of various types: *exogenous effects*, $\beta_G^t B_{ij}(t)$, such as global covariates, node covariates, edge covariates, as well as *endogenous effects*, $\beta_D^t s(\{N(\tau) | \tau < t\})$, where network statistics $s()$ capture endogenous quantities such as popularity, reciprocity, and triadic closure.

The parameter vector $\beta$ determines the relative importance of the various effects.

Conditional on the process $X$, the distribution of the interarrival time for interaction $i \to j$ are generalized exponentials, with instantaneous rates as described in (3.2) and interval rates

$$\mu_{k,ij}(x_k,\beta) = \int_{t_k^x}^{t_{k+1}^x} \lambda_{ij}(t,x,\beta)\,dt = e^{-d(x_i(t_k^x),x_j(t_k^x))}c_{ij}(k,\beta), \tag{3.3}$$

where $c_{ij}()$ is the remaining integral and latent distance $d()$ between the nodes is constant over the interval. The full log-likelihood of the complete process $\{X,N\}$, can be factorized in two components,

$$\ell(\beta,\Sigma) = \log p_\beta(n|x) + \log p_\Sigma(x), \tag{3.4}$$

where $\log p_\Sigma(x) = -\frac{n_x}{2}\log|\Sigma| - \frac{1}{2}\sum_{k=1}^{n_x}(x_k - x_{k-1})'\Sigma^{-1}(x_k - x_{k-1})$ and $\log p_\beta(n|x) = -\sum_{i\neq j}\sum_{k=1}^{n_x}\mu_{k,ij}(x_k,\beta) + \sum_{k=1}^{n_e}\log\lambda_{i_k j_k}(t_k^e, x_{t_k^e},\beta)$, where the generalized exponential formulation is the one adopted by Rastelli and Corneli [2021]. Although it is common in the REM literature to simplify inference by using the partial likelihood, we keep the generalized exponential component, as it can be estimated more easily in the M-step of the EM algorithm, described in section 3.5.

### 3.4.2 Discrete time relational event process $Y$

Often relational events are "published" only on prespecified discrete event times $\mathscr{T} = \{t_1^e,\ldots,t_n^e\}$. For simplicity of notation, we will assume that the relational event collection process and the jumps of the latent space are equal, i.e., $n = n_x = n_e$ and $\{t_1 = t_1^x = t_1^e,\ldots,t_n = t_n^x = t_n^e\}$. We make an additional assumption that the rate $\lambda$ is constant with respect to the endogenous and exogenous variables inside the collection intervals $(t_k, t_{k+1}]$. In fact, with respect to the endogenous variable $N$ it makes sense that no further information between the publication dates affects the rates. In other words, assuming a log link for the hazard, for $t \in (t_k, t_{k+1}]$

$$\log\lambda_{ij}(t,x,\beta) = -d(x_i(t_k),x_j(t_k)) + \beta_G^t B_{ij}(t_k) + \beta_D^t s(\{N(\tau)|\tau \le t_k\}). \tag{3.5}$$

As the interactions $i \to j$ are collected at $t_{k+1}$ from the observation intervals $(t_k, t_{k+1}]$, the resulting interval counts

$$y_{k,ij} = N_{ij}(t_{k+1}) - N_{ij}(t_k)$$

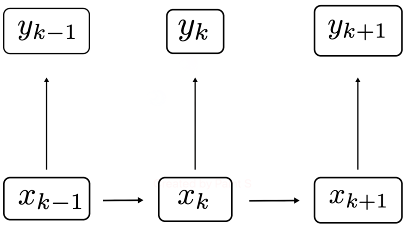


Figure 3.1. The observed counts $y_k$ are a result of the dynamics in nodes locations $x_k$. Hence, $y$ is independent conditionally to the latent locations $x$.

of the number of interactions between $i$ and $j$ are Poisson distributed with interval rate,

$$\mu_{k,ij}(x_k,\beta)=\int_{t_k}^{t_{k+1}}\lambda_{ij}(t,x,\beta)\,dt=(t_{k+1}-t_k)\lambda_{ij}(t_k,x,\beta). \tag{3.6}$$

An advantage of using discrete time is the reduction of the model complexity. It is not uncommon to observe thousands, even millions of links. Such numbers are not surprising when we consider $p(p-1)$ processes having an expected number of links $\mathbb{E}[\sum_{p(p-1)} N_{ij}(t)]$ that grows rapidly. The model can be written as a discrete-time state-space process,

$$\begin{cases} x_k & \sim & N(x_{k-1},\Sigma), \quad & k=1,\dots,n \\ y_{k,ij} & \sim & \mathrm{Poi}(\mu_{k,ij}(x_k,\beta)), & 1\le i\ne j\le p. \end{cases} \tag{3.7}$$

Given the complete observations $(x,y)$, the complete log-likelihood for the state space model in (3.7) can again be factorized in two components,

$$\ell(\beta,\Sigma)=\log p_\beta(y|x)+\log p_\Sigma(x), \tag{3.8}$$

where $\log p_\beta(y|x)=-\sum_{kij}\mu_{k,ij}(x_k,\beta)+\sum_{kij}y_{ij}(k)\log\mu_{k,ij}(x_k,\beta)$ and $\log p_\Sigma(x)$ as above, where the factorization is according to the directed graph in Figure 3.1, where $y_k \perp y_{-k}, x_{-k}|x_k$ and $x_{k+1}\perp x_{k-1}|x_k$. Similar to Butts [2008] and Perry and Wolfe [2013], who focused on non-homogeneous exponential waiting times, this approach focuses on non-homogeneous Poisson counts.

One advantage of the latent space formulation is the dimensionality reduction in the latent representation. As the number of nodes $p$ increases the number of observed counts $p(p-1)n$ grows quadratically while the latent space grows linearly as $pdn$.

Dynamic exponential family network model. Given the state space formulation in (3.7), it is possible to generalize the model considering connections drawn from any exponential family distribution without changing the inference procedure. In fact, ignoring the connection with any underlying counting process, we could define a temporal network process on discrete time intervals $k$ ($k \in \{1, \dots, n\}$) between nodes $i$ and $j$ as $f(y_{ij}(k)) = \exp((y_{ij}(k)\theta_{ij} - b(\theta_{ij}))/a(\varphi) + c(y_{ij}(k), \varphi)$, where $\theta_{ij}$ is the edge-specific canonical parameter. Using the canonical link function, we can specify the canonical parameter in a similar fashion to (3.5),

$$\theta_{ij}(x_k) = -d(x_{ki}, x_{kj})$$

where the values for $x$ are the latent states as before. It is also possible to add additional covariates, but we do not consider this case here. In Supplementary Materials 3.9.4 we show how to obtain the Kalman update equation for any exponential family. The inferential method presented in this manuscript remains mostly the same with a minimal change, effectively replacing the mean $\mu(x_k)$ and variance $R_k$ of the process by

$$\mu(x_k) = b'(\theta)|_{x_k} \text{ and } R_k = b''(\theta)a(\varphi)|_{x_k}.$$

This generalized temporal network model can be used to model import and export or other dynamic networks with weighted edges.

Marginalization One of the main advantages of our latent space network model is that, unlike many other network models, it is coherent under sampling a subset of nodes. Given that any subset $V'$ of $V$ maintains the same distances among nodes, the distribution of the restricted node set $P_{V'}$ is the same as the marginalized distribution of the full model $P_V|_{V'}$. This invariance means that it is unimportant to which node set the observed nodes actually belong. Therefore, for the true latent dimension $d$, as well as for any dimension higher than that, the model is invariant under marginalization. The only effect of subsampling is on inference, in that the conditional variance of the latent locations given the restricted nodes is larger than when given the full node set $V$, as they have fewer triangulation opportunities.

## 3.5 Inference

In this section, we develop all the necessary steps for making inferences on the latent states $x_k$ and the parameters $\Sigma$ an $\beta$. Since the latent process, $x_k$ is unobserved, we aim to maximize $\int_x L(\beta, \Sigma; y, x)dx$. We use the Expectation Maximization (EM) algorithm [Dempster et al., 1977]. EM algorithm is widely used in problems where certain variables are missing or latent. The EM algorithm consists of an iterative maximization of the conditional expectation of the latent process $X|N, \beta, \Sigma$ with respect to the data.

Due to the stepwise dynamic of the latent locations (3.1) the expectation step is equivalent for both models presented in Section 3.4.1 and Section 3.4.2. As the locations are constant within intervals $\mathcal{T}$, the continuous time non-homogeneous exponential relational event model $N$ reduces to a discrete-time Poisson model during the E-Step.

$$Q(\beta, \Sigma|\beta^*, \Sigma^*) = \mathbb{E}_X[\ell(\beta, \Sigma)|y].$$

where $\beta^*, \Sigma^*$ denote the parameters estimated at the previous EM iteration. In the maximization step $Q(\beta, \Sigma|\beta^*, \Sigma^*)$ is maximized with respect to the parameters $\beta, \Sigma$. The two steps above are iterated until convergence is reached. The expectation step is typically challenging due to the high dimensional nature of the integral.

The expectation of the log-likelihood can approximately be written as a function of the first two conditioned moments $\mathbb{E}[x_k|y_{1:n}]$ and $\mathbb{V}[x_k|y_{1:n}]$. Exploiting the state space formulation of the model (3.7) we can estimate these two quantities with a Kalman filter and smoother [Kalman, 1960]. The filter derives the mean and variance of the latent process $x_k$ conditioned to the information on $y$ up to time k,

$$\hat{x}_{k|k} = \mathbb{E}[x_k|y_{1:k}] \qquad V_{k|k} = \mathbb{V}[x_k|y_{1:k}].$$

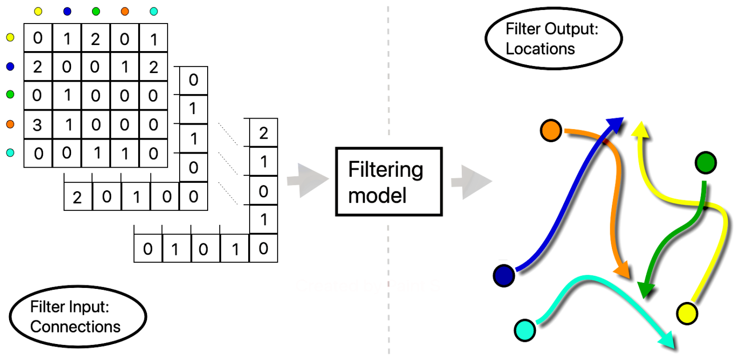
The smoother refines these quantities accounting for the complete information on $y$ up to time $n$,

$$\hat{x}_{k|n} = \mathbb{E}[x_k|y_{1:n}] \qquad V_{k|n} = \mathbb{V}[x_k|y_{1:n}].$$

The expected log-likelihood can be then calculated using these quantities obtained from the smoother.

### 3.5.1 E-Step: Extended Kalman Filter

The Kalman filter is one of the most popular algorithms for making inferences on state space models and it provides a solution that is both computationally cheap and accurate. Kalman filter is an iterative method that calculates the conditional



|  | ● | ● | ● | ● | ● |
|---|---|---|---|---|---|
| ● | 0 | 1 | 2 | 0 | 1 |
| ● | 2 | 0 | 0 | 1 | 2 |
| ● | 0 | 1 | 0 | 0 | 0 |
| ● | 3 | 1 | 0 | 0 | 0 |
| ● | 0 | 0 | 1 | 1 | 0 |

Figure 3.2. The filtering model takes as input a sequence of adjacency matrices and updates the node locations in the latent space.

distribution of the latent $x_k$. Given the causal DAG at Figure 3.1 $x_k$ depends on $x_{k-1}$ and the observed $y_k$. Assuming prior knowledge on the distribution of $x_{k-1}$ the conditional distribution of $x_k$ is calculated easily. The procedure is applied sequentially from time 1 to $n$, where the conditional distribution achieved at time $k$ becomes the prior knowledge for the next time point. An arbitrary distribution is specified for the initial $x_0$. Calculating the conditional distribution entirely could be difficult so the first moments are calculated only. The calculation of the conditional probability involves two steps that are universal in the filtering literature: predict and update. In order to be consistent with the aforementioned literature we denote $\hat{x}_{k|k} = \mathbb{E}[x_k|y_{1:k}]$ and $V_{k|k} = \mathbb{V}[x_k|y_{1:k}]$ as the expectation and variance conditioned of having observed $y_k$. Note that $x_k$ and $y_k$ are vectors of length $p_x = pd$ and $p_y = p(p-1)$ or $p(p-1)/2$ in case of an undirected network, respectively. These correspond to the vectorized coordinate and adjacency matrices at time $k$, respectively. $\Sigma$ is a $pd \times pd$ matrix and is constant over time. $R_k$ the observed data variance is a diagonal $p_y \times p_y$ matrix. The latent process conditional variance $V_k$ is a $p_x \times p_x$ matrix, whereas the Jacobian matrix $H_k$ is of size $p_x \times p_y$.

### Predict

Assume that at time $k-1$ the approximated conditional distribution of the latent locations is $x_{k-1|k-1} \sim N(\hat{x}_{k-1|k-1}, V_{k-1|k-1})$. For the initial case $k = 1$ we set

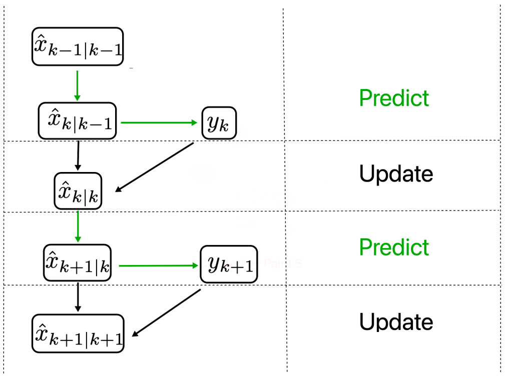


Figure 3.3. The filtering procedure can be summarized as a sequence of predictions and updates. At each time step a prediction on the observed links count is made. The prediction error is then propagated back to the nodes for updating their positions.

arbitrarily $x_{0|0} = v_0$ and $V_{0|0} = \Sigma_0$. The predict step calculates the first moments of $x_k$ conditioned to $y_{k-1}$. In fields such as physics, chemistry, or engineering it is common to employ a forward function $x_k = f(x_{k-1}) + v_k$ which is related to the physical properties of the system. In our case, the random walk formulation makes no constraints on the latent process evolution. The forward function is the identity with moments

$$\hat{x}_{k|k-1} = \mathbb{E}[x_{k-1} + v_k | y_{1:k-1}] = \hat{x}_{k-1|k-1}$$
$$V_{k|k-1} = \mathbb{V}[x_{k-1} + v_k | y_{1:k-1}] = V_{k-1|k-1} + \Sigma$$

These are called the apriori mean and variance of the latent locations before observing $y_k$. The prior distribution is $x_{k|k-1} \sim N(\hat{x}_{k|k-1}, V_{k|k-1})$.

### Update

The update step finalizes the calculation of the conditional distribution. We consider the mean vector of all the pairwise relationships $\mu(x_k, \beta) : \mathbb{R}^{p_x} \to \mathbb{R}^{p_y}$ described at (3.3) and (3.6) and covariance matrix $\mathbb{V}[y_k] = R_k$ where counts are independent with variance equal to the mean $R_k = \mu(x_k, \beta)\mathbb{I}_{p_y}$. In case a general dynamic network model using exponential family weighted edges, as described in Section 3.4.2, is considered then the mean $\mu(x_k)$ and variance $R_k$ vary accordingly.

Kalman filters assume that the observed process $y_k$ is Gaussian and the transformations involved are linear. The Extended Kalman Filter [Anderson and

Moore, 2012] overcomes the Kalman filter limitations. By means of a first-order Taylor expansion

$$\mu(x_k,\beta) = \mu(\hat{x}_{k|k-1},\beta) + H_k(x_k - \hat{x}_{k|k-1}), \qquad H_k = \frac{\partial \mu(x,\beta)}{\partial x}\Big|_{\hat{x}_{k|k-1}} \tag{3.9}$$

we calculate the expectation $\mathbb{E}[y_k|y_{k-1}] = \mu(\hat{x}_{k|k-1},\beta)$, variance $\mathbb{V}[y_k|y_{k-1}] = H_k V_{k|k-1} H_k' + R_k$ and covariance $\mathbb{C}ov[x_k, y_k|y_{k-1}] = V_{k|k-1}H_k'$ of the conditional predictive distribution of $y_k$.

The joint multivariate distribution of the observed and latent process is

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} \Big| y_{1:k-1} \sim \mathscr{L}\left(\begin{bmatrix} \hat{x}_{k|k-1} \\ \mu(\hat{x}_{k|k-1},\beta) \end{bmatrix}, \begin{bmatrix} V_{k|k-1} & H_k V_{k|k-1} \\ V_{k|k-1}H_k' & H_k V_{k|k-1} H_k' + R_k \end{bmatrix}\right)$$

where $\mathscr{L}$ is some probability law parametrized by the first two moments. Using the multivariate regression formulation we have the conditional moments of $x_k$

$$\begin{aligned} \hat{x}_{k|k} &= \mathbb{E}[x_k|y_{1:k}] = \hat{x}_{k|k-1} + K_k(y_k - \mu(\hat{x}_{k|k-1},\beta)) \\ V_{k|k} &= \mathbb{E}[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})'|y_{1:k}] = (\mathbb{I} - K_k H_k)V_{k|k-1}, \\ K_k &= V_{k|k-1}H_k'(R_k + H_k V_{k|k-1} H_k')^{-1}, \end{aligned} \tag{3.10}$$

see at Supplementary Materials 3.9.1 for more details. We hence obtain posterior distribution $x_{k|k} \sim N(\hat{x}_{k|k}, V_{k|k})$, which is approximated to be Gaussian. This will be the starting distribution for the inference at time $k+1$. The filtering procedure is shown in Algorithm 1. In Figure 3.2 we show a visual representation of the algorithm: at each time point the model takes as input an adjacency matrix and returns the locations in the latent space.

In the update step, the latent locations are updated according to the magnitude of the prediction error: a larger error in the prediction corresponds to a wider change in the locations. The filtering matrix $K_k$, capturing the linear relationship between the latent and observed processes, weights this prediction error. $K_k$ is the ratio between the noise $R_k$ and the latent variance $\Sigma$. Thus $K_k$ filters the prediction error according to the signal/noise ratio. Fahrmeir [1992] simply considers it as a single Fisher scoring step, see Supplementary Materials 3.9.4.

The Kalman filter can be interpreted both in a frequentist and Bayesian way. From a Bayesian perspective, the filtering procedure consists of a sequence of updates of the posterior mean and variance [Gamerman, 1991, 1992; West et al., 1985], whereas from a frequentist side, the estimation based on the posterior mode is equivalent to the maximization of a penalized likelihood [Fahrmeir and

**Algorithm 1** *Extended Kalman Filter*

```
Initialize $\hat{x}_{0|0} = v_0$ and $V_{0,0} = \Sigma_0$
for k = 1, ..., n do
    1. Filter prediction step
           $\hat{x}_{k|k-1} = \hat{x}_{k-1|k-1}$
           $V_{k|k-1} = V_{k-1|k-1} + \Sigma$
    2. Filter update step
           $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \mu(\hat{x}_{k|k-1}, \beta))$
           $V_{k|k} = (I - K_k H_k) V_{k|k-1}$
       where
           $K_k = V_{k|k-1} H_k^{'} (H_k V_{k|k-1} H_k^{'} + R_k)^{-1}$
           $H_k = \frac{\partial \mu(x,\beta)}{\partial x}\Big|_{\hat{x}_{k|k-1}}$
           $R_k = \mu(\hat{x}_{k|k-1}, \beta)\, \mathbb{I}_{p_y}$
```

Kaufmann, 1991; Fahrmeir, 1992], see Supplementary Materials 3.9.4. Approximating the posterior distribution with the same family of the prior, i.e., Gaussian, the posterior mean is equivalent to the posterior mode and hence the equivalence of the two approaches. This double interpretation makes Kalman filters appealing for both types of applications.

## Smoother

The smoother moves backward from the last prediction to the first. It calculates the first moments of the latent process conditioned to the information of all time points. Similarly, as the EKF, the backward matrix $B$ can be calculated considering the multivariate distribution of the latent locations at two consecutive time points,

$$\begin{bmatrix} x_{k-1} \\ x_k \end{bmatrix} \Big| y_{1:k-1} \sim N\left( \begin{bmatrix} \hat{x}_{k-1|k-1} \\ \hat{x}_{k|k-1} \end{bmatrix}, \begin{bmatrix} V_{k-1|k-1} & V_{k-1|k-1} \\ V_{k-1|k-1} & V_{k|k-1} \end{bmatrix} \right).$$

Using the multivariate regression formula we have the conditioned mean of $x_{k-1}$ over $x_k$

$$\mathbb{E}[x_{k-1}|x_k, y_{1:k-1}] = \hat{x}_{k-1|k-1} + B_k(x_k - \hat{x}_{k|k-1}) \quad \text{with} \quad B_k = V_{k-1|k-1} V_{k|k-1}^{-1}$$

According to the conditional independence in Figure (3.1) we have $(x_{k-1} \perp y_{k:n})|x_k$ since $x_k$ closes the dependency path. Using the iterated expectation rule

**Algorithm 2** *Smoother*

```
for k = n, ..., 1 do
    Backward step
        $\hat{x}_{k-1|n} = \hat{x}_{k-1|k-1} + B_k(\hat{x}_{k|n} - \hat{x}_{k|k-1})$
        $V_{k-1|n} = V_{k-1|k-1} + B_k(V_{k|n} - V_{k|k-1})B_k^{'}$
    where
        $B_k = V_{k-1|k-1}V_{k|k-1}^{-1}$
```

we have

$$\begin{aligned}\hat{x}_{k-1|n} &= \mathbb{E}[x_{k-1}|y_{1:n}] = \mathbb{E}[\mathbb{E}[x_{k-1}|x_k, y_{1:n}]\,|y_{1:n}] = \mathbb{E}[\mathbb{E}[x_{k-1}|x_k, y_{1:k-1}]\,|y_{1:n}]\\ &= \mathbb{E}\left[\hat{x}_{k-1|k-1} + B_k(x_k - \hat{x}_{k|k-1})|y_{1:n}\right]\\ &= \hat{x}_{k-1|k-1} + B_k(\hat{x}_{k|n} - \hat{x}_{k|k-1})\end{aligned}$$

where $\hat{x}_{k-1|k-1}$ and $\hat{x}_{k|k-1}$ are constants. In the same way, using the iterated variance rule

$$\begin{aligned}\mathbb{V}[x_{k-1}|y_{1:n}] &= \mathbb{E}[\mathbb{V}[x_{k-1}|x_k, y_{1:n}]\,|y_{1:n}] + \mathbb{V}[\mathbb{E}[x_{k-1}|x_k, y_{1:n}]\,|y_{1:n}]\\ &= V_{k-1|k-1} - B_k V_{k|k-1} B_k' + B_k V_{k|n} B_k'\\ &= V_{k-1|k-1} + B_k(V_{k|n} - V_{k|k-1})B_k',\end{aligned}$$

see at Supplementary Materials 3.9.2 for more details. The smoothing procedure is presented in Algorithm 2 and it is known as the Rauch-Tung-Striebel smoother. The final iteration of the smoother updates the starting values $\hat{x}_{0|0}$ and $V_{0|0}$. These values will be used as starting points for the successive EM iteration.

### 3.5.2 M-Step: generalized linear model

In the maximization step, we maximize the log-likelihood with respect to the parameters $\beta, \Sigma$ and we make the first distinction between the continuous (3.4) and discrete (3.8) time models. For the continuous time process $N$ the expected log-likelihood is

$$Q^N(\beta, \Sigma|\beta^*, \Sigma^*) = \mathbb{E}_X[\log p_\beta(N|X)|y_{1:n}] + \mathbb{E}_X[\log p_\Sigma(X)|y_{1:n}] = Q^E(\beta) + Q^G(\Sigma).$$

For the discrete-time process $Y$ the expected log-likelihood is

$$Q^Y(\beta, \Sigma|\beta^*, \Sigma^*) = \mathbb{E}_X[\log p_\beta(Y|X)|y_{1:n}] + \mathbb{E}_X[\log p_\Sigma(X)|y_{1:n}] = Q^P(\beta) + Q^G(\Sigma).$$

Notice that the Poisson component $Q^P(\beta)$ and exponential component $Q^E(\beta)$ do not depend on $\Sigma$, whereas the Gaussian component $Q^G(\Sigma)$ does not depend on the remaining parameters $\beta$. These quantities can therefore be optimized separately.

### Gaussian component

We can maximize the Gaussian component

$$Q^G(\Sigma) = -\frac{1}{2}\sum_{k=1}^{n} \mathbb{E}[(x_k - x_{k-1})'\Sigma^{-1}(x_k - x_{k-1})|y_{1:n}] - n\log|\Sigma| - \frac{n}{2}\log(2\pi).$$

finding the zero of the first derivative with respect to $\Sigma$. Rearranging the elements and taking the expectation as shown in Supplementary Materials 3.9.3 we obtain

$$\begin{aligned} \hat{\Sigma} &= \mathbb{E}\left[\frac{1}{n}\sum_{k=1}^{n}(x_k - x_{k-1})(x_k - x_{k-1})'\middle|y_{1:n}\right] \\ &= \frac{1}{n}\sum_{k=1}^{n} V_{k|n} + V_{k-1|n} + B_k V_{k|n} + V_{k|n}B_k' + (\hat{x}_{k|n} - \hat{x}_{k-1|n})(\hat{x}_{k|n} - \hat{x}_{k-1|n})' \end{aligned}$$

This result corresponds to the one presented in Fahrmeir [1994]. Substituting $V_{k|n}B_k' = \mathbb{C}\text{ov}(x_{k|n}, x_{k-1|n}\big|y_{1:n})$ we have the equivalence with the result of Watson and Engle [1983].

The estimate of $\Sigma$ plays a major role in the bias/variance trade-off. It can find interpretation in the univariate scenario. If the latent process has a small variance then a little portion of the prediction error is used to update the locations and therefore the latent process moves slowly and is delayed. When the variance is high the estimated latent process is heavily influenced by the last observation and has a tendency to overfit the observed process. In some practical fields, the variance is tuned manually by searching for overfitting or delayed behaviors in the errors. Our EM provides a precise solution and avoids manual tuning.

### Poisson component

For arbitrary exponential family distributed edges, as described in Section 3.4.2, the observed process component can be maximized numerically with a general optimization algorithm. However, for Poisson distribution a more elegant solution is available. Consider the conditional expected rate in the interval $t \in (t_k, t_{k+1}]$

$$\log(\mathbb{E}[\lambda_{ij}(t, x_k, \beta))|y_{1:n}] = \log(\mathbb{E}[e^{-d(x_{ki}, x_{kj})}|y_{1:n}]) + \beta_G^t B_{ij}(t_k) + \beta_D^t s(\{N(\tau)|\tau \le t_k\}), \tag{3.11}$$

with its associated expected cumulative hazard across the entire interval $\mu^*_{k,ij}(y_{1:n}, \beta) = (t_{k+1} - t_k)\mathbb{E}[\lambda_{ij}(t, x_k, \beta))|y_{1:n}]$. The expectation of the Poisson component for the discrete-time process $Y$ can then be rearranged as follows

$$\begin{aligned} Q^P(\beta) &= \sum_{kij} \mathbb{E}[-\mu_{k,ij}(x_k, \beta) + y_{k,ij} \log(\mu_{k,ij}(x_k, \beta)) - \log(y_{k,ij}!)|y_{1:n}] \\ &= \sum_{kij} -\mu^*_{k,ij}(y_{1:n}, \beta) + y_{k,ij} \log(\mu^*_{k,ij}(y_{1:n}, \beta)) - \log(y_{k,ij}!) + C \end{aligned}$$

which, up to an additive constant, is a Poisson log-likelihood parametrized by $\mu^*_{k,ij}(y_{1:n}, \beta)$. The optimization can be performed by fitting a generalized linear model [McCullagh, 2018] with the above linear predictor and the offset $\log(\mathbb{E}[e^{-d(x_{ki}, x_{kj})}|y_{1:n}])$. See Supplementary Materials 3.9.3 for the full derivation. The expected value in the offset cannot be further simplified. We use a second-order Taylor approximation, which can be expressed as a function of the first two moments of the latent locations, $\hat{x}_{k|n} = \mathbb{E}[x_k|y_{1:n}]$ and $V_{k|n} = \mathbb{V}[x_k|y_{1:n}]$. Consider $g_{ij}(x) = e^{-d(x_{ki}, x_{kj})}$, then the expectation within the off-set is approximately

$$\mathbb{E}[g_{ij}(x)|y_{1:n}] \approx g_{ij}(\hat{x}_{k|n}) + \frac{1}{2}\text{trace}\left(\frac{\partial^2 g_{ij}(x)}{\partial^2 x}\Big|_{\hat{x}_{k|n}} V_{k|n}\right),$$

since the expectation of the first derivative is zero. Simulation studies show that if the latent space changes smoothly, i.e., a low value on the diagonal of $\Sigma$, the approximation is almost perfect.

Above we have described the linear fixed effect case. In the case non-linear or random effects are required then generalized additive modeling [Wood, 2006] can be inserted in this part of the M-step. This formulation is very general and employs spline bases for estimating non-linear or time-varying effects.

### Exponential component

The expectation of the exponential component for the continuous time process $N$ is

$$Q^E(\beta) = \mathbb{E}\left[-\sum_{i \neq j} \sum_{k=1}^{n_x} \mu_{k,ij}(x_k, \beta) + \sum_{k=1}^{n_e} \log \lambda_{i_k j_k}(t_k, x_{t_k}, \beta)\right]$$

Note that, up to a multiplicative constant $y_{k,ij}$, the exponential log-likelihood factorizes similarly to that of the Poisson. Also in this case the expected log-likelihood can be rewritten as an exponential log-likelihood with the same offset

**Algorithm 3** *Expectation Maximization*

```
Initialize $\hat{x}_{0|0} = v_0$, $V_{0|0} = \Sigma$, $\Sigma = \Sigma_0$ and $\beta = \beta_0$
while not converged do
    1. Expectation:
        - Extended Kalman Filter
        - Smoother
    2. Maximization and update of starting values:
        $\beta = GLM$
        $\Sigma = \hat{\Sigma}$
        $\hat{x}_{0|0} = \hat{x}_{0|n}$
        $V_{0|0} = V_{0|n}$
    3. Check for convergence
```

as in equation (3.11). Inference involves survival regression with exponential waiting times. In case the hazard in equation (3.2) would also contain an unknown time-varying baseline hazard $\lambda_0(t)$ common to all nodes $V$, then the M-step could proceed using the partial likelihood as in Cox proportional hazard regression [Cox, 1972].

### 3.5.3 Computational aspects

The $p^2 \times p^2$ matrix inversion in (3.10) represents a computational bottleneck in many Kalman filter applications. However, there are cases where the dimension of the latent process is much smaller than the observed process dimension. The Sherman-Morrison-Woodbury identity can be employed

$$\left(R_k + H_k V_{k|k-1} H_k'\right)^{-1} = R_k^{-1} - R_k^{-1} H_k (V_{k|k-1}^{-1} + H_k' V_{k|k-1} H_k)^{-1} H_k' R_k^{-1}$$

and requires $p \times p$ matrices inversion only. As the latent space employed by our model has a cheap $p$-dimensional representation our scenario is particularly appealing for the application of the Sherman-Morrison-Woodbury identity. The identity is closely related to the Information Filter (see the Supplementary Materials 3.9.4). The overall computational cost of the algorithm is therefore dominated by the inversion of a $p \times p$ matrix [Mandel, 2006].

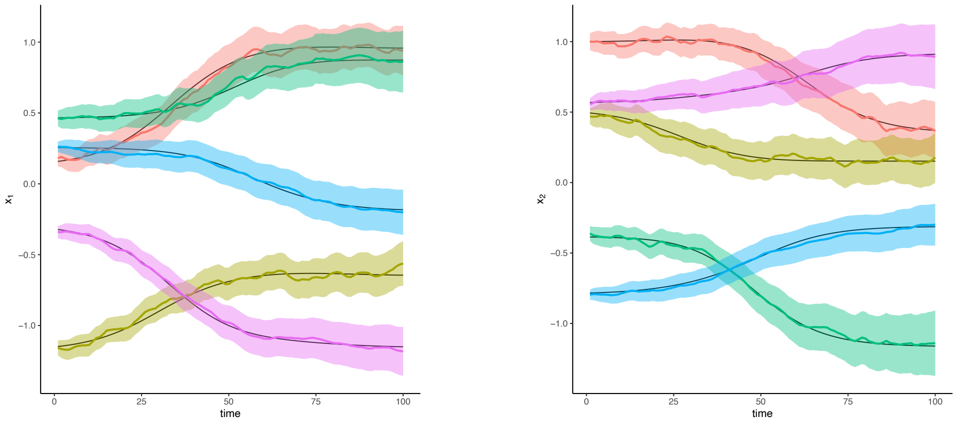


Figure 3.4. An example of the model fit of the latent space on simulated data with 10 nodes. The two plots represent the $d = 2$ latent space dimensions, $x_1$ and $x_2$, across time $k$ for 5 nodes, by plotting $\hat{x}_{k|n}$ and their variability bands $\hat{x}_{k|n} \pm 1.96\sqrt{V_{k|n}}$. Such quantities are produced by the Kalman smoother, allowing for a straightforward assessment of the model fit. The black line represents the true locations of the simulated data. Procrustes rotation is used to find the best match between the fit and the truth.

### 3.5.4 Goodness-of-fit and model selection

The conditional distribution of the latent space $x$ conditioned to the observed process $y$ can be used for assessing the uncertainty about the latent process. Variability bands can be drawn by using the quantiles of the distribution $x_{k|n} \sim N(\hat{x}_{k|n}, V_{k|n})$ and the user can visually check whether the dynamic locations are far from being a constant line, as shown in Figure 3.4.

Akaike Information Criterion. The dimension $d$ of the latent space can be selected by using some Information Criterion such as the cAIC

$$\mathrm{cAIC} = -2\log f(y|\hat{\beta}, \hat{x}) + 2\Phi$$

where $\Phi$ is the effective degrees of freedom of the fixed and random latent part of the model. Saefken et al. [2014] present a unifying approach for calculating the conditional Akaike information in generalized linear models that can be used in this context. This allows us to select the latent space dimension $d$ that minimizes the conditional Akaike criterion. The cAIC can also be used for choosing between different variance structures, e.g., a diagonal matrix $\Sigma$ with either the same or different diagonal elements, or for choosing between a static or a dynamic latent model. The static model, where all the locations are fixed in time, can be obtained by modifying our algorithm, as the static model can be viewed as a dynamic model with one single time interval, grouping together all time intervals. The filtering procedure is reduced to updating the locations with $\hat{\Sigma} = 0$.

### 3.5.5 Identifiability and divergence

The latent space formulation is identifiable with respect to the relative distances but unidentifiable in the locations [Hoff et al., 2002]: infinite combinations of rotations and translations have the same distances and therefore the same likelihood. This implies the non-identifiability of $\Sigma$, as the coordinate system rotates. Each update of the filter and smoother may involve a certain shift and rotation in the next location configuration. As a result when we update the starting points $x_{0|0}$ for the next EM iteration they may be shifted and rotated, with related rotation for $\Sigma$. These movements become stable as the starting points $x_{0|0}$ converge. It is however possible to make $\Sigma$ fully identifiable, by fixing $d+1$ constraints on the node locations. Alternatively, one can specify $\Sigma$ spherical or spherical within each node, to obtain an identifiable $\Sigma$. In principle, it is possible to extend the latent model to steps with time-varying $\Sigma_t$, but it would require additional assumptions. For example, assuming that the $d \times d$ diagonal submatrices of the $dp \times dp$ matrix $\Sigma_t$ are identical makes it identifiable. However, this is undesirable from a practical point of view as it would make each node equally variable, which is clearly not the case in many scenarios. Instead, we prefer to interpret the time-homogeneity of $\Sigma$ as a Bayesian prior on X: rather than being an assumption on the underlying generating process of X, it guarantees the “continuity” of $X$ as well as identifiability of a particular axis of rotation of the latent space. Clearly, this assumption affects the posterior distribution of $X$, but not strongly its posterior mean, which is our main quantity of interest.

A practical aspect Kalman Filter users may encounter when working on real data is divergency issues of the algorithm, defined as generating unbounded state value residuals within the procedure [Fitzgerald, 1971]. Many factors can influence the divergence tendency such as a wrong variance specification in $R_k$, poor approximation of non-linearity, inappropriate initial choice $\beta$, abrupt changes in link rates, too large variances on the diagonal of $V_{0|0}$ and $\Sigma$ or poor initial latent state values $x_0$. In case of bad starting points $x_0$ the update of locations might have abrupt changes because in a non-convex likelihood optimization locations jump to find a more stable configuration.

Fine-tuning parameters and starting points can resolve the above problems. Artificially inflating $R_k$ solves the overdispersion problems, although inferring the correct variance function of the data might take some extra effort. Sufficiently good $x_{0|0}$ points can be calculated via Multidimensional Scaling or reversing the time dimension and running the Kalman Filter backward. Furthermore, starting the EM close to the static model, by setting the diagonal values of $V_{0|0}$ and $\Sigma$ low, always leads to a stable Kalman update. In fact, the latent space variances can be seen as tuning parameters that can be expanded slowly to allow for more movement in the latent space. Where possible, one eventually expands them toward the maximum likelihood values. Otherwise, a profile maximum likelihood estimate will be the best alternative.

## 3.6 Simulation study

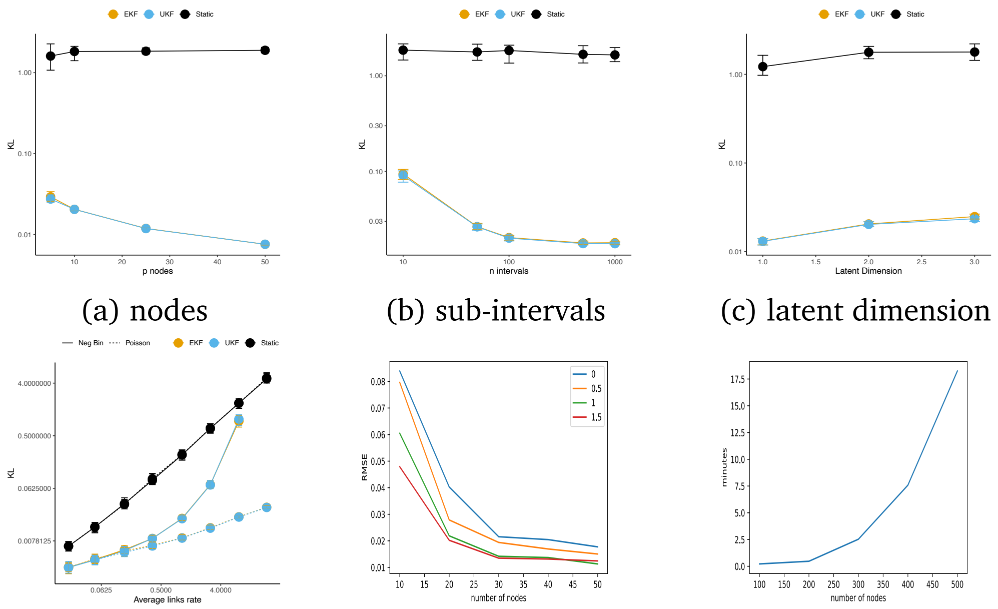
In order to assess the method performance we carry out a simulation study. We specify logistic functions for the latent location trajectories $x_k$, rescaling and shifting these functions in different ways. The link counts are generated from a Poisson distribution with $\log(\mu_{k,ij}(x_k)) = \alpha - \|x_{ki} - x_{kj}\|_2^2$ for $p$ nodes across $n$ intervals with $d$ latent dimensions. The simulation study involves varying the number of nodes, intervals, and dimensions. We also propose some challenges to the model such as the misspecification of the distribution family, high clustering, or sparsity behavior. Optimal starting points are calculated via the static model as described in Section 3.5.5. We use the out-of-fold Kullback Leibler divergence as a performance measure

$$\begin{aligned} KL(\hat{x}, x_{\text{true}}) &= \mathbb{E}_y\left[\log p(y|x_{\text{true}}) - \log p(y|\hat{x})\right] \\ &\approx \frac{\sum \log p(y_{\text{new}}|x_{\text{true}}) - \log p(y_{\text{new}}|\hat{x})}{np(p-1)/2} \end{aligned}$$

where $y_{\text{new}}$ denotes an additional sample that is generated from $x_{\text{true}}$. The Kulback-Leibler is a performance measure based on the distance matrix, which is invariant to rotations and translations of the locations.

**Varying the number of nodes p.** Figure 3.5a shows the results of varying the number of nodes $p = 5, 10, 25, 50$. The EKF performance improves as $p$ increases dramatically. This is a consequence of, on the one hand, a quadratic increase in the number of possible interactions and, on the other, a quadratic increase in the number of triangulation opportunities in the latent space.

**Varying the number of intervals n.** Figure 3.5b shows the results of varying the number of observed time sub-intervals $n = 10, 50, 100, 1000$. Again, the EKF



(a) nodes (b) sub-intervals (c) latent dimension

(d) overdispersion (e) reciprocity strength (f) computational time

Figure 3.5. Kullback-Leibler measure shows that the EKF and UKF both improve performance with (a) an additional number of nodes $p$ and (b) interval $n$, while slightly deteriorates when (c) increasing the latent dimension $d$. (d) Shows the effects of model misspecification, (e) the reliability of endogenous effects estimation in our latent space formulation. (f) Computational time grows markedly in the number of nodes $p$.

performance improves with the increase of $n$. The reason for the improvement is that when the same time interval is divided into a larger number of sub-intervals, it reduces the effective latent space variance and it increases the number of observations.

Varying the latent dimension d. Figure 3.5c shows a slight decrease in the performance when increasing the true latent dimension $d$. Clearly, when the latent dimension increases, the number of observations remains constant, but the dynamics become more complex, resulting in an increase in the KL divergence.

Effect of model misspecification: overdispersion. In Figure 3.5d we investigate the inference behavior under one type of model misspecification, namely, overdispersion. We simulate data from a negative binomial with mean $\mu_{k,ij}(x_k)$ and a quadratic variance function $\mu_{k,ij}(x_k) + \mu_{k,ij}(x_k)^2$ and compare the performance of the EKF to data simulated from a Poisson distribution with the same increasing mean $\mu_{k,ij}(x_k)$. For low rates, the negative binomial variance is almost the same as that of the Poisson, and here we observe the same EKF performances over the two distributions. For high rates, the fit on negative binomial counts deteriorates and starts to become comparable to that of the static model. For the highest rate in the simulation study, the signal-to-noise ratio in the data is so low that the inference procedure diverges in all the simulations. However, it is interesting to note that for highly sparse counts of relational events, the inference procedure always converges (for more details, see Supplementary Materials 3.9.6).

Alternative methods. In the various simulations we compare the EKF implementation with two possible competitors. The Unscented Kalman filter uses a so-called unscented transformation as an alternative to the EKF linear approximation of non-linear equations. For details, we remand the reader to the Supplementary Materials 3.9.5. The static model refers to the latent space implementation with non-dynamic states, described in Section 3.5.4. Figure 3.5 (a-d) shows that the EKF and UKF have very similar performances in terms of KL divergence, whereas the computational costs are very similar (Supplementary Materials 3.9.6). In general, it can be seen that ignoring state dynamics can be highly detrimental, as the KL divergence of the static model is typically much higher than that of the EKF. However, there is one exception: if the model is highly misspecified and the dispersion is much higher than that of a Poisson, then the static model becomes more robust and starts to become competitive.

Modeling endogenous effects. On the one hand, endogenous effects, such as reciprocity or triadic effect, are drivers of relational events that depend on the past structure of the network. Other the other hand, the latent space itself also encapsulates part of the network structure. Therefore, it is important to check whether endogenous effects are identifiable in the presence of latent dynamics. Figure 3.5e shows the mean squared error (MSE) of the estimated reciprocity for four different reciprocity strengths in a simulation study across an increasing number of nodes $p$. The results show that the MSE decreases roughly as $1/p$, which is consistent with the fact that the information grows quadratic with the number of nodes.

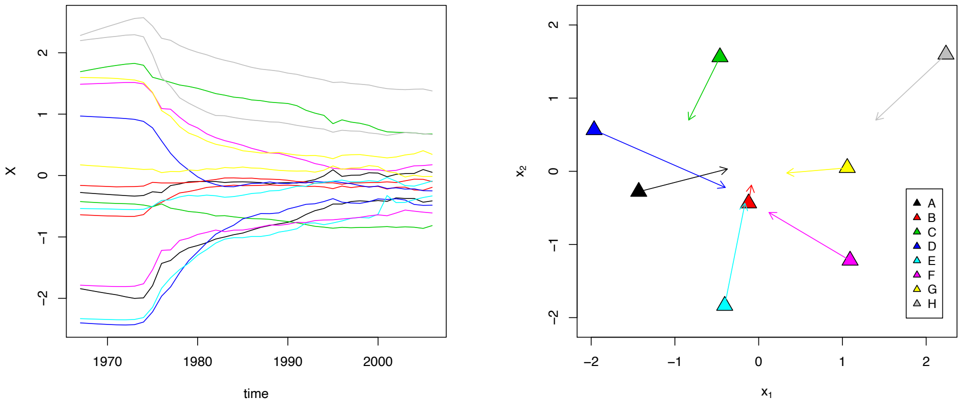
Modeling larger networks The simulations so far were performed on relatively small networks with $p \leq 50$, a dimension that is achievable for a custom implementation in the R language. For larger networks, we created an implementation in TensorFlow and performed the simulations on *Google Colab* using its free GPU resources. Figure 3.5f shows the computational time for larger networks. The 100 nodes model converges in roughly 22 seconds, whereas for networks with 500 nodes, roughly 20 minutes are needed. Computational time seems to increase roughly quadratically in the number of nodes. Another common computational bottleneck in large networks is that the number of observations carried by the adjacency matrix and the related machine operations grows quadratically with the number of nodes. In that case, stratified subsampling [Raftery et al., 2012] on the adjacency matrix elements could reduce the computational burden. Using this idea, a pilot Kalman filter can be run to calculate the stratum contribution via the increment in the expected log-likelihood. Other ideas, such as parallel Kalman Filters [Särkkä and García-Fernández, 2020] where multiple time points can be computed in parallel, can only be implemented if the memory consumption of each individual Kalman filter iteration is small, which is not our case.

## 3.7 Dynamics of patent citation patterns

The patent citation process introduced in Section 3.3 presents some peculiar characteristics with respect to the underlying relational event: patents are added in tranches to the system, and citations happen only at the moment of patent creation. Furthermore, patents can cite only those patents that have previously been created and not the ones that are added to the network in the future. Therefore, rather than focusing on the individual patents, we focus on the citations between groups of patents, such as the patent classes and subclasses, described above. Our aim is to describe the relative changing importance of each of these (sub)classes over time in being cited as prior art in novel patents. We consider the latent space model for the number of citations $y_{k,ij}$ from patents of field $i$ to patents of field $j$ at time $k$

$$\begin{aligned} y_{k,ij} &\sim \text{Poi}(\mu_{k,ij}(x_k,\beta)) \\ \log(\mu_{k,ij}(x_k,\beta)) &= \log C_i(k) + \alpha_0 - \|x_{ki} - x_{kj}\|_2^2 + \text{sender}_i + \text{receiver}_j \end{aligned} \tag{3.12}$$

where $\alpha_0$ is an intercept and $\text{sender}_i$ and $\text{receiver}_j$ are respectively the sender and receiver random effects. We include random effects in the linear predictor as the usual conditional formulation of the regression model. The citation rate is proportional to the number of patents $C_i(k)$ added in a field within a year. If in a certain year, there are no patents added in a field, the rate would clearly be



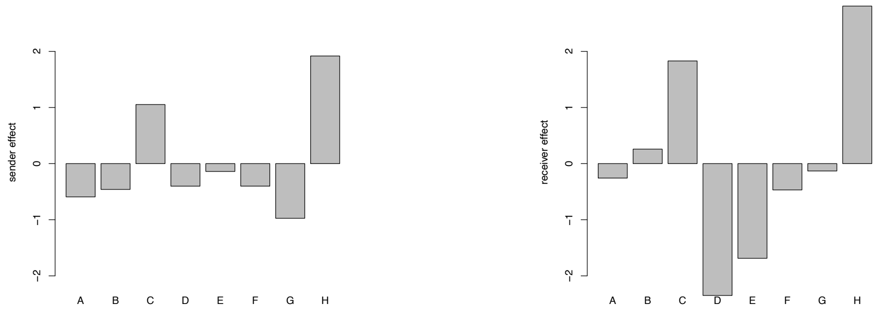
(a) (b)

Figure 3.6. Changes in latent space patent locations. (a) The two coordinates for each of the 8 main classes are shown in the same figure. The first ten years show a static behavior in citations. After that point the fields start moving closer as the citations between fields intensify; (b) The overall change in latent space locations of the 8 main classes over the entire period of 1967-2006.

zero. We, therefore, specify an additional offset $\log C_i(k)$ that accounts for the number of patents added in field $i$ at time $k$. The inclusion of the offset has the advantage that the interpretation of the latent space locations and other effects is with respect to a single patent in each (sub)class. As the aim is to explore the major relative movement of each of the (sub)classes, we consider here a bidimensional latent space. Optimal starting points are calculated via the static model as described in Section 3.5.5.

Figure 3.6 shows a peculiar behavior of the latent locations of the 8 main technology classes. They seem to be more or less static in the initial 10 years from 1967 until 1976. Patents can only cite back in time and therefore the first patents added to the system cannot cite patents submitted before the year 1967. The apparent stationarity may therefore be an artifact. The figure suggests that around 1976 the patent citation process start behaving more “normally”, i.e., it starts to represent more representatively the bulk of the citation process. This seems reasonable as patents cite an average of 10 years back in time, with a mode that is significantly less than 10 years.

In general, we observe that the exchange of citations between different fields increases over time, ending with a large cluster including the majority of the ICL



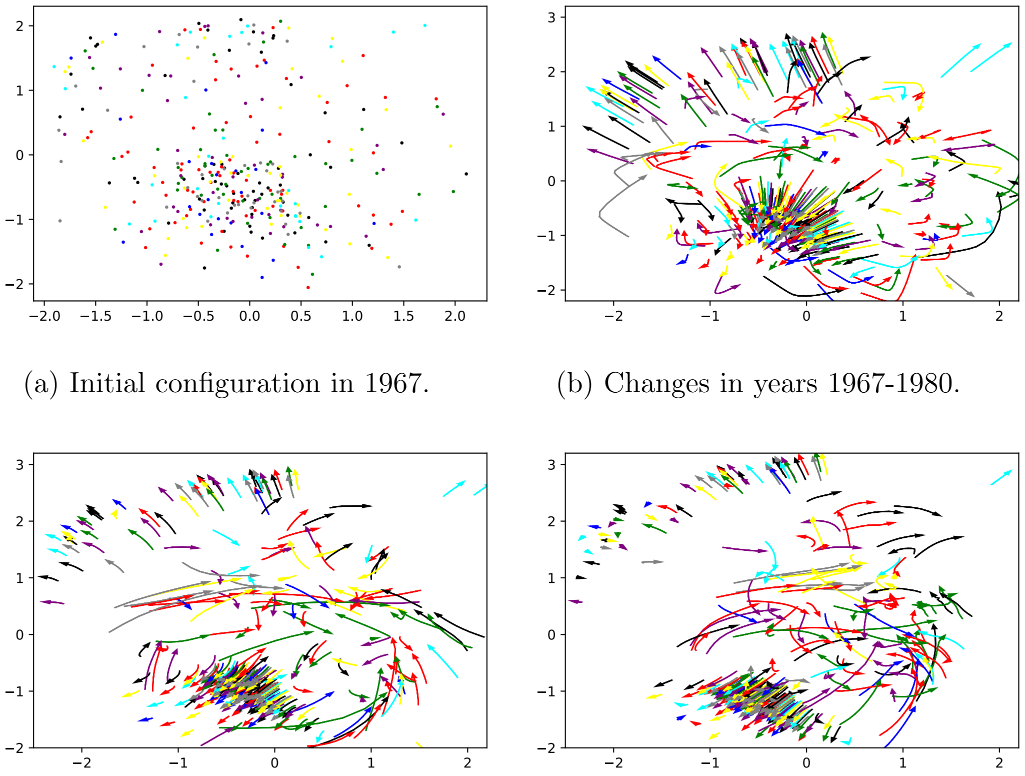
(a) Sender effect

(b) Receiver effect

Figure 3.7. Model inference on dynamic locations for the relational event model with sender and receiver effects. (a) shows a summary of the movement of the patent classes in the observed time interval.

categories. Only classes C (Chemistry and Metallurgy) and H (Electricity) remain somewhat separate from the other main classes. The overall conclusion is that except for classes C and H, the other main technology classes lose their specific characteristics and patents tend to cite more across technology class borders. This suggests that most technology classes are becoming less dissimilar: there is an increasing heterogeneity *within* the fields, as they communicate with other technology fields, and thus a higher homogeneity *between* the fields.

The sender and receiver effects can be interpreted as the asymmetry between fields citations that the symmetric latent space representation fails to capture. Figure 3.7(b) show how the Textile, Papers, and Fixed constructions classes are very low receiver classes, meaning that they are cited below average. Figure 3.7(s) shows that Physics patents have a low tendency to cite others. The high sending and receiving tendencies of the Chemistry, Metallurgy and Electricity patents must be seen in the context of Figures 3.6: the fact that we observe such huge effects jointly together with their distant location to the other patent classes might suggest some violation of the model assumptions. The two locations should be closer to the main cluster but there does not exist a 2D latent configuration that makes a good fit. An analysis without sender and receiver effects (Supplementary Materials 3.9.7) indeed shows that those two classes would be apparently closer, joining the other technology classes.



(a) Initial configuration in 1967.

(b) Changes in years 1967-1980.

(c) Changes in years 1980-1993

(d) Changes in years 1993-2006

Figure 3.8. Dynamic latent locations for the 487 technology subclasses. The colors correspond to the original 8 main technology classes.

### 3.7.1 Extending the analysis to subclass dynamics

The 8 main technology classes give a rough overview of the patent dynamics. However, given that we analyze more than 23 million citations, a finer analysis should reveal more detailed results. We, therefore, extend the analysis to the subclass level of the patent classification system. The 8 main technology classes consist of a total of 487 more specific subclasses.

Figure 3.8 shows the latent dynamics for all the 487 subclasses, where the color refers to the original 8 main technology classes. What is immediately clear is that the dynamics within a single technology class are quite diverse. Figure 3.8 (a) shows that the subclasses are evenly spread in the latent plane. Moreover, by inspecting single subclass trajectories, it emerges that a subclass tends to move with few subclasses from within the same main technology class, but also with some subclasses from another class. This is consistent with the raw data, as approximately half of a patent's connectivity is within the same class, while the rest is towards other classes.

Figure 3.8 also shows that subclasses are heterogeneous in their citation behavior from the beginning and that not all the subclasses converge to a single cluster. Technology subclasses end up forming three heterogeneous clusters, as evidenced by 3.8 (d). As time passes the bottom left nodes separate from the center and converge into a dense cluster, revealing an increasing heterogeneity in their citation. On the top left something similar happens although this cluster is less dense as its nodes do not seem to shorten their distances. Nodes belonging to clusters with such flatten shapes typically present high connectivity with the immediate neighbor but this connectivity does not extend to distant nodes, creating a chain where the two poles share little similarity. At the center, by looking at the inward arrows, it is possible to spot a third, low-density cluster that is separating from the other two. We conclude that the increment of heterogeneity in patent citations is not uniform across all subclasses. There is some coordinated movement from the three clusters of subclasses. Patents within these clusters tend to get more similar citing behavior, whereas patents between these clusters tend to cite each other less. It is interesting to note that the apparent converging behavior of the main technology classes in Figure 3.6 is simply the result of aggregating the subclasses where the diverging movements are averaged out.

## 3.8 Conclusion

In the last decade, REMs have been used for describing the drivers of dynamic network interactions. Traditional approaches focus on endogenous and exogenous drivers, which may not always be able to capture all heterogeneity in the data. Our aim has been to extend relational event modeling by letting their interactions depend on dynamic locations in a latent space.

The model defines the latent locations as missing states, where the observations are the time-stamped relational events or aggregates of those events within a certain interval. We use an EM algorithm, whereby a Kalman filter calculates their conditional expectation, and a generalized linear model formulation performs the maximization step. Kalman Filters are effective methods for estimating latent dynamic processes. Their simplicity and computational efficiency make them suitable for many problems common in engineering contexts. The filter relies on a sequence of linear operations and easily calculates the Expectation step, typically untractable for non-trivial cases. The Kalman filter dual interpretation in the Bayesian and frequentist literature would also make an effective Gibbs possible. Current Bayesian approaches, such as Sewell and Chen [2015], rely on a simplified stratified case-control sampling of non-events. As there are many more non-events with distant nodes, mid-distances are either never sampled or sampled and overweighted by an inappropriate case-control weight. Although this reduces computational complexity, this produces a bias in the inference procedure.

It is easy to extend the linearity of the exogenous and endogenous effects in the model formulation (3.2) to smooth effects. The generalized linear model approach for the M-step can easily be replaced by a generalized additive set-up for incorporating smooth and time-varying effects as well as random effects [Wood, 2006]. The simulation results show that the modeling and inference set-up is accurate, computationally feasible, and insightful under different scenarios.

We applied the model to 23 million patent citations from the US patent office in order to investigate the innovation dynamics in the period 1967-2006. Focusing on the 8 main technology classes suggests that there is an overall convergence in the latent space, meaning that the patents classes are becoming either more similar or more internally dissimilar. A subsequent analysis of the 487 subclasses revealed that the second hypothesis explains most of the apparent convergence: it seems that the subclasses within each main technology class have coordinated, but diverging dynamics, which suggest that the main technology classes have become more dissimilar over time. This may be because the original class denominations refer to distinctions that have become less relevant over time. For this reason, it would probably be good to avoid using the main technology classes as important descriptors of patents and instead focus on their subclass denominations.

## 3.9 Supplementary material

### 3.9.1 EKF implementation details

In (3.7) $x_k$ and $y_k$ are vectors of length $p_x = pd$ and $p_y = p(p-1)$ or $p(p-1)/2$ in case of an undirected network respectively. These are the $p \times d$ location matrix and $p \times p$ adjacency matrix that have been vectorized. At time $k$ we have

$$\mu(x_k,\beta) = \begin{bmatrix} \mu_{12}(x_k,\beta) \\ \vdots \\ \mu_{p-1,p}(x_k,\beta) \end{bmatrix}, \qquad x_k = \begin{bmatrix} x_{k1} \\ \vdots \\ x_{kp} \end{bmatrix}, \qquad x_{ki} = \begin{bmatrix} x_{ki1} \\ \vdots \\ x_{kid} \end{bmatrix},$$

where $x_{ki}$ is the $d$-dimensional location of node $i$. The choice of using the Euclidean distance is arbitrary and other distance measures can be selected. The dimension of the latent space is commonly chosen as $d = 2$ or 3 for the sake of visual inspection, but more formal criteria can be used to select a proper dimension.

The matrix $H_k$ of the first derivatives is structured as follows

$$H_k = \frac{\partial}{\partial x}\mu(x,\beta)\mid_{\hat{x}_{k|k-1}} = \begin{bmatrix} \frac{\partial}{\partial x}\mu_{k,12}(x,\beta)\mid_{\hat{x}_{k|k-1}} \\ \vdots \\ \frac{\partial}{\partial x}\mu_{k,ij}(x,\beta)\mid_{\hat{x}_{k|k-1}} \\ \vdots \\ \frac{\partial}{\partial x}\mu_{k,p-1,p}(x,\beta)\mid_{\hat{x}_{k|k-1}} \end{bmatrix}$$

$H_k$ is a $p_y \times p_x$ block matrix, where the row indexed by the interaction $(i,j)$ is composed of $d$-dimensional vectors $\frac{\partial}{\partial x_k}\mu_{k,ij}(x,\beta)$ for $k = 1,\dots,p$ as follows

$$\frac{\partial}{\partial x}\mu_{k,ij}(x,\beta) = \begin{cases} \frac{\partial}{\partial x_i}\mu_{k,ij}(x,\beta) = 2(x_j - x_i)e^{-\|x_i - x_j\|_2^2 + f_{ij}^F(\beta,k) + f_{ij}^R(\beta,k)}, \\ \frac{\partial}{\partial x_j}\mu_{k,ij}(x,\beta) = -2(x_j - x_i)e^{-\|x_i - x_j\|_2^2 + f_{ij}^F(\beta,k) + f_{ij}^R(\beta,k)}, \\ \frac{\partial}{\partial x_k}\mu_{k,ij}(x,\beta) = 0, \end{cases}$$

The posterior variance is calculated keeping the Taylor local approximation $\mu(x_k,\beta) \approx H_k x_k$

$$\begin{aligned} V_{k|k} &= \mathbb{E}[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})'] \\ &= \mathbb{E}[(x_k - \hat{x}_{k|k-1} - K_k(y_k - H_k\hat{x}_{k|k-1}))(x_k - \hat{x}_{k|k-1} - K_k(y_k - H_k\hat{x}_{k|k-1}))'] \\ &= \mathbb{E}[(x_k - \hat{x}_{k|k-1} - K_k(H_k x_k + \epsilon_k - H_k\hat{x}_{k|k-1}))(x_k - \hat{x}_{k|k-1} - K_k(H_k x_k + \epsilon_k - H_k\hat{x}_{k|k-1}))'] \\ &= \mathbb{E}[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})'] + \mathbb{E}[K_k(H_k x_k - H_k\hat{x}_{k|k-1})(H_k x_k - H_k\hat{x}_{k|k-1})'K_k'] + \mathbb{E}[K_k\epsilon_k\epsilon_k'K_k'] \\ &\quad - \mathbb{E}[K_k H_k(x_k - \hat{x}_{k|k-1}))(x_k - \hat{x}_{k|k-1}))'] - \mathbb{E}[(x_k - \hat{x}_{k|k-1}))(x_k - \hat{x}_{k|k-1}))'H_k'K_k'] \\ &= V_{k|k-1} + K_k H_k V_{k|k-1}H_k'K_k' + K_k R_k K_k' - K_k H_k V_{k|k-1} - V_{k|k-1}H_k'K_k' \end{aligned}$$

where

$$K_k H_k V_{k|k-1}H_k'K_k' + K_k R_k K_k' = K_k(H_k V_{k|k-1}H_k' + R_k)K_k' = V_{k|k-1}H_k'K_k'$$

thus

$$V_{k|k} = V_{k|k-1} - K_k H_k V_{k|k-1} = (\mathbb{I} - K_k H_k)V_{k|k-1}.$$

### 3.9.2 Smoother

$$\begin{aligned}
&\mathbb{E}[\mathbb{V}[x_{k-1}|x_k,y]|y]=\mathbb{E}[\mathbb{V}[x_{k-1}|x_k,y_{1:k-1}]|y]\\
&\quad=\mathbb{E}\big[\mathbb{V}[x_{k-1}|y_{1:k-1}]-\mathbb{C}ov(x_{k-1},x_k|y_{1:k-1})\mathbb{V}(x_k|y_{1:k-1})^{-1}\mathbb{C}ov(x_{k-1},x_k|y_{1:k-1})'|y\big]\\
&\quad=\mathbb{E}\big[V_{k-1|k-1}-B_kV_{k|k-1}B_k'|y\big]=V_{k-1|k-1}-B_kV_{k|k-1}B_k'\\
&\mathbb{V}[\mathbb{E}[x_{k-1}|x_k,y]|y]=\mathbb{V}\big[\hat{x}_{k-1|k-1}+B_k(x_k-\hat{x}_{k|k-1})|y\big]=B_kV_{k|n}B_k'
\end{aligned}$$

### 3.9.3 Maximization

Poisson component

$$\begin{aligned}
Q(\beta,\Sigma)=\sum_{tij}&\mathbb{E}[-\mu_{k,ij}(x_k,\beta)]+\mathbb{E}[y_{ij}(k)\log(\mu_{k,ij}(x_k,\beta))]-\log(y_{ij}(k)!)+C_2=\\
&\sum_{tij}-\mathbb{E}[e^{-d(x_{ki},x_{kj})}]e^{f_{ij}^F(\beta,k)+f_{ij}^R(\beta,k)}+\\
&+y_{ij}(k)(\mathbb{E}[-d(x_{ki},x_{kj})]+f_{ij}^F(\beta,k)+f_{ij}^R(\beta,k))-\log(y_{ij}(k)!)+C_2
\end{aligned}\tag{3.13}$$

Notice that adding and subtracting $y_{ij}(k)\log(\mathbb{E}[e^{-d(x_{ki},x_{kj})}])$

$$\begin{aligned}
&y_{ij}(k)(\mathbb{E}[-d(x_{ki},x_{kj})]+f_{ij}^F(\beta,k)+f_{ij}^R(\beta,k))\\
&=y_{ij}(k)(\log(\mathbb{E}[e^{-d(x_{ki},x_{kj})}])+f_{ij}^F(\beta,k)+f_{ij}^R(\beta,k))+\\
&+y_{ij}(k)\mathbb{E}[-d(x_{ki},x_{kj})]-y_{ij}(k)\log(\mathbb{E}[e^{-d(x_{ki},x_{kj})}])
\end{aligned}\tag{3.14}$$

thus

$$\begin{aligned}
Q(\beta,\Sigma)&=\sum_{tij}-\mathbb{E}[e^{-d(x_{ki},x_{kj})}]e^{f_{ij}^F(\beta,k)+f_{ij}^R(\beta,k)}+\\
&+y_{ij}(k)(\log(\mathbb{E}[e^{-d(x_{ki},x_{kj})}])+f_{ij}^F(\beta,k)+f_{ij}^R(\beta,k))-\log(y_{ij}(k)!)+C_3\\
&=\sum_{tij}-\mu_{k,ij}^*(x_k,\beta)+y_{ij}(k)(\log(\mu_{k,ij}^*(x_k,\beta))-\log(y_{ij}(k)!)+C_3
\end{aligned}\tag{3.15}$$

Gaussian component

$$\begin{aligned}\hat{\Sigma} &= \mathbb{E}\left[\frac{1}{n}\sum_{1}^{n}(x_k - x_{k-1})(x_k - x_{k-1})'\middle|y_{1:n}\right] = \frac{1}{n}\sum_{1}^{n}\mathbb{E}\left[(x_k - x_{k-1})(x_k - x_{k-1})'\middle|y_{1:n}\right]\\ &= \frac{1}{n}\sum_{1}^{n}\mathbb{E}\left[x_k x_k'\middle|y_{1:n}\right] + \mathbb{E}\left[x_{k-1}x_{k-1}'\middle|y_{1:n}\right] - \mathbb{E}\left[x_{k-1}x_k'\middle|y_{1:n}\right] - \mathbb{E}\left[x_k x_{k-1}'\middle|y_{1:n}\right]\\ &= \frac{1}{n}\sum_{1}^{n}V_{k|n} + V_{k-1|n} + B_k V_{k|n} + V_{k|n}B_k' + (\hat{x}_{k|n} - \hat{x}_{k-1|n})(\hat{x}_{k|n} - \hat{x}_{k-1|n})'\end{aligned} \tag{3.16}$$

$$\begin{aligned}\mathbb{E}\left[x_k x_k'\middle|y_{1:n}\right] &= \mathbb{E}\left[((x_k - \hat{x}_{k|n}) + \hat{x}_{k|n})((x_k - \hat{x}_{k|n}) + \hat{x}_{k|n})'\middle|y_{1:n}\right]\\ &= \mathbb{E}\left[(x_k - \hat{x}_{k|n})(x_k - \hat{x}_{k|n})'\middle|y_{1:n}\right] + \hat{x}_{k|n}\hat{x}_{k|n}' = V_{k|n} + \hat{x}_{k|n}\hat{x}_{k|n}'\\ \mathbb{E}\left[x_k x_{k-1}'\middle|y_{1:n}\right] &= \mathbb{E}\left[x_k\mathbb{E}\left[x_{k-1}'\middle|x_k, y_{1:k-1}\right]\middle|y_{1:n}\right] = \mathbb{E}\left[x_k(\hat{x}_{k|k} + B_k(x_k - \hat{x}_{k|k-1}))'\middle|y_{1:n}\right]\\ &= \mathbb{E}\left[((x_k - \hat{x}_{k|n}) + \hat{x}_{k|n})(\hat{x}_{k-1,k-1} + B_k((x_k - \hat{x}_{k|n}) + \hat{x}_{k|n} - \hat{x}_{k|k-1}))'\middle|y_{1:n}\right]\\ &= \mathbb{E}\left[(x_k - \hat{x}_{k|n})(x_k - \hat{x}_{k|n})'\middle|y_{1:n}\right]B_k' + \hat{x}_{k|n}(\hat{x}_{k-1,k-1} + B_k(\hat{x}_{k|n} - \hat{x}_{k|k-1}))'\\ &= V_{k|n}B_k' + \hat{x}_{k|n}\hat{x}_{k-1|n}'\end{aligned}$$

### 3.9.4 Alternative derivation of EKF

The Poisson distribution can be written in the natural exponential family formulation [McCullagh, 2018]:

$$\begin{aligned}&p(x_k|x_{k-1}) = \frac{1}{\sqrt{2\pi}}|\Sigma|^{-1}e^{-\frac{1}{2}(x_k - x_{k-1})'\Sigma^{-1}(x_k - x_{k-1})}\\ &p(y_k|x_k) = c(y_k)e^{\theta' y_k - b(\theta)}\\ &b(\theta) = 1'e^{\theta}\\ &\theta(x_k) = \log\mu(x_k, \beta)\\ &\mu(x_k, \beta) = \mathbb{E}[y_k|x_k] = \frac{\partial}{\partial\theta}b(\theta)\\ &R_k = \mathbb{V}[y_k|x_k] = \frac{\partial^2}{\partial\theta^2}b(\theta).\\ &b(\theta) : \mathbb{R}^{p_y} \to \mathbb{R}\\ &\theta(\mu) = \left[\frac{\partial}{\partial\theta}b(\theta)\right]^{-1} : \mathbb{R}^{p_y} \to \mathbb{R}^{p_y}.\end{aligned}$$

The advantage of writing the Poisson distribution in the natural exponential family form is that further developments will be valid for any distribution of the natural exponential family. Other exponential family distributions are possible specifying differently the functions $\theta(\cdot)$ and $b(\cdot)$. The likelihood can be then written as

$$L(\beta,\Sigma;y,x)=\prod_{k=1}^{n}\frac{1}{\sqrt{2\pi}}|\Sigma|^{-1}e^{-\frac{1}{2}(x_k-x_{k-1})'\Sigma^{-1}(x_k-x_{k-1})}c(y_k)e^{\theta' y_k-b(\theta)} \tag{3.17}$$

We obtain the correction step via maximum likelihood. The likelihood that we are treating here is different than the one presented in (3.17). We are taking the single likelihood contribution at time $k$ conditioned to the inference at the previous time point. Thus the marginal distribution of the latent process is substituted with its conditional distribution, i.e., the distribution that we calculated in the prediction step. The likelihood is presented as

$$l_k(x_k)=-\frac{1}{2}(x_k-\hat{x}_{k|k-1})'V_{k|k-1}^{-1}(x_k-\hat{x}_{k|k-1})+\theta' y_k-b(\theta) \tag{3.18}$$

were $V_{k|k-1}$ represent the variance of the latent process conditioned to $y_{k-1}$. From a frequentist point of view (3.18) is a penalized likelihood, composed by the Poisson probability of the observations and a penalty term for the latent process. In a Bayesian setting, it can be considered a posterior distribution, where the penalty represents the prior distribution. The penalty/prior regulates the smoothness of the process via the covariance matrix $\Sigma$. The maximization of the posterior density is equivalent to the maximization of the penalized likelihood [Fahrmeir, 1992]. We maximize this likelihood according to $x_k$, to obtain $\hat{x}_{k|k}$. This clearly is not equivalent to the conditional mean, except in case the posterior mode coincides with the posterior mean. This is true for the Gaussian density, which is not our case. The posterior is therefore approximated with the same family distribution of the prior, i.e., Gaussian, see Gamerman [1991] and Fahrmeir [1992]. Thus we are approximating the posterior mean with the posterior mode.

Using the chain rule, we take the derivative of the likelihood with respect to $x_k$ and transpose it we have

$$\frac{\partial}{\partial x_k}l_k(x_k)=-V_{k|k-1}^{-1}(x_k-\hat{x}_{k|k-1})+\frac{\partial\mu(x_k,\beta)}{\partial x_k}'\frac{\partial\theta(\mu)}{\partial\mu}(y_k-\frac{\partial}{\partial\theta}b(\theta)).$$

A first order Taylor expansion is applied on the mean of $y_k$

$$\frac{\partial}{\partial\theta}b(\theta)=\mu(x_k,\beta)=\mu(\hat{x}_{k|k-1})+\frac{\partial\mu(x_k,\beta)}{\partial x_k}(x_k-\hat{x}_{k|k-1}) \tag{3.19}$$

obtaining

$$\frac{\partial}{\partial x_k}l_k(x_k)=-V_{k|k-1}^{-1}(x_k-\hat{x}_{k|k-1})+\frac{\partial\mu(x_k,\beta)}{\partial x_k}'\frac{\partial\theta(\mu)}{\partial\mu}\left(y_k-\mu(\hat{x}_{k|k-1})-\frac{\partial\mu(x_k,\beta)}{\partial x_k}(x_k-\hat{x}_{k|k-1})\right)$$

Setting $\frac{\partial}{\partial x_k} l_k(x_k) = 0$ and rearranging the members of the equation we have

$$x_k = \hat{x}_{k|k-1} + \left[ V_{k|k-1}^{-1} + \frac{\partial \mu(x_k, \beta)'}{\partial x_k} \frac{\partial \theta(\mu)}{\partial \mu} \frac{\partial \mu(x_k, \beta)}{\partial x_k} \right]^{-1} \left[ \frac{\partial \mu(x_k, \beta)'}{\partial x_k} \frac{\partial \theta(\mu)}{\partial \mu} \right] \big( y_k - \mu(\hat{x}_{k|k-1}) \big).$$

We evaluate the derivatives at $\hat{x}_{k|k-1}$ and use the property that the second derivative of $b(\theta)$ is equal to the variance of $y_k|x_k$. Since $x_k$ is unknown, we approximate it with $\hat{x}_{k|k-1}$.

$$\frac{\partial \theta(\mu)}{\partial \mu}\Big|_{\hat{x}_{k|k-1}} = \left( \frac{\partial^2 b(\theta)}{\partial \theta^2} \right)^{-1} \Big|_{\hat{x}_{k|k-1}} = \mathbb{V}(y_k|x_k)^{-1}\Big|_{\hat{x}_{k|k-1}} = R_k^{-1}. \tag{3.20}$$

Setting

$$\frac{\partial \mu(x_k, \beta)}{\partial x_k}\Big|_{\hat{x}_{k|k-1}} = H_k$$

and considering that

$$\mu(\hat{x}_{k|k-1}) = H_k \hat{x}_{k|k-1}$$

we obtain the update

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + [V_{k|k-1}^{-1} + H_k' R_k^{-1} H_k]^{-1} [H_k' R_k^{-1}](y_k - H_k \hat{x}_{k|k-1}) \\ &= \hat{x}_{k|k-1} + K_k (y_k - H_k \hat{x}_{k|k-1}). \end{aligned}$$

The last equation comes under the name of Information Filter. $V_{k|k-1}^{-1}$ is the information matrix on $x_k$ given $y_{1:k-1}$, $H_k' R_k^{-1} H_k$ is the information on $x_k$ contributed by the last observation $y_k$ and the sum of the two is the information on $x_k$ given $y_{1:k}$. Considering that the numerator $[H_k' R_k^{-1}](y_k - H_k \hat{x}_{k|k-1})$ is the first derivative, the correction step has the form of a single Fisher scoring step [Fahrmeir, 1992]. The formula of the filter can be rearranged in the following way

$$\begin{aligned} K_k &= (V_{k|k-1}^{-1} + H_k' R^{-1} H_k)^{-1} H_k' R_k^{-1} \\ &= (V_{k|k-1}^{-1} + H_k' R^{-1} H_k)^{-1} H_k' R_k^{-1} (R_k + H_k V_{k|k-1} H_k')(R_k + H_k V_{k|k-1} H_k')^{-1} \\ &= (V_{k|k-1}^{-1} + H_k' R_k^{-1} H_k)^{-1} (V_{k|k-1}^{-1} +' R_k^{-1}) V_{k|k-1} H_k' (R_k + H_k V_{k|k-1} H_k')^{-1} \\ &= V_{k|k-1} H_k' (R_k + H_k V_{k|k-1} H_k')^{-1} \end{aligned}$$

obtaining the filtering matrix for the EKF.

### 3.9.5 Higher order approximation: Unscented Kalman filter

In this section, we want to present a possible competitor to the EKF. The EKF is based on a first-order Taylor expansion in (3.9). We can approximate the $\mu$ function with an order higher. A popular solution is the Unscented Transformation, the key solution of the Unscented Kalman Filter (UKF) [Julier and Uhlmann, 1996, 1997]. The algorithm has a similar shape as the EKF with the difference that the filtering matrix $K_k$ is calculated empirically. We begin with a fixed number of points to approximate a Gaussian by creating a discrete distribution having the same first and second (and possibly higher) moments. Each point in the discrete approximation can be directly transformed. The mean and the covariance of the transformed ensemble can then be computed as the estimate of the nonlinear transformation of the original distribution.

Given a $pd$-dimensional Gaussian having covariance $V_{k|k-1}$ we can construct a set of points having the same sample covariance from the columns (or rows) of the matrices $\sqrt{(\kappa + pd)V_{k|k-1}}$. The square root of the matrix is typically done via a Cholesky decomposition. Adding and subtracting these points to $\hat{x}_{k|k-1}$ yields a symmetric set of $2pd + 1$ points (central point included) having the desired sample mean and covariance. This is the minimal number of points capable of encoding this information [Julier and Uhlmann, 1996]. We then calculate the sample mean and covariance of the transformed points. Finally, the filtering matrix $K_k$ can be calculated as the rate between the sample covariance and the sample variance.

$$K_k = \widehat{\mathbb{C}ov}(x_k, y_k|y_{1:k-1})\widehat{\mathbb{V}}(y_k|y_{k-1})^{-1}.$$

The Unscented Kalman Filter is presented in Algorithm 4. The prediction and the update step are the same as those of the EKF. The $\kappa$ parameter regulates both the weight of the central point and the spreading of the other points: a large $\kappa$ leads to a wider spreading of the points. Julier and Uhlmann [1997] suggests a useful heuristic to select $pd + \kappa = 3$. The use of the Unscented Kalman filter makes the computation of (3.5.2) straightforward by simply taking the sample mean of the transformed ensemble.

In section 3.6 we show that the EKF performs approximately equivalent to this competitor. This further extension to the Kalman filter is therefore useful for showing that the linear approximation in the EKF is sufficiently good for our purpose.

**Algorithm 4** *Unscented Kalman Filter*

```
  Initialize $\kappa = \kappa_0$
  $w_0 = \kappa/(p_x + \kappa)$
  $w_j = 1/2(p_x + \kappa), \quad j = 1, \ldots, 2p_x$
  for k = 1, ..., n do
    1. Filter prediction step
    2. Filtering matrix calculation
          $A = V_{k,k-1}^{\frac{1}{2}}$
          $s_0 = \hat{x}_{k,k-1}$
          $s_j = \hat{x}_{k,k-1} + \sqrt{pd + \kappa A}_j, \quad j = 1, \ldots, p_x$
          $s_{j+p_x} = \hat{x}_{k,k-1} - \sqrt{pd + \kappa A}_j, \quad j = 1, \ldots, p_x$
          $\hat{\mu}_k = \sum_{j=0}^{2p_x} w_j \mu(s_j, \beta)$
          $R_k = \hat{\mu}_k \ I_{p_y}$
          $S_k = \sum_{j=0}^{2p_x} w_j (\mu(s_j, \beta) - \hat{\mu}_k)(\mu(s_j, \beta) - \hat{\mu}_k)' + R_k$
          $C_k = \sum_{j=0}^{2p_x} w_j (s_j - \hat{x}_{k|k-1})(\mu(s_j, \beta) - \hat{\mu}_k)'$
          $K_k = C_k S_k^{-1}$
    3. Filter update step
```

### 3.9.6 Simulation study

Figure 3.9 shows a set of estimated locations overlaid with the black, underlying true locations. The observed $Y$ process was simulated 200 times from these true trajectories with $p = 10$ nodes, $n = 100$ intervals, and $d = 2$ dimensions. The colored lines are the 200 trajectories estimated by the EM-EKF.

Here the complete results of the simulation study are given both in terms of KL divergence and computational time for varying the number of nodes (Figure 3.10), the number of time intervals (Figure 3.11), the latent dimension (Figure 3.12), the overdispersion level (Figure 3.13).

### 3.9.7 Patent analysis

For comparison, we also fit the model without a random sender and receiver effects: Figure 3.14(b) shows that the distances of the Chemistry, Metallurgy and Electricity patent classes were inflated and that the random sender and receiver effects were indeed capturing the misrepresentation. The Physics class

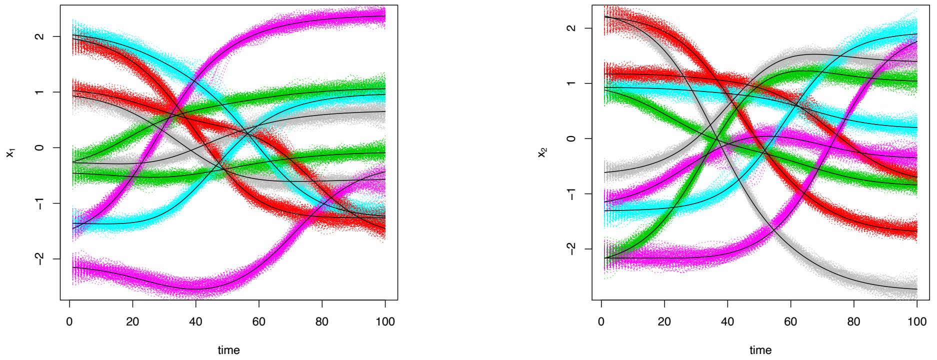
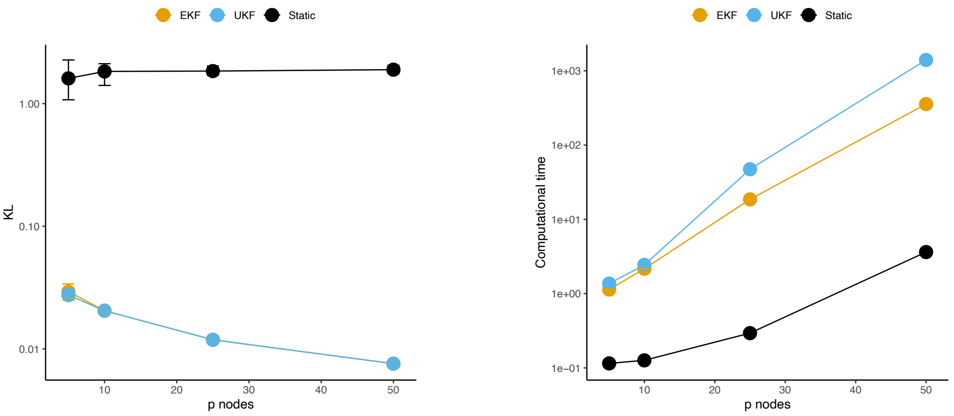


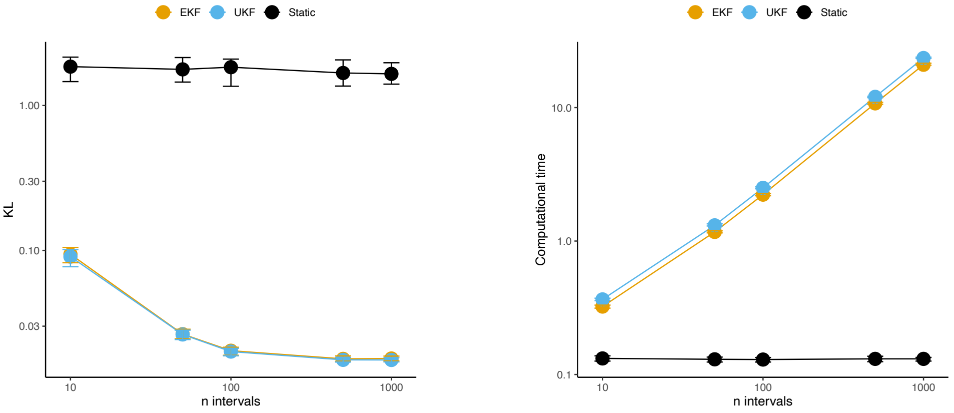
Figure 3.9. Model fit on 200 simulated datasets. The figure shows that the estimated latent locations are centered at their true values with relatively high precision. Black lines represent the true locations in time. Colored lines represent node trajectories estimated by the model for each simulation. For the sake of visual inspection, a Procusteus transformation has been used for rotating and shifting the estimates over the true line.
We consider $p = 10, n = 100, d = 2$ and $x_1, x_2$ are respectively the first and the second dimension.

comes now very close to Electricity, whereas the Chemistry and Metallurgy class overlaps with Human necessities. By looking back at the discrepancy between sender and receiver effects in Figure 3.7 we see that Chemistry and Metallurgy patents have the tendency to receive more from Human necessities, whereas the Physics patents receive more citations from Electricity. In Figure 3.14(b) Textile, Papers and Fixed constructions classes are pushed far away as the latent space now attempts to account for their negative receiver effects.



(a) (b)

Figure 3.10. a. The kullback-Leibler measure shows that whereas the static model shows a stable misfit to the dynamic latent model, the EKF and UKF both improve performance with an additional number of nodes $p$; b. Computational time grows markedly in the number of nodes $p$.



(a) (b)

Figure 3.11. a. With the increasing number of time points $n$ the Kullback-Leibler fit improves similarly for UKF and EKF, whereas the static model fit stays unchanged; b. the computational time grows linearly in $n$ for the UKF and EKF.

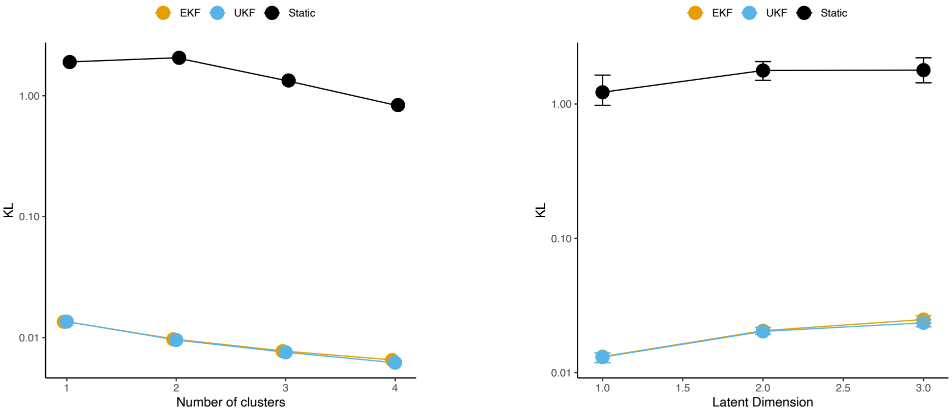


Figure 3.12. Two scenarios where the performance does not change substantially. KL measures by varying the number of clusters in the simulated data and increasing the latent dimension. The model fit approximately does not deteriorate with a higher dimension $d$ and does not change when we have clusters formed in the latent space.

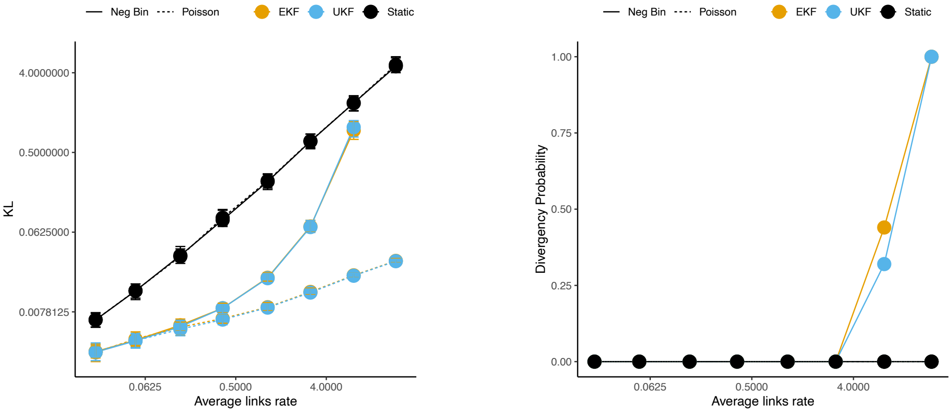
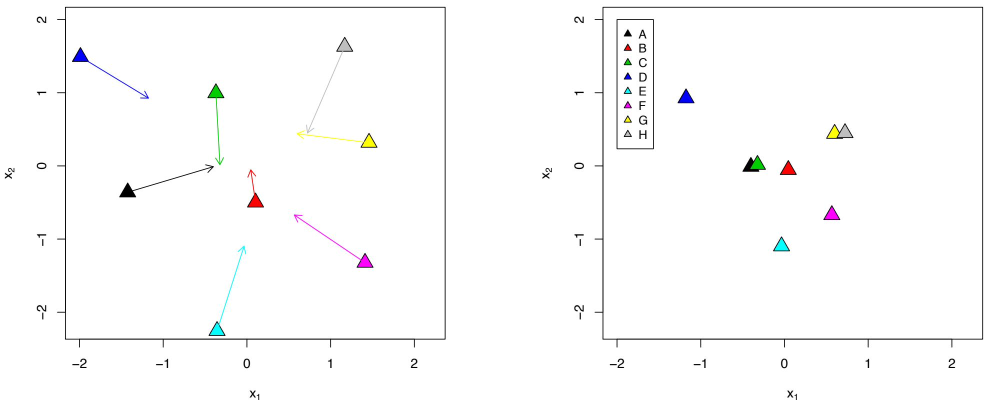


Figure 3.13. Overdispersion vs correct family specification performances varying the rate of links in the network. The divergence frequency suggests the level of overdispersion for which the model cannot retrieve the signal in the data.



(a) Changes in patent citation pattern. Interval years 1967-2006.

(b) Final configuration.

Figure 3.14. Model inference on dynamic locations for the relational event model without sender and receiver effects.

# Chapter 4

# Fast inference of latent space dynamics in huge relational event networks

I declare that the content of this chapter comes from the original pre-print [Artico and Wit, 2023a] on Arxiv in collaboration with E.C. Wit.

## 4.1 Summary

Relational events are a type of social interaction, that sometimes are referred to as dynamic networks. Its dynamics typically depend on emerging patterns, so-called endogenous variables, or external forces, referred to as exogenous variables. Comprehensive information on the actors in the network, especially for huge networks, is rare, however. A latent space approach in network analysis has been a popular way to account for unmeasured covariates that are driving network configurations. Bayesian and EM-type algorithms have been proposed for inferring the latent space, but both the sheer size of many social network applications as well as the dynamic nature of the process, and therefore the latent space, make computations prohibitively expensive. In this work, we propose a likelihood-based algorithm that can deal with huge relational event networks. We propose a hierarchical strategy for inferring network community dynamics embedded into an interpretable latent space. Node dynamics are described by smooth spline processes. To make the framework feasible for large networks we borrow from machine learning optimization methodology. Model-based clustering is carried out via a convex clustering penalization, encouraging shared trajectories for ease of interpretation. We propose a model-based approach for separating macro-microstructures and performing a hierarchical analysis within successive hierarchies. The method can fit millions of nodes on a public Colab

GPU in a few minutes.

## 4.2 Introduction

Networks are ubiquitous in various fields, such as gene regulation [Signorelli et al., 2016], finance [Cook and Soramaki, 2014], psychopathology symptoms [De Vos et al., 2017], political collaboration [Signorelli and Wit, 2018], and contagion [Užupytė and Wit, 2020]. The analysis of networks is crucial for understanding intricate relationships and interactions among the system components. However, the analysis can be challenging due to various endogenous and exogenous factors that may affect the network's formation. Therefore, statistical modeling aims to capture the underlying generative process to identify the drivers of these complex interactions. Such models can help filter out noise from the data and assist in learning certain features of the process, making interpretation possible.

The focus of our manuscript is on temporal random networks, where nodes create instantaneous directed or undirected connections with time stamps. Examples of such networks include email exchanges, bank loans, phone calls, and article citations. Traditionally, researchers have flattened the time variable and analyzed the resulting static network. However, this approach oversimplifies the temporal structure of the process and results in a loss of information. Most real-world networks are dynamic, where actors repeatedly form and break ties over time, and the adjustment of ties is influenced by the existence or non-existence of other ties [Brandes et al., 2009]. As such, the network is both the dependent and explanatory variable in this process.

To capture the temporal aspect of the network, we view the generative process as a network structure in which actors interact with each other over time through instantaneous events. This framework is known as *relational event modeling* and allows for quantitative analysis of the dynamic nature of the network.

A foundational form of the relational event model, which utilizes event history modeling, was introduced by Butts [2008] with an application to communication patterns during the World Trade Center disaster. The model was subsequently extended by Brandes et al. [2009] to accommodate weighted networks, where actors such as countries, international organizations, or ethnic groups engage in events that are assigned positive or negative weights based on cooperative or hostile interactions, respectively.

Relational event modeling has been applied to various domains, such as healthcare organizations in a regional community of patient transfers by Vu et al.

[2017] or social interactions between animals by Tranmer et al. [2015].

Relational event models allow for network connectivity to depend on the past evolution of the network. However, tracking the past configurations of a dynamic network can be challenging due to the vast number of possible configurations (e.g., k-stars, k-triangles) and their closure time, which can continue to affect future configurations. To address this challenge, we propose utilizing a dynamic latent space to summarize past configurations.

The idea behind a dynamic latent space is to describe the underlying structure of a network by placing vertices in a space where the distance between two points reflects their tendency or lack of tendency to connect. Social scientists often refer to this as a "social space," where actors who have more interactions are situated closer to one another and vice versa [Bourdieu, 1989]. The locations of the vertices are allowed to change over time, such that new connections are formed and subjects develop attraction/repulsion that forces them to adjust their social space configuration. The resulting configuration best reflects the new connectivity behavior, thereby providing a snapshot of the social history of the subjects, their preferences, and the groups they might join or leave. While this approach approximates the past information, it provides an effective summary of the network's evolution within the limits of the latent space formulation.

Many authors have studied the problem of tracking latent locations, especially in the static case where locations are assumed to be fixed over time. For instance, Hoff et al. [2002] provide a framework for inference for static binary random graphs. However, the limitations of the latent space formulation have led to the development of some extensions of that model, such as the bilinear mixed-effects model [Hoff, 2005], the multiplicative random graph model [Hoff, 2008, 2009], and the stochastic block model, which groups actors together based on their similarity. DuBois et al. [2013] have extended the stochastic block model to relational event data.

Sarkar and Moore [Sarkar and Moore, 2005] introduced a method for modeling dynamic binary networks using latent space. Their approach involves two phases: the first is a preprocessing step that uses generalized multidimensional scaling to obtain initial estimates of node locations, and the second is an estimation step where the dynamic locations are optimized using a conjugate gradient method. In this model, distances between nodes are approximated by thresholding larger ones, and a penalty is added to force distant nodes to be closer.

Several approaches have been proposed to model dynamic latent spaces with node-specific parameters, including the method by Sewell and Chen [2015], which uses the Metropolis-Hastings algorithm and case-control sampling for scalable inference, and the Bayesian approach by Durante and Dunson [2016], which utilizes Polya-Gamma data augmentation for binary connections and sequential learning of Gaussian processes for node dynamics. In a frequentist perspective, Artico and Wit [2022] presented a Relational Event Model that estimates Gaussian processes via a Kalman filter within an EM algorithm, providing a more robust convergence without requiring data augmentation. However, these methods have limited scalability to large networks.

### The methodology presented

The aim of this manuscript is to develop an efficient inference scheme for latent dynamic processes underlying an extremely high-dimensional relational event process. The framework is very general and can be extended to networks with weighted edges of any exponential family distribution. There are two dual representations of the process, either as a continuous time exponential or as discrete Poisson counts. Depending on the sparsity of the observed process, one or the other can be selected in the inference procedure. Interpretation of the huge dynamic latent space is made possible thanks to a clustering component that groups nodes with shared trajectories. The inference is performed under the stochastic variational inference framework, where the marginal lower-bound is directly maximized via parallel computing.

In Section 2 we propose the structure of the latent space and the relational event modeling background with the dual representation of the process. In Section 3 we present the penalized likelihood approach and stress the convex clustering penalization. Section 4 is dedicated to the optimization methodology. We consider a mini-batch stochastic gradient descent, a popular neural network optimization framework, and adapt it for graph data. The algorithm works on subsampling the data, hence particular care is given to sparse information handling. In Section 5 we leverage a variational approach to fit jointly both the model parameters and hyperparameters, such as smoothness and clustering. In Section 6 we show that the model can be run repeatedly within the detected clusters to fit a nested latent space. In Section 7 we present a simulation study. Section 8 is an application of our model to the complete Wikipedia history of edited pages.

## 4.3 Latent space relational event models

In this section, we introduce a general version of a latent space relational event model (REM). We consider a set of actors, defined as a finite vertex set $V = \{1, \dots, p\}$, that can exchange links or edges in time. In principle, we will consider the exchange of relational events, such as discrete interaction, e.g., sending an email or citing a patent, but one can also consider extensions to the quantitative exchanges, such as import and export. As drivers of the exchange process, we consider both endogenous, such as reciprocity, and exogenous variables, such as vertex characteristics. One particular exogenous variable is the relative location of the vertices in some similarity latent space, which itself is defined as a dynamic process.

We consider a non-homogeneous multivariate Poisson counting process $N = \{N_{ij}(t) \mid i, j \in V, t \in [0, T]\}$ and a smooth process $Z = \{Z_i(t) \in \mathbb{R}^d \mid t \in [0, T], i = 1, \dots, p\}$ relative to some standard filtration $\mathscr{F}$. In particular, we consider $\mathscr{F}$-measurable rate functions $\lambda_{ij}(t)$ that drive the components of the counting process. In particular, we assume that the rates $\lambda_{ij}(t)$ are functions of the underlying positions $Z_i(t)$ and $Z_j(t)$, besides possible other features. The features can be of various types: *exogenous* $x_{ij}(t)$, such as global covariates, node covariates, edge covariates, as well as *endogenous* $\mathscr{F}_t$-measurable $s_{ij}(t)$, where network statistics capture endogenous quantities such as popularity, reciprocity, and triadic closure. The parameter vector $\beta(t) = (\beta_0(t), \beta_1(t))$ determines the relative importance of the various effects. The rate function between nodes $i$ and $j$ at time $t$ is assumed to be

$$\log \lambda_{ij}(t) = m(z_i(t), z_j(t)) + \beta_0(t)^t x_{ij}(t) + \beta_1(t)^t s_{ij}(t) \tag{4.1}$$

where $m(z_i(t), z_j(t))$ is a similarity measure between node specific latent variables. The dynamics are assumed to follow a spline process

$$\begin{aligned} z_i(t) &= b(t)^t \alpha_i^z \qquad i = 1, \dots, p & (4.2)\\ \beta(t) &= b(t)^t \alpha^\beta, & (4.3)\end{aligned}$$

for some $m$ dimensional vector of basis functions $b(t)$. $\alpha_i^z$ is the $m \times d$ parameter matrix for a $d$-dimensional spline. The basic type is taken to be P-splines as a cheap representation of a Gaussian process. Node-specific splines correspond to $z_i(t)$ while $\beta(t)$ are splines shared by all nodes. The similarity measure $m(z_i(t), z_j(t))$ can be $z_i(t)^t \Lambda z_j(t)$ or $-||z_i(t) - z_j(t)||^2$. The measure $z_i(t)^t \Lambda z_j(t)$ comes from Hoff's eigen model [Hoff, 2008]. This measure can model multiple similarity forms:

- $\Lambda$ is a $k \times k$ matrix and $||z_i(t)||_2 = 1$: hyper-cube latent space, i.e., a stochastic block model.
- $\Lambda$ scalar and $||z_i(t)||_2^2 = 1$: hyper-sphere latent space where the distance measure is the angle between two nodes. This measure can be approximated locally by the Euclidian distance.

- $\Lambda$ scalar: latent space where the inner product defines the degree of similarity between two nodes. This model also expresses block modeling effects embedded into a similarity space. This measure finds interpretation in the angle between two points as a distance, whereas the norm of the single node describes the subjective tendency to make connections.

The first two measures, as well as the Euclidian distance, identify a non-convex optimization problem while the last one is convex. Although using a convex measure is appealing for the theoretical convergence guaranteed, it suffers from high dimensional saddle points which turn, from a practical perspective, to be similar to a non-convex optimization problem.

We assume a nested latent space, i.e., nodes form communities with common trajectories. These communities can be decomposed into sub-communities that have shared movements within the mother community. This can be repeated for many levels with a progression from the macro scale to the micro-scale. We do not make any specific assumption about the shape of these clusters. For most of this manuscript, we focus on detecting only the macro cluster level, while in Section 6 we describe the extension of the nested levels.

Given the joint formulation $(Z, N)$ of the state-space and interaction process, we will assume that only the interaction process $N$ is observed and the main aim of this paper is to infer the structure of the smooth process $Z$ and the rate functions $\lambda$, or more specifically, the parameters $\alpha$ associated with their functional form. We will consider two cases of the interacting point process defined above. First, we consider the general case, in which the relational events are observed in continuous time. This is the traditional setting for relational events. We will also define a relational event model where the interactions can only happen at specific times. For example, bibliometric citations or patent citations only happen at prespecified publication dates. Furthermore, this model allows a generalization to non-binary relational events, such as export between countries, that can be dealt with in the same inferential framework.

### 4.3.1 Continuous time relational event process $N$

We consider a sequence of $n$ relational events, $E_{\text{cont}} = \{(i_k, j_k, t_k) \mid t_k \in [0, T],\ i_k, j_k \in V, k = 1, \ldots n\}$ observed according to the above defined relational counting process $N$. Conditional on the smooth process $Z$, the distribution of the interarrival time for interaction $i \to j$ is a generalized exponential, with instantaneous rates as described in (4.1). The conditional log-likelihood of the process

$Z \mid N$

$$\ell(\alpha) = \sum_{i,j} \left[ \sum_{t \in E_{\text{cont}}(i,j)} \log \lambda_{i,j}(t) \right] - \int_0^T \lambda_{i,j}(t) dt \tag{4.4}$$

where the generalized exponential formulation is the one adopted by Rastelli and Corneli [2021]. This likelihood is commonly simplified in the REM literature with the partial likelihood [Perry and Wolfe, 2013] relative to the equivalent Cox process [Cox, 1972].

### 4.3.2 Discrete time relational event process $Y$

Often relational events are "published" only on prespecified discrete event times $\mathcal{T} = \{t_1, \dots, t_n\}$. We consider a sequence of $n$ relational events, $E_{\text{disc}} = \{y_{k,ij} \mid t_k \in [0,T],\ i_k, j_k \in V, k = 1, \dots n\}$ where the interactions $i \rightarrow j$ are collected at $t_{k+1}$ from the observation intervals $(t_k, t_{k+1}]$, with resulting interval counts

$$y_{k,ij} = N_{ij}(t_{k+1}) - N_{ij}(t_k).$$

We assume that the rate $\lambda$ is constant with respect to the endogenous and exogenous variables inside the collection intervals $(t_k, t_{k+1}]$. In fact, with respect to the endogenous variable $N$ it makes sense that no further information between the publication dates affects the rates. In other words, we assume that the log link at equation (4.1) for the hazard is conditioned to the past information up to time $t_k$.

The interval counts $y_{k,ij}$ of the number of interactions between $i$ and $j$ are Poisson distributed with interval rate,

$$\int_{t_k}^{t_{k+1}} \lambda_{ij}(t)\, dt = \lambda_{ij}(t_k)\Delta t_k, \tag{4.5}$$

where $\Delta t_k = t_{k+1} - t_k$. An advantage of using discrete time is the reduction of the model complexity. In certain real-world processes, it is not uncommon to observe thousands, even millions of links. A discrete-time representation reduces the computational complexity from the number of links to the number of collection intervals.

Given the complete observations $(Z, Y)$, the complete log-likelihood for the discrete-time latent space model is

$$\ell(\alpha) = \sum_{k, i \neq j} -\lambda_{i,j}(t_k)\Delta t_k + y_{k,ij} \log \lambda_{i,j}(t_k)\Delta t_k \tag{4.6}$$

Similar to Perry and Wolfe [2013], who focuses on non-homogeneous exponential waiting times, this approach focuses on non-homogeneous Poisson counts.

This approach can be further generalized to any exponential family [Artico and Wit, 2023b] or the zero-inflated exponential family [Sewell and Chen, 2016].

## 4.4 A penalized likelihood approach

For inferring the above model we aim to maximize the following penalized log-likelihood

$$\ell^P(\alpha) = \ell(\alpha) + P_{\text{smooth}}(\alpha) + P_{\text{clust}}(\alpha) \tag{4.7}$$

where $\ell(\lambda)$ is either (4.4) or (4.6) depending on the case. $P_{\text{smooth}}$ is a smoothness penalty on the spline process, and $P_{\text{clust}}$ is a convex clustering penalty for forcing nodes to be closer. Although in the classic formulation of generalized additive models [Wood, 2006] the process smoothness is regulated by penalizing the second derivative $\int_0^T \alpha^2 b(t)'' dt$, for dynamic systems it is more important to consider the first derivative as it regulates the difference between a static or a dynamic model. Moreover, the latent space is not identifiable due to rotations: the resulting dynamics are hence the original nodes' trajectories plus infinite infra-time rotations. A first derivative penalty reduces rotations that are misinterpreted as node dynamics. For P-splines the penalty has a convenient form

$$P_{\text{smooth}}(\alpha) = -\gamma_{\text{smooth}} \sum_{i=1}^{p} \sum_{k=2}^{m} \left\| \alpha_{i,k} - \alpha_{i,k-1} \right\|^2$$

with the first order differences on the basis heights. P-splines are a low-rank, smooth representation of a Gaussian process. The basis captures the local temporal structure of the process and a finer granularity can be achieved by increasing the number of basis $m$. For $m = n$ we obtain a Gaussian process. Taking $m < n$ has both computational benefit and a potential overfitting reduction.

### 4.4.1 Convex clustering penalty for community detection

A common problem that arises when using large dimensional models is that results are dense. It is hard to interpret a large number of parameters. Therefore we simplify our model fit by grouping together nodes into communities that share common movements. It is often more sensible to spot common movements across different nodes in order to separate them from nodes with independent trajectories. We cluster node trajectories with the popular convex clustering penalty

[Pelckmans et al., 2005; Hocking et al., 2011; Chen et al., 2015; Weylandt et al., 2020]

$$P_{\text{clust}}(\alpha) = -\gamma_{\text{clust}} \sum_i \int_0^T (z_i(t) - c_i(t))^2 dt - \gamma_{\text{dist}} \sum_{i<j} w_{ij} \int_0^T (c_i(t) - c_j(t))^2 dt.$$

Similarly to the smoothness penalty, this penalty finds a discrete simplification

$$P_{\text{clust}}(\alpha) = -\gamma_{\text{clust}} \sum_i ||\alpha_i^z - c_i||^2 - \gamma_{\text{dist}} \sum_{i<j} w_{ij} ||c_i - c_j||^2 \tag{4.8}$$

$$\text{where} \qquad w_{ij} = \mathbb{I}_{[0,\gamma_{\text{radius}}]}(||\alpha_i^+ - \alpha_j^+||) \tag{4.9}$$

thanks to the P-spline low-rank process representation. This penalty yields a unique solution to a combinatorial problem, which is typically non-convex. This formulation [Hocking et al., 2011; Sun et al., 2021] shrinks the closest nodes in a hierarchical sequence. It consists of a vector of features $\alpha_i^z$ and a vector of auxiliary variables $c_i$ that corresponds to node $i$ centroid. $\alpha^+$ are considered a reliable estimate of the true parameters. The first component $\sum_i ||\alpha_i^z - c_i||^2$ ensures that the centroids are sufficiently close to the respective nodes while the second component $\sum_{i<j} w_{ij} ||c_i - c_j||^2$ enforces closer centroids to shorten their distance. The parameters $\gamma_{\text{dist}}$ and $\gamma_{\text{clust}}$ regulate the amount of shrinkage for the centroid-centroid and the centroid-$\alpha$ distance respectively. We can group together centroids that are closer than a certain threshold $\epsilon$. Faster convergence and different cluster shapes can be achieved by altering the kernel $w_{ij}$. The kernel aims to increment the penalty locally, its radius is regulated by $\gamma_{\text{radius}}$. Common choices for the kernel are Gaussian or discrete, as in (4.8), whose performances are approximately equivalent.

In the original convex clustering formulation $\alpha$ corresponds to observed features and the kernel is calculated using them as input. A popular attempt at clustering unobserved features comes from Lindsten et al. [2011] who clustered the latent states of a Kalman Filter model. Similarly, we estimate these $\alpha^+$ by a pilot optimization phase where we fit the *vanilla* model including the smoothness penalty only. These estimates $\alpha_i^+$ will be considered as fixed in the further inference. In case we have convexity in both the likelihood similarity measure and in the penalty, we obtain a double-convex optimization problem. The clustering path can be computed by increasing the kernel radius $\gamma_{\text{radius}}$ or by the shrinkage $\gamma_{\text{dist}}$ in different strategies. As the radius increases, more nodes are included in the kernel and are shrunken, leading to a hierarchical procedure that ends in a single cluster.

### 4.4.2 A fast convex clustering penalty

The inclusion of a clustering and distance penalty in the original convex clustering formulation produces, however, a near unidentifiability between $\gamma_{\text{clust}}$ and $\gamma_{\text{dist}}$. Given a fixed radius, multiple combinations of $\gamma_{\text{clust}}$ and $\gamma_{\text{dist}}$ have nearly identical predictive performance without any preference on whether aggregating nodes or not. From a geometrical perspective the amount of shrinkage on $\alpha$ can be held constant for any value of $c$ that follows the path from $c = \alpha$, hence $\gamma_{\text{dist}} = 0$, to the point of centroid aggregation at $\gamma_{\text{dist}} \to +\infty$. We can bypass the problem by “dropping" entirely the distance component. The aim is to cluster all the nodes that enter into the kernel. For $\gamma_{\text{dist}} \to +\infty$ groups of centroids have perfect matching and the minimization of the convex clustering penalty (4.8) finds analytic solution as

$$P_{\text{clust}}(\alpha) = -\gamma_{\text{clust}} \sum_{i=1}^{p} \|\alpha_i - c_i\|^2 \tag{4.10}$$

$$\text{where} \qquad c_i = \sum_{j=1}^{p} \alpha_j \mathbb{I}\{i - j\} / \sum_{j=1}^{p} \mathbb{I}\{i - j\} \tag{4.11}$$

which has computational complexity linear in $p$ rather than quadratic as before. The value $c_i$ is the average coordinate among all nodes belonging to the same cluster as $i$, which needs to be calculated once for each cluster. $\mathbb{I}\{i - j\}$ simply indicates the cluster assignment or, more precisely, if there exists a path of kernels that connects $i$ to $j$. Thus $\mathbb{I}\{i - j\}$ indicates that $i$ and $j$ belong to the same connected component in the graph constructed by kernel $w_{ij}$. This can be done by updating the kernel adjacency list as the sequence of samples $B$ is filtered by the kernel $w_{ij}$. This implies that not all the pairwise relationships $w_{ij}$ need to be observed, just the ones that relate a node to at least one other node of the same cluster.

Convex clustering can be considered as a hard clustering method where nodes with unique dynamics are modeled independently, instead of being considered as outliers or abusively allocated to the closest cluster. An alternative approach is proposed by Handcock et al. [2007] with a finite Gaussian mixture model, which may suffer from local minima or high dimensionality. Furthermore, the latter can only detect circular clusters, while in our method we do not specify the cluster distribution.

Alternatively to kernel aggregation, a useful heuristic exists. The fast convex clustering penalty (4.10) can be seen as the analytic equivalent to the *hdbscan* heuristic [Schubert et al., 2017] where nodes belonging to the same discrete kernel are sequentially aggregated as the kernel enlarges. This heuristic can suggest good candidate radii to test and offer a more robust allocation. Moreover the $\gamma_{\text{dist}} \to \infty$ convex clustering version can be interpreted in a more general perspective where any clustering or aggregation algorithm can be used and the resulting cluster allocation can be plugged in the model. Thus our approach opens the door to a supervised clustering selection method for a wide range of existing algorithms.

## 4.5 Optimization

The computational complexity for optimizing the model described in section (4.4) is prohibitive when the data dimension is very large. In these cases, it is necessary to restrict the inference over subsamples of the data. A method that we borrow from machine learning is the so-called mini-batch gradient descent. It consists of taking a random subsample from the data named mini-batch $B$, where $B \subset E$ and $E = E_{\text{cont}}$ or $E = E_{\text{disc}}$, according to the case. The mini-batch has typically a small size $n_b =\mid B \mid$. The fast computation, mostly matrix operations, is restricted to the mini-batch. Over this subset the likelihood $\ell(\alpha)_B$ is calculated and a gradient step is taken, such as $\alpha \leftarrow \alpha + \psi \nabla \ell(\alpha)_B$. The procedure is repeated, sampling new mini-batches B, until convergence. As a result of the subsampling, the gradient is an unbiased estimator of the full gradient. The mini-batch gradient trades variance for computational and memory costs. For a certain mini-batch size, stochastic gradient descent reaches the minimum faster than a deterministic gradient. The gradient update step is a Newton step where the costly second derivative matrix is substituted by a cheap but unknown $\psi$ parameter. As a result, the missing Hessian leads to the gradient elements having the wrong individual scale, hence the wrong global direction in the gradient vector. The past literature, e.g. [Ruder, 2016; Duchi et al., 2011], has focused on two main issues: decreasing the gradient variance and rescaling the gradient estimate. Both problems are solved by the popular *Adam* [Kingma and Ba, 2014]. In *Adam* the gradient update is formulated as a state-space model, where the gradient moments are thought of as latent states. Leveraging a simple, univariate form of the Kalman Filter, known as Exponentially Weighted Moving

Average (EWMA), the update has the form

$$\begin{aligned} g &\leftarrow \nabla\ell(\alpha)_B \\ m_k &\leftarrow \xi_1 m_{k-1} + (1-\xi_1)g \\ v_k &\leftarrow \xi_2 v_{k-1} + (1-\xi_2)g^2 \\ \alpha_k &\leftarrow \alpha_{k-1} + \psi\frac{m_k}{v_k} \end{aligned}$$

at iteration $k$, $m_k$ and $v_k$ are the gradient first and second moments, respectively. Hence the moments are a weighted average with the past moments, where the weights decrease exponentially in time. The $\xi$ parameters regulate how much of the past information is used to update the current moments. Thus *Adam* provides an estimator for the first two gradient moments. The benefit from the averaging is the variance reduction of these moments, although some bias might be introduced if the process relies too much on the past. Moreover, leveraging the Bartlett identity $E[\frac{\partial^2}{\partial^2\alpha}\ell(\alpha)] = E[(\frac{\partial}{\partial\alpha}\ell(\alpha))^2]$ we have that $v_k$ is an estimator of the diagonal elements of the Hessian matrix. Imposing locally, i.e. at iteration $k$, the assumption of a spherical covariance matrix between the parameters, the inverse of the diagonal Hessian applies an effective rescaling to the gradient elements. The algorithm can also tackle high parameter correlation or ridge problems by learning the correct direction from the past steps. The lack of the off-diagonal Hessian elements is hence replaced by the gradient averaging over the past noisy directions. The optimization is performed until the algorithm reaches the maximum or, more precisely, a stationary distribution at the maximum. This stationary distribution has been extensively studied and in some cases it can be considered as a posterior distribution [Mandt et al., 2017]. The optimization is stopped if the algorithm does not find a new maximum after a reasonably high number of iterations.

Although *Adam* has shown to be effective in many scenarios, it has some side effects. The algorithm can suffer from pathological cases of severe parameter scale imbalance or large gradients variance (see Section 4.5.1 about sparsity). The problem of scale is commonly tackled in machine learning via parameter normalization. In our case, it can be mitigated by using basis splines that share a similar scale in the weights, such as P-splines.

### 4.5.1 A sparse gradient update problem

When working with high-dimensional problems, the amount of information contained in the mini-batch determines the success of the optimization. In our model, the shortage of information corresponds to the problem of sparsity. In this section, we tackle two types of sparsity: sparsity in the sampled connectivity and sparsity in the sampled parameters. *Adam*, by increasing the long term memory parameters $\xi$, is designed for solving sparse update problems. However, in extremely sparse scenarios the gradient variance can become too high and the EWMA cannot recover a decent signal from the noise.

### Sparsity in the parameters

The mini-batch size determines how many nodes and time points, hence parameters $\alpha_i$, are included in the current iteration. The gradient over the missing parameters is zero, therefore the EWMA performs a smooth averaging over a sparse vector. A way for reducing the gradient variability is to include as many parameters as possible in the mini-batch. A mini-batch of size $n_b$ on average contains $0.632 \times 2n_b$ nodes, where 0.632 is the resampling bootstrap ratio. Given the local structure of the P-spline basis, every time point corresponds to 4 non-zero basis. We hence update an average of $0.632 \times 2n_b \times 4 \times d$ parameters over a total of $pmd$ parameters. Fixing $m = 10$ allows us to fit a 10 degrees of freedom function, a value that is sufficiently high in most applications. The gradient is sufficiently dense as long as the ratio $0.632 \times 8n_b/pm$ is close to 1. The size of the mini-batch should hence grow linearly with the nodes $p$. Possible choices are between $n_b = p$ and $n_b = 2p$ for a ratio of approximately 0.5 and 1 respectively. These values correspond to a sparsity level that *Adam* can handle easily, see Figure 4.2. Moreover, the calculations are made under the worst-case scenario where all the degrees of freedom are necessary. In case the effective degrees of freedom are less than $m$ the smoothness penalty defines a dependency chain over the basis parameters, i.e., parameters are more correlated and move together. The level of smoothness regulates how local is this kind of dependency: the higher the smoothness, the lower the effective number of parameters.

A similar reasoning applies to centroids $c_i$. The simplification in (4.10) solves another important sparsity problem. If we were using the original penalty (4.8) the quadratic cost of the distance component would require some sort of subsampling, i.e., a mini-batch penalty $\sum_{i,j \in B} w_{ij} ||c_i - c_j||^2$. Since the chances of randomly sampling two close nodes are almost zero for large networks, the vast majority of elements would be excluded by the kernel. As a consequence, the level of sparsity of the gradient with respect to $c$ would be even higher than for the splines. This results in an ineffective shrinking of centroids. Instead (4.10) solves the problem by removing this component. The gradient is calculated over all the centroids and they are aggregated by the kernel only. In Appendix 4.11.1

we propose an alternative mini-batch convex clustering penalty.

Rastelli and Corneli [2021] constructed the mini-batch by sampling a set of nodes, rather than edges like our case, including all the dependencies with the remaining nodes. This produces a node-wise update where the information tends to focus too much on a single node and very little on the others. The algorithm needs to cycle over all the nodes before focusing on the same nodes again. The optimization is carried by a memory-less Stochastic Gradient Descent that cannot compensate for the imbalance. These two factors might result in slow or false convergence.

### Sparsity in the links

Sparsity not only occurs in sampling nodes but also in the observed data and in the information of the gradient. We refer to this as gradient sparsity in a more general sense. The problem with independent sampling in a sparse large network is that distant nodes are sampled more often, which do not interact. The large number of zeros that overcrowd the mini-batch is redundant, hence very little information. As a result, the gradient taken over the mini-batch rarely contains information about the connectivity between two nodes. The redundancy lies in the fact that the macro-level structure of a large network can be summarized by a few "compound" zeros that connect macro components.

Some authors have tried to solve the problem by partitioning the latent space into blocks. Hence the overall number of interactions can grow only linearly with the number of nodes [Rastelli et al., 2018]. Case-control sampling overcomes the redundancy in the data by including in the sample as many links as possible (cases), with minimal inclusion of zeros (controls). The idea consists of dropping the majority of zeros and making a few of them representatives of the entire non-interacting population. The only consequence of case-control sampling is the increase of variance in the estimates, but this is commonly compensated by a large amount of data. Raftery et al. [2012] give a detailed procedure on how to perform stratified case-control sampling for static binary networks. The shortest path distances are used as a proxy of the latent distance, allowing for the stratification of controls at different lengths. Controls are sampled in each stratum for each node. Particular care must be paid to sampling the same control for the two nodes in order to avoid unnecessary biases in the case-control weights, as the two pushing forces might differ substantially if the two nodes have substantially different centralities. The procedure approximates the likelihood and successfully captures both macro and micro-structure in the latent space. However, the preprocessing phase where controls are sampled is both computationally and memory expensive.

A cheaper solution is proposed in the Supplementary Material of Sewell and Chen [2016], applied to temporal networks. The stratification is dropped and the controls are sampled at random, capturing mainly the macro-structure. An additional control set contains all the non-interactions of nodes with at least one interaction during the time span. Although this set accounts for a minimal micro-structure, its memory requirements can explode easily. The set size indeed increases as time goes to infinity since it is more likely to observe at least one interaction between two nodes.

In our approach, we drop the micro-community structure since we have a clustering formulation. We therefore can make further simplifications in the case-control sampling. Sampling controls at random capture mainly the macro-structure as you sample more frequently distant nodes. We propose two different model formulations. Depending on the level of the sparsity of the process, a continuous time or a discrete-time formulation.

### A discrete-time model for dense data

The model in (4.6) can be used when the network presents many interactions. Clearly storing the adjacency matrix elements (the square of the nodes × the number of time intervals) is unfeasible for large networks, hence we restrict this usage only to cases where the interactions can be calculated on line. For such cases, there is no need of storing all the pairwise interactions as they can be calculated during the sampling phase.

### A continuous time model for sparse data

We propose the case-control version for the inference of a continuous time relational event model (4.4). A popular approach in the REM literature [Butts, 2008; Brandes et al., 2009; Vu et al., 2017] is to maximize the so-called partial likelihood

$$PL(\alpha) = \prod_{tij \in E} \frac{\lambda_{ij}(t)}{\sum_{kl} \lambda_{kl}(t)}$$

of the Cox process $N$ at (4.4). As the risk set in the denominator is computationally challenging, Vu et al. [2015] following Borgan et al. [1995] show that a random subset of the risk set yields a consistent estimator for the model parameters. Lerner and Lomi [2020] pushed this concept to the limit by showing that sampling one single control is a sufficient statistic for the risk set, fitting successfully a REM over millions of nodes. The partial likelihood in that case is

$$\ell(\alpha) = \sum_{tij \in E} \log \frac{\lambda_{ij}(t)}{\lambda_{ij}(t) + \lambda_{i^* j^*}(t)} \tag{4.12}$$

where $i^*, j^*$ is a sampled control at time $t$. This case-control sampling hence allows storing in memory only the history of links. The mini-batch $B$ is composed of sampling half links and half controls, where new controls are sampled at each likelihood evaluation. The only drawback of subsampling one single element is the increase of variance in the estimates, as it is inversely proportional to the number of controls subsampled. This is compensated by the vast amount of data that comes from a large network. Similarly to *Adam*, the case-control likelihood trades variance for computational efficiency.

In case the links are dense within communities a case-control discrete-time model is considered in Appendix 4.11.2.

### 4.5.2 Mini-batch model

The calculation of mini-batch loss should be computed efficiently. We require that the matrix operations grow linearly with the number of nodes $p$. At each iteration, we sample a mini-batch $B \subset E$, where $E$ is either $E = E_{\text{cont}}$ or $E = E_{\text{disc}}$, consisting of randomly sampled pairs $i, j$ and time $t$ from the data set. We set the mini-batch size $\mid B \mid = 2p$ to ensure that the gradient is calculated over the majority of parameters. Lower sizes might update only a little portion of nodes, destabilizing the optimization algorithm as discussed in Section 4.5.1. All the matrix operations and gradients are computed over the mini-batch penalized likelihood

$$\ell(\alpha)_B^p = \frac{\mid E \mid}{\mid B \mid} \ell(\alpha)_B + P_{\text{smooth}}(\alpha) + P_{\text{clust}}(\alpha), \tag{4.13}$$

where $\ell(y, \lambda)_B$ is the likelihood evaluated over B, given in (4.12) or (4.6) for sparse or dense network scenarios. Similarly to case-control weights, $\frac{|E|}{|B|}$ rescales the likelihood component accounting for the downsampling. $P_{\text{smooth}}$ and $P_{\text{clust}}$ do not require any subsampling since they have a computational complexity that is linear in $p$. Additionally, they yield a faster optimization as the full parameters dependencies are included.

## 4.6 Stochastic Variational Inference

In this section, we discuss how to estimate both the model parameters $\alpha$ and the hyper-parameters $\gamma = (\gamma_{\text{smooth}}, \gamma_{\text{clust}}, \gamma_{\text{radius}})$. Given the full parameter vector

$\theta = (\alpha, \gamma)$ a naive choice for maximizing the marginal likelihood $p(y) = \int p(y \mid \theta)p(\theta)d\theta$ can be k-fold cross-validation. Validation sets are iteratively removed from the model inference and hyper-parameters are selected as the best performing in these sets. Although cross-validation is a good way for assessing hyper-parameter tuning in dense networks, it can be unreliable for sparse scenarios. In order to avoid removing relevant information about single-node dynamics, the validation set should be as small as possible. This leads to a high number of validation sets, hence a high computational burden. Moreover, the number of hyper-parameters is recommended to be either low or weakly dependent, which is not our case.

Our proposed approach for maximizing $p(y)$ is via stochastic variational inference [Kingma and Welling, 2013; Hoffman et al., 2013; Blei et al., 2017; Kucukelbir et al., 2017]. Variational inference aims to maximize the following lower-bound of the marginal likelihood

$$\begin{aligned}
\log p(y) &= \log \int p(y \mid \theta)p(\theta)d\theta = \log \int p(y \mid \theta)q_{\mu,\sigma}(\theta)\frac{p(\theta)}{q_{\mu,\sigma}(\theta)}d\theta \\
&= \log \mathbb{E}_{q_{\mu,\sigma}}[p(y \mid \theta)\frac{p(\theta)}{q_{\mu,\sigma}(\theta)}] \\
&\geq \mathbb{E}_{q_{\mu,\sigma}}[\log p(y \mid \theta)] + \mathbb{E}_{q_{\mu,\sigma}}[\log p(\theta) - \log q_{\mu,\sigma}(\theta)] \qquad (4.14) \\
&= \mathbb{E}_{q_{\mu,\sigma}}[\log p(y \mid \theta)] - D_{\mathrm{KL}}[q_{\mu,\sigma} \| p] = \mathscr{L}(\mu, \sigma) \qquad (4.15)
\end{aligned}$$

where the unknown true density $p(\theta)$ is in practice replaced by an arbitrary prior distribution and the posterior distribution is approximated by the variational density $q_{\mu,\sigma}(\theta)$. A common choice is independent Gaussian $q_{\mu,\sigma}(\theta) = \prod_{i=1}^{p+3} q_{\mu_i,\sigma_i}(\theta_i)$ where all posterior dependencies are ignored and inference reduces to the first two posterior moments $\mu, \sigma^2$. Differently from the mean-field approach that aims to find a recursive closed form of $q_{\mu,\sigma}(\theta)$, stochastic variational inference aims to directly maximize (4.14) where the untractable components of the lower-bound are approximated via Monte Carlo integration [Kingma and Welling, 2013; Kucukelbir et al., 2017]. All parameters can hence be updated simultaneously using *Adam* stochastic gradient optimization. The only element that requires Monte Carlo evaluation is the mini-batch likelihood. As shown by Kingma and Welling [2013] in the expectation $\mathbb{E}_{q_{\mu,\sigma}}[\ell_B(\alpha)] = \frac{1}{H}\sum_{h=1}^{H} \ell_B(\alpha^h)$ the number $H$ of Monte Carlo replicates drawn from $q_{\mu,\sigma}(\alpha)$ can be reduced to 1 when the mini-batch size is sufficiently large and the optimization is performed via moving average gradient scheme. At each iteration we draw one Monte Carlo sample $\alpha^*$ from the variational density $q_{\mu,\sigma}(\alpha)$ obtaining the mini-batch lower-bound

$$\mathscr{L}_B(\mu,\sigma) = \frac{|E|}{|B|}\ell_B(\alpha^*) + \mathbb{E}_{q_{\mu,\sigma}}[P_{\text{smooth}}] + \mathbb{E}_{q_{\mu,\sigma}}[P_{\text{clust}}] - D_{\text{KL}}[q_{\mu,\sigma}\|p]$$
$$\text{where} \qquad \alpha^* = \mu + \sigma\epsilon \qquad \epsilon \sim N(0,1),$$

which is the quantity we maximize. The reparametrization $\alpha^* = \mu + \sigma\epsilon$ ensures that the gradient is not affected by noise in updating the parameters $\mu,\sigma$. Moreover, we recommend initializing $\sigma$ small, as the single-sample Monte Carlo integration is prone to diverge for large variance. In a variational context, the two penalties naturally translate into Bayesian priors. The three remaining expectations $D_{\text{KL}}[q_{\mu,\sigma}(\theta)\|p(\theta)]$, $\mathbb{E}_{q_{\mu,\sigma}}[P_{\text{smoooth}}]$, $\mathbb{E}_{q_{\mu,\sigma}}[P_{\text{clust}}]$ have simple close form solutions thanks to the Gaussianity and independence, see Appendix 4.11.3 for details.

Variational inference works particularly well in settings where $q_{\mu,\sigma}(\theta)$ provides a sufficiently good approximation of the posterior, i.e., the lower bound reaches a sufficiently close value to the marginal. The independence assumption on $q_{\mu,\sigma}(\theta)$ is appropriate for a posterior that is approximately independent or, like in our case, locally dependent. The conditional dependency induced by observing the data, i.e., the posterior covariance, is locally present for close nodes and adjacent time points. Hence latent network representations combined with a Gaussian process are particularly suited for variational inference, as it ignores a relatively small amount of information when approximating with an independent posterior. Once again we fit the macro scale by sacrificing the micro-scale dependencies. Finally, our model can be seen as variational autoencoder [Kingma and Welling, 2013] with the addition of penalties. Despite its most common usage as an image generator, a variational autoencoder is a more general framework for representing any Bayesian inference problem as an encoder-decoder. For our model, the decoding side is fully structured by the link function while the encoder reduces to a selector operator that associates an edge to the respective posterior node positions in the latent space.

## 4.7 Marginalization: A hierarchical community model

For static networks, repeated community detection can be used to detect hierarchies of nodal communities. Our methodology can be seen as a dynamic model-based partitioning of the nodes. By repeated application of our method, we can obtain nested communities in dynamic networks. The concept of nested communities is appealing to practitioners, where interpretation is simplified via nested structures.

This divide-and-conquer approach suits well the model's purpose. Given the set of clusters, the latent space model is estimated recursively inside each cluster. This nested procedure can be iterated multiple times as long as the variance of the locations allows for meaningful community discovery. This procedure is performed over clusters of reasonable size: unassigned nodes or small communities are left untouched. This fitting procedure can be seen as adding a random effect to the model for explaining within-cluster variance.

Under the latent space assumption, any marginalization or sub-sampling of the original network is a coherent estimator of the locations and therefore the inference in the micro-structure can be done regardless of the macro-structure. Given that any subset $V'$ of $V$ maintains the same distances among nodes, the distribution of the restricted node set $P_{V'}$ is the same as the marginalized distribution of the full model $P_V \mid_{V'}$. This invariance means that it is unimportant to which node set the observed nodes actually belong. The model is therefore invariant under marginalization.

The micro-communities formulation offers various advantages. In case the community is sufficiently small we can account for all the dependencies with a full covariance matrix for the variational parameters as proposed in Blei et al. [2017] or a low-rank approximation of it backed by importance sampling [Zhang et al., 2021]. Moreover, time dynamics can have a finer granularity, thus they can be captured with a higher number of spline basis or a Gaussian process. The Extended Kalman filter model proposed in Artico and Wit [2023b] performs sequential learning of Gaussian processes embedded in dynamic networks. The model can be thought of as a special case of variational Expectation Maximization where the posterior is approximated by a multivariate Gaussian matching the first two moments.

## 4.8 Simulation study

We dedicate this section to investigating the features of the estimation procedure. We are particularly interested in exploring: the goodness of fit and computational time as the number of nodes varies, the convergence behavior for different minibatch sizes, the comparison of different models for different sparsity scenarios, and the accuracy of classification in different clustering settings. As the locations are not identifiable up to an arbitrary rotation, translation, and mirroring, MSE is calculated by pre-processing results via a Procrustes transformation, search-

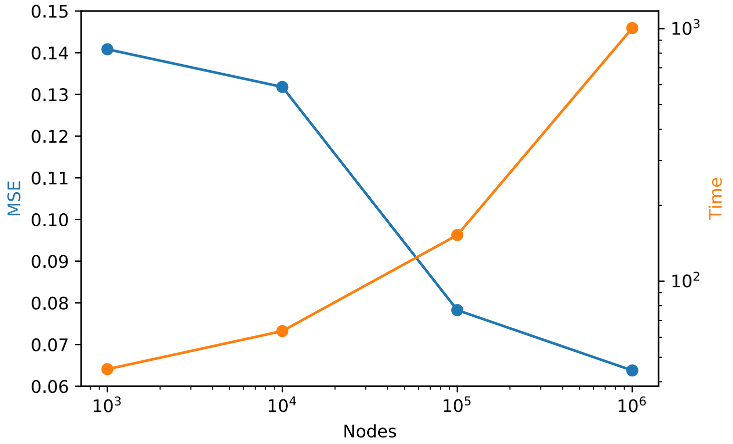


Figure 4.1. Model average performance and computational time. The goodness of fit (MSE) improves as the nodes increase. Computational time (in seconds) in the log-log plot express a sub-linear increase, showing that the model scales at most linearly with the number of nodes. A network with $10^5$ nodes takes approximately 5 minutes of training while a network with $10^6$ nodes takes approximately 20 minutes.

ing for the best rotation and translation that match the truth. Simulations are repeated 10 times and nodes starting points are set at 0.

Vary number of nodes The first scenario is presented in Figure 4.1 where the average MSE between the fitted and true trajectories is calculated. The goodness of fit improves with the nodes. This supports the consistency of the latent location estimator as it converges to the true locations for a large number of nodes [Shalizi and Asta, 2017].

Average computational time, in a log-log plot, follows a sub-linear increase showing that the model scales at most linearly with the number of nodes. The increasing angle indicates the limitations of the GPU used in these analyses. One million nodes indeed require a significant use of memory, which slows down computations. All our analysis have been conducted with a standard and free Colab GPU, which struggle beyond 4 million nodes. We suggest switching to more powerful GPUs for larger settings.

Vary mini-batch size The second set of experiments consists of varying the mini-batch size. We use as a standard setting a network with $10^5$ nodes. Figure

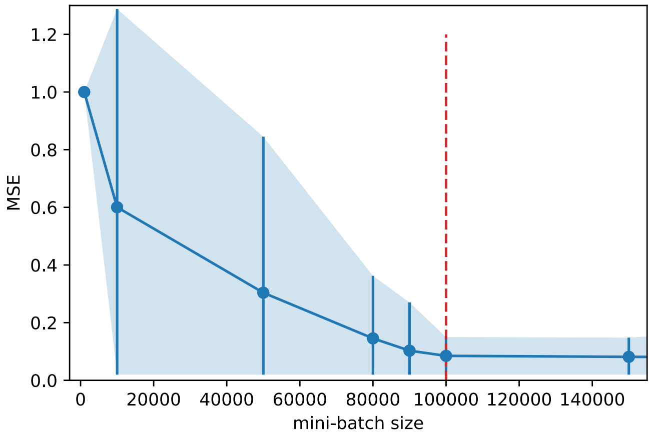


Figure 4.2. Vary the mini-batch size. MSE (blue line) improves as the mini-batch increases. The high standard deviation (blue bars and shades) highlights false convergence behavior below the safe threshold(vertical red line) where the sparse parameter update is not sufficiently informative. For the lowest mini-batch size, the algorithm does not make any meaningful movement from the starting points.

4.2 shows how a low mini-batch size can cause false convergence as the level of sparsity in the parameter update does not carry enough information for a proper gradient direction recovery, as mentioned in Section (4.5.1). For the lowest mini-batch size considered ($10^3$) the fit has both poor MSE and low standard deviation. This means that the algorithm does not move. By increasing the size we have a gradual improvement of the MSE, however the high standard deviation points to a serious instability, which might or not converge to a good value. The behavior stabilizes above a mini-batch size of $10^5$, giving both low MSE and stability. Hence a mini-batch size $n_b = h \times p$, with $h > 1$, can be considered a safe ratio for ensuring the fitting.

Vary sparsity in the links We compare the behavior of the algorithm under different sparsity levels for some models presented in 4.5.1. We compare the Poisson model for dense network activity with the Cox model for the sparse case, showing that they have comparable performance. The Poisson model performs optimally in dense scenarios, however, it deteriorates as sparsity increases, in a behavior very similar to Figure 4.2. In Figure 4.3 we show that the Poisson model in the sparse scenario performs inevitably worse than in the dense scenario. The

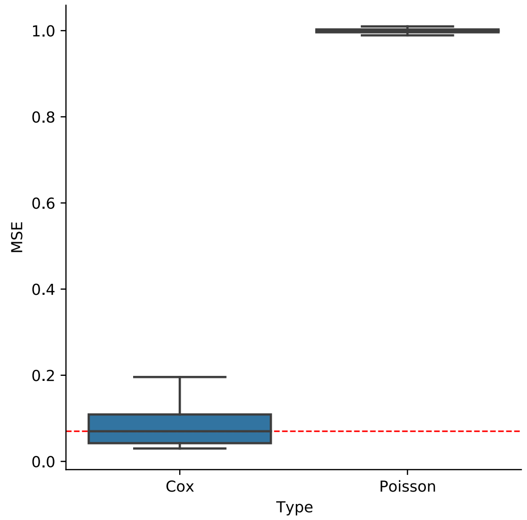


Figure 4.3. High sparsity scenario. The red line is the Poisson model performance for the dense scenario, we keep it as a benchmark. The Poisson fit deteriorates showing inadequacy for catching sparse behaviors. The Cox model shows to meet the benchmark with comparably fit.

dense Poisson fit is represented by the dotted red line. The sparse Cox model presents an MSE very similar to the dense Poisson, showing that the case-control sampling in the risk set does not deteriorate the fit significantly and hence the partial likelihood correctly channels the information necessary for inference.

This case study underlines the relationship between the two types of sparsity mentioned in Section (4.5.1). Sparsity, whereas in the parameters or in the data, results in a partial recovery of the true dynamics. By increasing the sparsity we have a worsening as more spline basis parameters never leave the starting point at 0.

Vary clusters vicinity We conclude by showing the clustering accuracy as the scale of the synthetic latent space reduces toward 0, letting nodes become closer. In Figure 4.4 we show that the proposed method correctly allocates nodes as long as the space is sufficiently separable. The first point indeed shows perfect classification. The more the nodes are closer, the more the individual node variance becomes influential, and the harder the model discriminates between different clusters.

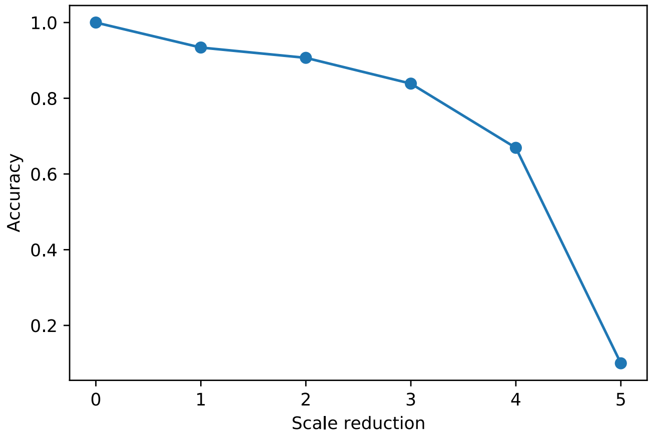


Figure 4.4. Clustering performance. Perfect allocation for separable clusters. As shrinkage increases, clusters are put closer, letting higher chances for node trajectories to overlap. Clustering accuracy deteriorates as a consequence.

## 4.9 Data analysis: Wikipedia editing network

Wikipedia's editing history consists of the history of all the editing events of articles by the editors since the foundation of Wikipedia in 2001. This massive bimodal event history data set includes approximately 361 million links, 6.7 million editors, 5.5 million articles, and hence 40 trillion of possible dyadic interactions. The focus is modeling the latent drivers that might explain user editing behavior. Lerner and Lomi [2020] successfully fitted a Cox proportional hazard model, where endogenous effects such as repetition, two-step reciprocity, individual strength, and assortativity are taken under consideration. Their model includes a total of 5 parameters. In this manuscript, we propose a more complex form of endogenous effect, the dynamic latent space, where we fit several millions of parameters. The model we propose is the following

$$\log \lambda_{ij}(t) = -\|z_i(t) - z_j\|_2^2 + \text{propensity}_i + \text{propensity}_j \tag{4.16}$$

that describes a Euclidean latent space where user $i$ and article $j$ have a subjective propensity of editing and being edited respectively. We chose to keep articles static in time. This improves interpretation as the space becomes a *latent topic space* where editors move when they approach new articles. The interpretation of such space is powerful as certain regions correspond to topics that have a certain degree of relationship, i.e., the similarity induced by the heterogeneity of editors' backgrounds.

The use of the propensity random effect, the Euclidian distance, and the static article is justified by a substantial improvement in the model fit. Without these assumptions the model places the editors and articles into two separate clouds, with minimal dynamics. We filtered the data as most of the editors modify a few articles only at the beginning of their subscription. We retain editors that have at least 15 interactions. To characterize the topic space is sufficient to keep the most popular articles, edited at least 100 times. The overall network contains 209.737.058 links, 706.820 articles, and 572.586 editors for a total of 1.279.406 nodes.

In the analysis three patterns can be identified among the editor trajectories: (1) independent editors with trajectories that explore a wide range of articles, see Figure 4.6, (2) active editors who edited several articles with a possibly curved trajectory, see Figure 4.5, and (3) temporary editors that entered, interacted, and exited using a straight trajectory.

Independent editors These editors are likely highly experienced or specialized in various areas, as they edit a wide range of articles. They are difficult for others to replicate their latent patterns. These editors do not commonly belong to any cluster. They may be considered experts within their domain, and their contributions to Wikipedia may be highly valuable due to their depth of knowledge and expertise. They are shown mainly in Figure 4.6 although a few examples are successfully clustered in Figure 4.5.

Active editors These editors enter or leave the cloud of articles with a possibly curved trajectory, see Figure 4.5. They may be more casual or novice contributors who are focused on a specific set of articles. They may have a lower level of expertise than the editors with independent trajectories and may not engage with as many articles, but they can contribute valuable edits and improvements to the articles they interact with.

Temporary editors These editors enter, interact, and exit using straight trajectories. Some examples are shown in Figure 4.5 and more in detail in Figure 4.7. These editors are considered snapshot editors. They make interactions only in a very short period of time. The arrow's distance, which is approximately two years, highlights how fast these nodes move in space. The proposed framework naturally models these trajectories as straight lines as no data supports a possible curvature outside the interaction interval.

Figure 4.5 and Figure 4.6 present clusters and outliers respectively. These are obtained from the optimal radius selected by our method. The optimal radius hence captures a mixture of three behaviors. Alternatively, the use of a sub-optimal radius can focus on one single behavior. Figure 4.7 shows the clustering for a larger radius, capable of capturing the snapshot editors only. The usage of different radii, although sub-optimal from the model formulation perspective, can hence be informative. The fact that we need to use multiple radii to identify different behaviors in the trajectories may be due to the complexity and diversity of the data, as the editors' trajectories exhibit a wide range of behaviors and patterns.

## 4.10 Conclusions

The main contribution of this manuscript is the development of an efficient inference scheme for latent dynamic processes underlying a relational event process. The framework is general and can be extended to networks with weighted edges of any exponential family distribution, making it a useful tool for analyzing a wide range of data.

One key aspect of the model is the use of smooth spline functions to capture the latent trajectories of nodes in dynamic networks. This allows for a more accurate representation of the underlying dynamics than traditional static models. The model also employs a smoothness penalty for regulating the smoothness of the spline and a clustering penalty for detecting shared trajectories among the nodes. This makes the model more interpretable and allows for the identification of patterns and behaviors within the network. The model can be run within the detected clusters and fit a nested latent space, which allows for revealing different levels of granularity of the relationships.

Another important aspect of the model is its scalability. The optimization is performed by the popular *Adam* algorithm, which is not memory intensive and is very fast in computation. It can optimize nearly any function and learn the Hessian via the past gradients' history. This allows the model to handle large networks with millions of nodes/parameters, which is going to be a common problem for future network analysis. Additionally, particular care has been given to handling sparse data and sparse parameter updates, which makes the model more robust.

The inference is conducted via Variational Bayes, finding an effective approximation of the posterior distribution for the complete set of parameters, including smoothness magnitude and clustering shrinkage. Under the Variational formulation, the inference problem translates into a classic optimization problem, finding *Adam* as a good ally.

This manuscript includes a simulation study that confirms the claims made in the manuscript, showing that the model behaves correctly in scenarios such as sparsity in the data, sparsity in the parameter update, clustering accuracy, and consistency of the location estimator.

We applied the model to the Wikipedia complete edited page history. Differently to Lerner and Lomi [2020], which analyzed this data with a 5-parameter model, our latent space model successfully fitted several millions of parameters. The application of the model to the Wikipedia data revealed various shared behaviors that are coherent with natural expectations. For example, some editors consistently modify Wikipedia pages over time, while others are more temporary editors. This differentiation between experts and non-experts shows that the model correctly identifies important behaviors in the Wikipedia editing patterns, which could help to understand the dynamics of the Wikipedia community and improve the quality of the articles.

Overall, the proposed model provides a powerful and interpretable tool for analyzing the dynamics of networks and can help reveal the underlying patterns and behaviors of the nodes. The interpretability of the results makes it a valuable tool for understanding the underlying dynamics and making predictions about future behavior. This can be useful for a wide range of applications such as social network analysis, recommender systems, and biological networks. Given the popularity of a latent space representation in various emerging fields, possible extensions for our model include financial time series analysis, moving object detection, language generation, and translation.

## 4.11 Supplementary material

### 4.11.1 A mini-batch cluster penalty for finite $\gamma_{\text{dist}}$

When we construct the mini-batch $B$, the chance of randomly sampling two close is almost zero for large networks. As a result, when $w_{ij}$ has a relatively small radius only a few elements are included in the kernel or, in the case of a continuous kernel, have a sufficiently high weight. Therefore the vast majority of elements in the mini-batch are excluded. As a consequence, the level of sparsity for the gradient with respect to $c$ is even higher than for the splines. In order to make the gradient dense we propose to save the history of pairs that entered in the kernel at the previous iterations, then randomly sample $B^*$ in this set, where $\mid B^* \mid = p$. The resulting mini-batch penalty is

$$P^B_{\text{clust}} = \gamma_{\text{aux}} \sum_{i=1}^{p} ||\alpha_i - c_i||^2 + \gamma_{\text{dist}} \sum_{i,j \in B^*} w_{ij} ||c_i - c_j||^2. \tag{4.17}$$

which has complexity linear in $p$. The mini-batch $B^*$ is sampled over the history of pairs for which $w_{ij}$ is positive. Notice that, similarly to the smoothness penalty, the full-time sequence of the sampled nodes is included.

## 4.11.2 A discrete-time model for sparse data

We use this model formulation for the specific case when there exists dense connectivity within and sparse connectivity between communities. Hence one benefits from aggregating the data. We employ a non-stratified case-control formulation of the Poisson likelihood 4.6. In order to obtain an unbiased estimate of the intercept, the likelihood term for the non-events needs to be overweighted by $\frac{N_0}{n_0}$

$$\ell(\alpha) = \sum_{y_{t,ij}>0} \left[ -\lambda_{i,j}(t)\Delta t + y_{i,j}(t) \log \lambda_{i,j}(t)\Delta t \right] + \frac{N_0}{n_0} \sum_{y_{t,ij}=0} -\lambda_{i,j}(t)\Delta t. \tag{4.18}$$

Alternatively, the intercept absorbs the bias leading to the correct latent node positions.

## 4.11.3 Variational inference details

In the Monte Carlo Variational approach, some expectations can be solved analytically, leaving the Monte Carlo integration for those who are intractable. Besides the non-tractable log-likelihood component, the remaining expectations can be solved as

$$\mathbb{E}[P_{\text{smooth}}] = \frac{pd(k-1)}{2} \mathbb{E}[\log \gamma_{\text{smooth}}] - \mathbb{E}[\gamma_{\text{smooth}}] \sum_{i=1}^{p} \sum_{k=2}^{m} \mathbb{E}[\left\| \alpha_{i,k} - \alpha_{i,k-1} \right\|^2]$$

$$\mathbb{E}[P_{\text{clust}}] = \frac{pdk}{2} \mathbb{E}[\log \gamma_{\text{clust}}] - \mathbb{E}[\gamma_{\text{clust}}] \sum_{i=1}^{p} \mathbb{E}[||\alpha_i - c_i||^2]$$

where $(\log \gamma_{\text{smooth}}, \log \gamma_{\text{clust}})$ are Gaussian densities with log-normal expectations $\mathbb{E}[\gamma_{\text{smooth}}] = e^{\mu_{\text{smooth}} + 0.5\sigma^2_{\text{smooth}}}$ and $\mathbb{E}[\gamma_{\text{clust}}] = e^{\mu_{\text{clust}} + 0.5\sigma^2_{\text{clust}}}$. The expectations of the normalizing constants are $\mathbb{E}[\log \gamma_{\text{smooth}}] = \mu_{\text{smooth}}$ and $\mathbb{E}[\log \gamma_{\text{clust}}] = \mu_{\text{clust}}$. The other expectations are simply functions of the first two moments $\mathbb{E}[\alpha_i] = \mu_i$ and $\mathbb{E}[\alpha_i^2] = \sigma_i^2 + \mu_i^2$. The centroids $c_i$ can be safely held to be constant for various reasons. The first is that $c_i$ is an averaging between several trajectories and hence its variance must be negligible compared to $\alpha_i$. The second is that $\mathbb{E}[\|\alpha_i - c_i\|^2]$ is minimized by taking $c_i$ as degenerate, or nearly degenerate since the prior would prevent the estimation of degenerate random variables. The last remaining parameter $\gamma_{\text{radius}}$ cannot be updated by gradient, however, is particularly easy to find a grid of candidate points from visual inspection of the latent space. We then select $\gamma_{\text{radius}}$ that maximizes the lower-bound.

The final component, $D_{\text{KL}}[q(\theta)\|p(\theta)]$ can also find close form as done in the appendix of Kingma and Welling [2013]

$$D_{\text{KL}}[q(\theta)\|p(\theta)] = -\frac{1}{2}\sum_{i=1}^{p+2} 1 + \log\frac{\sigma_i^2}{\sigma_0^2} - \frac{\sigma_i^2}{\sigma_0^2} - \frac{(\mu_i - \mu_0)^2}{\sigma_0^2}$$

where $\mu_0, \sigma_0^2$ are respectively the mean and variance of a Gaussian prior.

When running the Variational inference, we might have the $\gamma_{\text{clust}}$ estimate being misleading for the case when both the number of clusters and the number of links are low. This is because the penalty might become such big that it is the major contributor to the lower-bound. The model hence prioritizes the minimization of $\mathbb{E}[\|\alpha_i - c_i\|^2]$ collapsing all trajectories and making $\gamma_{\text{clust}}$ unreasonably big. The lower-bound still makes a correct clustering selection although some estimates of $\gamma_{\text{clust}}$ might not be coherent with the expectations, i.e. expecting $\gamma_{\text{clust}}$ low for a low number of clusters. This behavior is paired with a substantial worsening in the likelihood, which reflects the introduction of the bias.

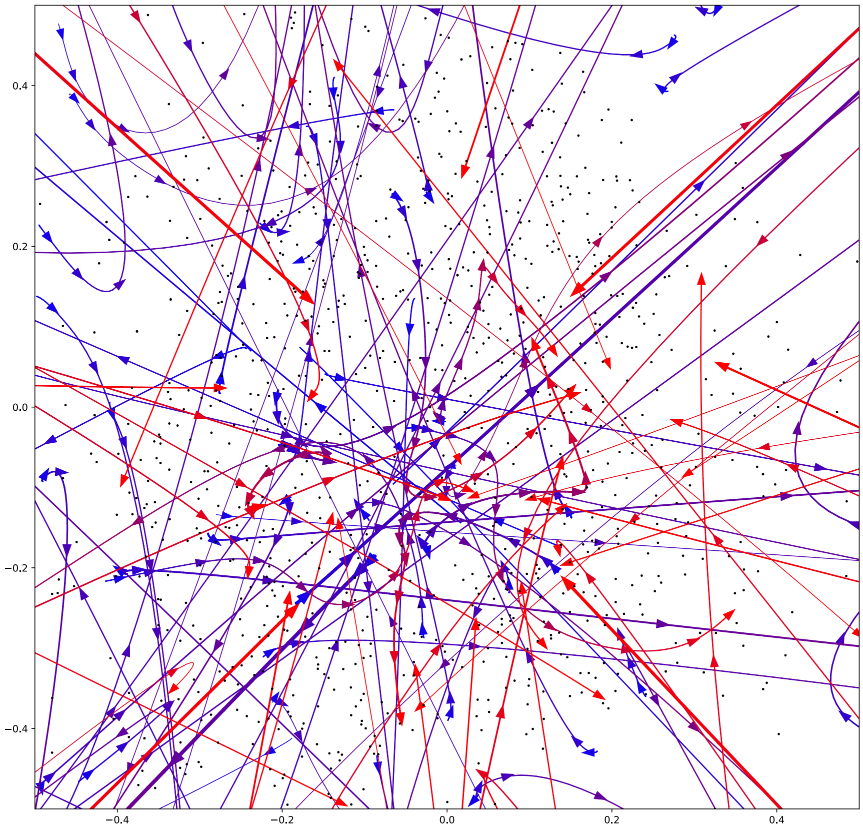


Figure 4.5. Clustering trajectories of editors. The trajectory size is proportional to the cluster population size. The largest trajectories contain approximately 3000 editors. The color progression from blue to red corresponds to the observed time. This figure presents both active editors and temporary editors. Active editors edit articles for a sustained period, highlighted by the color progression, which typically presents a curved trajectory. They are characterized by a mid-size background and an important contribution to the articles. Alternatively, temporary editors make fewer interactions, have a little background, and move faster with typically a straight line.

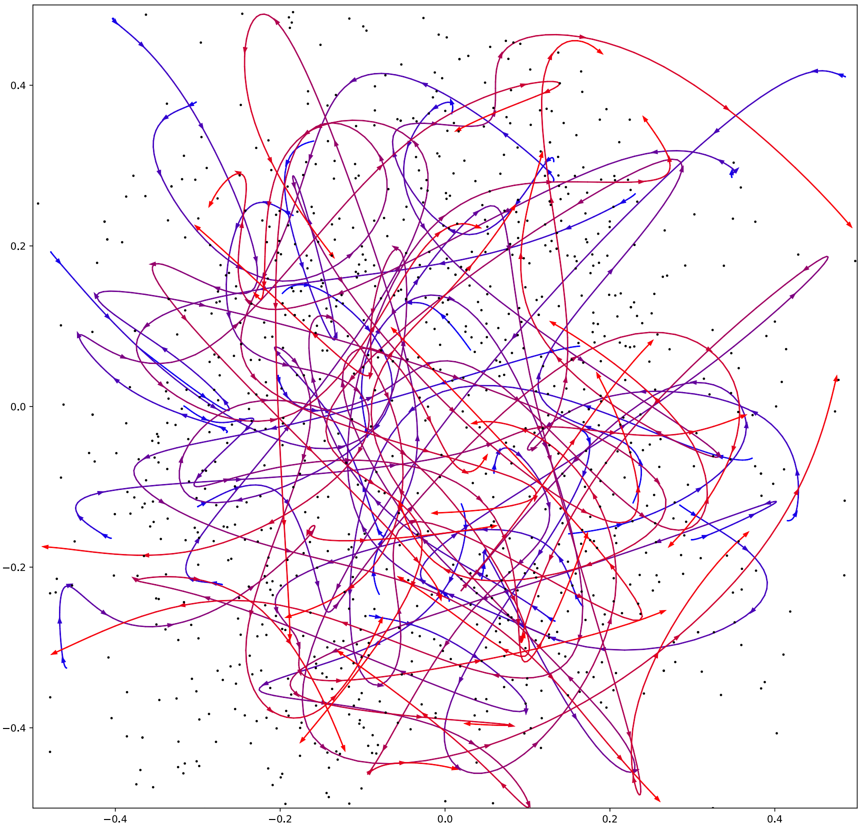


Figure 4.6. Independent trajectories: these editors do not have a cluster belonging. Their trajectory represents an independent behavior backed by strong expertise in their competence area. This figure presents only a subset of independent editors. We selected those with high centrality by filtering trajectories within [-0.5, 0.5]. The color progression from blue to red corresponds to the observed time.

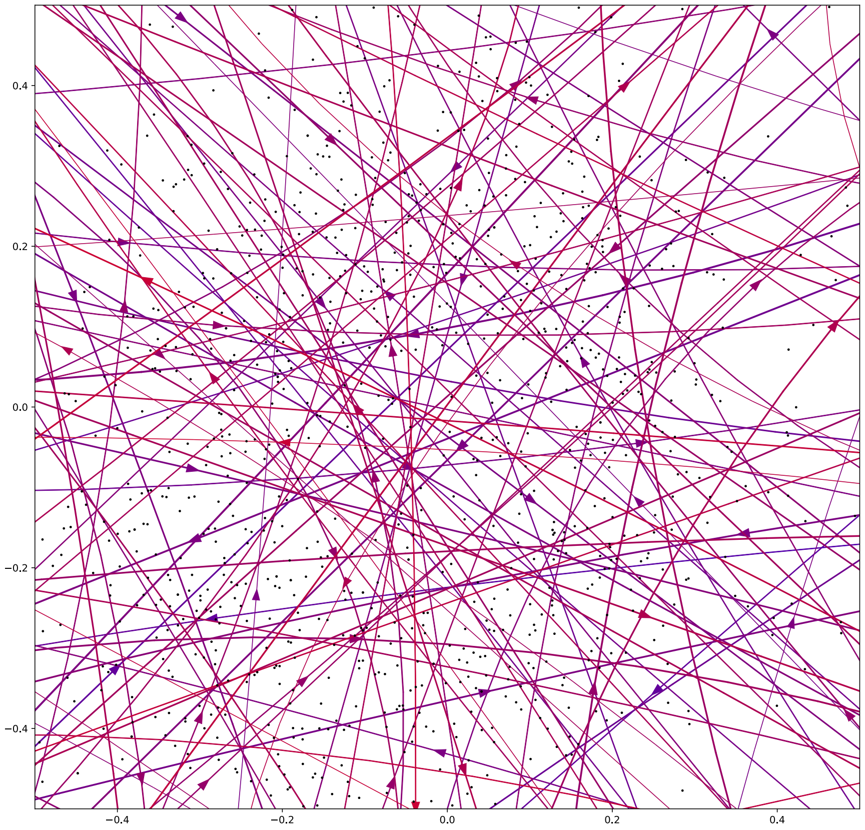


Figure 4.7. Snapshot editors. In-and-out editors are active for a short period of time. Their competence area is little as they focus on a few articles or topics. They typically have a straight trajectory. This figure presents the largest distance between the arrowheads, which is approximately two years. This implies these editors are the fastest movers observed.

# Chapter 5

# Conclusions

This thesis is focused on exploring the dynamics of evolving networks through a statistical approach. Networks are used to represent complex systems that change and evolve over time, such as social networks, transportation systems, and the Internet. Understanding the dynamics of networks has become an important research topic in fields ranging from physics to sociology to computer science.

The thesis is built upon three key projects, each focusing on a different aspect of network dynamics. The first project involves testing the power-law marginal distributions of growing networks and delves into the complexities of degree distribution testing in networks. The second project explores latent dynamics in networks by developing a dynamic latent space relational event model. The third project proposes an extension for analyzing huge networks.

Through these projects, the thesis shows the power of statistical techniques in uncovering the latent drivers of the evolution of networks.

Power-law marginal distributions of networks: This project proposes a statistical testing procedure to determine whether the degree distribution of a given network follows a power-law distribution, a result of a preferential attachment process. We modify the Kolmogorov-Smirnov test to account for dependent degree sequences and ensure sufficient power. They apply this method to many empirical degree distributions and find that almost 65% of the tested networks have a power-law tail with at least 80% power. This work contributes to the ongoing discussion around the putative scale-free nature of real-world networks and the existence of an underlying "network law." Its preferential attachment process has ethical implications, such as the meritocracy of research publications.

Latent dynamics in networks: This project focuses on dynamic networks where the relational events constitute time-stamped edges. The authors propose a dynamic latent space relational event model, where nodes are associated with dynamic locations and their relative distances drive their interaction tendencies. The goal is to infer the locations of the nodes, which can change over time, using the Expectation Maximization algorithm and an extension of the universal Kalman filter. We also include fixed and random effects in their model to suit a large variety of applications. This work is significant because it offers an efficient method for modeling dynamic networks that reflect underlying dynamics in some latent space.

Latent dynamics for large networks: This project addresses the challenge of dealing with huge relational event networks. We propose a likelihood-based algorithm that infers network community dynamics embedded into an interpretable latent space. The node dynamics are described by smooth spline processes, and the framework is made feasible for large networks through machine learning optimization methodology. We use a convex clustering penalization for model-based clustering to encourage shared trajectories for ease of interpretation. This additionally aims to separate macro-microstructures and perform a hierarchical analysis within successive hierarchies. This work is significant because it offers a practical and efficient method for dealing with large-scale dynamic networks, which are increasingly prevalent in many social network applications.

The three projects under discussion delve into the complex phenomenon of network growth and transformation, shedding light on the role of latent drivers that shape the structure of observed networks. While the first project focuses on specific types of drivers for growing networks, the second and third projects delve into the latent drivers that produce any kind of transformation in the observed network.

The observed network and its marginal distribution are the visible outcomes of underlying latent drivers, which are not always directly observable. In the case of the power-law, which characterizes many real-world networks, the preferential attachment process is the latent popularity of nodes that grows over time.

Often, since the latent drivers are unknown, we can only observe the network's final configuration, which is the outcome of an evolutionary process driven by latent drivers. The first project makes a hypothesis on the latent drivers, specifically the preferential attachment process, and tests for it in the network marginal distribution. Other times networks are observed as time-stamp data and the observed sequence might be informative on the hidden drivers. Focusing on this data type, the second and third projects do not impose any specific constraint on the latent driver's structure and allow the model to estimate these drivers.

Thus, the latent approach allows for a deeper investigation of the hidden drivers that produce a particular observed network. By studying the underlying drivers, analysts can better understand how networks grow and transform, and, ultimately, how they impact various domains, from social networks to biological systems.

# Bibliography

Albert, R. and Barabási, A.-L. [2002]. Statistical mechanics of complex networks, *Reviews of modern physics* **74**(1): 47.

Albert, R., Jeong, H. and Barabási, A.-L. [1999]. Internet: Diameter of the world-wide web, *nature* **401**(6749): 130.

Amaral, L. A. N., Scala, A., Barthelemy, M. and Stanley, H. E. [2000]. Classes of small-world networks, *Proceedings of the national academy of sciences* **97**(21): 11149–11152.

Anderson, B. D. and Moore, J. B. [2012]. *Optimal filtering*, Courier Corporation.

Anderson, T. W. and Darling, D. A. [1954]. A test of goodness of fit, *Journal of the American statistical association* **49**(268): 765–769.

Artico, I., Smolyarenko, I., Vinciotti, V. and Wit, E. C. [2020]. How rare are power-law networks really?, *Proceedings of the Royal Society A* **476**(2241): 20190742.

Artico, I. and Wit, E. [2023a]. Fast inference of latent space dynamics in huge relational event networks, *arXiv preprint arXiv:2303.17460* .

Artico, I. and Wit, E. C. [2022]. Dynamic latent space relational event model, *arXiv preprint arXiv:2204.04753* .

Artico, I. and Wit, E. C. [2023b]. Dynamic latent space relational event model, *Journal of the Royal Statistical Society Series A: Statistics in Society* . qnad042.
**URL:** *https://doi.org/10.1093/jrsssa/qnad042*

Barabási, A. [2018]. Love is all you need: Clauset's fruitless search for scale-free networks, *Blog post available at https://www. barabasilab. com/post/love-is-all-you-need* .

Barabási, A.-L. and Oltvai, Z. N. [2004]. Network biology: understanding the cell's functional organization, *Nature reviews genetics* **5**(2): 101.

Blei, D. M., Kucukelbir, A. and McAuliffe, J. D. [2017]. Variational inference: A review for statisticians, *Journal of the American statistical Association* **112**(518): 859–877.

Bloom, H. S. [1995]. Minimum detectable effects: A simple way to report the statistical power of experimental designs, *Evaluation review* **19**(5): 547–556.

Borgan, O., Goldstein, L. and Langholz, B. [1995]. Methods for the analysis of sampled cohort data in the cox proportional hazards model, *The Annals of Statistics* pp. 1749–1778.

Bourdieu, P. [1989]. Social space and symbolic power, *Sociological theory* **7**(1): 14–25.

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

Broido, A. D. and Clauset, A. [2019]. Scale-free networks are rare, *Nature communications* **10**(1): 1017.

Bscan [2013]. Own work, cc0, https://commons.wikimedia.org/w/index.php?curid=25222928.

Butts, C. T. [2008]. 4. a relational event framework for social action, *Sociological Methodology* **38**(1): 155–200.

Chen, G. K., Chi, E. C., Ranola, J. M. O. and Lange, K. [2015]. Convex clustering: An attractive alternative to hierarchical clustering, *PLoS computational biology* **11**(5): e1004228.

Chicheportiche, R. and Bouchaud, J.-P. [2011]. Goodness-of-fit tests with dependent observations, *Journal of Statistical Mechanics: Theory and Experiment* **2011**(09): P09003.

Chicheportiche, R. and Bouchaud, J.-P. [2012]. Weighted Kolmogorov-Smirnov test: Accounting for the tails, *Physical Review E* **86**(4): 041115.

Clauset, A., Shalizi, C. R. and Newman, M. E. [2009]. Power-law distributions in empirical data, *SIAM review* **51**(4): 661–703.

Cook, S. and Soramaki, K. [2014]. The global network of payment flows. [Online; No. 2012-006 (September 23, 2014).].
**URL:** *https://ssrn.com/abstract=2503774*

Cox, D. R. [1972]. Regression models and life-tables, *Journal of the Royal Statistical Society: Series B (Methodological)* **34**(2): 187–202.

Crane, H. [2018]. *Probabilistic foundations of statistical network analysis*, Chapman and Hall/CRC.

Crowder, M. J. [1976]. Maximum likelihood estimation for dependent observations, *Journal of the Royal Statistical Society: Series B (Methodological)* **38**(1): 45–53.

De Solla Price, D. J. [1965]. Networks of scientific papers, *Science* pp. 510–515.

De Vos, S., Wardenaar, K. J., Bos, E. H., Wit, E. C., Bouwmans, M. E. and De Jonge, P. [2017]. An investigation of emotion dynamics in major depressive disorder patients and healthy persons using sparse longitudinal networks, *PLoS One* **12**(6): e0178586.

Deijfen, M., Van Den Esker, H., Van Der Hofstad, R. and Hooghiemstra, G. [2009]. A preferential attachment model with random initial degrees, *Arkiv för matematik* **47**(1): 41–72.

Dempster, A. P., Laird, N. M. and Rubin, D. B. [1977]. Maximum likelihood from incomplete data via the em algorithm, *Journal of the Royal Statistical Society: Series B (Methodological)* **39**(1): 1–22.

DuBois, C., Butts, C. and Smyth, P. [2013]. Stochastic blockmodeling of relational event dynamics, *Artificial Intelligence and Statistics*, pp. 238–246.

Duchi, J., Hazan, E. and Singer, Y. [2011]. Adaptive subgradient methods for online learning and stochastic optimization., *Journal of machine learning research* **12**(7).

Durante, D. and Dunson, D. B. [2016]. Locally adaptive dynamic networks, *The Annals of Applied Statistics* **10**(4): 2203–2232.

Efron, B. [1992]. Bootstrap methods: another look at the jackknife, *Breakthroughs in statistics*, Springer, pp. 569–593.

Fahrmeir, L. [1992]. Posterior mode estimation by extended kalman filtering for multivariate dynamic generalized linear models, *Journal of the American Statistical Association* **87**(418): 501–509.

Fahrmeir, L. [1994]. Dynamic modelling and penalized likelihood estimation for discrete time survival data, *Biometrika* **81**(2): 317–330.

Fahrmeir, L. and Kaufmann, H. [1991]. On kalman filtering, posterior mode estimation and fisher scoring in dynamic exponential family regression, *Metrika* **38**(1): 37–60.

Faloutsos, M., Faloutsos, P. and Faloutsos, C. [1999]. On power-law relationships of the internet topology, *ACM SIGCOMM computer communication review* **29**(4): 251–262.

Feigelson, E. and Babu, G. J. [2013]. Beware the Kolmogorov-Smirnov test!, https://asaip.psu.edu/Articles/beware-the-kolmogorov-smirnov-test.

Fitzgerald, R. [1971]. Divergence of the kalman filter, *IEEE Transactions on Automatic Control* **16**(6): 736–747.

Gamerman, D. [1991]. Dynamic bayesian models for survival data, *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **40**(1): 63–79.

Gamerman, D. [1992]. A dynamic approach to the statistical analysis of point processes, *Biometrika* **79**(1): 39–50.

Gao, F. and van der Vaart, A. [2017]. On the asymptotic normality of estimating the affine preferential attachment network models with random initial degrees, *Stochastic Processes and their Applications* **127**(11): 3754–3775.

Gay, D. M. [1990]. Usage summary for selected optimization routines, *Computing science technical report* **153**: 1–21.

Goldman, M. and Kaplan, D. M. [2016]. Evenly sensitive KS-type inference on distributions, *Technical report*, Working paper, available at http://faculty. missouri. edu/˜ kaplandm.

Handcock, M. S., Raftery, A. E. and Tantrum, J. M. [2007]. Model-based clustering for social networks, *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **170**(2): 301–354.

Hanneke, S., Fu, W. and Xing, E. P. [2010]. Discrete temporal models of social networks, *Electronic journal of statistics* **4**: 585–605.

Hocking, T. D., Joulin, A., Bach, F. and Vert, J.-P. [2011]. Clusterpath an algorithm for clustering using convex fusion penalties, *28th international conference on machine learning*, p. 1.

Hoff, P. [2008]. Modeling homophily and stochastic equivalence in symmetric relational data, *Advances in neural information processing systems*, pp. 657–664.

Hoff, P. D. [2005]. Bilinear mixed-effects models for dyadic data, *Journal of the american Statistical association* **100**(469): 286–295.

Hoff, P. D. [2009]. Multiplicative latent factor models for description and prediction of social networks, *Computational and mathematical organization theory* **15**(4): 261.

Hoff, P. D., Raftery, A. E. and Handcock, M. S. [2002]. Latent space approaches to social network analysis, *Journal of the american Statistical association* **97**(460): 1090–1098.

Hoffman, M. D., Blei, D. M., Wang, C. and Paisley, J. [2013]. Stochastic variational inference, *Journal of Machine Learning Research* .

Holme, P. [2019]. Rare and everywhere: Perspectives on scale-free networks, *Nature communications* **10**(1): 1016.

Jeong, H., Mason, S. P., Barabási, A.-L. and Oltvai, Z. N. [2001]. Lethality and centrality in protein networks, *Nature* **411**(6833): 41.

Julier, S. J. and Uhlmann, J. K. [1997]. New extension of the Kalman filter to nonlinear systems, *in* I. Kadar (ed.), *Signal Processing, Sensor Fusion, and Target Recognition VI*, Vol. 3068, International Society for Optics and Photonics, SPIE, pp. 182 – 193.
**URL:** *https://doi.org/10.1117/12.280797*

Julier, S. and Uhlmann, J. K. [1996]. A general method for approximating nonlinear transformations of probability distributions, *Technical report*, Department of Engineering Science, University of Oxford.

Kalman, R. E. [1960]. A new approach to linear filtering and prediction problems, *Journal of Basic Engineering* **82**(1): 35–45.

Khanin, R. and Wit, E. [2006]. How scale-free are biological networks, *Journal of computational biology* **13**(3): 810–818.

Kingma, D. P. and Ba, J. [2014]. Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* .

Kingma, D. P. and Welling, M. [2013]. Auto-encoding variational bayes, *arXiv preprint arXiv:1312.6114* .

Kolaczyk, E. D. and Csárdi, G. [2014]. *Statistical analysis of network data with R*, Vol. 65, Springer.

Kolmogorov, A. [1933]. Sulla determinazione empirica di una legge di distribuzione, *Inst. Ital. Attuari, Giorn.* **4**: 83–91.

Krapivsky, P. L. and Redner, S. [2001]. Organization of growing random networks, *Physical Review E* **63**(6): 066123.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A. and Blei, D. M. [2017]. Automatic differentiation variational inference, *Journal of machine learning research* .

Lafond, F. and Kim, D. [2019]. Long-run dynamics of the us patent classification system, *Journal of Evolutionary Economics* **29**(2): 631–664.

Laherrere, J. and Sornette, D. [1998]. Stretched exponential distributions in nature and economy:“fat tails” with characteristic scales, *The European Physical Journal B-Condensed Matter and Complex Systems* **2**(4): 525–539.

Lerner, J. and Lomi, A. [2020]. Reliability of relational event model estimates under sampling: How to fit a relational event model to 360 million dyadic events, *Network science* **8**(1): 97–135.

Lima-Mendez, G. and van Helden, J. [2009]. The powerful law of the power law and other myths in network biology, *Molecular BioSystems* **5**(12): 1482–1493.

Lindsten, F., Ohlsson, H. and Ljung, L. [2011]. Clustering using sum-of-norms regularization: With application to particle filter output computation, *2011 IEEE Statistical Signal Processing Workshop (SSP)*, IEEE, pp. 201–204.

Mandel, J. [2006]. *Efficient implementation of the ensemble Kalman filter*, University of Colorado at Denver and Health Sciences Center, Center for Computational Mathematics.

Mandt, S., Hoffman, M. D. and Blei, D. M. [2017]. Stochastic gradient descent as approximate bayesian inference, *arXiv preprint arXiv:1704.04289* .

Mansuy, R. and Yor, M. [2008]. *Aspects of Brownian motion*, Springer Science & Business Media.

McCullagh, P. [2018]. *Generalized linear models*, Routledge.

Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P. and Bhattacharjee, B. [2007]. Measurement and analysis of online social networks, *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, ACM, pp. 29–42.

Mitzenmacher, M. [2004]. A brief history of generative models for power law and lognormal distributions, *Internet mathematics* **1**(2): 226–251.

Newman, M. E. [2002]. Spread of epidemic disease on networks, *Physical review E* **66**(1): 016128.

Newman, M. E. [2003]. The structure and function of complex networks, *SIAM review* **45**(2): 167–256.

Newman, M. E. [2005]. Power laws, pareto distributions and zipf's law, *Contemporary physics* **46**(5): 323–351.

Newman, M. E. and Girvan, M. [2004]. Finding and evaluating community structure in networks, *Physical review E* **69**(2): 026113.

Pelckmans, K., De Brabanter, J., Suykens, J. A. and De Moor, B. [2005]. Convex clustering shrinkage, *PASCAL workshop on statistics and optimization of clustering workshop*.

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.

Popper, K. [1962]. *Conjectures and refutations: The growth of scientific knowledge*, routledge.

Raftery, A. E., Niu, X., Hoff, P. D. and Yeung, K. Y. [2012]. Fast inference for the latent space network model using a case-control approximate likelihood, *Journal of computational and graphical statistics* **21**(4): 901–919.

Rastelli, R. and Corneli, M. [2021]. Continuous latent position models for instantaneous interactions, *arXiv preprint arXiv:2103.17146* .

Rastelli, R., Maire, F. and Friel, N. [2018]. Computationally efficient inference for latent position network models, *arXiv preprint arXiv:1804.02274* .

Redner, S. [1998]. How popular is your paper? an empirical study of the citation distribution, *The European Physical Journal B-Condensed Matter and Complex Systems* **4**(2): 131–134.

Ruder, S. [2016]. An overview of gradient descent optimization algorithms, *arXiv preprint arXiv:1609.04747* .

Saefken, B., Kneib, T., van Waveren, C.-S. and Greven, S. [2014]. A unifying approach to the estimation of the conditional akaike information in generalized linear mixed models, *Electronic Journal of Statistics* **8**(1): 201–225.

Sahoo, P. [2013]. Probability and mathematical statistics, *University of Louisville* .

Sarkar, P. and Moore, A. W. [2005]. Dynamic social network analysis using latent space models, *Acm Sigkdd Explorations Newsletter* **7**(2): 31–40.

Särkkä, S. and García-Fernández, Á. F. [2020]. Temporal parallelization of bayesian smoothers, *IEEE Transactions on Automatic Control* **66**(1): 299–306.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P. and Xu, X. [2017]. Dbscan revisited, revisited: why and how you should (still) use dbscan, *ACM Transactions on Database Systems (TODS)* **42**(3): 1–21.

Sewell, D. K. and Chen, Y. [2015]. Latent space models for dynamic networks, *Journal of the American Statistical Association* **110**(512): 1646–1657.

Sewell, D. K. and Chen, Y. [2016]. Latent space models for dynamic networks with weighted edges, *Social Networks* **44**: 105–116.

Shalizi, C. R. and Asta, D. [2017]. Consistency of maximum likelihood for continuous-space network models, *arXiv preprint arXiv:1711.02123* .

Signorelli, M., Vinciotti, V. and Wit, E. C. [2016]. Neat: an efficient network enrichment analysis test, *BMC bioinformatics* **17**(1): 1–17.

Signorelli, M. and Wit, E. C. [2018]. A penalized inference approach to stochastic block modelling of community structure in the italian parliament, *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **67**(2): 355–369.

Smolyarenko, I. [2019]. Are network degree distributions observable?, *Proceedings of the 8th International Conference on Complex Networks and their Applications*, Lisbon, Portugal.

Snijders, T. A. and Pickup, M. [2017]. Stochastic actor-oriented models for network dynamics, *Annual review of statistics and its application* **4**(1).

Sun, D., Toh, K.-C. and Yuan, Y. [2021]. Convex clustering: Model, theoretical guarantee and efficient algorithm., *J. Mach. Learn. Res.* **22**(9): 1–32.

Thorndike, R. L. [1953]. Who belongs in the family?, *Psychometrika* **18**(4): 267–276.

Tranmer, M., Marcum, C. S., Morton, F. B., Croft, D. P. and de Kort, S. R. [2015]. Using the relational event model (rem) to investigate the temporal dynamics of animal social networks, *Animal behaviour* **101**: 99–105.

Užupytė, R. and Wit, E. C. [2020]. Test for triadic closure and triadic protection in temporal relational event data, *Social Network Analysis and Mining* **10**(1): 1–12.

Van der Hofstad, R. [2016]. *Random graphs and complex networks*, Vol. 1, Cambridge university press.

Voitalov, I., van der Hoorn, P., van der Hofstad, R. and Krioukov, D. [2019]. Scale-free networks well done, *Physical Review Research* **1**(3): 033034.

Vu, D., Lomi, A., Mascia, D. and Pallotti, F. [2017]. Relational event models for longitudinal network data with an application to interhospital patient transfers, *Statistics in medicine* **36**(14): 2265–2287.

Vu, D., Pattison, P. and Robins, G. [2015]. Relational event models for social learning in moocs, *Social Networks* **43**: 121–135.

Wang, T. and Resnick, S. I. [2020]. Degree growth rates and index estimation in a directed preferential attachment model, *Stochastic Processes and Their Applications* **130**(2): 878–906.

Watson, M. W. and Engle, R. F. [1983]. Alternative algorithms for the estimation of dynamic factor, mimic and varying coefficient regression models, *Journal of Econometrics* **23**(3): 385–400.

West, M., Harrison, P. J. and Migon, H. S. [1985]. Dynamic generalized linear models and bayesian forecasting, *Journal of the American Statistical Association* **80**(389): 73–83.

Weylandt, M., Nagorski, J. and Allen, G. I. [2020]. Dynamic visualization and fast computation for convex clustering via algorithmic regularization, *Journal of Computational and Graphical Statistics* **29**(1): 87–96.

Wheeler, N. R., Xu, Y., Gok, A., Kidd, I. V., Bruckman, L. S., Sun, J. and French, R. H. [2014]. Data science study protocols for investigating lifetime and degradation of pv technology systems, *IEEE PVSC*, Vol. 40, Citeseer.

Wood, S. N. [2006]. *Generalized additive models: an introduction with R*, chapman and hall/CRC.

Younge, K. A. and Kuhn, J. M. [2016]. Patent-to-patent similarity: A vector space model, *Available at SSRN 2709238* .

Zhang, L., Carpenter, B., Gelman, A. and Vehtari, A. [2021]. Pathfinder: Parallel quasi-newton variational inference, *arXiv preprint arXiv:2108.03782* .