MCMC Methods: Graph Samplers, Invariance Tests and Epidemic Models

James Alexander Scott

Department of Mathematics
Imperial College London
180 Queen's Gate
London SW7 2AZ

This dissertation is submitted for the degree of
Doctor of Philosophy of Imperial College London
and the Diploma of Imperial College London

August 2023

I would like to dedicate this thesis to my loving parents, Aileen and David, who have supported me throughout this endeavour.

Declaration

I certify that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

James Alexander Scott
August 2023

© The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Acknowledgements

First, I would like to thank my supervisor, Professor Axel Gandy, for his unwavering support throughout this process. Axel is an exemplary supervisor who invariably found time to discuss our research. His supervisions were both fun and fruitful, and led me to develop a deeper passion for my field, and to progress as a researcher. His encouragement and enthusiasm for my work made the entire process enjoyable, and an experience that I will find difficult to leave behind.

I am indebted to both Professor Pierre Jacob and Professor Nitesh Pillai, who kindly hosted me in the Statistics department at Harvard University for several months during the PhD. Being involved with Pierre's research group during that time was invaluable to me, as I became familiar with interesting topics that I was otherwise unlikely to have had the opportunity to research.

Most of all, this thesis would not have been completed without the love and support of my closest friends and family. My parents have been supportive and patient throughout. My partner, Ira, has been a pillar of support, and I decided to pursue this path largely due to her encouragement. I will be forever grateful as in hindsight it was the right decision.

Abstract

Markov Chain Monte Carlo (MCMC) techniques are used ubiquitously for simulation-based inference. This thesis provides novel contributions to MCMC methods and their application to graph sampling and epidemic modeling. The first topic considered is that of sampling graphs conditional on a set of prescribed statistics, which is a difficult problem arising naturally in many fields: sociology (Holland and Leinhardt, 1981), psychology (Connor and Simberloff, 1979), categorical data analysis (Agresti, 1992) and finance (Squartini et al., 2018, Gandy and Veraart, 2019) being examples. Bespoke MCMC samplers are proposed for this setting. The second major topic addressed is that of modeling the dynamics of infectious diseases, where MCMC is leveraged as the general inference engine.

The first part of this thesis addresses important problems such as the uniform sampling of graphs with given degree sequences, and weighted graphs with given strength sequences. These distributions are frequently used for exact tests on social networks and two-way contingency tables. Another application is quantifying the statistical significance of patterns observed in real networks. This is crucial for understanding whether such patterns indicate the presence of interesting network phenomena, or whether they simply result from less interesting processes, such as nodal-heterogeneity. The MCMC samplers developed in the course of this research are complex, and there is great scope for conceptual, analytic, and implementation errors. This motivates a chapter that develops novel tests for detecting errors in MCMC implementations. The tests introduced are unique in being exact, which allows us to keep the false rejection probability arbitrarily low.

Rather than develop bespoke samplers, as in the first part of the thesis, the second part leverages a standard MCMC framework **Stan** (Stan Development Team, 2018) as the workhorse for fitting state-of-the-art epidemic models. We present a general framework for semi-mechanistic Bayesian modeling of infectious diseases using renewal processes. The term semi-mechanistic relates to statistical estimation within some constrained

mechanism. This research was motivated by the ongoing SARS-COV-2 pandemic, and variants of the model have been used in specific analyses of Covid-19. We present **epidemia**, an R package allowing researchers to leverage the epidemic models. A key goal of this work is to demonstrate that MCMC, and in particular, Stan's No-U-Turn (Hoffman and Gelman, 2014) sampler, can be routinely employed to fit a large-class of epidemic models. A second goal is to make the models accessible to the general research community, through **epidemia**.

Table of contents

1 Introduction

1.1 Preamble

This thesis employs Markov Chain Monte Carlo (MCMC) to tackle problems appearing in both graph sampling and epidemic modeling. Our primary objective is to extend the class of models to which MCMC can be leveraged as an effective inference engine. The graph sampling problems addressed are known to be challenging. Our contribution is to develop bespoke algorithms which can be used for efficient inference in this setting. In the context of epidemic modeling, we do not develop new MCMC methods, rather, the goal is to show that a broad class of state-of-the-art epidemic models can be fit routinely with MCMC. We introduce a framework for Bayesian, regression-oriented models for infectious disease dynamics, and an R-package allowing routine specification and fitting of these models.

A secondary objective of our work is to test, statistically, if an MCMC implementation has a given invariant distribution. This task is motivated by the MCMC algorithms developed in Part I, which rely on involved derivations and numerical implementations. We believe that such tests should be a routine part of research that uses MCMC - and hope that our methods contribute towards this. Chapter 4 addresses this topic.

The structure of the thesis is as follows. Part I develops MCMC algorithms for sampling graphs conditional on a model's sufficient statistics. This general problem appears in a number of distinct fields. Section 1.1.1 motivates the task and highlights several applications of the samplers that will be presented in Chapters 2 and 3. As mentioned, Chapter 4 addresses the challenge of testing whether a given MCMC sampler indeed has a desired invariant distribution. This research was motivated by our experience developing the graphs sampling algorithms, and also the observation that much current statistical research leverages samplers that have been either poorly or informally tested.

Part II moves away from the development of new MCMC samplers, and towards building a general modeling framework that leverages MCMC, and in particular Stan's (Stan Development Team, 2018) implementation of the No-U-Turn sampler (Hoffman and Gelman, 2014), as the underlying inference engine. More concretely, this part of

the thesis presents a general framework for Bayesian, regression-oriented modeling of infectious diseases using renewal processes. During the early stages of the SARS-CoV-2 pandemic, there was a critical need for high quality statistical models that were capable of inferring transmission rates, the effects of mitigation efforts, and of forecasting. Part II presents the modeling framework designed to address these challenges and culminates in an R-package that allows quick and flexible implementation of these models. We motivate this line of research in Section 1.1.2.

1.1.1 Part I: MCMC for Conditional Graph Sampling

Sampling graphs conditional on a set of prescribed statistics is a difficult problem arising in many fields, including sociology (Holland and Leinhardt, 1981), psychology (Connor and Simberloff, 1979), categorical data analysis (Agresti, 1992) and finance (Squartini et al., 2018, Gandy and Veraart, 2019). The first part of this thesis is dedicated to constructing new MCMC samplers for this setting. The samplers are designed to tackle the unique challenges posed by such problems, and are broadly applicable to the fields listed above.

Consider the example of testing the goodness-of-fit of statistical network models. Holland and Leinhardt (1981) introduced a class of log-linear models, known as p_1-models, for modeling social networks. These attempt to explain differing nodal sociability and popularity, and also the phenomenon of reciprocity in networks, which has been observed repeatedly in the context of social networks. Letting x denote the adjacency matrix of a directed graph, the likelihood has an exponential family form

$$\log P(x; \alpha, \beta, \rho) \propto \sum_i \alpha_i x_{i\cdot} + \sum_j \beta_j x_{\cdot j} + \rho \sum_{i<j} x_{ij} x_{ji}, \tag{1.1}$$

where α_k and β_k determine the sociability and popularity of the k^{th} node respectively, while the parameter ρ measures the intensity of link reciprocation. We can test the goodness-of-fit of the submodel with no tendency for link reciprocation through testing the hypothesis that $\rho = 0$. The uniformly most powerful unbiased (UMPU) test for this considers the conditional distribution of $T(x) := \sum_{i<j} x_{ij} x_{ji}$ given the sufficient statistics, which are the node degrees $x_{k\cdot}$ and $x_{\cdot k}$.

Sampling from this distribution entails simulating adjacency matrices uniformly from the set of all such matrices with the same row and column marginals as observed in the data. These samples can then be used to perform a Monte Carlo test. More generally, ρ and $T(x)$ could be replaced by another parameter δ and sufficient statistic $Z(x)$, and one could test whether $Z(x)$ is observed more frequently than expected under the null model which conditions on the node degrees. This is a common approach for detecting network motifs (Milo et al., 2002).

Similar problems involving sampling of matrices with prescribed row and column marginals show up in categorical data analysis, where the technique is used to test for

independence between categorical random variables (Diaconis and Sturmfels, 1998). This sampling problem is difficult, and MCMC methods are generally not scalable to sparse tables, and often require tools from algebraic statistics to ensure irreducibility of the samplers. There is a need for faster mixing and more flexible methods to tackle this problem, and Chapter 2 takes a step in this direction.

The goodness-of-fit problem described above can be extended to weighted graphs. A natural extension of (1.1) to weighted and directed graphs includes both degrees and strengths

$$\log P(x;\theta) \propto \sum_i \alpha_i a_{i\cdot} + \sum_i \beta_j a_{\cdot j} + \sum_i \phi_i x_{i\cdot} + \sum_j \psi_j x_{\cdot j} + \delta T(x),$$

where x is the graph's weight matrix, and $a := (x^0_{ij})$ is its adjacency matrix, where the convention $0^0 := 0$, is used. The parameter vector θ simply collects all parameters on the right-hand side. This model is employed in Mastrandrea et al. (2014a), where it is used to reconstruct networks from node-level data. The UMPU test of $\delta = 0$ proceeds similarly to before, but conditions on both node degrees *and* strengths. Chapter 3 develops approaches for sampling from this conditional distribution, and extends the work in Chapter 2 to the case of graphs with real-valued edge weights.

Although not pursued in this thesis, the sampler presented in Chapter 3 can also be used for *network tomography* (Castro et al., 2004, Tebaldi and West, 1998, Squartini et al., 2018). In this setting, the network of interest is not fully observed, and instead only aggregate statistics are available. The task is to *reconstruct* networks consistent with the set of observed statistics. In what follows, we discuss the specific case of reconstructing economic and financial networks.

Economists and policymakers have long been interested in estimating the risk of contagion in financial networks (see Gai and Kapadia, 2010, Haldane and May, 2011, Staum et al., 2016, Elliott et al., 2014, Acemoglu et al., 2013, Glasserman and Young, 2016). It is natural to model such networks as a graph, with institutions being nodes, and edges between nodes representing the type of exposure. These exposures could be interbank lending, derivative exposures or equity cross-holdings, each of which represents its own contagion channel. It is well known that the structure of links in these networks has a large effect on how and if shocks propagate, and ultimately on the level of systemic risk in the network.

In practice, however, these networks are unobserved, even to central banks. Instead, only *aggregated network data* is available. For example, in the interbank setting aggregate exposures will be available through public balance sheet information while bilateral exposures between institutions are not observed. This corresponds to only observing node-level data, rather than the graph edges themselves. The question then arises of how to reconstruct the missing links. This is known to be a difficult problem, and there is a need for principled methods to recover such networks from partial information.

Formally, assume we observe k statistics $T_i(G) = t_i$ from the unknown graph G. Previous approaches to network reconstruction use *exponential random graph models* (Park and Newman, 2004, Squartini et al., 2017). These methods assume that G is drawn from an ensemble Ω according to some unknown law P. This distribution is estimated by maximizing Shannon entropy whilst satisfying the observed data $t_1, ..., t_k$ in expectation. This is a constrained maximization problem of the functional

$$\mathcal{L}(P) = -\mathrm{E}(\log(P(G)) - \sum_{i=0}^{k} \lambda_i(\mathrm{E}(T_i(G)) - t_i), \tag{1.2}$$

with $T_0 = 1$ and $t_0 = 1$. The solution to (1.2) then has exponential family form

$$P(G; \lambda) = \frac{1}{Z(\lambda)} \exp\left(-\sum_{i=1}^{k} \lambda_i T_i(G)\right). \tag{1.3}$$

This approach is equivalent to assuming the parametric family (1.3), and estimating parameters λ_i using MLE's $\hat{\lambda}_i$ based on one sample. Networks can then be simulated from $P(\cdot; \hat{\lambda})$. Such approaches fail to *condition* on the observed data $t_1, ..., t_k$. It is reasonable to expect inference to place zero measure on graphs not satisfying this data. In addition, most methods assign deterministic weights to edges once the topology has been constructed. This has obvious and serious consequences for estimating systemic risk.

The above problem can alternatively be tackled using Bayesian methods. For example, Gandy and Veraart (2016) propose a block Gibbs sampler for a special case of this problem. The sampler in Chapter 3 can be seen as a generalization of this Gibbs sampler.

1.1.2 Part II: Semi-Mechanistic Modeling of Infectious Disease Dynamics

The emergence of SARS-CoV-2 triggered extensive research into statistical models that are capable of providing insights on the temporal dynamics of the disease. Examples include quantifying transmission rates and the effects of control measures, such as lockdown. Models have also been used extensively to forecast the evolution of key count time series including latent infections, hospitalizations and deaths (see https://covid19forecasthub.org/ and https://covid19forecasthub.eu/). In order to address these inferential tasks, a number of articles have used a Bayesian approach that explicitly models disease dynamics; in particular, employing self-exciting processes to propagate infections over time (Flaxman et al., 2020a, Vollmer et al., 2020, Mellan et al., 2020, Unwin et al., 2020, NYS Press Office, 2020, Olney et al., 2021, The Scottish Government, 2020, Mishra et al., 2020b). At the time of writing, models of this sort continue to be popular and used to inform time-critical policy decisions. Part II of this thesis presents a general version of these models and motivates them through continuous-time counting

processes. In particular, we discuss a number of important model extensions. This discussion paves the way for presenting what is the highlight of Part II: a novel R package **epidemia** for specifying and fitting these models.

Models of infectious disease dynamics are commonly classified as either mechanistic or statistical (Myers et al., 2000). Mechanistic models derive infection dynamics from theoretical considerations over how diseases spread within and between communities. An example of this are deterministic compartmental models (DCMs) (Kermack et al., 1927, Kermack, 1932, 1933), which propose differential equations that govern the change in infections over time. These equations are motivated by contacts between individuals in susceptible and infected classes. Purely statistical models, on the other hand, make few assumptions over the transmission mechanism, and instead infer future dynamics from the history of the process and related covariates. Examples include Generalized Linear Models (GLMs), time series approaches including Auto Regressive Integrated Moving Average (ARIMA) (Box and Jenkins, 1962), and more modern forecasting methods based on machine learning.

The models in Part II cannot be classified as exclusively mechanistic or statistical. Instead, they are often termed *semi-mechanistic*. They are fully *Bayesian*, i.e. unknown quantities are assigned priors and inferred through the posterior. They are *regression-oriented* and flexibly parameterize key unknown epidemiological quantities in terms of covariates and autocorrelation processes. These three features are the key defining properties of the class of models studied in Part II. Therefore, we finish this section by discussing each of them in more depth.

- *semi-mechanistic*: This term refers to statistical models that explicitly describe infection dynamics. The models are *statistical* in the sense that they define a likelihood function for the observed data. They are *mechanistic* because self-exciting processes are used to propagate infections in discrete time. Previous infections directly precipitate new infections. Moreover, the memory kernel of the process allows an individual's infectiousness to depend explicitly on the time since infection. This approach has been used in multiple previous works (Fraser, 2007, Cori et al., 2013, Nouvellet et al., 2018, Cauchemez et al., 2008) and has been shown to correspond to a Susceptible-Exposed-Infected-Recovered (SEIR) model when a particular form of the generation distribution is used (Champredon et al., 2018).

- *Bayesian*: The Bayesian approach has certain advantages in this context. Several aspects of these models are fundamentally unidentified (Roosa and Chowell, 2019). For most diseases, infection counts are not fully observable and suffer from under-reporting (Gibbons et al., 2014). Recorded counts could be explained by a high infection and low ascertainment regime, or alternatively by low infections and high ascertainment. If a series of mitigation efforts are applied in sequence to control an

epidemic, then the effects may be confounded and difficult to disentangle (Bhatt et al., 2020). Bayesian approaches using MCMC allow full exploration of posterior correlations between such coupled parameters. Informative, or weakly informative, priors may be incorporated to regularize, and help to mitigate identifiability problems, which may otherwise pose difficulties for sampling (Gelman et al., 2008, 2015).

- *Regression-oriented*: The models use a flexible regression-based framework for parameterizing transmission and ascertainment rates. This allows the models to be tailored towards the specific inferential task. For example, transmission rates can be inferred by parameterizing them as a random walk. Alternatively, including covariates allows estimating the effect of control measures (Cowling et al., 2020, Flaxman et al., 2020a) or mobility (Badr et al., 2020, Miller et al., 2020) on transmission rates. Multilevel models (Gelman and Hill, 2006, Hox et al., 2010, Kreft and de Leeuw, 2011) can be employed to better infer these effects by leveraging information from multiple regions simultaneously. Ascertainment rates such as the infection ascertainment rate (IAR) and infection fatality rate (IFR) are important unknown quantities that link the infection process to observed data. Most modeling approaches assume either full ascertainment (Cori et al., 2013) or constant rates (Flaxman et al., 2020a). In practice, however, these rates are spatio-temporally dependent, and flexible parameterization of them allows for more realistic observational models.

1.2 Chapter Summaries

We outline the structure of the thesis, and the contents of each chapter. Part I consists of three chapters, the first two of which develop new MCMC samplers for conditional graph sampling. This challenge was briefly motivated in Section 1.1.1. The samplers developed in these chapters are complex, often requiring detailed mathematical derivations and proofs to justify their theoretical properties, including for example irreducibility and ergodicity of the chains. This motivates a general question: how can we empirically test the validity of a given implementation of a MCMC sampler? Or more generally, the validity of Monte Carlo methods? These questions are considered in the final chapter of Part I. We now briefly outline the contents of each of these three chapters.

Chapter 2 proposes new and efficient algorithms for two problems: sampling conditional on node degrees in unweighted graphs, and conditional on node strengths in integer-weighted graphs. The resulting conditional distributions provide the basis for exact tests on social networks and two-way contingency tables. The algorithms are able to sample conditional on the presence or absence of an arbitrary set of edges. Existing samplers based on MCMC or sequential importance sampling are generally not scalable; their efficiency can degrade in large graphs with complex patterns of known edges. MCMC

methods usually require explicit computation of a Markov basis to navigate the state space; this is computationally intensive even for small graphs. The samplers presented in this chapter do not require a Markov basis, and are efficient both in sparse and dense settings. The key idea is to carefully select a Markov kernel on the basis of the current state of the chain. We demonstrate the utility of our methods on a real network and contingency table.

Chapter 3 extends the work in Chapter 2 to the case where networks have real-valued edge weights, and may also exhibit sparsity. A method is proposed to sample from the set of weighted graphs exactly conditional on given node strengths, and approximately conditional on both degrees and strengths. This conditioning reduces the influence of nuisance parameters in the resulting distribution. Strengths can be maintained exactly, and degrees ± 1 of their observed values. The ergodicity of the sampler is considered, and proved for the case of conditioning only on strengths. The chapter applies the algorithm to evaluate the statistical significance of network patterns such as community structure. The sampler can also be used for reconstructing financial networks (see Section 1.1.1).

Chapter 4 is the final chapter of the first part of this thesis, and develops approaches for testing implementations of MCMC methods as well as of general Monte Carlo methods. Based on statistical hypothesis tests, these approaches can be used in a unit testing framework to, for example, check if individual steps in a Gibbs sampler or a reversible jump MCMC have the desired invariant distribution. Two exact tests for assessing whether a given Markov chain has a specified invariant distribution are discussed. These and other tests of Monte Carlo methods can be embedded into a sequential method that allows low expected effort if the simulation shows the desired behavior and high power if it does not. Moreover, the false rejection probability can be kept arbitrarily low. For general Monte Carlo methods, this allows testing, for example, if a sampler has a specified distribution or if a sampler produces samples with the desired mean. The methods have been implemented in the R-package **mcunit**.

Part II of this thesis tackles the second topic described in the preamble, which is the statistical modeling of infectious disease dynamics. This consists of two chapters, the first of which describes a framework for a class of models that have been used extensively throughout the SARS-CoV-2 pandemic. The second introduces an R-package that allows users to flexibly specify, fit, and analyze these models using MCMC. Both chapters are described in more detail below.

Chapter 5 introduces the mathematical framework behind the epidemic models, and in particular discusses advantages and limitations over competing approaches. The model discussed grew out of specific analyses conducted during the pandemic, in particular an analysis concerning the effects of mitigation measures on reducing SARS-CoV-2 transmission in 11 European countries. It parameterizes the time varying reproduction number R_t through a regression framework in which covariates can be, for example, governmental interventions or changes in mobility patterns. This allows a joint fit across

regions and partial pooling to share strength. The framework provides a fully generative model for latent infections and observations deriving from them, including deaths, cases, hospitalizations, ICU admissions and seroprevalence surveys.

Chapter 6 introduces **epidemia**, an R package implementing the modeling framework of Chapter 5. The implemented models define a likelihood for all observed data while also explicitly modeling transmission dynamics: an approach often termed as *semi-mechanistic*. Infections are propagated over time using renewal equations. This approach is inspired by self-exciting, continuous-time point processes such as the Hawkes process. A variety of inferential tasks can be performed using the package. Key epidemiological quantities, including reproduction numbers and latent infections, may be estimated within the framework. The models may be used to evaluate the determinants of changes in transmission rates, including the effects of control measures. Epidemic dynamics may be simulated either from a fitted model or a "prior" model; allowing for prior/posterior predictive checks, experimentation, and forecasting.

The thesis is concluded in Chapter 7, where we in particular discuss possible directions for future research.

1.3 List of Publications

The contents of the thesis are available as preprints, and have either been published or submitted for publication:

- Scott, J. and Gandy, A., (2020), "State-Dependent Kernel Selection for Conditional Sampling of Graphs," *Journal of Computational and Graphical Statistics*, 29, 847-858.
  doi: 10.1080/10618600.2020.1753529

- Scott, J. and Gandy, A., (2021), "Approximate Conditional Sampling for Pattern Detection in Weighted Networks," *arXiv:2109.09104 [stat.ME]*, submitted.

- Gandy, A. and Scott, J., (2021), "Exact Tests of MCMC and Other Monte Carlo Methods," *arXiv:2001.06465 [stat.ME]*, submitted.

- Bhatt, S., Ferguson, N., Flaxman, S., Gandy, A., Mishra, S., Scott, J., (2021), "Semi-Mechanistic Bayesian Modeling of SARS-CoV-2 with Renewal Processes," *arXiv:2012.00394 [stat.AP]*, accepted by the *Journal of the Royal Statistical Society: Series A* as a discussion paper.

- Scott, J. , Gandy, A., Swapnil, M., Unwin, J., Flaxman, S., Bhatt, S., Ish-Horowicz, J. (2021), "Epidemia: An R Package for Semi-Mechanistic Bayesian Modeling of Infectious Diseases using Point Processes," *arXiv:2110.12461 [stat.CO]*, submitted.

PART I

The first part of this thesis has two objectives. The first is to develop new MCMC samplers for particular problems involving conditional graph sampling. Chapter 2 considers this in the context of unweighted and integer-weighted graphs, while Chapter 3 extends to the case of graphs with real-valued edge weights and potential sparsity. Motivated by these two chapters, the final chapter considers the second objective, which is developing tests for whether an MCMC implementation has a given invariant distribution.

2 State-Dependent Kernel Selection for Conditional Sampling of Graphs

2.1 Introduction

Inference on graphs conditional on vertex-level data arises in sociology (Holland and Leinhardt, 1981), psychology (Rasch, 1960), community ecology (Connor and Simberloff, 1979) and categorical data analysis (Agresti, 1992). Testing in this setting can be based on asymptotic results. However, these approximations can be poor in sparse graphs. An alternative approach is to use sampling to approximate the distribution of test statistics. This leads to two difficult problems: sampling graphs with given degrees and sampling weighted graphs with given strengths. Researchers often additionally need to condition on the presence or absence of certain edges in the graphs.

Several existing methods construct Markov chains in this setting. Unfortunately, if the null distribution conditions on known edges, it is difficult to construct a connected Markov chain on the relevant state space. Existing methods either specialize to particular patterns of known edges, or in the general case, use techniques from computational algebra to compute a Markov basis (Diaconis and Sturmfels, 1998, Aoki and Takemura, 2005, Rapallo, 2006). These methods are computationally intensive and are impractical for graphs with more than a few vertices.

We propose new MCMC methods that use *state-dependent mixing* of Markov kernels. The idea is to intelligently select a 'good' kernel for the current state of the chain. This technique allows us to construct samplers that require little tuning to the problem at hand, and do not require computation of a Markov basis. The samplers are irreducible in the face of arbitrary patterns of known edges, and are efficient both in sparse and dense graphs.

The first focus of this chapter is on sampling unweighted graphs with prescribed vertex degrees. This problem arises in carrying out exact tests. Consider, for example, social network analysis. A social network equipped with a dichotomous relation can be expressed as a digraph. Vertices represent actors, with edges representing the applicability of the

relation between actors. Frequently, researchers are interested in testing the presence of reciprocity in the network; defined loosely as a preference for mutual dyads in the digraph. Holland and Leinhardt (1981) introduce an exponential family model under which the UMPU test for reciprocity conditions on the observed degree sequences. Conditioning removes nuisance parameters from the null, and the resulting distribution is then uniform on the reference set.

The complex interactions that result from conditioning render analytic analysis of the null distribution difficult or impossible. Efforts have been made to develop recursive formulas to enumerate all graphs in the reference set (Wasserman and Faust, 1994), however these are impractical for even moderately sized graphs. If we can sample graphs uniformly, then we can approximate the null distribution of a test statistic. Thus, the literature has focused on simulation, whose methods can broadly be divided into two camps; Markov Chain Monte Carlo (MCMC) (Rao et al., 1996, Roberts, 2000, Milo et al., 2002, McDonald et al., 2007, Verhelst, 2008) and sequential importance sampling (SIS) (Snijders, 1991, Zhang and Chen, 2013, Chen et al., 2005, Bayati et al., 2010).

Sampling binary tables with given margins is equivalent to sampling undirected bipartite graphs with given vertex degrees. This is applied in community ecology to test for patterns in co-occurrence tables, and in psychometrics to test the Rasch hypothesis (see Gustafsson, 1980). Thus, there exists a substantial parallel literature along these lines.

Most MCMC algorithms proposed for sampling graphs are adaptations of methods proposed for zero-one tables. Typically, they use a combination of 'switch' moves (Ryser, 1963) and additional moves to maintain irreducibility in the face of structural zeros. Rao et al. (1996) and McDonald et al. (2007) consider 'compact alternating hexagon' and 'hexad' updates respectively. Most proposed methods suffer from poor mixing in unbalanced matrices, rendering them impractical for moderate to large graphs. Additionally, they are not extensible to arbitrary known edges. SIS methods build the graph sequentially, at each iteration choosing a candidate edge with probability proportional to the vertex degrees. Early methods for this application include (Snijders, 1991, Chen et al., 2005). Most of these samplers get stuck, and the probability of restarting approaches 1 as the degree sequences grow. Bezáková et al. (2012) provide examples where such algorithms are slow. More recent methods avoid the issue of restarting and often come with better theoretical guarantees (Bayati et al., 2010, Blitzstein and Diaconis, 2011, Zhang and Chen, 2013). Our approach to this problem is to construct an MCMC sampler using a symmetric decomposition of Markov kernels; this is a concept defined in Section 2.2.2.

The second focus of this chapter is on sampling integer-weighted graphs with prescribed vertex strengths. This can be used to conduct network tomography in the case of a star network topology. However, the motivating application is approximating the null distribution for evaluating exact tests on two-way contingency tables. This is a

classical problem in statistics, which is important because standard asymptotics justifying approximate tests (notably Pearson's χ^2 test of independence) do not hold for tables with cells with low expected frequencies (see Agresti, 2013).

In conditional tests of independence, one is interested in sampling tables with given margins. This corresponds to sampling integer-weighted bipartite graphs conditional on vertex strengths. Diaconis and Sturmfels (1998) proposed a simple 'switch' Markov chain to sample from such tables. We describe this in more detail in Section 2.2. It suffers slow mixing in sparse tables.

Diaconis and Sturmfels (1998) also proposed an algebraic algorithm to construct a connected Markov chain in the context of incomplete tables. Other MCMC methods proposed to sample incomplete tables also rely on computing a Markov basis (Aoki and Takemura, 2005, Rapallo, 2006). The computational cost of this is exponential in the size of the table. Additionally, the computation is example-specific; i.e., a new basis must be computed for each pattern of structural zeros considered. Chen et al. (2005) introduced the first SIS method for uniform sampling of contingency tables with given marginals. Chen (2007) extended this to incomplete tables. Eisinger and Chen (2017) develop a sampler with improved efficiency, particularly in sparse graphs. We propose an auxiliary variable MCMC sampler which overcomes many of the aforementioned limitations and compare the sampler to SIS approaches in Section 2.6.

This chapter begins by setting notation in Section 2.1.1. Section 2.2 introduces state-dependent kernel selection, and presents practical strategies for ensuring chains using this technique have the correct invariant distribution. Sections 2.3 and 2.4 propose samplers in the unweighted and weighted graph settings respectively. We present a detailed simulation study in Section 2.5, and apply our samplers to real data in Section 2.6. Finally, we conclude in Section 2.7. All proofs can be found in the appendix.

2.1.1 Notation

An undirected graph $G := (V, E)$ is a pair with V being a labelled vertex set and E a collection of distinct unordered pairs of vertices. If the graph is directed, then E consists of distinct ordered vertex pairs. An integer-weighted graph is a triple $G := (V, E, c)$. The function $c\colon V \times V \to \mathbb{N}_0$ assigns a non-zero weight to each $uv \in E$, and 0 to each $uv \notin E$. If the context requires clarification, we use $V(G)$, $E(G)$ and c_G to denote the objects belonging to G.

The in- and out-degrees of a vertex are the number of edges to and from the vertex respectively. The in- and out-strengths of a vertex of a weighted graph are the total weight of edges to and from the vertex respectively. If the graph is undirected, there is no distinction between in and out, so we simply use the terms degree and strength of a vertex. Two undirected graphs with the same vertex set have the same degree (strength) sequence if every vertex has the same degree (strength) in each graph. We use the same

terminology for directed graphs, where both the in and out values must be equal for every vertex.

2.2 State-Dependent Kernel Selection

Before discussing state-dependent kernel selection in general terms, we give a concrete example. Let $r := (r_1, ..., r_I)$ and $c := (c_1, ..., c_J)$ be non-negative integer vectors, and let $\mathcal{X}$ denote the set of all $I \times J$ non-negative integer matrices such that the row and column marginals equal r and c respectively. Assume $\mathcal{X}$ is non-empty. The task is to construct a Markov chain ergodic with respect to the uniform distribution on $\mathcal{X}$. Diaconis and Sturmfels (1998) describe a simple Markov chain for this purpose. Given X_n, pick a pair of rows and a pair of columns uniformly at random. The chain proceeds by sampling from the conditional distribution of the delineated subtable given all other entries. An update takes the form

$$\begin{matrix} +\Delta & -\Delta \\ -\Delta & +\Delta \end{matrix}$$

for Δ sampled uniformly from integers which do not induce negative values in the subtable.

A Markov chain on $\mathcal{X}$ is completely characterized by its kernel Q, a regular conditional distribution, where $Q(x, A) := P[X_{n+1} \in A \mid X_n = x]$ for A measurable. In this example Q can be viewed as randomly selecting from a set of other kernels. Indeed, let $\mathcal{Z}$ be the collection of indices of all 2×2 sub-arrays of $I \times J$ tables. The Gibbs update on each $z \in \mathcal{Z}$ defines a kernel K_z on $(\mathcal{X}, \mathcal{B})$, with $\mathcal{B}$ being the Borel algebra. A scan order is a method of choosing a particular kernel from this collection. The aforementioned chain is an example of *random scan*, where kernels are chosen irrespective of the current state. The chain's kernel is $Q := \sum_z K_z/|\mathcal{Z}|$.

The chain suffers poor mixing in sparse matrices, as Δ is often degenerate at 0. As we will see, we can use a state-dependent scan order to improve mixing whilst maintaining ergodicity.

2.2.1 General Setup

State-dependent kernel selection can be defined in general terms as follows. Let $(\mathcal{Z}, \mathcal{F})$ and $(\mathcal{X}, \mathcal{B})$ be Borel spaces. $\mathcal{X}$ is the state space and $\mathcal{Z}$ is the index set of $K := \{K_z \colon z \in \mathcal{Z}\}$, a collection of kernels on $(\mathcal{X}, \mathcal{B})$. We assume throughout that the map $(z, x) \mapsto K_z(x, B)$ is jointly measurable for each B. The kernel selection mechanism is defined via a set $w := \{w_x : x \in \mathcal{X}\}$ where each w_x is a probability measure on $\mathcal{F}$ and the map $x \mapsto w_x(F)$ is measurable for each F. A set satisfying these requirements is often referred to as a *probability kernel* from $(\mathcal{X}, \mathcal{B})$ to $(\mathcal{Z}, \mathcal{F})$. If the current state is x, the chain proceeds to sample a kernel K_z according to w_x, and then samples the next state from $K_z(x, \cdot)$. The

kernel of this chain is defined through

$$Q(x,\cdot) := \int K_z(x,\cdot) w_x(\mathrm{d}z) \text{ for all } x \in \mathcal{X}. \tag{2.1}$$

If (2.1) holds, then we call (K, w) a decomposition of the kernel Q. The decomposition of a kernel is not necessarily unique. Any kernel Q has an 'identity' decomposition, given by $(\{K_1\}, w)$ with $K_1 = Q$ and $w_x(\{1\}) = 1$ for all $x \in \mathcal{X}$. In Sections 2.2.2 and 2.2.3 we give techniques for constructing kernels with a desired invariant distribution π using decompositions. These strategies are then used to develop the samplers in Sections 2.3 and 2.4 respectively.

2.2.2 Using a Symmetric Decomposition

A decomposition (K, w) where each K_z is π-reversible does not imply that Q is π-reversible. One notable exception to this is when w is the random scan order, where each w_x is the uniform distribution on $\mathcal{Z}$. We now define a class of decompositions, which we call symmetric decompositions, for which the resulting Q will be π-reversible. Loosely speaking, it requires that if a state x' is reachable in one step from a state x via a kernel K_z then the likelihood that K_z is selected from state x is the same as in state x'.

Definition 2.2.1 (Symmetric Decomposition)**.** *A decomposition (K, w) is symmetric if there exist a σ-finite measure μ and for every x densities $f_x = \mathrm{d}w_x/\mathrm{d}\mu$ such that for each z and x, $f_x(z) = f_{x'}(z)$ for $K_z(x,\cdot)$-almost every x'.*

Any state-independent kernel selection is symmetric: for example, random scan and the 'identity' decomposition. As an example of a non-trivial decomposition, consider a three-state state space, as depicted in Figure 2.1. The left figure defines three kernels on this space. A naive chain might pick from these uniformly, irrespective of the current state. However, if the chain is in state i, then K_i cannot change the state. A faster mixing chain Q randomly selects between the other two kernels so that the state changes. This (state-dependent) strategy has a symmetric decomposition. Each of the three kernels shown in Figure 2.1 (a) is reversible with respect to the uniform distribution, and by Lemma 2.2.2 so is Q.

Lemma 2.2.2. *Q is π-reversible if it has a symmetric decomposition (K, w) where every $K_z \in K$ is π-reversible.*

The reverse of Lemma 2.2.2 holds trivially through the identity decomposition. This method is used to support the validity of the sampler developed in Section 2.3.

2.2.3 Kernel Selection as an Auxiliary Variable

Here, we present an alternative way of constructing a π-invariant chain. The technique described is used in Section 2.4. Suppose we have a set of statistics $\{T_z : z \in \mathcal{Z}\}$ on

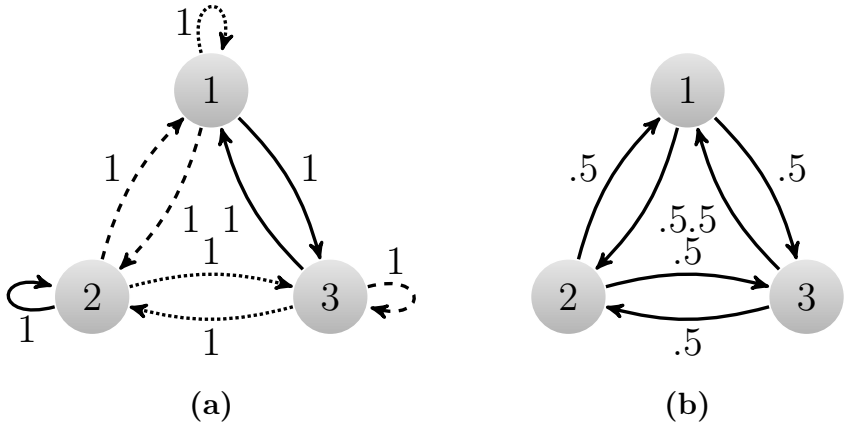


Fig. 2.1 Depiction of the kernel decomposition outlined in Section 2.2.2. (a) shows transition probabilities for K_1 (dotted), K_2 (solid) and K_3 (dashed). (b) shows transitions probabilities for Q.

$\mathcal{X}$. Defining a selection law w, we can construct a chain whose updates keep part of the state fixed. At each iteration, this chain proceeds by selecting a feature T_z and changing the current state using some kernel that is π-invariant and keeps T_z fixed. If w does not depend on x this can be viewed as a Gibbs sampler with 'generalized' conditioning statistics. Intuitively, however, a state-dependent w may lead to better mixing. Unfortunately, in this case, the chain will not generally be π-invariant.

We can maintain π-invariance by treating kernel selection as an auxiliary variable. Consider the product space $(\mathcal{Z} \times \mathcal{X}, \mathcal{F} \otimes \mathcal{B})$. The iterated integrals

$$\tilde{\pi}(f) := \int \int f(z,x) w_x(\mathrm{d}z) \pi(\mathrm{d}x), \tag{2.2}$$

for all non-negative measurable f define a distribution on $\mathcal{F} \otimes \mathcal{B}$. $\tilde{\pi}$ is the joint distribution for the coordinates $Z(z,x) = z$ and $X(z,x) = x$, while w is the conditional distribution of Z given X and π is the marginal of X.

We now construct a chain on the extended space that is $\tilde{\pi}$-invariant. This will imply the marginal chain of interest is π-invariant. From the current state (z,x), the chain first samples $z' \sim w_x$. Then we sample x' using a kernel $K_{z'}$ which keeps both $T_{z'}$ and Z fixed and is $\tilde{\pi}$-invariant. An obvious choice for each K_z is

$$K_z(x,\cdot) := P[X \in \cdot \mid T_z(X) = T_z(x), Z = z], \tag{2.3}$$

assuming, of course, that we can sample directly from this distribution. Otherwise, if the density of (2.3) is known up to a normalizing constant, we could use Metropolis-Hastings with proposals that keep T_z and Z fixed.

2.3 Sampling Unweighted Graphs

Sampling unweighted graphs conditional on vertex degrees arises in many disciplines. In exponential random graph models, the degrees are often sufficient statistics for nuisance parameters in the null distribution (Snijders, 1991). Other applications include analysis of co-occurrence tables in ecology, and testing the Rasch model in psychometrics (Gustafsson, 1980).

We formalize the sampling problem as follows. Let G_0 be a given directed or undirected graph with a finite vertex set V. Let $\mathcal{F}$ be a subset of possible edges of a graph with vertex set V. Let $\mathcal{G}$ be the set of all graphs G with the same vertex set and degree sequence as G_0, and additionally satisfying $E(G) \cap \mathcal{F} = E(G_0) \cap \mathcal{F}$. Our goal is to sample from the uniform distribution π on $\mathcal{G}$.

Intuitively, $\mathcal{F}$ represents edges known *by design* to be present or absent. Given vertices u and v, if uv belongs to $\mathcal{F}$ then uv is either present in all graphs in $\mathcal{G}$, or in none. We stress *by design* because the constraints imposed by the degree sequence and $\mathcal{F}$ may imply that further edges are present or absent in all graphs of $\mathcal{G}$. We call this set $\tilde{\mathcal{F}}$ the set of known edges, and formally define it as

$$\tilde{\mathcal{F}} = \{\text{possible edges } uv : uv \in G_0 \Leftrightarrow (uv \in G \text{ for all } G \in \mathcal{G})\}.$$

We show in Section 2.3.2 a method of computing $\tilde{\mathcal{F}}$.

Algorithm 1 gives one step of the sampler we propose. It needs two 'neighborhood' sets associated to each vertex in a graph G. The set $N_G(u)$ are the in-neighbors of u, excluding any vertex v for which the edge vu is known. $M_G(u)$ is the set of all vertices v which are not out-neighbors of u, and for which the absence of uv is not known. These are defined as

$$N_G(u) := \{v \in V : vu \in E(G), vu \notin \tilde{\mathcal{F}}\},$$

$$M_G(u) := \{v \in V : uv \notin E(G), uv \notin \tilde{\mathcal{F}}\}.$$

Here is a sketch of one run of Algorithm 1. Let $n = 0$, and sample a_0 uniformly from the set of all vertices v for which $N_G(v)$ is non-empty. Sample a_1 uniformly from $N_G(a_0)$, then sample a_2 uniformly from $M_G(a_1)$. Replace a_1a_0 in $E(G)$ with a_1a_2. Letting $n = n + 2$, iterate this procedure, however in each subsequent step a_{n+1} cannot be a_{n-1}; this prevents the sampler adding the edge $a_{n+1}a_{n+2}$, and removing it in the next iteration, and should improve state space exploration. Iterate until a_n is a_0, at which point all degrees have been maintained. The computational cost of Algorithm 1 is proportional to the random length of the sampled vertex sequence a. This does not imply that longer sequences are worse; they tend to reduce the correlation between the current and next state of the chain. Figure 2.2 shows a simple example step.

Algorithm 1: One iteration of the unweighted graph sampler (UGS). The astrisk $*$ denotes a dummy node, that is not in the node set V.

G, $\tilde{\mathcal{F}}$;
$a_{-1} \leftarrow *$;
$a_0 \sim U(\{v \in V : N_G(v) \neq \emptyset\})$;
$n \leftarrow 0$;
repeat
  $a_{n+1} \sim U(N_G(a_n) \setminus \{a_{n-1}\})$;
  $a_{n+2} \sim U(M_G(a_{n+1}))$;
  $E(G) \leftarrow E(G) \setminus \{a_{n+1}a_n\}$;
  $E(G) \leftarrow E(G) \cup \{a_{n+1}a_{n+2}\}$;
  $n \leftarrow n + 2$;
until $a_n = a_0$;
return G

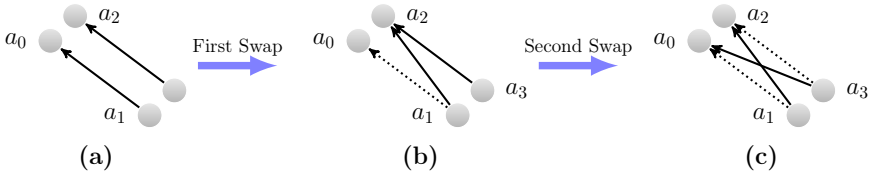


Fig. 2.2 One iteration of Algorithm 1 with two iterations in the inner loop. Figures (a) and (b) give the graph and quantities prior to the first and second edge swaps respectively. Figure (c) depicts the returned graph. Dashed edges are edges removed through the sampling step.

2.3.1 Properties of Algorithm 1

Let a^r denote the reverse of a finite sequence a. Given a graph, let $u_1v_1 \leftrightarrow u_2v_2$ denote the operation of replacing edge u_1v_1 with edge u_2v_2. We will refer to this operation as a *swap*. We call $u_1v_1 \leftrightarrow u_2v_2$ *viable* if and only if u_1v_1 is an edge, u_2v_2 is not an edge and both u_1v_1 and u_2v_2 are not in $\tilde{\mathcal{F}}$.

A single iteration of Algorithm 1 samples a random sequence of vertices a. Proposition 2.3.1 implies that the expected length of this is finite, so that a takes the form $a_0a_1...a_ka_0$ for some k odd. Let $\mathcal{A}$ be the collection of sequences taking this form.

Proposition 2.3.1. *For any input graph $G \in \mathcal{G}$ and any $\mathcal{F}$, the expected length of the vertex sequence a formed by Algorithm 1 is finite.*

Let two sequences be *equivalent* if and only if they are either identical or they are each others' reverse. We let $\mathcal{Z}$ be the quotient set of $\mathcal{A}$ by this equivalence relation.

We will associate each class $z \in \mathcal{Z}$ with a kernel on $\mathcal{G}$. Fix any z and let a be a representative of z. Consider the following Markov chain on $\mathcal{G}$. From the current state, attempt to iteratively perform $a_1a_0 \leftrightarrow a_1a_2$, $a_3a_2 \leftrightarrow a_3a_4$, ... , $a_ka_{k-1} \leftrightarrow a_ka_0$ to obtain the next state. We say this move is *viable* if and only if all the swaps are viable when applied iteratively. We refer to this sequence of swaps as the *swaps corresponding to* a. If the swaps are not viable, attempt the swaps corresponding to a^r. If neither swap sequence is viable, then the next state of the chain is unchanged. We define K_z as the

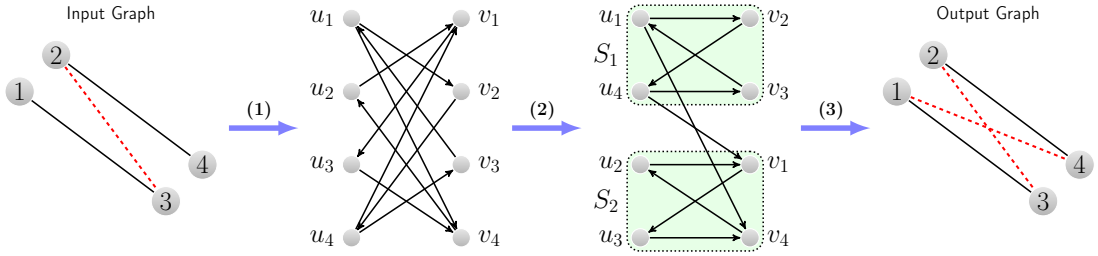


Fig. 2.3 Algorithm for computing $\tilde{\mathcal{F}}$, shown for an undirected graph with four vertices. Input graph G is on the left, where the dashed line represents a prohibited edge (i.e. $\mathcal{F} = \{23\}$). In stage (1) B_G is constructed. In (2) the components S_1 and S_2 are computed, and after (3) we observe $\tilde{\mathcal{F}} = \{23, 14\}$.

kernel of this chain. Remark 1 implies that K_z is well-defined; specifically, the definition is independent of the chosen representative of z.

Remark 1. *If the sequences a and a^r are distinct and the swaps corresponding to a are viable, then the swaps corresponding to a^r are not viable.*

Let K be the collection of these kernels. The conditional distribution w on $\mathcal{Z}$ is defined implicitly by the law of a given through Algorithm 1. Formally, the sampler selects K_z by sampling either a or a^r belonging to z. Lemma 2.3.2 implies that Q is π-reversible on $\mathcal{G}$.

Lemma 2.3.2. *(K, w) is a symmetric decomposition of Q, and each $K_z \in K$ is π-reversible.*

In practice, we consider a lazy version of the chain, which ensures aperiodicity. Fix some small $\alpha \in (0, 1)$ and define $\tilde{Q} := (1 - \alpha)Q + \alpha I$ where I is the identity kernel. Proposition 2.3.3 follows by Lemma 2.3.2 and through additionally showing that the chain is connected.

Proposition 2.3.3. *A Markov chain with kernel $\tilde{Q}$ has limiting distribution π.*

2.3.2 Identifying all Known Edges/Non-Edges

We show how to determine $\tilde{\mathcal{F}}$ from $\mathcal{F}$ and the degree sequence using an auxiliary graph. Given any $G \in \mathcal{G}$ with n vertices labelled $1, ..., n$ we construct an auxiliary bipartite digraph $B := (U, V, E)$. Let $U = \{u_i\}$ and $V = \{v_i\}$ for $i = 1, ..., n$. We define the graph's edge set as

$$E(B) := \{v_j u_i : ij \in E(G) \setminus \mathcal{F}\} \cup \{u_i v_j : ij \notin E(G) \cup \mathcal{F}\}.$$

Figure 2.3 shows an example of one such graph.

Let $B_{\mathcal{G}}$ denote the collection of all graphs generated this way from the set $\mathcal{G}$. Proposition 2.3.4 shows that we can identify $\tilde{\mathcal{F}}$ prior to sampling by identifying all strongly

connected components of any graph in $B_{\mathcal{G}}$. This can be done using a depth-first search on B_G, followed by another depth-first search on the transposed graph.

Proposition 2.3.4. *Fix any graph $B \in B_{\mathcal{G}}$. The vertex pair ij belongs to $\tilde{\mathcal{F}}$ if and only if there is no edge incident to both u_i and v_j, or if u_i and v_j belong to different strongly connected components.*

The complexity of this preprocessing procedure is $\Theta(2n + n^2)$ (Cormen et al., 2009, chap. 22). It is difficult to formally compare this to the cost of sampling. Empirically, the average cost of each iteration of Algorithm 1 appears to grow roughly linearly in n, while the number of iterations needed for sampling grows super-linearly in n. Thus, in practice, the cost of sampling dominates the edge identification procedure.

2.4 Sampling Weighted Graphs

Sampling weighted graphs with given vertex strengths arises in the analysis of two-way contingency tables. In this context, sampling is used to approximate the null distribution of test statistics (see Agresti, 1992). The general problem is stated as follows. Let G_0 be a given integer-weighted, directed or undirected graph with a finite vertex set V. Let $\mathcal{F}$ be a subset of possible edges of a graph with vertex set V. Let $\mathcal{G}$ be the set of all weighted graphs with the same vertex set and strength sequence as G_0, and additionally assigning weight $c_{G_0}(uv)$ to each $uv \in \mathcal{F}$. Our goal is to sample from π, the uniform distribution on $\mathcal{G}$. We use the auxiliary variable method proposed in Section 2.2.3. The method first requires defining a set of conditioning statistics $\{T_z : z \in \mathcal{Z}\}$ and a selection law w over them. This is the focus of Sections 2.4.1 and 2.4.2. We derive the kernel set K in Section 2.4.3.

2.4.1 Conditioning Statistics

We start by specifying the conditioning statistics T_z for a given $z \in \mathcal{Z}$. Define $\mathcal{A}$ and $\mathcal{Z}$ as in Section 2.3.1, and associate each $a = a_0a_1...a_ka_0 \in z$ with a vector of vertex pairs

$$e(a) := (a_1a_0, a_1a_2, a_3a_2, a_3a_4, ..., a_ka_{k-1}, a_ka_0),$$

of length $k + 1$. We refer to z as *valid* if it satisfies two conditions. Firstly vertex pairs in $e(a)$ must be distinct and secondly they must not be in $\mathcal{F}$. If these properties hold for $e(a)$ then they hold for $e(a^r)$, and so it suffices that they hold for any $a \in z$. If z is invalid we condition on the whole graph so that no update can occur (i.e. $T_z(G) := G$). Otherwise fix any $a \in z$ and define $T_z(G) := (c_G(uv) : uv \notin e(a))$. This quantity does not depend on which a is chosen because $uv \in e(a)$ if and only if $uv \in e(a^r)$. This statistic conditions on the weight of all edges outside $e(a)$, so that we only update along $e(a)$.

2.4.2 Selection Law

Before defining the selection law w, we provide intuition as to which z we wish to sample. From the current state G we intuitively wish to select z that allow us to change the state space. This translates to avoiding z for which the level set $\{T_z = T_z(G)\} = \{G\}$, and therefore implies avoiding all invalid z and some valid z. Assuming z is valid, fix $a \in z$ and consider the vector $e := e(a)$. Letting $s := (+1, -1, ..., +1, -1)$, any graph in $\{T_z = T_z(G)\}$ must assign weights $c_G(e) + s\Delta$ to e and for some Δ for which the resulting weights are non-negative. We denote the range of Δ by $\{\Delta_l, \dots, \Delta_u\}$. If $\Delta_u = \Delta_l = 0$ then the only graph satisfying this is G. This will happen if there exists i odd and j even such that $c_G(e_i) = c_G(e_j) = 0$. If G is sparse then only a small proportion of z can avoid this. Moreover, which z avoid this depends on the current state and so any state-independent w will be inefficient. Our state-dependent w, which we now define, is designed to limit this.

Occasionally the sampling strategy will fail to sample a kernel index z. Let id refer to an arbitrarily chosen invalid z^*, to be chosen by default if this happens. We start by redefining the vertex sets $N_G(u)$ and $M_G(u)$ from Section 2.3 as

$$N_G(u) := \{v \in V : vu \in E, vu \notin \mathcal{F}\},$$

$$M_G(u) := \{v \in V : uv \notin \mathcal{F}\}.$$

Given the current state G we sample z as follows. Let $n = 0$, $a_{-1} = *$ and sample a_0 uniformly from the set of vertices v for which $N_G(v)$ is non-empty. Repeat the following until termination.

1. If $N_G(a_n) \setminus \{a_{n-1}\}$ is empty return id, else sample a_{n+1} uniformly from this set.
2. If $M_G(a_{n+1}) \setminus \{a_n\}$ is empty return id, else if a_0 is in this set then return $[a_0 ... a_{n+1} a_0]$. Otherwise sample a_{n+2} uniformly from this set and let $n = n + 2$.

The chain cannot move if the above procedure returns id or an invalid $[a]$. The former is rare and occurs in cases of extreme sparsity. The latter will be more likely with a large set of fixed edges.

2.4.3 The Kernel Set

We now derive the kernel set. First define $\tilde{\pi}$ as in (2.2), as an iterated integral of non-negative measurable functions on $\mathcal{Z} \times \mathcal{G}$. Each K_z will take the form of (2.3). That is, we sample *directly* from the conditional probability of the joint $\tilde{\pi}$ given T_z and the coordinate $Z(z, G) = z$. Therefore if z is invalid then K_z must be the identity kernel. However if z is valid we saw in Section 2.4.2 that the update can be parameterised by Δ taking values in $[\Delta_l, \Delta_u]$. Therefore, it suffices to derive the distribution of Δ. This is

the focus of this section. Throughout we let $G_{\Delta'}$ refer to the graph obtained from the current state at $\Delta = \Delta'$.

Suppose that we sample a vertex sequence a and $z := [a]$ is valid. Since π is the uniform distribution, $P[\Delta = \Delta']$ is proportional to $w_{G_{\Delta'}}(z)$ for each $\Delta' \in [\Delta_l, \Delta_u]$. It is often not possible to sample a from G_{Δ_l}, or to sample a^r from G_{Δ_u}. This is why we collapse a and a^r into z; doing so ensures we can always sample z from each $G_{\Delta'}$ for $\Delta' \in [\Delta_l, \Delta_u]$. Suppose $\Delta_u - \Delta_l > 1$. Fix any $\Delta_l < \Delta' < \Delta_u$ and define $\alpha := w_{G_{\Delta'}}(z)$. This is interpreted as the probability of sampling z from $G_{\Delta'}$. Inspecting the kernel selection law defined in Section 2.4.2 we see that $w_{G_1} = w_{G_2}$ for any G_1 and G_2 with the same topology. Since $E(G_{\Delta'})$ is the same for any $\Delta_l < \Delta' < \Delta_u$, it follows that α does not depend on the specific Δ' chosen. If on the other hand $\Delta_u - \Delta_l \leq 1$ we give α an arbitrary finite value. Then

$$\text{Law}(\Delta) = \frac{1}{A}\left[w_{G_{\Delta_l}}(z)\delta_{\Delta_l} + w_{G_{\Delta_u}}(z)\delta_{\Delta_u} + \sum_{\Delta_l<\Delta'<\Delta_u} \alpha\delta_{\Delta}\right], \tag{2.4}$$

where $A := w_{G_{\Delta_l}}(z) + w_{G_{\Delta_u}}(z) + \alpha \max(\Delta_u - \Delta_l - 1, 0)$. $w_{G_{\Delta_l}}(z)$, $w_{G_{\Delta_u}}(z)$ and α are easily computed by following the details of Section 2.4.1.

2.4.4 Summary

Algorithm 2 gives pseudo-code for one iteration of the sampler. By construction, the chain is π- invariant. Proposition 2.4.1 holds by additionally showing the chain is connected.

Proposition 2.4.1. *The chain defined by (K, w) has limiting distribution π.*

Q can be readily adapted to sample distributions whose density is known up to a normalizing constant by using Metropolis-Hastings to sample Δ. The computation cost of Algorithm 2 is proportional to the number of edges updated.

2.5 Simulation Study

Methods used in this section, and in Section 2.6, were programmed in C++ and wrapped to R. The algorithms were run on an Intel Core i5-6360U 2GHz CPU. In Section 2.5.1 we investigate the effect of graph density and size on the performance of the proposed samplers, while Section 2.5.2 looks at the effect of fixed edges/non-edges. An R-package implementing the new algorithms is available in the supplemental materials.

2.5.1 Effect of Size and Sparsity

Consider the Erdős-Rényi model $G(n, \theta)$ for directed graphs with self-loops. The parameter n denotes the number of vertices, and each ordered vertex pair is an edge with probability θ, independent of all other edges. If $G \sim G(n, \theta)$ then the conditional

Algorithm 2: One Iteration of the weighted graph sampler (WGS).

G, $\mathcal{F}$;
sign$(uv) \leftarrow 0$;
edges $\leftarrow \{\}$;
$(\Delta_l, \Delta_u) \leftarrow (-\infty, \infty)$;
$a_{-1} \leftarrow *$;
$n \leftarrow 0$;
$a_0 \sim U(\{v \in V : N_G(v) \neq \emptyset\})$;
repeat
  if $N_G(a_n) \setminus \{a_{n-1}\} = \emptyset$ **then**
    return G;
  else
    $a_{n+1} \sim U(N_G(a_n) \setminus \{a_{n-1}\})$;
  if $a_0 \in M_G(a_{n+1}) \setminus \{a_n\}$ **then**
    $a_{n+2} \leftarrow a_0$;
  else if $M_G(a_{n+1}) \setminus \{a_n\} = \emptyset$ **then**
    return G;
  else
    $a_{n+2} \sim U(M_G(a_{n+1}) \setminus \{a_n\})$;
  if *edges* $\cap \{a_{n+1}a_n, a_{n+1}a_{n+2}\} \neq \emptyset$ **then**
    return G;
  edges $\leftarrow$ edges $\cup \{a_{n+1}a_n, a_{n+1}a_{n+2}\}$;
  sign$(a_{n+1}a_n) \leftarrow +1$; sign$(a_{n+1}a_{n+2}) \leftarrow -1$;
  $n \leftarrow n + 2$;
until $a_n = a_0$;
forall *edge* $\in$ *edges* **do**
  if *sign*(*edge*) $= +1$ **then**
    $\Delta_l \leftarrow \max(\Delta_l, -c_G(\text{edge}))$;
  else
    $\Delta_u \leftarrow \min(\Delta_u, c_G(\text{edge}))$;
Sample Δ according to (2.4);
forall *edge* $\in$ *edges* **do**
  $c_G(\text{edge}) \leftarrow c_G(\text{edge}) + \text{sign(edge)}\Delta$;
return G;

distribution of G given its degree sequences is uniform over all graphs with the same degrees. This observation provides us with a useful strategy for assessing convergence of samplers in the unweighted graph setting, which we now detail.

Fixing a particular value of n and θ, we first simulate N independent graphs $G_i \sim G(n, \theta)$. For each G_i, we construct a new graph G_i^0 with the same degrees as G_i using a maximum flow algorithm. This is done to find an initial graph which is far from the mode of the posterior distribution. The algorithm we use is adapted from that described in Gandy and Veraart (2016), Appendix A. To test the performance of a given sampler we use it to simulate N Markov chains. The i^{th} chain is given initial state G_i^0 and run to obtain samples $G_i^1, ..., G_i^M$. If the chain has converged to its target distribution after t iterations, then the distribution of G_i^t and G_i should be statistically indistinguishable. Moreover, if we have access to a statistic T then we can compare, for each t, the empirical

distribution of $T(G_1^t), ..., T(G_N^t)$ to the distribution of $T(G)$. If the sampler mixes rapidly, we expect these to be similar for small t.

The same approach can be used in the weighted graph setting. An analog to the Erdős-Rényi model for weighted and directed graphs with self-loops assigns edge weights to G according to a geometric distribution, i.e. the event $c_G(uv) = k$ occurs with probability $\theta^k(1-\theta)$ independent of all other edge weights. Conditioning on G's vertex strengths then yields the uniform distribution over all graphs with the same strengths.

It remains to specify T in the unweighted and weighted settings. In the former we estimate reciprocity. Letting X denote the adjacency matrix of G, then we use $T = T_u$ with $T_u(G) := \sum_{i<j} X_{ij}X_{ji}$. This statistic is interpreted as the total number of mutual dyads in the graph. In the latter case $T = T_w$ with

$$T_w(G) = \frac{\sum_{u \neq v} \min\left(c_G(uv), c_G(vu)\right)}{\sum_{u \neq v} c_G(uv)}.$$

This is a measure of reciprocity for weighted graphs, first proposed in Squartini et al. (2013).

Recall that for each t we wish to compare the empirical distribution of $T(G_1^t), ..., T(G_N^t)$ to that of $T(G)$. $T_u(G)$ is distributed Bin $\left(n(n-1)/2, \theta^2\right)$, and so in the unweighted setting we undertake M Chi-squared tests and record the sequence of p-values $p_1, ..., p_M$. This allows us to formally assess convergence of the samplers. The distribution of $T_w(G)$ is not known analytically, and so for the weighted setting we draw 10^5 samples from this distribution and undertake two-sample Kolmogorov-Smirnov tests instead.

We repeat the above procedure for various combinations of n and θ to uncover the effect of graph size and density on the statistical efficiency of the samplers. Algorithm 2 (WGS) is compared to the Diaconis & Sturmfels chain (DS) detailed in Section 3. The DS chain operates on the adjacency matrix of the graphs. Algorithm 1 (UGS) is compared to a simple and widely used randomization procedure that works as follows. At each stage select two edges uv and wx at random from the current graph G. If $ux \notin E(G) \cup \mathcal{F}$ and $wv \notin E(G) \cup \mathcal{F}$ then replace uv and wx with ux and wv, else do not change G. We refer to this randomization procedure as the Switch chain. In each setting, we use $M = 10^4$ and $N = 500$. Thinning used for each chain was chosen to make the computation time per sample comparable.

The results are displayed in Table 2.1. The efficiency of the Switch chain deteriorates relative to UGS as θ increases. DS becomes inefficient compared to WGS in sparse graphs, as depicted in Figure 2.4. The proposed methods perform comparatively well across all graph sizes and densities considered.

2.5.2 Incomplete Tables

UGS and WGS are irreducible in the face of arbitrary fixed edges/non-edges. Here we investigate the ability of the samplers to traverse the state space under such constraints.

Table 2.1 Results of the simulations outlined in Section 2.5.1. We record $t^* := \min\{t : p_t > 0.1\}$, i.e. the smallest t for which the null that $T(G_1^t), ..., T(G_N^t)$ is drawn from the distribution of $T(G)$ was not rejected at the 10% level. Mixing rate is the estimated proportion of sampling steps that changed the state of the chain. Mean ESS/s reports the mean time-normalized effective sample size across all M chains.

	Unweighted						*Weighted*					
	t^*		*Mixing Rate*		*mean ESS/s*		t^*		*Mixing Rate*		*mean ESS/s*	
Setup	*UGS*	*Switch*	*UGS*	*Switch*	*UGS*	*Switch*	*WGS*	*DS*	*WGS*	*DS*	*WGS*	*DS*
$n = 20$ & $\theta =$												
0.1	5	4	1.00	0.53	7.01×10^4	5.60×10^4	4	189	0.49	0.00	2.60×10^4	3.48×10^2
0.5	6	11	1.00	0.13	2.15×10^4	1.21×10^4	79	427	0.58	0.14	3.50×10^3	7.57×10^2
0.9	3	30	1.00	0.00	3.51×10^4	1.43×10^3	244	460	0.86	0.58	1.11×10^3	3.98×10^2
$n = 50$ & $\theta =$												
0.1	10	8	1.00	0.72	1.83×10^4	1.62×10^4	21	2232	0.51	0.01	4.19×10^3	3.70×10^1
0.5	29	46	1.00	0.21	6.81×10^3	5.33×10^3	473	2385	0.57	0.16	2.45×10^2	2.00×10^1
0.9	6	193	1.00	0.006	1.50×10^4	5.99×10^2	1531	2603	0.84	0.59	2.64×10^1	9.51
$n = 100$ & $\theta =$												
0.1	20	15	1.00	0.77	1.45×10^3	1.32×10^3	64	7433	0.51	0.01	7.87×10^2	2.30
0.5	65	111	1.00	0.23	5.97×10^2	5.95×10^2	1600	10,359	0.59	0.05	1.47×10^1	2.64
0.9	13	534	1.00	0.002	1.36×10^3	6.00×10^1	4992	11218	0.84	0.57	8.62	3.12

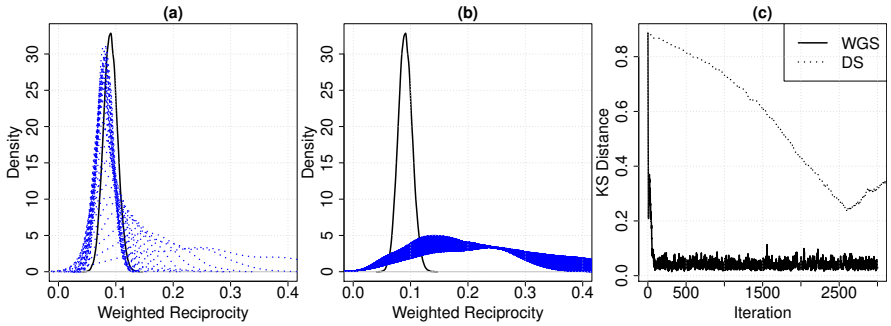


Fig. 2.4 In Figures (a) and (b) the black line is the density of $T_w(G)$ estimated using 10^5 samples when $n = 10^2$ and $p = 0.1$. In (a) the dotted lines show the evolution of the empirical density of $\{T(G_j^l)\}$ for $l \leq 40$ using WGS. In (b) this quantity using DS for $l \leq 10^3$. (c) shows the Kolmogorov-Smirnov distance between samples $\{T(G_j^l)\}$ and Monte Carlo samples for each l using both WGS and DS.

We do this in the context of incomplete binary and contingency tables, which are tables with some entries fixed at zero. They arise in several contexts. In the contingency table setting, particular combinations of the two variables may be impossible, forcing zero entries in the corresponding cells. Alternatively there may be missing observations or in some contexts, researchers may wish to fit composite models by partitioning the cells into subsets, and fitting a separate log-linear model for each group (Goodman, 1963, 1968, Fienberg, 1969). See Bishop and Fienberg (1969) for extensive examples of incomplete tables.

We construct 10^3 6×6 incomplete contingency tables using the following procedure. To construct the i^{th} table, we randomly place half of the table coordinates into the fixed set $\mathcal{F}_i$. We then use a maximum flow algorithm to construct a table x_i^0 in the set $\mathcal{X}_i$, which consists of all 6×6 tables x with all margins equal to 3, and additionally satisfying $x_{\mathcal{F}_i} = 0$. If $\mathcal{X}_i$ is empty then $\mathcal{F}_i$ is re-sampled until it is not. We use the LattE software (Kahle et al., 2017) to count the number of tables in $\mathcal{X}_i$. 10^5 samples are obtained using

WGS and DS with thinning of 10 and with initial state x_i^0, and we record the proportion of tables in $\mathcal{X}_i$ visited by each sampler. This is repeated for incomplete binary tables to compare UGS and Switch, however using table margins equal to 1. The results are shown in Figure 2.5.

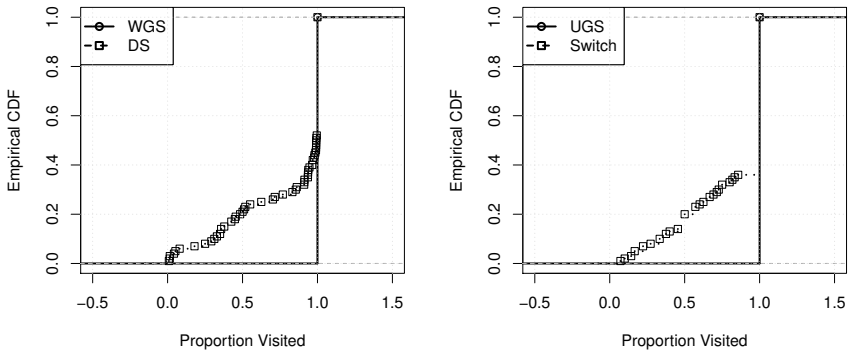


Fig. 2.5 Empirical CDF of the proportion of tables visited by each sampler over the 10^3 runs. Left: results for incomplete contingency tables. Right: results for binary tables.

Figure 2.5 provides empirical evidence that WGS and UGS can traverse the state space. DS and Switch appear to be reducible for particular patterns of fixed entries. Diaconis and Sturmfels (1998) propose an alternative chain which uses techniques from computational algebra to compute a Markov basis for the state space. Other approaches using computational algebra include Aoki and Takemura (2005) and Rapallo (2006). Unfortunately the cost of computing the basis is exponential in the size of the table, and these methods are feasible only for tables with only a few rows and columns. Chen (2007) propose SIS algorithms for uniform sampling of incomplete binary and contingency tables. The authors provide an implementation of their method for binary tables, and we use this to test their sampler (labelled SIS_CP1). In each of the 10^3 cases, we collected 10^5 samples and found their method visited all graphs in the state space. However, we show in Section 2.6.1 that the algorithm is not always reliable.

2.6 Applications

We consider the comparative performance of the new samplers on real datasets. In Section 2.6.1 we use Algorithm 1 to detect compartmentalization in an ecological network, and in Section 2.6.2 we use Algorithm 2 to investigate nestedness in a large affiliation network.

Reported standard errors were computed using spectral methods from R's coda package. These estimates were compared to those obtained using batch means, and where feasible, bootstrapping. These latter estimates are not reported as there was little discernible difference from those obtained by spectral methods. Thinning used in each

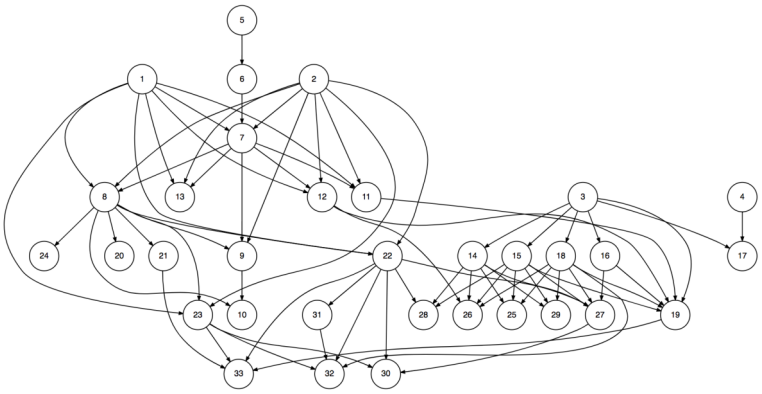


Fig. 2.6 Food web of the Chesapeake bay ecosystem.

method was set to approximately equate the resulting standard errors. We used burn-in equivalent to 20% of samples obtained.

2.6.1 Ecological Networks

A food web is a digraph encoding predator-prey relationships within a group of species. Each species is a node and a link exists from species A to species B if and only if B consumes A.

Ecologists wish to identify and explain structural patterns in observed food webs including motifs, diet contiguity, intervality, connectance and compartmentalization. We will focus on assessing the tendency towards compartmentalization in food webs. Compartmentalization describes the extent to which species can be partitioned into distinct groups such that linkage density *within* groups is greater than that *between* groups (Girvan and Newman, 2002, Krause et al., 2003).

Figure 2.6 depicts the food web of 33 species in the Chesapeake bay in the summer. The data was collected by Baird and Ulanowicz (1989). Pimm and Lawton (1980) proposed a statistic $\bar{C}$ to measure the level of compartmentalization in a food web. Here we describe a directed analogue of this statistic. Let G represent a food web of n species, and i and j be two species. Let c_{ij} be the number of shared predators of species i and j as a proportion of the total number of predators of i and j. $\bar{C}$ is then the mean of the off-diagonal elements of (c_{ij}), defined by

$$\bar{C} := \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} c_{ij}.$$

$\bar{C}$ takes values in $[0, 1]$ and higher values are associated with greater levels of compartmentalization.

We begin by testing whether the observed level of $\bar{C}_0 = 0.0260$ can be considered high when compared to the set of all graphs with the same degree sequence as G. With thinning of 5, Algorithm 1 (UGS) took around 2 second to obtain 10^5 samples. The

estimated p-value was $0.0163 \pm 4.3 \times 10^{-4}$, complementing previous results suggesting food webs have high compartmentalization when compared to random graphs where species have an equal probability of linking to each other species (Krause et al., 2003, Rezende et al., 2009). The sequential importance sampling algorithm SIS-CP1 (Chen, 2007) took 33 seconds to obtain 10^5 samples, estimating a p-value of $0.0158 \pm 4.3 \times 10^{-4}$.

Guimerà et al. (2010) find that compartmentalization observed in real food webs is not unusual when compared to networks generated under niche models, and conclude that 'compartmentalization can be explained solely by the niche-valued ranking of species'.

We attempt to test this hypothesis for the Chesapeake bay food web. We compute the chain averaged trophic level (Williams and Martinez, 2004) for each species, and assume any given species is forbidden from consuming other species with a higher trophic level. The resulting forbidden links should help to control for the food web's trophic structure. The assumption induces 565 forbidden edges in the null distribution.

Again using thinning of 5, UGS took 2 seconds to obtain 10^5 samples. Empirically, the time taken for UGS to randomize a graph depends on the density of the graph rather than the number of nodes. Sparse graphs are quick to randomize, and if the graph density falls as nodes increase, the randomization time scales sub-linearly in the number of nodes. The estimated effective sample size was over 9.5×10^3, giving an estimated p-value of $0.0568 \pm 7.5 \times 10^{-4}$. At a significance of $\alpha = 0.05$, we can no longer conclude that compartmentalization in the Chesapeake food web is unusual under the null distribution. Our method of determining trophic structure is crude, and a closer analysis of the food web is warranted before drawing any conclusions.

SIS-CP1 took 24 seconds to run and over 97% of the samples produced were discarded as invalid, leaving only 3,069 to be used for estimation. The estimated p-value was $0.0558 \pm 6.3 \times 10^{-3}$. Using alternative methods to calculate the species' trophic levels gives rise to other sets of forbidden edges. For some such patterns, SIS-CP1 was unable to construct a single valid sample.

2.6.2 Affiliation Networks

In social network analysis, an affiliation network represents membership or participation data between a set of actors and a set of groups. For example, a link may indicate participation of an actor in an event. Dyadic data of this type include board membership (eg. Mizruchi, 1983), participation in online forums (eg. Allatta, 2003) and authorship of research chapters (eg. Newman et al., 2001). Affiliation networks are bipartite graphs, and can therefore be represented as a contingency table, with rows corresponding to actors and with columns denoting the groups.

Social scientists are interested in detecting network structure through particular metrics. Example patterns of interest include 'small-world effects', clustering and degree distributions. Here we focus on detecting nestedness in data collected by Opsahl (2013) on messages sent by users of an online social platform to online forums. This data is a 899

by 522 contingency table whose $(i, j)^{\text{th}}$ entry is the number of messages posted by user i in forum j. Informally speaking, nestedness is the degree to which neighbors of nodes with low degree are a subset of the neighbours of nodes with higher degree. Nestedness has been detected in a number of network systems including ecological interaction networks, social media information networks and socio-economic networks, and has been shown to have important implications for the robustness and stability of a system.

Several measures for nestedness have been developed for integer contingency tables. Galeano et al. (2009) propose to use the weighted-interaction nestedness (WIN) estimator, which is a matric based on a weighted Manhattan distance and takes values between 0 and 1. Higher values indicate higher levels of nestedness. We will test whether the observed WIN statistic is unusually high when compared to a suitable null distribution.

We assume under the null hypothesis that the table is a uniform draw from the set of all tables with the same margins. This procedure of fixing the margins is widely used in both binary and integer matrices (Connor and Simberloff, 1979, Gotelli and Entsminger, 2001, Ulrich and Gotelli, 2007). Alternative null models are available; for example fixing one margin or only satisfying the observed margins in expectation.

We obtain 10^3 samples using WGS and DS with thinning set to 10^6. WGS estimated the average WIN distance of the sampled tables at $0.0539 \pm 7.3 \times 10^{-6}$. This estimated standard error is equivalent to that from 10^3 independent samples, indicating good mixing. DS estimated $0.056 \pm 1.7 \times 10^{-3}$ and has a high correlation between successive samples, giving an effective sample size of 12. The estimate exhibits high bias because the chain shows non-stationary behaviour for the first 300 iterations. The observed statistic was 0.157, and so both methods give a p-value of 0. Taken at face value this indicates strong evidence for nestedness. However, this is more likely down to misspecification of the null model. WIN is sensitive to overall matrix density and the sampled tables are systematically denser than the observed table. It would be instructive therefore to consider alternate null distributions which better preserves this property.

Eisinger and Chen (2017) develop efficient SIS methods for sampling tables uniformly over all tables with given margins. The authors provide code for a cell-by-cell method labelled SIS-G* (coded in C). Using SIS-G* on this example we were unable to produce a valid sample. It appears the method is not scalable to large tables.

2.7 Discussion

This chapter has developed new MCMC samplers for two important problems. First, for sampling from the set of unweighted graphs respecting prescribed vertex degrees. Second, for sampling from the set of weighted graphs respecting prescribed vertex strengths. The samplers work when conditioning on the presence or absence of a set of edges. We have shown examples where alternative MCMC methods are infeasible as they rely on computing a Markov basis, and where existing SIS methods perform poorly. In contrast,

our methods do not require computing a Markov basis, and are orders of magnitude more efficient in these examples.

State-dependent mixing of Markov kernels is a general concept, and the specific implementation of our samplers is not unique. The technique could be used to develop alternative samplers specialized to particular setting. The samplers can be readily extended to sample from arbitrary distributions known up to a normalization constant. For example, they can be adapted to carry out Bayesian network tomography in the case of a star network topology. In contrast, SIS methods are not readily adaptable to more general distributions. A theoretical analysis of the mixing times of the new samplers is beyond the scope of this chapter, and is left for future work.

3

Approximate Conditional Sampling for Pattern Detection in Weighted Networks

3.1 Introduction

This chapter develops an approach to measuring the significance of patterns observed in weighted networks. The proposed method compares the network of interest to graphs drawn from a null model. The null model is designed to account for node heterogeneity, including both heavy-tailed degree and strength distributions. Unknown nuisance parameters are dealt with by approximate conditioning, and samples are drawn using a novel MCMC method. This algorithm uses similar techniques to those used in Chapter 2, extended to the case of graphs with real-valued edge weights and potential sparsity. The development just outlined mirrors approaches that have long been used to successfully to detect patterns in unweighted graphs (Connor and Simberloff, 1979, Newman et al., 2001, Milo et al., 2002, Maslov et al., 2004, Stouffer et al., 2007), which was described in detail in Chapter 2.

Take, for example, the task of detecting community structure in weighted networks. In general, the community membership of nodes is unknown and must be recovered. Typically, this is done by optimizing some criterion, which could be a quality function like modularity (Newman, 2004). An alternative approach is to fit a statistical model which permits community structure using maximum likelihood. Possible models include the stochastic block model (Nowicki and Snijders, 2001) and the degree-corrected stochastic block model (Karrer and Newman, 2011).

Although popular, modularity is not based on any notion of the significance of a partition; rather it is defined as the absolute difference between observed inter-community links, and those expected under a given null model. As a result, it suffers from the *resolution limit* (Fortunato and Barthélemy, 2007, Kumpula et al., 2007), whereby smaller modules cannot be detected in large networks. A number of methods attempt to overcome this by explicitly defining notions of significance (Aldecoa and Marín, 2011, Miyauchi

and Kawase, 2016, Traag et al., 2013, Reichardt and Bornholdt, 2006, Palowitch et al., 2018, He et al., 2020), which can be optimized over network partitions.

Nonetheless, these methods consider the p-value of a fixed partition and are invalid when assessing the significance of a partition which results from optimizing an objective function. It is possible to find partitions with low p-values in random graphs with no embedded community structure (Guimerà et al., 2004, Reichardt and Bornholdt, 2006, Fortunato, 2010). The p-values are incorrect unless they account for the selection process. This phenomenon parallels that of inference post model selection, which is a widely studied problem that has recently garnered much attention within the statistics community (Taylor and Tibshirani, 2015, Hastie et al., 2019).

In this chapter, we introduce a null model which can be used to quantify the significance of general patterns found in weighted graphs. For example, the approach can be used to determine the significance of community structure *after having identified a partition* with an optimization method. The approach is based on a generalization of 'rewiring' Markov chains (Ryser, 1963, Hakimi, 1962, Rao et al., 1996) to weighted graphs, and is inspired by a recently developed Markov chain (Gandy and Veraart, 2016) for weighted graphs.

After introducing terminology, Section 3.3 motivates the problem by first reviewing a common approaches in the unweighted case. Section 3.4 formulates the general sampling problem, and Section 3.5 introduces the novel MCMC method for sampling the null model. Section 3.6 considers the stochastic stability of the proposed sampler, and Section 3.7 performs an extensive simulation study to test the performance of the method against competing alternatives. Finally, we conclude in Section 3.8.

3.2 Terminology

This chapter is only concerned with directed graphs. Occasionally, we consider unweighted graphs, which are denoted $G := (N, A)$ where $N := \{1, \dots, n\}$ is a set of nodes and $A := (a_{uv})$ is the adjacency matrix. For weighted graphs, the binary adjacency matrix is replaced with a weight matrix. Formally, $G := (N, W)$, where $W := (w_{uv})$ and $w_{uv} \in [0, \infty)$. The topology is implicit: $a_{uv} = 0$ if and only if $w_{uv} = 0$, or alternatively, $a_{uv} = w_{uv}^0$ with the convention that $0^0 = 0$.

Define a node's *out-degree* and *in-degree* by $d_u^- := \sum_v a_{uv}$ and $d_u^+ := \sum_v a_{vu}$ respectively, and collect them into vectors $d^- := (d_1^-, \dots, d_n^-)^t$ and $d^+ := (d_1^+, \dots, d_n^+)^t$. For weighted graphs, we define the node's out- and in-strength by $s_u^- := \sum_v w_{uv}$ and $s_u^+ := \sum_v w_{vu}$, which are also collected into vectors $s^- := (s_1^-, \dots, s_n^-)^t$ and $s^+ := (s_1^+, \dots, s_n^+)^t$. If the graph to which an object belongs is unclear, we explicitly denote its dependence on the graph. For example, we might write $W(G)$ instead of W.

3.3 Motivating a Null Model for Weighted Graphs

Null models have long been used to detect statistically significant patterns in unweighted networks. Such models have found application in a number of diverse fields, including sociology, ecology, categorical data analysis, systems biology, and community detection. While there exists extensive literature for the unweighted case, very little has been developed for both defining and sampling similar models for weighted graphs. We therefore start by reviewing the unweighted case, which motivates our development for weighted graphs.

3.3.1 Null Models for Unweighted Graphs

We define a family of distributions on the space $\mathcal{G}$ of unweighted graphs with n nodes. Formally, let

$$\mathrm{P}_\theta(G) := \kappa(\theta)^{-1} \exp\left(\alpha^t d^- + \beta^t d^+\right), \tag{3.1}$$

where $\kappa(\theta)$ is a normalizing constant and $\theta = (\alpha^t, \beta^t)^t$. The degree vectors are the sufficient statistics, or energies, of the distribution. The parameters $\alpha := (\alpha_1, \ldots, \alpha_n)^t$ and $\beta := (\beta_1, \ldots, \beta_n)^t$ control the distribution of out-degrees and in-degrees, with α_u and β_u representing the sociability and popularity of node u. This is an exponential model, and may be viewed as a directed analogue of the β-model (Chatterjee et al., 2011), or as a special case of the p_1-family (Holland and Leinhardt, 1981), whereby the reciprocity parameters are uniformly taken to be $\rho_{uv} = 0$. These p_1 models were introduced in the context of social network analysis, and were extended to the class of Markov Graphs by Frank and Strauss (1986), and eventually to the class of p^*, or exponential random graph models (ERGMs) (Wasserman and Pattison, 1996).

The model, and its undirected equivalent, are routinely used to measure the significance of properties observed in real-world networks. Measuring significance is useful for a number of tasks; including for use in hypothesis testing, which is used to find evidence of local graph patterns (Milo et al., 2002). An example of such a pattern is reciprocity (Holland and Leinhardt, 1981), which is often evident in social networks. Significance can also be optimized directly by including it in an objective function. This approach helps to discover network patterns, and is widely used for community detection (Newman, 2004).

In general, a practitioner will measure a property of interest in a network, which may be community structure, clustering, or a network motif, for example. This is usually summarized by a statistic $T : \mathcal{G} \to \mathbb{R}$, with large T implying greater prevalence of the property. The observed value t_0 can only be interpreted in the context of *the distribution of T under a suitable null model*. To put this in a formal framework, we embed (3.1) in a larger exponential family

$$\mathrm{P}_{(\theta,\delta)}(G) := \kappa(\theta, \delta)^{-1} \exp\left(\alpha^t d^- + \beta^t d^+ + \delta T(G)\right),$$

which includes the statistic of interest. The null hypothesis that (3.1) provides a good fit of the network, i.e. that t_0 is not significant, is equivalent to testing $\delta = 0$ against the alternative $\delta \neq 0$. This is the approach suggested in Holland and Leinhardt (1981) to test the goodness of fit of the p_1-model, but can equally be interpreted as quantifying the extent to which t_0 is surprising under (3.1).

Notice that the hypothesis $\delta = 0$ is composite because the null depends on the unknown nuisance parameters θ. The typical way to deal with this is to *condition on the sufficient statistics*, which in this case are $d := (d^{-t}, d^{+t})^t$. It is shown in Lehmann and Romano (2006) that tests based on this conditional distribution are optimal, i.e. the uniformly most powerful unbiased (UMPU) test of $\delta = 0$ against $\delta \neq 0$. If in fact the observed graph $G_0 \sim \mathrm{P}_{\theta_0}$ for some θ_0, then the conditional distribution of G_0 given degrees is uniform on

$$\mathcal{G}(d_0) := \{G \in \mathcal{G} : d(G) = d_0\},$$

where $d_0 := d(G_0)$. This is the set of all graphs with the same degree sequence as G_0. This fact is obvious because (3.1) only depends on G through the degrees.

In general, the conditional distribution of the test function is not available analytically, and so we typically resort to drawing samples $G_1, \ldots, G_N \sim \mathrm{Uniform}(\mathcal{G}(d_0))$. Significance $p \in [0, 1]$ is then computed by comparing t_0 to the associated empirical distribution, i.e.

$$p := \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{[t_0,\infty)}(T(G_i)). \tag{3.2}$$

The algorithms used to sample $G_1, \ldots G_N$ depend on our initial assumptions on the graph space. If $\mathcal{G}$ permits non-simple graphs, i.e. allows both self-loops and multiple edges, then it is straightforward to draw independent samples using the *pairing model* (also known as the configuration model), which was first discussed by Bollobás (1980), Bender and Canfield (1978). However, in practice most networks are simple and if we restrict $\mathcal{G}$ accordingly, the situation becomes more complex. In particular, there is no straightforward method for drawing independent and exactly uniform samples. A common approach is to construct a Markov chain based on randomly rewiring edges, such that node degrees are exactly maintained (Ryser, 1963, Hakimi, 1962, Rao et al., 1996). This yields correlated samples which are asymptotically uniform, and can be treated as approximately independent if the chain is thinned appropriately. An alternative approach is to construct samples using sequential importance sampling (Bayati et al., 2010, Chen, 2007, Snijders, 1991, Blitzstein and Diaconis, 2011, Zhang and Chen, 2013).

Why Conserve Degrees?

The most obvious starting point for a null model would be the directed Erdős-Rényi model. However, this implies that node degrees are i.i.d. Binomial, and in particular that all nodes have the same expected degrees. In practice, degree distributions are rarely

binomial, and are instead often heavy-tailed. This is a problem because the prevalence of many graph structures is tied to heterogeneity between nodes, and in particular the degree distribution. Practitioners are typically not interested in structure that arises purely as an artifact of this, and are instead looking for evidence of higher-order processes governing the formation of the network. Since the Erdős-Rényi model cannot faithfully model degree distributions, it does not provide an adequate baseline with which to compare real networks to. By including d^- and d^+ as sufficient statistics in (3.1), the parameters α and β can explicitly account for nodal heterogeneity, making P_θ more suitable as a null model.

3.3.2 Extending to Weighted Graphs

In the weighted case, a natural question is whether the strength sequences could substitute for the degrees in (3.1). This approach has been proposed in the statistical mechanics literature, and is often referred to as the *weighted configuration model* (Squartini et al., 2011, Serrano and Boguñá, 2005, Serrano et al., 2006). It fails to faithfully model the topology of real networks. When w_{uv} is continuous, all mass is on complete networks. When integer-valued, the probability of each edge existing approaches one for most real networks. The upshot is that degrees are important for conveying a graph's topology, and should be used *in addition* to the strengths. Therefore, we consider

$$\mathrm{P}_\theta(G) := \kappa(\theta)^{-1} \exp\left(\alpha^t d^- + \beta^t d^+ + \phi^t s^- + \psi^t s^+\right), \tag{3.3}$$

which is an exponential family and an extension of (3.1). $\kappa(\theta)$ denotes the normalizing constant and $\theta := (\alpha^t, \beta^t, \phi^t, \psi^t)^t$. We assume that $\phi_u < 0$ and $\psi_u < 0$ for all $u \in N$ for reasons that will soon be clear. This model has appeared in Mastrandrea et al. (2014b), where it was employed to reconstruct networks from node-level data. Following these authors, we refer to it as the *directed enchanced configuration model* (DECM).

Clearly, this model is neither elegant nor parsimonious. For any given node, there is likely to be high correlation between its fixed effects, and one naturally wonders whether the many parameters could be reduced by, for example, positing a simple functional relationship between degrees and strengths. The point, however, is that the model is general; by including fixed effects for both strengths and degrees it contains as sub-models many reasonable processes governing nodal-heterogeneity. This generality is essential for controlling for nodal effects when testing for higher-order processes that might explain network formation.

Unlike most exponential random graph models, the model is tractable and has a simple edge-level interpretation. An edge exists with probability

$$P\{w_{uv} > 0\} = \frac{e^{\alpha_u + \beta_v}}{e^{\alpha_u + \beta_v} + \lambda_{uv}}, \tag{3.4}$$

where $\lambda_{uv} := -\phi_u - \psi_v$. A link is more likely to form if u and v are (topologically) sociable and popular, respectively. The probability increases with ϕ_u and ψ_v, showing that edge formation also depends on the strength parameters.

Conditional on the edge uv existing, its weight w_{uv} follows an exponential distribution with rate λ_{uv}. The constraints on ϕ and ψ ensure that this is positive. The exponential distribution is *memoryless*, and so the probability of reinforcing an existing link by one unit is

$$P\{w_{uv} \geq x + 1 \mid w_{uv} \geq x\} = e^{\phi_u + \psi_v},$$

for all $x > 0$. Since this is invalid when a link does not exist (i.e. when $x = 0$) there is a different cost for reinforcing an edge as opposed to forming a new edge. This permits network sparsity, and makes the model more suitable for modeling real networks than the weighted configuration model.

As in the unweighted case, the task is to use (3.3) as a null model for quantifying the significance of a property of interest, which is measured by a statistic $T : \mathcal{G} \to \mathbb{R}$. Heuristic approaches have been proposed for this, using maximum likelihood estimation (Mastrandrea et al., 2014b, Gabrielli et al., 2019). Since (3.3) is an exponential family, the MLE $\hat{\theta}$ can be found numerically as the solution to the $4n$ coupled equations given by setting observed sufficient statistics to their expectation. It is then straightforward to generate independent samples from the model with parameter $\hat{\theta}$. The observed statistic t_0 can then be compared to the sampled networks.

The aforementioned approach does not consider uncertainty around the MLE. In analogy to Section 3.3.1, a more formal approach considers the extended model

$$\mathrm{P}_{\theta,\delta}(G) := \kappa(\theta, \delta)^{-1} \exp\left(\alpha^t d^- + \beta^t d^+ + \phi^t s^- + \psi^t s^+ + \delta T(G)\right), \tag{3.5}$$

and formulates the problem as assessing the probability of observing t_0 given that $\delta = 0$. Within the likelihood framework, one approach is to appeal to Wilks' theorem, which states that the likelihood ratio statistic is, under regularity conditions, asymptotically Chi-squared with one degree of freedom. Unfortunately, the conditions required to apply Wilks' theorem, and indeed even for appealing to the asymptotic consistency of the MLEs, do not hold in this model. The data $\{w_{uv}\}$ are not identically distributed under (3.3), and the number of independent parameters grow linearly with n. This observation has been made repeatedly for the unweighted case (3.1) (Holland and Leinhardt, 1981, Snijders, 1991, McDonald et al., 2007), but to our knowledge has received little attention in articles using (3.3).

Recall that the optimal test of $\delta = 0$ conditions on the sufficient statistics. The resulting null model would then be uniform on

$$\mathcal{G}(d_0, s_0) := \{G \in \mathcal{G} : d(G) = d_0, s(G) = s_0\},$$

where $d_0 := d(G_0)$ and $s_0 := s(G_0)$. This is the set of graphs conserving both degree and strength sequences exactly. Sampling uniformly from this set is, in general, a difficult problem, and we are not aware of any methods that have been proposed to achieve this. For this reason, our approach is to *approximately condition* on the degrees, and consider instead the set

$$\mathcal{G}_m(d_0, s_0) := \{G \in \mathcal{G} : \|d(G) - d_0\|_\infty \leq m, s(G) = s_0\}, \tag{3.6}$$

for $m > 0$. This maintains strengths exactly, and keeps all node degrees within m of the observed values.

3.4 The General Setup

We have motivated the task of sampling from the conditional distribution of (3.3) given degrees and strengths. Indeed, this is the focus of the chapter. Nonetheless, other reasonable weighted null models exist. For example, Palowitch et al. (2018) recently introduced the *continuous configuration model*, which is a weighted extension of the Chung-Lu model (Chung and Lu, 2002a,b). Since alternatives could be used, we extend the discussion of the previous section to allow for other null models.

The general setup is as follows. We observe a graph $G_0 \in \mathcal{G}$ and wish to measure the significance of structures in the graph. This is done by comparing G_0 to graphs from a null model P on $\mathcal{G}$. This may prohibit edges in a set $\mathcal{F} \subseteq N^2$. That is, $G \in \mathcal{G}$ only if $w_{uv} = 0$ for all $uv \in \mathcal{F}$. This is typically employed to disallow self-loops, but can also be used to match any pattern of non-edges, including none at all.

We do not use P directly, and instead wish to condition on $d_0 := d(G_0)$ and $s_0 := s(G_0)$ to control for nodal heterogeneity. As mentioned, doing this exactly is a difficult problem. Instead, we opt to *approximately* condition on the degrees. As we will see, this provides enough 'slack' to construct an MCMC sampler to draw graphs from the distribution. Define, for each integer $m > 0$, the function

$$d_m(G) := \mathbb{1}_{N_m}(d(G)), \tag{3.7}$$

where $N_m := \{d' : \|d' - d_0\|_\infty \leq m\}$ is a neighborhood of d. Conditioning on this leads to graphs where each node has degrees that are within m of the same node in G_0. For a given $m > 0$, the target distribution of our sampler conditions P on the functions d_m and s. Its support is $\mathcal{G}_m(d_0, s_0)$.

We now briefly discuss the assumptions made on the null model before turning to the main task; that of constructing the MCMC sampler.

3.4.1 Assumptions on the Null Model

Typically, we will prefer to view P as defined on the space of weight matrices $[0,\infty)^{n\times n}$ rather than on $\mathcal{G}$ directly. We allow P to exhibit sparsity. That is, it is assumed that P has a density f with respect to

$$\Lambda := \sum_{A\in\{0,1\}^{n\times n}} \lambda_A, \tag{3.8}$$

where λ_A is $\|A\|_0$-dimensional Lebesgue measure on $\{w \in [0,\infty)^{n\times n} : w^0_{uv} = a_{uv}\}$. These sets are disjoint and so $\{\lambda_A\}$ are mutually singular. The requirement ensures that if an edge exists, i.e. if $w_{uv} > 0$, then it is continuous. It also permits network sparsity by allowing different topologies to have positive probability.

3.5 Randomizing Weighted Graphs

Sampling from $\mathcal{G}_m(d_0, s_0)$ is difficult because the space is highly constrained. Here, we develop a Markov chain approach to the problem. The algorithm is inspired by the rewiring chains that are already widely applied in the literature for unweighted graphs. It relies on repeatedly applying local moves, referred to as k-cycles (Gandy and Veraart, 2016).

3.5.1 Introducing k-cycles

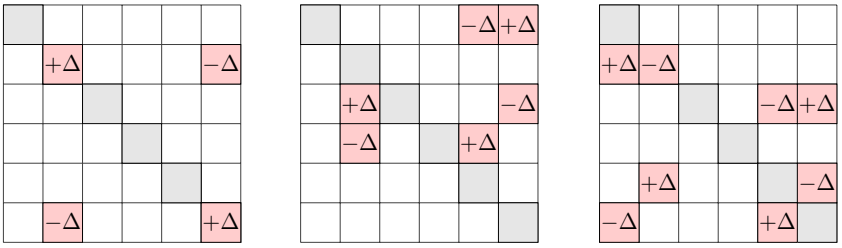
Consider the following 'rewiring' update used to randomize simple unweighted directed graphs while preserving degrees exactly. Select two edges u_1v_1 and u_2v_2 uniformly at random. If u_1, u_2, v_1 and v_2 are not all distinct, or if either of u_1v_2 or u_2v_1 are already edges, then reject and start again. Otherwise, remove u_1v_1 and u_2v_2 from the edge set and replace them with u_1v_2 and u_2v_1. This local procedure is applied continually to randomize the network.

Here, we introduce analogous updates for weighted graphs, referred to as k-cycles. These originally appeared in Gandy and Veraart (2016). First, fix two vectors of distinct nodes $(u_1,\dots,u_k)$ and $(v_1,\dots,v_k)$, where k is between 2 and n. A k-cycle attempts to update the weight matrix along the $2k$ coordinates

$$(u_1v_1, u_1v_2, u_2v_2, \dots, u_kv_k, u_kv_1), \tag{3.9}$$

conditional on all other values. Figure 3.1 depicts examples of these coordinates for different k. If $k = 2$ then four edges are potentially updated, which is similar to the rewiring move in unweighted graphs. It turns out, however, that we need to allow longer updates $k > 2$ to ensure irreducibility of the Markov chain.

Let $w := (w_1, w_2, \dots, w_{2k})^t$ refer to the delineated edge weights corresponding to the coordinates (3.9). Throughout this chapter, we refer to these weights as the *cycle-weights*.



(a) Updates along $\{22, 26, 66, 62\}$.

(b) Updates along $\{45, 42, 32, 36, 16, 15\}$

(c) Updates along $\{36, 35, 65, 61, 21, 22, 52, 56\}$.

Fig. 3.1 Example k-cycles of different lengths on a graph with 6 nodes. Self-loops are disallowed, as indicated by the gray squares. Therefore, the proposed cycles in Figures (a) and (c) would be rejected.

These values must be updated to remain within $\mathcal{G}_m(d_0, s_0)$. In particular, conserving the strengths is equivalent to maintaining the marginals of the weight matrix. Because all edges outside a k-cycle are considered fixed, conserving strengths is equivalent to conserving the consecutive sums

$$(w_1 + w_2, w_2 + w_3, \ldots, w_{2k-1} + w_{2k}), \tag{3.10}$$

exactly. Figure 3.1 should help to convince the reader of this statement. We argue in Section 3.5.2 that any update to w must take the form $w + a\Delta$, where $a = (+1, -1, +1, \ldots, -1)^t$ and where the scalar Δ lies within a bounded interval that we are yet to define. This fact is also visualized in Figure 3.1.

Relationship to Rewiring Moves

For intuition, we clarify the relationship between rewiring moves in unweighted graphs and 2-cycles. First, assume that the weight matrix of an unweighted graph is synonymous with its adjacency matrix. A 2-cycle would update the coordinates $\{u_1v_1, u_1v_2, u_2v_2, u_2v_1\}$. If in fact $w = (1, 0, 1, 0)^t$, then setting $\Delta = -1$ leads to a new value $(0, 1, 0, 1)^t$, performing the same edge replacement as a rewiring move. $\Delta = 0$ corresponds to rejecting a move, and would occur if say $w = (1, 1, 1, 0)^t$. In unweighted graphs Δ could only ever lie in $\{-1, 0, 1\}$, rather than within a real interval as in the weighted case.

3.5.2 Conditional Distribution along a k-cycle

So far, we have characterized k-cycles as updating certain subsets of the weight matrix while keeping all other entries fixed. Their definition is incomplete, as we are yet to describe how to make them reversible with respect to the target distribution. Since k-cycles are block updates of the weight matrix, it suffices for them to be reversible with respect to the *full conditionals* of the updated entries (cycle-weights). Here we derive these full conditionals.

Our approach to finding the full conditional of a given k-cycle consists of two steps. We first condition the model P on non-cycle weights (entries of the weight matrix not delineated by the k-cycle). In the second step, we proceed to condition on the strengths and degrees. Formally

- Denote by Q the conditional distribution of P given all non-cycle weights.
- Condition Q on the node strengths, followed by the node degrees to obtain the full conditional.

It is important to recognize that Q *is not the full conditional* of interest, because it does not yet account for node strengths and degrees. Q lives on $(\Omega_{2k}, \mathcal{B}(\Omega_{2k}))$, where $\Omega_d := [0, \infty)^d$ is the d-dimensional non-negative orthant.

For intuition, we give an example of Q in the case that the model P is the DECM (defined by (3.3)). In this case

$$Q = \prod_{uv} \left(\frac{e^{\alpha_u + \beta_v}}{e^{\alpha_u + \beta_v} + \lambda_{uv}} Q_{uv} + \frac{\lambda_{uv}}{e^{\alpha_u + \beta_v} + \lambda_{uv}} \delta_{\{0\}} \right),$$

where the product is over the coordinates (3.9). Notation here is as in Section 3.3.2. Q_{uv} is the exponential distribution with rate λ_{uv}, and $\delta_{\{0\}}$ is the Dirac measure at zero.

We now proceed with the second step, which is to condition Q on the nodes strengths and then, approximately, on the degrees.

Conditioning on node strengths

As argued in Section 3.5.1, conditioning Q on the strengths is the same as conditioning on the consecutive sum (3.10). This sum is formalized as a statistic $T : \Omega_{2k} \to \Omega_{2k-1}$. Rigorously proving the conditional distribution of Q given T is a difficult task, because Q is neither fully discrete nor continuous. This motivates a general definition of the conditional distribution, which is as follows.

Definition 3.5.1 (Conditional Distribution)**.** *A family $\mathcal{Q} := \{Q_t : t \in \Omega_{2k-1}\}$ of probability measures on $\mathcal{B}(\Omega_{2k})$ is the conditional probability distribution of Q given T if*

1. *$Q_t\{T \neq t\} = 0$ for TQ-almost all t in Ω_{2k-1}, and*
2. *if $g : \Omega_{2k} \to \mathbb{R}$ is nonnegative and measurable then $t \mapsto \int g(x) Q_t(\mathrm{d}x)$ is measurable and*

$$\int g(x) Q(\mathrm{d}x) = \int \int g(x) Q_t(\mathrm{d}x) TQ(\mathrm{d}t). \tag{3.11}$$

First consider the level sets $\{T = t\}$ on which each Q_t lives. It is easy to see that $\{T = t\}$ is a closed line segment L_t that can be parameterized by

$$L_t(\Delta) := x + \Delta a, \tag{3.12}$$

where x is an arbitrary element of L_t and $a := (+1, -1, +1, \ldots, -1)^t$ is the alternating vector described in Section 3.5.1. The scalar parameter Δ must lie in $[\Delta_l, \Delta_u]$ where $-\Delta_l$ and Δ_u are the smallest odd and even elements of x respectively. The boundary points are $x_l := L_t(\Delta_l)$ and $x_u := L_t(\Delta_u)$.

Proposition 3.5.2 states the conditional distribution. The proof is provided in Appendix B.1. The proposition defines each Q_t in terms of another distribution μ_t on $(\Omega_{2k}, \mathcal{B}(\Omega_{2k}))$, whose support is L_t. This is defined through

$$\mu_t(B) := \frac{\int_{L_t \cap B} f(x)ds}{\int_{L_t} f(x)ds},$$

for each $B \in \mathcal{B}(\Omega_{2k})$. Recall that P has density f with respect to (3.8), In the above equation, $f(x)$ denotes f evaluated at the weight matrix implied by letting the cycle-weights take the value x, and keeping all other entries the same. Both integrals in this expression are line integrals, and the denominator serves as a normalizing constant.

Proposition 3.5.2. *Fix any $t \in \Omega_{2k-1}$. If the boundary points satisfy $\|x_l\|_0 = \|x_u\|_0 = 2k-1$ then let*

$$Q_t := \frac{1}{\kappa_t}\left(f(x_l)\delta_{\{x_l\}} + f(x_u)\delta_{\{x_u\}} + \alpha_t \mu_t\right), \tag{3.13}$$

where $\alpha_t = \frac{1}{\sqrt{2k}} \int_{L_t} f(x)ds$ and $\kappa_t := f(x_l) + f(x_u) + \alpha_t$. Otherwise, let

$$Q_t := \begin{cases} \delta_{\{x_l\}} & \text{if } \|x_l\|_0 < \|x_u\|_0 \\ \delta_{\{x_u\}} & \text{if } \|x_l\|_0 > \|x_u\|_0 \\ \kappa_t^{-1}\left(f(x_l)\delta_{\{x_l\}} + f(x_u)\delta_{\{x_u\}}\right) & \text{if } \|x_l\|_0 = \|x_u\|_0 < 2k-1, \end{cases} \tag{3.14}$$

where $\kappa_t := f(x_l) + f(x_u)$. The collection $\mathcal{Q} := \{Q_t : t \in \Omega_{2k-1}\}$ is the conditional distribution of Q given T.

Approximate conditioning on degrees

Suppose now that we consider degrees *in addition* to the strengths, i.e. we wish to approximately condition Q_t on the degrees. To formalize this, first fix an arbitrary $x \in L_t$ and let G_x refer to the graph obtained by letting the cycle-weights take the value x, whilst keeping all other weights fixed. We then condition on $x \mapsto d_m(G_x)$, which is a map from $L_t \to \{0, 1\}$. This statistic depends implicitly on the topology of the graph outside the k-cycle, which is, of course, fixed. We assume that this topology is such that

$$\{x \in L_t : d_m(G_x) = 1\}, \tag{3.15}$$

is non-empty. The set of graphs for which this is empty is π-negligible because the graphs cannot lie within $\mathcal{G}_m(d_0, s_0)$. Therefore, this case can be safely ignored.

Conditioning Q_t on $x \mapsto d_m(G_x)$ is equivalent to restricting it to (3.15), i.e. the points at which the associated graph has degrees close enough to the target vector. These graphs can have one of at most three topologies. If $x_l \neq x_u$ then the topologies of G_{x_l} and G_{x_u} are different, because the zero elements of x_l and x_u are distinct. If x_1 and x_2 are both in the interior of L_t, then the topology of G_{x_1} and G_{x_2} are the same, because all entries in x_1 and x_2 are positive. This shows that conditioning may assign zero probability to either of the boundary points, or to the entire interior of L_t.

Special attention should be given to the case where (3.15) is Q_t-negligible. This would happen, for example, if x_l has more zeros than x_u, but also $d_m(G_{x_l}) = 0$. Another possibility is that x_l and x_u have the same number of zeros, but more than one, and $d_m(G_{x_l}) = d_m(G_{x_u}) = 0$. In both cases, it is easy to see that all points in (3.15) must have ties among positive elements, which is a π-negligible event. Therefore, the conditional distribution can be defined arbitrarily in this case.

Example: Conditional Distribution for the DECM

Here we specialize to the DECM. The resulting conditional distribution will be easy to sample directly, providing a convenient way to perform k-cycles.

First, suppose that x_l and x_u each have one zero entry. To compute the line integral appearing in the conditional Q_t, first observe that $L_t(\Delta + \mathrm{d}\Delta) = L_t(\Delta) + a\,\mathrm{d}\Delta$ for any $\Delta \in (\Delta_l, \Delta_u)$, and so

$$\mathrm{d}s = \|L_t(\Delta + \mathrm{d}\Delta) - L_t(\Delta)\|_2 = \sqrt{2k}\,\mathrm{d}\Delta,$$

where $\mathrm{d}s$ is the differential on L_t. Therefore,

$$\begin{aligned}\int_{L_t} f(x)\,\mathrm{d}s &= \sqrt{2k}\int_{\Delta_l}^{\Delta_u} f(x + a\Delta)\,\mathrm{d}\Delta \\ &= \sqrt{2k} f(x + a\Delta^*)(\Delta_u - \Delta_l),\end{aligned}$$

where $\Delta^* \in (\Delta_l, \Delta_u)$. Here we have used that $f(x+a\Delta) = f(x+a\Delta^*)$ for all $\Delta \in (\Delta_l, \Delta_u)$. This is true because f depends only on degrees and strengths, which are invariant over such Δ. It is also easy to verify that

$$f(x_l) = e^{-(\alpha_{u_1} + \beta_{v_1})} f(x + a\Delta^*) \tag{3.16}$$

$$f(x_u) = e^{-(\alpha_{u_2} + \beta_{v_2})} f(x + a\Delta^*), \tag{3.17}$$

where u_1v_1 and u_2v_2 are the edges corresponding to the zero weights in x_l and x_u respectively. Putting this together, (3.13) reduces to

$$Q_t := \frac{1}{\kappa_t}\left(e^{-(\alpha_{u_1} + \beta_{v_1})}\delta_{\{x_l\}} + e^{-(\alpha_{u_2} + \beta_{v_2})}\delta_{\{x_u\}} + (\Delta_u - \Delta_l)\mu_t\right),$$

Algorithm 3: A k-cycle for the DECM.

Input: G, z, $\hat{\alpha}$ and $\hat{\beta}$;
if $z \cap \mathcal{F} \neq \emptyset$ **then return** G;
$x \leftarrow W_z(G)$;
$\Delta_l \leftarrow -\min_i(x_{2i+1})$ and $\Delta_u \leftarrow \min_i(x_{2i})$;
if $\Delta_u = \Delta_l = 0$ **then return** G;
Let z_l and z_u be edges corresponding to elements of $x + a\Delta_l$ and $x + a\Delta_u$ that are zero respectively;
$n_l \leftarrow |z_l|$ and $n_u \leftarrow |z_u|$;
$p_l \leftarrow p_u \leftarrow p_{\text{int}} \leftarrow 0$;
if $d_m(G_{x+a\Delta_l}) = 1$ *and* $n_l \geq n_u$ **then** $p_l \leftarrow \prod_{uv \in z_l} e^{-\hat{\alpha}_u - \hat{\beta}_v}$;
if $d_m(G_{x+a\Delta_u}) = 1$ *and* $n_u \geq n_l$ **then** $p_u \leftarrow \prod_{uv \in z_u} e^{-\hat{\alpha}_u - \hat{\beta}_v}$;
Let $\Delta^* \in (\Delta_l, \Delta_u)$;
if $d_m(G_{x+a\Delta^*}) = 1$ *and* $n_u = n_l = 1$ **then** $p_{int} \leftarrow \Delta_u - \Delta_l$;
$p^* \leftarrow p_l + p_u + p_{\text{int}}$;
if $p^* = 0$ **then return** G;
$u \sim \text{Unif}[0, p^*]$;
if $u < p_l$ **then** $\Delta \leftarrow \Delta_l$;
else if $u < p_l + p_u$ **then** $\Delta \leftarrow \Delta_u$;
else $\Delta \sim \text{Unif}(\Delta_l, \Delta_u)$;
return $G_{x+a\Delta}$;

NOTES: G is the current state of the chain, while z is a set of coordinates of the form (3.9). $\hat{\alpha}$ and $\hat{\beta}$ are n-vectors. Here, G_x refers to the graph obtained by assigning weights x to edges along the k-cycle, i.e. edges in z.

where μ_t is the uniform distribution on L_t, and $\kappa_t = e^{-(\alpha_{u_1} + \beta_{v_1})} + e^{-(\alpha_{u_2} + \beta_{v_2})} + \Delta_u - \Delta_l$. Q_t in the remaining cases (shown in 3.14) are found similarly. Of course, Q_t is not the distribution of interest, as we also need to approximately condition on the degrees. This is straightforward and consists of restricting Q_t to the appropriate parts of L_t, as was outlined in Section 3.5.2. Direct sampling from both Q_t and Q_t given approximate degrees is straightforward. For general densities, however, the line integral would typically need to be computed by numerical integration. Furthermore, direct sampling from μ_t may not be possible, and could require more sophisticated methods like rejection sampling.

3.5.3 Performing the k-cycle

We are now ready to describe the full k-cycle for the DECM. π-invariance is automatically satisfied since we sample directly from the full conditional. One remaining issue, however, is that the parameter vectors α and β in (3.3) are unobserved. The full conditional still depends on these because the degrees have not been conditioned on exactly. Nonetheless, their influence is small when conditioning on d_m for small m. One option is to assume $\alpha = \beta = 0$ so that they do not appear in the distribution. Another option is to estimate them via maximum likelihood. Algorithm 3 gives pseudocode for the algorithm, which uses some assumed values $\hat{\alpha}$ and $\hat{\beta}$.

3.5.4 Combining k-cycles

This section introduces an auxiliary variable method of selecting k-cycles. The k-cycle chosen at each iteration *depends on the current state of the chain*. This allows for better mixing in both sparse and dense graphs.

Motivating kernel selection

To motivate our method, first recall the rewiring moves (discussed in Section 3.5.1) used for randomizing unweighted directed graphs. The move selects two edges u_1v_1 and u_2v_2 randomly and attempts to replace them with u_1v_2 and u_2v_1. This is only possible if both u_1v_2 and u_2v_1 are not already in the edge set. If the network is sparse, then the edge replacement has a high probability of succeeding. If it were dense, however, similar performance could be achieved by instead selecting u_1v_1 and u_2v_2 from the set of non-edges, rather than edges.

Closely related to the rewiring chains is a random walk that operates directly on the graph's adjacency matrix. This is often referred to as a checkerboard swap or tetrad move (Artzy-Randrup and Stone, 2005, Stone and Roberts, 1990, Verhelst, 2008, Rao et al., 1996, Diaconis and Gangolli, 1995). It selects a 2×2 submatrix at random and attempts to modify it with either

$$\begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} -1 & +1 \\ +1 & -1 \end{pmatrix},$$

and rejects if the resulting adjacency matrix is invalid. When successful, the move performs the same update as a rewiring move. However, the nodes are in effect chosen randomly, and so the performance will be poor in both sparse and dense graphs as the rejection rate is prohibitively high.

These local moves can be seen as *Markov kernels*, and the method of selecting them is referred to as *kernel selection*. The above discussion highlights the impact that kernel selection has on the practical efficiency of the resulting chain. Such considerations are exacerbated in the context of k-cycles; Proposition 3.5.2 shows that for P-almost all graphs, a k-cycle is unable to propose a new graph if there is more than one zero weight along the cycle.

The naive approach would be to first sample $k \in \{2, \dots, n\}$ and then each of $(u_1, \dots, u_k)$ and $(v_1, \dots, v_k)$ uniformly from the node set N without replacement. The k-cycle is then formed as in (3.9). This is analogous to the checkerboard/tetrad moves previously discussed, and is the approach used in Gandy and Veraart (2016). In sparse graphs, however, the chance of only one zero cycle-weight is small, and the sampler can be prohibitively slow. Our aim in this section is to define a better strategy.

Cycle selection as an auxiliary variable

The method of selecting a k-cycle can be interpreted as an auxiliary variable. Formally, let $\mathcal{Z}$ be the index set of all possible k-cycles, which corresponds to the collection of all sets of the form (3.9). Note that permutations of (3.9) are considered equivalent here. We want the selected cycle $Z \in \mathcal{Z}$ to depend on the current state of the chain $G \in \mathcal{G}_m(d_0, s_0)$. Therefore, we let $Z \sim q_G$ where q_G is the *state-dependent distribution* of Z.

Selecting k-cycles this way does not generally maintain π-invariance. For this, we must extend the state space to include the selection variable, and consider the properties of the Markov chain on the joint space. Formally, define the product space $\mathcal{Z} \times \mathcal{G}_m(d_0, s_0)$. The iterated integrals

$$\tilde{\pi}(g) := \int \int g(z, G) q_G(\mathrm{d}z) \pi(\mathrm{d}G),$$

for all non-negative Borel-measurable g define a distribution $\tilde{\pi}$ on the joint space. Starting from (z, G), the extended chain first samples $z' \sim q_G$, and proceeds to update the edge weights along the k-cycle defined by z'. If the extended chain is $\tilde{\pi}$-invariant then the marginal chain on $\mathcal{G}_m(d_0, s_0)$ is π-invariant. To maintain $\tilde{\pi}$-invariance, we simply need to adjust the full conditional of the weights along a k-cycle to additionally condition on Z.

An efficient selection strategy

We prioritize selecting cycles that have some chance of moving the chain to a new state. The main limitation of naively selecting k-cycles is sparsity. We therefore select new nodes by constructing an 'alternating' cycle of out-edges and in-edges. We require two neighborhood sets associated with each node. These are

$$N_G^-(u) := \{v \in N : a_{uv} = 1\}$$
$$N_G^+(u) := \{v \in N : a_{vu} = 1\},$$

which are the out-neighbours and in-neighbours of u respectively. We start by sampling k according to some distribution Γ on $\{2, \ldots, n\}$. This should be positive everywhere to improve the stochastic stability of the sampler. We then sample $u_1 v_1$ uniformly from the set of all edges in the graph. Starting from u_1, the remaining nodes are sampled by alternately walking through the out-neighbours and in-neighbours of the previous node. The full strategy is shown in Algorithm 4.

If the algorithm terminates at line 10, the resulting k-cycle has at most one zero weight and no fixed edges. If instead it returns $\emptyset$, then the strategy has failed to select a k-cycle and the Markov chain remains at the current state. If k is small in comparison to the size of the network, nodes generally have more than two edges, and the pattern of prohibited edges $\mathcal{F}$ is not particularly complex, then the chance of failing to find a k-cycle is small.

Algorithm 4: k-cycle selection strategy.

Input: G;
1 $k \sim \Gamma(\{2, \dots, n\})$;
2 $u_1 v_1 \sim \text{Unif}(\{uv : a_{uv} = 1\})$;
3 **for** $l = 2$ *to* k **do**
4 **if** $d^-_{u_{l-1}}(G) \leq 1$ **then return** $\emptyset$;
5 $v_l \sim \text{Unif}(N^-_{u_{l-1}}(G) \setminus \{v_{l-1}\})$;
6 **if** $d^+_{v_l}(G) \leq 1$ **then return** $\emptyset$;
7 $u_l \sim \text{Unif}(N^+_{v_l}(G) \setminus \{u_{l-1}\})$;
8 **if** $\exists i, j$ *distinct such that either* $u_i = u_j$ *or* $v_i = v_j$ **then return** $\emptyset$;
9 **if** $u_k v_1 \in \mathcal{F}$ **then return** $\emptyset$;
10 **return** $\{u_1 v_1, u_1 v_2, \dots u_k v_k, u_k v_1\}$

NOTES: G is the current state of the chain. Line 9 only checks if $u_k v_1$ is a prohibited edge because all other edges have positive weights, which by assumption implies they are not prohibited (see Section 3.4).

Algorithm 5: One iteration of the complete sampler.

Input: G, $\hat{\alpha}$, $\hat{\beta}$
1 $z \leftarrow$ output of Algorithm 4 applied to G;
2 **if** $z = \emptyset$ **then return** G;
3 $G' \leftarrow$ output of Algorithm 3 to G, z $\hat{\alpha}$ and $\hat{\beta}$, but adding $p_l \leftarrow \gamma_l p_l$ and $p_u \leftarrow \gamma_u p_u$ after line 9 ;
4 **return** G'

NOTES: Input are the same as in Algorithm 3. The adjustment to Algorithm 3 in line 3 accounts for the state-dependent selection of the k-cycle.

3.5.5 The Overall Sampler

One iteration of the full sampler tries to select a k-cycle with Algorithm 4. If successful, it then samples cycle-weights from its full conditional. Recall, however, that this must be adjusted to also condition on Z, which is the cycle selection variable.

To achieve this, let $z \in \mathcal{Z}$ be a cycle chosen by Algorithm 4, and recall the notation where G_x refers to the graph obtained by allowing cycle-weights to take the value x. For such cycles $\Delta_u - \Delta_l > 0$. Let γ_l and γ_u be the ratio of the probability of selecting z from G_{x_l} and G_{x_u} to the chance of selecting it from some graph $G^* := G_{x+a\Delta}$ satisfying $\Delta \in (\Delta_l, \Delta_u)$. Assume also that there are no positive ties along the cycle-weights (positive ties are P-null). Then by following Algorithm 4, one can deduce that

$$\gamma_l = \frac{M}{M-1} \frac{(d^-_{u_1}(G^*) - 1)(d^+_{v_1}(G^*) - 1)}{\sum_{uv \in z}(d^-_u(G^*) - 1)(d^+_v(G^*) - 1)} \tag{3.18}$$

$$\gamma_u = \frac{M}{M-1} \frac{(d^-_{u_2}(G^*) - 1)(d^+_{v_2}(G^*) - 1)}{\sum_{uv \in z}(d^-_u(G^*) - 1)(d^+_v(G^*) - 1)}, \tag{3.19}$$

where M is the total number of edges in G^*, and $u_1 v_1$ and $u_2 v_2$ are the edges corresponding to the zero weights in x_l and x_u respectively. Conditioning on the cycle selection strategy simply requires adjusting the boundary probabilities by the factors γ_l and γ_u. This is shown in Algorithm 5, which presents the full sampler.

3.6 Stochastic Stability

Here we attempt to provide conditions under which the chain we have introduced is *ergodic*; i.e. that it admits a unique invariant distribution. All proofs are provided in Appendix B. Ergodicity justifies the use of Monte Carlo averages through Birkhoff's ergodic theorem, which states that if $\{G_l\}_{l=0}^{\infty}$ is a Markov chain with unique invariant distribution π, and h is integrable, then

$$\frac{1}{L}\sum_{l=0}^{L-1} h(G_l) \to \mathbb{E}(h(G))$$

as $L \to \infty$, and where the expectation is taken under π.

Throughout this section, we fix degrees and strengths (d, s) and some $m > 0$, and consider the chain designed to sample from $\mathcal{G}_m(d, s)$. Proving irreducibility (Definition 3.6.3) in the general case is difficult, and we can only provide results for $m \geq n$, i.e. when there is in effect no conditioning on degrees. Further work is required to establish conditions for $m < n$. Nonetheless, our simulations appear to show that the sampler can traverse many topologies even when $m = 1$, and is capable of rapidly reaching the mode of π when the initial state is far in the tail of π. An example of this is provided in Section 3.7.1.

Now assume that $m \geq n$. It turns out that the chain is not ergodic for all strength sequences. Nonetheless, ergodicity holds for strength sequences produced by *P-almost all* graphs. To formalize this idea, let $\{U_i \times V_i \subseteq N^2\}_{i\in I}$ be a collection of non-empty and distinct sets for which

$$\sum_{u\in U_i} s_u^- = \sum_{v\in V_i} s_v^+, \tag{3.20}$$

and such that there does not exist non-empty $U \times V \subset U_i \times V_i$ on which (3.20) holds. Let $\tilde{\mathcal{G}}_m(d, s) \subseteq \mathcal{G}_m(d, s)$ be the set of graphs for which uv is an edge only if $uv \in U_i \times V_i$ for some $i \in I$. Ergodicity will hold for admissible strengths, as defined in Definition 3.6.1.

Definition 3.6.1 (Admissible Strengths)**.** *The strength vector s is admissible if $\{U_i\}_{i\in I}$ and $\{V_i\}_{i\in I}$ each partition N and $\tilde{\mathcal{G}}_m(d, s)$ is non-empty.*

Notice that (3.20) is always satisfied for $U_i = V_i = N$. If these are the unique sets satisfying (3.20), then admissibility simply requires that the reference set is non-empty.

Proposition 3.6.2 verifies that if the network is generated from some law absolutely continuous with respect to P, then the observed strengths will not be inadmissible. It also has implications for the topology of graphs that produced the strengths.

Proposition 3.6.2. *The set of graphs producing inadmissible sequences is P-negligible, where P is as defined in Section 3.4. Moreover, for any admissible s the set of graphs not in $\tilde{\mathcal{G}}_m(d, s)$ is P-negligible.*

Loosely speaking, ergodicity of a chain requires that it is irreducible, aperiodic and recurrent. By construction, the chain is aperiodic and has a unique invariant distribution, which will imply that it is recurrent. Therefore, the work is in demonstrating the property of irreducibility, defined as follows.

Definition 3.6.3 (φ-irreducibility)**.** *A Markov chain on $(\mathcal{X}, \mathcal{B})$ with kernel Φ is φ-irreducible if there exists a measure φ on $\mathcal{B}$ such that for all $x \in \mathcal{X}$ and A for which $\varphi(A) > 0$, there exists some $n > 0$ satisfying $\Phi^n(x, A) > 0$.*

Definition 3.6.3 shows that one can choose an arbitrary measure φ when establishing irreducibility. If the property is satisfied, then there exists a unique (up to null sets) 'maximal' irreducibility measure ψ, in the sense that any other irreducible measure must be absolutely continuous with respect to ψ. For more details on this, see Meyn et al. (2009, Chapter 4). The next proposition ties irreducibility to admissibility of the strength sequence.

Proposition 3.6.4. *If the strength vector s is admissible, then the resulting Markov chain is φ-irreducible.*

Suppose s is admissible. By construction, π is an invariant distribution of the chain. Since the chain is also ψ-irreducible, it is recurrent (Meyn et al., 2009, Proposition 10.1.1) and the invariant distribution is unique (Meyn et al., 2009, Proposition 10.4.4). This distribution is then the maximal irreducibility measure. This discussion is formalized in the following corollary.

Corollary 3.6.4.1. *If s is admissible, then the resulting chain has π as a unique invariant distribution.*

3.7 Experiments

Here we assess the performance of the sampler introduced in Section 3.5. The sampler was coded in C++, and all experiments were performed on an Intel Core i5 2GHz CPU. We first empirically analyze its efficiency in Section 3.7.1. This will demonstrate its ability to randomize large networks. In Section 3.7.2, we use the sampler as a null model for detecting patterns in weighted networks and compare its performance to competing methods.

3.7.1 Efficiency of the Sampler

We use the sampler to randomize a large, sparse, and highly structured network. The randomization maintains strengths exactly and keeps all node degrees within $\pm$ 1 of the initial network. The graph to be randomized has $n = 10^3$ nodes, 250 of which are assigned as 'core' nodes, and 750 as 'periphery' nodes. The core is partitioned into 5 cliques of 50 nodes, while the periphery is partitioned into 75 cliques of 10 nodes. The

subgraph of each clique is complete; that is every node has a directed link to all other nodes in the community. All 80 clique are connected by a single bridge to the rest of the network. Specifically, we add 79 'bridge' links by creating a single link from the first clique to the second, a link from the second to the third, etc. Weights of all edges are sampled independently from the exponential distribution with mean 10^3.

The adjacency matrix of this network is shown in the top left panel of Figure 3.2. This initial state is far from the mode of the conditional distribution, which is concentrated on matrices similar to that in the bottom right panel. The network is particularly difficult to randomize because creating links between cliques requires first choosing a k-cycle that includes a bridge link. Nonetheless, the sampler reached the network in the bottom right panel in under 35 seconds. We also attempted the randomization without the auxiliary kernel selection method of Section 3.5.4, instead selecting cycles as in Gandy and Veraart (2016). However, this approach was unable to reach the mode within a reasonable time. This demonstrates the importance of the state-dependent kernel selection for the efficiency of the sampler.

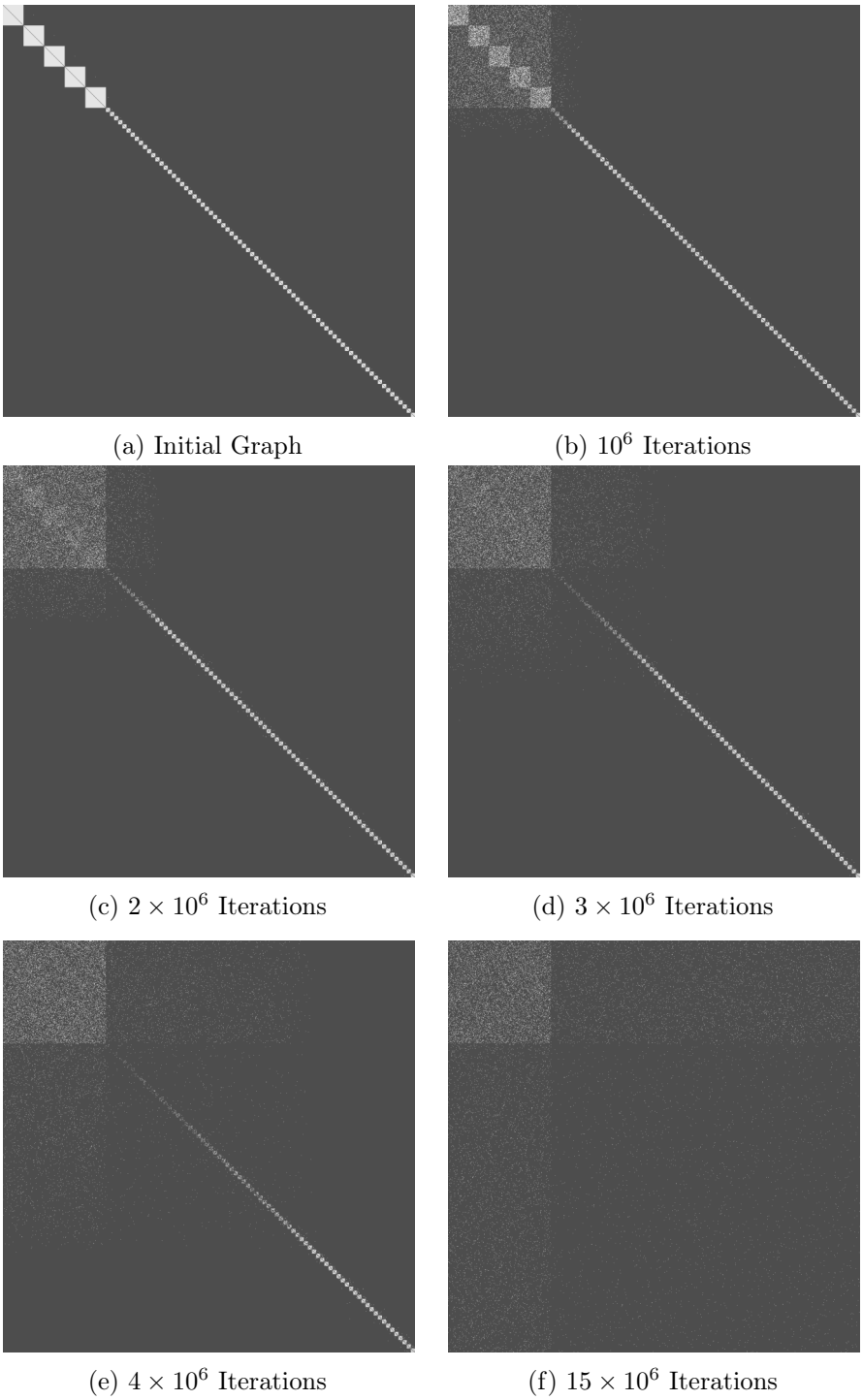
Recall that in Section 3.6 we considered the irreducibility of the chain. We were, however, only able to obtain results for $m \geq n$, i.e. when there is no conditioning on the degrees. This experiment has almost exactly conditioned on the degrees, and shows that the sampler remains capable of rapidly randomizing the network, and also of traversing different graph topologies. Although this certainly does not constitute a proof of irreducibility, it warrants further research in this direction.

3.7.2 Significance of Community Structure in Benchmark Networks

This section uses the sampler to assess community structure in simulated networks, and compares the method's performance to alternative null models in a power study. The ground-truth community structure in the networks is known. Such 'ground-truth' networks are usually simulated and are often termed benchmark models. Below, we introduce the benchmark model used and detail the parameters used for the simulation. We then describe the power study, competing methods and present the results.

Degree and Strength Corrected Stochastic Block Model

An early benchmark for unweighted and undirected graphs was suggested by Girvan and Newman (2002). Although simple, it does not account for heterogeneous community sizes and degrees. Modeling realistic degree distributions, which are heavy-tailed, is critical to the suitability of a benchmark. Heavy-tailed degrees can lead algorithms to group nodes with large degrees, irrespective of their true memberships (Karrer and Newman, 2011). Lancichinetti et al. (2008) introduced the LFR benchmark, which overcomes these shortcomings. This was extended to weighted and directed graphs in Lancichinetti and Fortunato (2009). Here we use a benchmark model that accounts for heterogeneous group sizes, degrees and strengths. The benchmark is simple and similar in spirit to the



(a) Initial Graph

(b) 10^6 Iterations

(c) 2×10^6 Iterations

(d) 3×10^6 Iterations

(e) 4×10^6 Iterations

(f) 15×10^6 Iterations

Fig. 3.2 Adjacency matrices of networks at different stages of the randomisation.

weighted stochastic block models (WSBMs) proposed in Aicher et al. (2015) and Palowitch et al. (2018).

The model is a straightforward generalization of the null model introduced in Section 3.3.2, with an additional parameter to control tendency towards clustering. It is defined for $n > 1$ nodes and $K > 1$ possible community assignments. Recall from Section 3.3.2 the n-vectors α, β, ϕ and ψ, where we constrained $\phi_u < 0$ and $\psi_u < 0$. Let c be a n-vector representing a community partition of the nodes, so that $c_u \in \{1, \cdots, K\}$. The strength of community structure is controlled by a scalar parameter $\theta \geq 1$ that, roughly speaking, represents the relative edge formation probability (or average edge weights) for intra-community compared to inter-community links. Each potential edge uv is associated with a factor $\theta_{c_u c_v}$, which is equal to θ if $c_u = c_v$, and is otherwise 1.

We describe the model at the edge-level, which will suggest a generative approach to drawing samples from it. As in Section 3.3.2, edges are assumed conditionally independent given the parameters, and thus this description will fully define the likelihood for the network. The probability that an edge forms between two distinct nodes is

$$P\{w_{uv} > 0\} = \min\left(\frac{e^{\alpha_u+\beta_v}}{e^{\alpha_u+\beta_v} + \lambda_{uv}}\theta_{c_u c_v}, 1\right), \tag{3.21}$$

where $\lambda_{uv} = -\phi_u - \psi_v$. Conditional on edge existence, the weights are exponentially distributed with mean

$$\mathbb{E}[w_{uv} \mid w_{uv} > 0] = \frac{\theta_{c_u c_v}}{\lambda_{uv}}.$$

This model can be seen as a stochastic block model generalized to account for a wide range of degree and strength distributions. When $\theta = 1$ it collapses to the null model of Section 3.3.2. The model can be extended in multiple ways. We have assumed no background nodes and no overlapping communities. For possible ways to extend in this direction, please see Palowitch et al. (2018). In addition, θ could be replaced with group-specific parameters.

Simulation Parameters

The distribution of group sizes, degrees and strengths are chosen to reflect the heavy-tailed nature of these quantities in real networks. For this, we follow an approach that is close to Lancichinetti and Fortunato (2009). Formally, we iteratively draw group sizes from a discrete power law with exponent α_1 truncated to between $s_{\min}$ and $s_{\max}$. Continue drawing new communities until the sum of all sizes is at least n, and then scale sizes proportionately until the total size is n. We then randomly assign nodes to the communities, such that the group sizes are respected.

We now describe all parameter values used in the simulations. The simulation requires applying our sampler thousands of times to different simulated networks. For this reason, we consider only relatively small networks by letting $n = 10^2$. For drawing group sizes, we

let $\alpha_1 = 2$, $s_{\min} = n/5$ and $s_{\max} = 3s_{\min}/2$. The parameter θ, which induces community structure, will be varied on a grid to assess the power of different methods at detecting deviations from the null.

Competing Null Models

The proposed method is compared to two alternative null models. The first is a weighted version of the Erdős-Rényi model (WER). Let G be a draw from the benchmark model, and $a_T := \sum_{u,v} a_{uv}(G)$ and $w_T := \sum_{u,v} w_{uv}(G)$ be the number of edges and total edge weights in G respectively. WER draws independent and identically distributed edges according to

$$P\{w_{uv} > 0\} = \frac{a_T}{n(n-1)},$$

with u and v distinct. Weights are then exponential with mean

$$\mathbb{E}(w_{uv} \mid w_{uv} > 0) = \frac{w_T}{a_T}.$$

The second model considered is the *continuous configuration model* (CCM) introduced in Palowitch et al. (2018). This is a weighted extension of the Chung-Lu model (Chung and Lu, 2002a,b) and, unlike WER, has the advantage of matching the degrees and strengths of G in expectation. Edges are formed independently with probability

$$p_{uv} := P\{w_{uv} > 0\} = \min\left(\frac{d_u^-(G)d_v^+(G)}{a_T}, 1\right),$$

and weights are exponential with mean

$$\mathbb{E}(w_{uv} \mid w_{uv} > 0) = \frac{s_u^-(G)s_v^+(G)}{w_T}\frac{1}{p_{uv}}.$$

CCM must permit self-loops, else the degrees and strengths of G are not correctly matched.

The Power Study and Results

The study is split into two phases. In both parts, we are interested in assessing the power of the competing methods at correctly detecting community structure, where the level of such structure is controlled by θ. This parameter will be varied from no clustering to levels where the clustering is quite apparent. Formally, we consider $\theta \in \{\theta_1, \ldots, \theta_L\}$ where $1 = \theta_1 < \ldots < \theta_L$. For each method and θ_l, we repeatedly complete the following three steps.

- Draw G according to the benchmark model for θ_l, and with all other parameters as described in Section 3.7.2. Compute $t_0 = T(G)$, where T is some statistic measuring the strength of clustering.

- Draw samples $G^{(1)}, \ldots, G^{(N)}$ using the method and let $t_1, \ldots, t_N$ be the associated test statistics.
- Compute the empirical significance (p-value) as in (3.2).

This process is repeated 5×10^3 times to obtain a distribution over the significance statistics. If a method performs well, then the p-values should be roughly uniformly when $\theta = 1$ and have high power for $\theta > 1$.

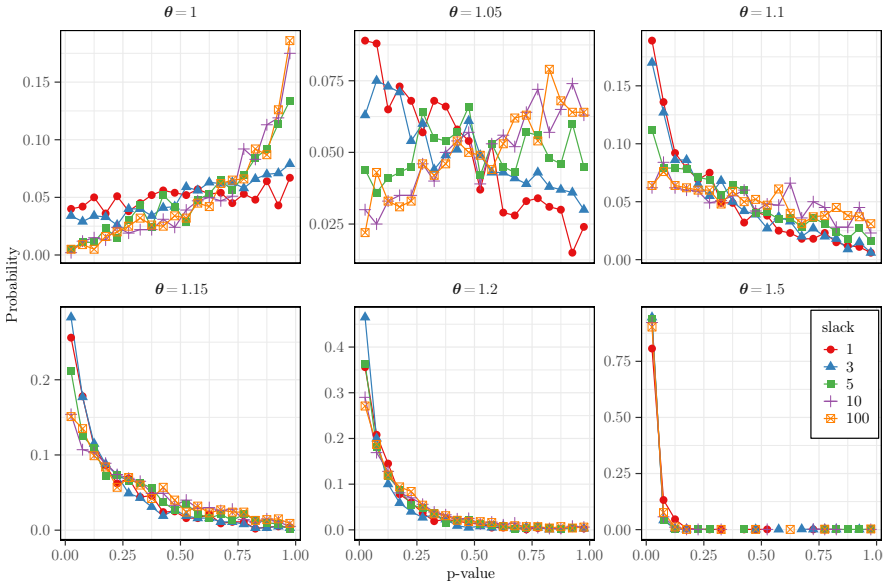
The first phase of the study pretends that the true communities in the benchmark graphs are unknown, and applies a standard community detection algorithm to recover the structure. For this, we employ *WalkTrap* (Pons and Latapy, 2005), however note that numerous alternatives could be used instead. The algorithm returns a graph partition, and we let T be modularity computed on this partition. Figure 3.3 shows the comparative performance of different methods as θ is varied from 1 to 2 in increments of 0.2. The figure shows that the proposed method outperforms the competing null models. In particular, when $\theta = 1$ the null model which conditions tightly on degrees (± 1) is close to uniform, as desired. This is not the case for either WER or CCM. Our method has a power advantage over both WER and CCM when the clustering effect is quite slight. This is expected: conditioning can act to improve relevance to the data at hand, and improve power against subtle alternatives. All methods perform well when θ is large.

The second phase looks to better understand the effect that approximate conditioning on the degrees has on the performance of the method.To illustrate this, we use a statistic that is deliberately sensitive to graph density. This is

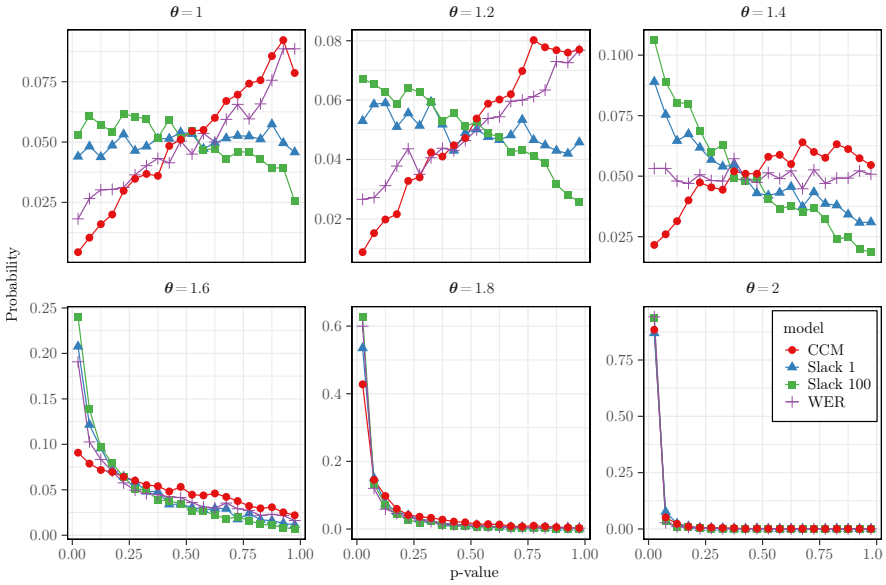
$$T(G) := \sum_{u,v} a_{uv}(G)\delta(c_u, c_v),$$

where δ is the Kronecker delta function. This measures total within-community edges. In practice, this statistic would not be used because we have modularity, which explicitly measures clustering *relative* to the configuration model and thus accounts for degree distributions. Nonetheless, it is not possible to make general graph statistics invariant to degree distributions, and so this example still has strong practical implications.

Figure 3.3 presents the results for the second phase. When $\theta = 1$, none of the null models are exactly uniform and there appears to be a bias towards high p-values. This is expected, as T is highly sensitive to degrees. Nonetheless, when degrees are conditioned ± 1 we get quite close to uniform because the effect of the unknown parameters (which were estimated by MLE) is minimized. Again, we see that approximate conditioning improves power against subtle alternatives.



(a) First phase.



(b) Second phase.

Fig. 3.3 Comparative performance of different null models in the power study. *Slack m* refers to the null model where node degrees are maintained $\pm m$. When $m = 1$ degrees are almost exactly conditioned on. *Slack 100* in effect performs no degree conditioning.

3.8 Discussion

This chapter has suggested a null model for weighted graphs. The model fixes node strengths and approximately fixes node degrees to within ± 1 of the values of an observed network. It can be employed to assess the statistical significance of patterns observed in networks. We have proposed an MCMC sampler for drawing samples from the model, and have shown empirically that it is capable of sampling large and sparse networks. This sampler uses similar techniques to those employed in Chapter 2 to achieve this. We performed an extensive power study to compare the performance of the null model to alternatives. The model compares favorably and appears capable of detecting subtle patterns, while also effectively controlling for nodal heterogeneity. The work in this chapter can be extended in a number of ways: a discussion of these is deferred to Chapter 7.

4

Exact Tests for the Correctness of MCMC and Other Monte Carlo Methods

The previous two chapters developed novel MCMC methods for conditional graph sampling. These rely on involved derivations of conditional distributions, and of theoretical properties. Such algorithms have a large scope for error, either in these derivations, or in the actual implementation of the sampler. This raises a general question of how we can test that the method indeed has the correct invariant distribution. This chapter develops new approaches for this, which, we believe, should be used as a routine part of a Bayesian workflow. The proposed methods are generally applicable to both MCMC algorithms and other Monte Carlo methods.

When constructing unit tests for MCMC samplers, there is a sensitivity/specificity trade off. The tests we introduce are unique in being exact; we can bound the type 1 error from above. This allows us to embed the test in a sequential framework to make the type 1 error arbitrarily small while maintaining high power in detecting genuine errors.

4.1 Introduction

Markov chain Monte Carlo methods are the main workhorse of Bayesian statistics. These methods are used to approximate posterior expectations which are otherwise analytically intractable. While there exist numerous diagnostics to assess convergence of the Monte Carlo estimates to *some* value, few articles address whether they converge to the *correct* values (Geweke, 2004, Cook et al., 2006, Talts et al., 2020).

MCMC often requires difficult derivations of marginal and conditional distributions (Geman and Geman, 1984), and derivatives of log densities (Roberts and Stramer, 2002, Duane et al., 1987, Girolami and Calderhead, 2011). Increasingly sophisticated algorithms raise the scope for analytic errors in these derivations, as well as of implementation errors. Testing for such errors should be an integral and routine part of the workflow of any Bayesian analysis using MCMC. This chapter proposes new hypothesis tests

to accomplish this. These tests are unique in being exact; they have guaranteed false rejection probability, which can theoretically be made as small as desired.

MCMC algorithms yield dependent samples, limiting the usefulness of existing procedures for detecting sampler errors (Geweke, 2004, Cook et al., 2006, Talts et al., 2020). This is because the exact distribution of the test statistics under the null measure is not known, and as a consequence, there is no guarantee over the false rejection probability. This has important practical implications. The obvious consequence is that a researcher applying the methods cannot always determine whether the test failed because of dependency between samples, or alternatively because of actual sampler errors that require further investigation. This could lead to a waste of valuable researcher time if they try to find errors that do not exist. Alternatively, errors could go undetected as they are explained away by correlation between samples.

One solution to sample dependency is to thin the chain, i.e. to subsample at given intervals to obtain approximately independent samples. This is the technique suggested in Talts et al. (2020). The integrated autocorrelation time is the number of steps required for a chain to forget its initial state. If this can be estimated well, then subsampling can be used to yield effectively independent samples. Unfortunately, reliable estimation of the quantity is widely considered challenging (Sokal, 1997). The target distribution is often multi-modal and incomplete sampling can lead to underestimating the quantity. Even supposing access to a good estimate, it will often be too large to be of practical use.

Many sampler errors can be detected in fewer iterations than required for independent samples. Because independence is not required, we can detect these faster than alternate methods, making the tests more efficient in many practical scenarios. The two suggested tests use ideas already present in the literature. One test relies on ideas suggested in Besag and Clifford (1989). The theoretical results of this paper are extended by allowing ties in the observations and a more general definition of ranks. The other test generalizes a method proposed in Gandy and Veraart (2016) to test a specific sampler.

We envisage that the new methods would be particularly useful in unit testing of MCMC and Monte Carlo methods. Unit testing is a standard part of the software development process (Runeson, 2006). Individual units of a piece of software are being tested, To demonstrate their functionality. Frameworks for implementing unit tests are available in many programming languages (Wikipedia contributors, 2019), for example Wickham (2011) in the R language. Tests are generally re-run after changes to the software, to ensure continued functionality. There can be a substantial number of tests in any piece of software - so it is important to keep the computational effort reasonable for a test that passes. Once a test fails, debugging of the code will usually be needed to pinpoint (and fix) the source of the error.

When used for unit testing, the tests for MCMC chains could be used for individual types of updates of e.g. a Gibbs sampler or of a reversible jump MCMC sampler. The

tests are constructed to test if the chain (or the step of the chain) has the correct invariant distribution. It is not testing if the chain is recurrent.

When using the above-mentioned methods or other (goodness-of-fit) tests based on simulated data in unit testing, one faces a trade-off between the false rejection rate, the power, and the sample size. Typically, one would like to have a (very) low false rejection probability, as investigating potential errors is time-consuming. Also, as mentioned above, the computational effort if no errors are present should be low. This immediately places bounds on the alternatives that one can detect. We present a sequential method that improves the position in this regard. It sequentially executes the test, and repeats the test only if the test yields moderate evidence for departure. This sequential approach is useful for general Monte Carlo tests and not just the two MCMC approaches.

Previous methods introduced to tackle this problem are discussed in Section 4.1.1. Section 4.2 proposes the new exact tests for MCMC samplers. Section 4.3 discusses how to embed exact tests into a sequential testing procedure to increase power and reduce the false rejection rate. As mentioned, this is useful for unit testing and applies to more general Monte Carlo methods. Section 4.4 presents a simulation study comparing our approach to previous methods. Section 4.5 applies our methods to an RJ-MCMC algorithm proposed by Andrieu and Doucet (1999). Conclusions are summarized in Section 4.6. The tests have been implemented in an easy to use R-package that immediately slots into the existing unit testing framework for R (Wickham, 2011). This is available at https://bitbucket.org/agandy/mcunit. Proofs can be found in Appendix C.1.

4.1.1 Related Literature for Testing Samplers

Geweke (2004) was the first article to formally consider the problem of detecting errors in MCMC samplers. Their method compares samples obtained using two techniques for drawing from the joint distribution of parameters and data. The first simulates directly from the generative model. The second is a Gibbs sampler, alternating between drawing parameters given data (using the MCMC sampler) and data given parameters. Z-tests are used to compare estimates of moments of the joint distribution. The downside of this approach is that the Gibbs sampler will generate dependent samples. In practical applications, the parameters and data can be highly correlated, and a high computational effort is required to control the false rejection rate.

Cook et al. (2006) propose tests based on sampled posterior quantiles in the Bayesian framework. The authors crucially observe that drawing θ from the prior and y from the likelihood implies that θ is an exact sample from the posterior given y. A sample $\theta_{1:L}$ from this posterior distribution is simulated using the sampler to be tested, and the empirical quantile of θ is computed among this sample. Unfortunately, the suggested limiting distribution of this quantile is incorrect (Gelman, 2017), and the proposed tests are not applicable when there is sample dependency, as is the case with MCMC.

Talts et al. (2020) proceed identically to Cook et al. (2006), but instead of using the empirical quantile of θ among $\theta_{1:L}$, they compute its rank. Due to discretization effects, the empirical quantile cannot be expected to be uniform on $[0, 1]$, however if the samples are independent and continuous, then the rank statistic is exactly uniform on $\{1, ..., L\}$. Repeating this procedure multiple times gives a sample of ranks which can be compared to this uniform distribution. Rather than constructing a formal test, the authors advocate visually assessing goodness of fit using histograms. The authors propose using thinning to deal with dependent samples when using an iterative simulator like MCMC. Unfortunately, this leaves the method prone to the aforementioned problems associated with subsampling Markov chains.

4.2 Exact Tests for Errors in MCMC Samplers

In this section, we describe two tests for detecting sampler errors for MCMC samplers. Analogous tests for simple Monte Carlo methods would be standard statistical tests such as goodness-of-fit tests.

Assume parameters $\theta \in \Theta$ and data $y \in \mathcal{Y}$ are modeled as a product of prior and likelihood $\pi(\theta)p(y \mid \theta)$, and that one can independently draw parameters from the prior and data from the likelihood.

Further, assume that the MCMC implementation is designed to work for all possible data $y \in \mathcal{Y}$. In a Bayesian analysis, we would observe y_{obs} and construct a Markov chain with kernel $K_{y_{obs}}$ to estimate expectations of functions with respect to the posterior $\pi(\cdot \mid y_{obs})$. If the data is implemented as an argument, then the sampler is a collection of kernels $\{K_y : y \in \mathcal{Y}\}$ such that each K_y is expected to have invariant distribution $\pi(\cdot \mid y)$.

This motivates the null hypothesis that K_y is $\pi(\cdot \mid y)$-invariant for all $y \in \mathcal{Y}$. The tests do not specifically check $K_{y_{obs}}$, but rather the viability of the sampler over all possible data values. For example, if only the kernels corresponding to a null set of data has errors, then the tests would not be able to detect this.

The null hypothesis will be false if there are errors in the sampler, broadly characterized as either *design* or *implementation* errors. Design errors correspond to having a wrong model for the sampler, and may include mistakes in derived quantities required for sampling, or a mistake in understanding of how a particular sampler works. Implementation errors refer to an incorrect execution of a given design, regardless of whether that design is correct. These are likely to be errors in the written code.

Both proposed methods are essentially goodness of fit tests which compare a computed sample of statistics to another distribution. By exact, we mean to say that the distribution of the sample is exactly known under the null hypothesis. We do not mean to imply that the p-value is computed exactly. In practice, cheaper inexact methods may be used to compute the p-values; for example, using a χ^2 test in the discrete case. This is of little

consequence because the sample size can be explicitly controlled in the test. For a large enough sample, the p-value will be as if exact.

Our tests are not designed to investigate the mixing behavior or the ergodicity of the Markov chain. A Markov chain can be correctly implemented yet slow mixing. Researchers wishing to diagnose slow mixing can instead refer to the vast literature on the subject (Cowles and Carlin, 1996). Properties required for full ergodicity, including irreducibility and aperiodicity, are typically easy to establish for continuous distributions, and may require proof otherwise.

Section 4.2.1 details a basic test which uses the Markov chain to yield samples which should be indistinguishable from independent samples drawn from the generative model under the null. This idea generalizes a method described in the supplementary material of Gandy and Veraart (2016) to test a specific MCMC sampler. Section 4.2.2 considers a more elaborate test based on uniformity of rank statistics. This uses ideas from Besag and Clifford (1989).

4.2.1 Exact Two-Sample Tests

This method samples from the model in two different ways. The first simply samples directly using the generative model, while the second starts by sampling directly, but then propagates the sample parameters L steps forward using the MCMC sampler. Formally, samples are generated with the sequence of steps

$$\begin{aligned}\theta' &\sim \pi(\cdot),\\ y' &\sim p(\cdot \mid \theta'),\\ \theta &\sim K_{y'}^{L}(\theta', \cdot).\end{aligned}$$

θ' is a perfect sample from $\pi(\cdot \mid y')$, and so initiating the chain at θ' implies that θ is also exactly from the posterior under the null hypothesis. Since y' is marginally correct, this implies that θ is unconditionally a sample from the prior. Moreover, if the procedure is repeated, each sample will be independent. Samples generated this way are described as *fitted* samples, while those generated directly from the model are *direct* samples. Algorithm 6 details the generation of these samples.

Any appropriate goodness of fit test can be employed to compare the fitted and direct samples. Under the null, these are independent and identically from the joint distribution of data and parameters. The most appropriate test to use will depend on the alternative hypotheses considered, and so we avoid prescribing a specific test here. If the samples are continuous, one may use the two-sample Kolmogorov-Smirnov test, the Cramer-von Mises test or the Wilcoxon signed rank test. If discrete, a likelihood ratio test or the Pearson's χ^2-test could be used. If the form of the prior is particularly simple, there may be no need to sample from it, and one could instead use a parametric one-sample test.

Algorithm 6: General algorithm to perform a two-sample test as described in Section 4.2.1.

1 **for** $n = 1$ *to* N_1 **do**
2 Draw $\tilde{\theta} \sim \pi(\cdot)$;
3 Draw $\tilde{y}_n \sim p(\cdot \mid \tilde{\theta})$;
4 Run Markov chain L steps from $\tilde{\theta}$ to obtain $\tilde{\theta}_n \sim K^L_{\tilde{y}_n}(\tilde{\theta}, \cdot)$;
5 **for** $n = 1$ *to* N_2 **do**
6 Draw $\theta_n \sim \pi(\cdot)$;
7 Draw $y_n \sim p(\cdot \mid \theta_n)$
8 Compare independent samples $\{(\tilde{\theta}_n, \tilde{y}_n)\}$ and $\{(\theta_n, y_n)\}$;

NOTES: N_1 and N_2 are the number of fitted and direct samples respectively.

Algorithm 6 is similar to that proposed by Geweke (2004), the key difference being that the data is resampled before each MCMC step. This guarantees independence of samples, which will be useful for controlling the false rejection rate whenever there is high correlation between data and parameters.

The method can be extended by iteratively updating both data and parameters. Line 4 could be replaced by repeating, L times, the step $\tilde{\theta} \sim K_{\tilde{y}_n}(\tilde{\theta}, \cdot)$ followed by $\tilde{y}_n \sim p(\cdot \mid \tilde{\theta})$. This is just a Gibbs sampler and, letting $\tilde{\theta}_n$ be the final parameter, $(\tilde{\theta}_n, \tilde{y}_n)$ clearly has the same distribution as (θ_n, y_n) under the null. This extension may improve power in certain circumstances, as is shown in Section 4.4.2.

4.2.2 An Exact Rank Test

Algorithm 6 may suffer low power for detecting certain errors. Sometimes, there may be mistakes in each conditional, which when aggregated are undetectable in the joint distribution of data and parameters. An example of this is shown in Section 4.4.1. Here a test is proposed that, similar to Talts et al. (2020), compares a sample of rank statistics to the uniform distribution. Each statistic is computed using multiple samples from a single posterior distribution, and so it may better detect divergences that might be averaged out in the joint.

This comes at the expense of requiring each Markov kernel K_y to be reversible with respect to $\pi(\cdot \mid y)$. This is not particularly restrictive: most MCMC algorithms are reversible by design, because showing reversibility is the easiest way to prove invariance with respect to a target distribution. Many of the most commonly used algorithms are reversible, including Metropolis Hastings, Hamiltonian Monte Carlo and slice sampling. Although samplers using composition of kernels are not reversible (for example, systematic scan Gibbs sampling), the constituent kernels can still be tested if they are each individually reversible.

Rank statistics which break ties in a vector must be considered, so that the null distribution of the rank is exactly uniform. The generalization is as follows. In the

following S_n is the set of permutations of $\{1, \ldots, n\}$, i.e. the set of vectors $s \in \{1, \ldots, n\}^n$ such that $s_i \neq s_j$ for all $i \neq j$.

Definition 4.2.1. *A function $R : \Theta^n \to S_n$ is an ordinal ranking for vectors $\theta_{1:n} \in \Theta^n$.*

Any function which can assign the same rank to two elements of a vector $\theta_{1:n}$ does not satisfy Definition 4.2.1.

The general idea behind the test is as follows. First draw θ from the prior and y from the likelihood. The kernel K_y is used to draw samples from the posterior, and the rank of θ among these samples is computed. Replicating this procedure multiple times, the resulting rank statistics will be exactly uniform under the null. Any of a number of goodness-of-fit tests can then be used. Algorithm 7 details the generation of a single rank statistic.

How the posterior samples are drawn has important implications for the uniformity of the rank statistics. Imagine, for example, using the MCMC sampler to realize a Markov chain $\theta_{1:L}$ initiated at $\theta_1 = \theta$. Given some ordinal ranking R, the null distribution of $R_1(\theta_{1:L})$ is generally not uniform on $\{1, \ldots, L\}$. Although each element of the chain is of course marginally $\pi(\cdot \mid y)$, the chain has Markovian dependence and its components are not exchangeable.

Assuming only reversibility, this can be rectified using a technique suggested in Besag and Clifford (1989), which is extended here to allow for possible ties in the Markov chain. Instead of initiating the chain from θ, sample M uniformly in $\{1, ..., L\}$ and let $\theta_M = \theta$. Then run the chain twice, once forward $L - M$ steps from θ_M, and then backwards $M - 1$ steps from θ_M. Letting R_M denote the M^{th} component of $R(\theta_{1:L})$, then by Proposition 4.2.3, R_M will be exactly uniform under the null. Before giving this proposition, a generalization of the Lemma of Besag and Clifford (1989) is stated.

Lemma 4.2.2. *Suppose $R(\theta_{1:L})$ is a random vector with values in S_L. If $M \sim$ Uniform$\{1, ..., L\}$ independently of $R(\theta_{1:L})$ then $R_M(\theta_{1:L})$ is uniformly distributed on $\{1, ..., L\}$.*

Proposition 4.2.3. *Let $R_M(\theta_{1:L})$ be the rank statistic returned from Algorithm 7. If for every y the kernel K_y is $\pi(\cdot \mid y)$-reversible then $R_M(\theta_{1:L}) \sim$ Uniform$\{1, ..., L\}$.*

The canonical example of an ordinal ranking that we have in mind first maps each component of $\theta_{1:n}$ to the real line with a function $h : \Theta \to \mathbb{R}$, computes the ranks of $h(\theta_1), \ldots, h(\theta_n)$, breaking ties in some order.

Importantly, the ordinal ranking in Algorithm 7 can be chosen based on any quantity which is independent of M. This allows, for example, randomly breaking ties as follows. If you had a collection of 'canonical' rankings, with the only difference between them being that the order of breaking ties is different, you could uniformly select a ranking from this set, thus breaking ties randomly. The ranking could also be selected based on y because it is independent of M. Specifically, the function h could be the likelihood

Algorithm 7: Computing a rank statistic using the method described in Section 4.2.2.

1 Draw $M \sim \text{Uniform}\{1, ..., L\}$;
2 Draw $\theta_M \sim \pi(\cdot)$;
3 Draw $y \sim p(\cdot \mid \theta_M)$;
4 Choose an ordinal ranking R such that R and M are independent;
5 **for** $l = 1$ *to* $M - 1$ **do**
6 $\quad \theta_{M-l} \sim K_y(\theta_{M-l+1}, \cdot)$;
7 **for** $l = M + 1$ *to* L **do**
8 $\quad \theta_l \sim K_y(\theta_{l-1}, \cdot)$;
9 **return** $R_M(\theta_{1:L})$

NOTES: L is the number of MCMC samples to use.

function mapping $\theta \mapsto p(y \mid \theta)$. This particular statistic is used in the simulation study of Section 4.4.1.

The proposed test may be generalized. In lines 5 and 7 of Algorithm 7 one could, with some fixed probability, update y given the current value of θ rather than updating θ using the Markov kernel. This would give samples on the joint space which can be compared using an ordinal ranking $R : (\Theta \times \mathcal{Y})^L \mapsto \mathcal{S}_L$. Proposition 4.2.3 still holds because this is simply testing a random scan Gibbs sampler on the joint space of parameters and data, which is of course reversible under the assumptions of the proposition. This can improve power in detecting certain subtle errors, as is shown in Section 4.4.2.

Another extension is to replace K_y by K_y^k for some $k > 1$. This has the effect of thinning the chain and reducing autocorrelation, and will be useful to increase power against more subtle alternatives. The important point, however, is that such thinning is not required for the null distribution of the ranks to be correct. Extending Algorithm 7 to multiple testing is simple.

4.3 Sequential Implementation for Unit Tests

Unit tests should have a low false rejection probability and a reasonable computational effort if the sampler works. Moreover, the tests ought to have high power and if errors exists, we should be willing to spend a larger effort detecting them. Here, a sequential testing procedure is proposed which achieves these goals.

Algorithm 8 immediately rejects the null under very low p-values, does not reject the null for higher p-values and continues simulations for p-values that are low, but not extremely low. The method runs for a maximum of k steps, and multiplies the sample size by Δ after the first iteration, which serves to detect errors more easily in subsequent iterations.

There are many possible variations on Algorithm 8. For example, one could define the probability of early rejection via "spending sequences" as in Gandy (2009). If using

Algorithm 8: Sequential wrapper around the methods.

$\beta_1 = \alpha/k$;
$\gamma = \beta^{1/k}$;
for $i = 1$ *to* k **do**
  $p^{(i)}$=vector of p-values from one of the algorithms (sample size n);
  $q_i = \min p^{(i)}/d$;
  if $q_i \leq \beta_i$ **then return** *fail*;
  if $q_i > \gamma + \beta_i$ **then** break;
  $\beta_{i+1} = \beta_i/\gamma$;
  if $i = 1$ **then** $n = \Delta n$;
return *OK*;

NOTES: d, is the dimension of the p-value vectors $p^{(i)}$; α, is the overall desired false rejection rate; k, the maximum number of sequential steps; Δ, the factor by which to multiple the sample size after the first iteration.

Algorithms 6 or 7 to generate the p-values, instead of adjusting the number of chains (through Δ), one could instead increase the amount of thinning within chains. This would also raise the power in subsequent iterations.

As mentioned, the proposed method has an overall false rejection rate of at most α, as the following theorem shows.

Theorem 4.3.1. *Suppose* $p^{(1)}, \ldots, p^{(k)}$ *are independent* d*-variate random vectors with values in* $[0,1]^d$*. If* $\mathrm{P}\{p_j^{(i)} \leq p\} \leq p$ *for all* $p \in [0,1]$ *and* $i = 1, \ldots, k$*,* $j = 1, \ldots, d$ *then* $\mathrm{P}\{\mathit{fail}\} \leq \alpha$*.*

The added effort of Algorithm 8 compared to the non-sequential case is modest if the p-values are generated under the null. Assuming that they are exactly uniform, the expected increase in iterations under the null for general k compared to $k = 1$ is

$$\Delta \sum_{i=1}^{k} \gamma^{i-1} = \Delta\gamma \left(\frac{1 - \gamma^{k-1}}{1 - \gamma} \right). \tag{4.1}$$

More generally, if only the inequality for p-values under the null is assumed (i.e. the probability of a p-value being below any bound q is at most q), then the expected increase in effort is bounded from above by

$$\Delta \sum_{i=2}^{k} \prod_{j=1}^{i-1} (\gamma + \beta_j). \tag{4.2}$$

Motivated by the simulation study in Appendix C.2, the default values for Algorithm 8 in the R-package *mcunit* are $\alpha = 10^{-5}$, $k = 7$, and $\Delta = 4$. This leads to $\gamma \approx 0.15$, and $\beta_1 \approx 1.4 \cdot 10^{-6}$, $\beta_2 \approx 9.8 \cdot 10^{-6}$, $\beta_3 \approx 6.6 \cdot 10^{-5}$, $\beta_4 \approx 4.6 \cdot 10^{-4}$, $\beta_5 \approx 3.1 \cdot 10^{-3}$ $\beta_6 \approx 2.1 \cdot 10^{-2}$, $\beta_7 = \gamma \approx 0.15$.

For these default parameter values, both (4.1) and (4.2) give the expected additional effort at around 68.5%. For $\Delta = 1$ and the other parameters unchanged, both formulas give around 17.1%. The difference between the two formulas is negligible when α is chosen to be small.

4.4 Simulations

To demonstrate the performance of the proposed and existing tests, this section presents the results of a power analysis using a stylized model, and a sampler in which errors have been intentionally introduced. The tests considered are exact two-sample and rank tests, and the methods of Geweke (2004) and Talts et al. (2020). Although Talts et al. (2020) propose graphically checking the distribution of their rank statistics to the uniform distribution, here a formal Kolmogorov-Smirnov test is used to allow consistent comparisons with other methods.

Consider the model

$$y \sim \theta_1 + \theta_2 + \varepsilon, \tag{4.3}$$

where $\theta := (\theta_1, \theta_2)$ is apriori independent, zero-mean normal with standard deviation $\sigma = 10$. The white noise term ε is independent of θ and also zero-mean normal but with variance $\sigma_\varepsilon^2 = 0.1$. While inference is easy in this model, we consider drawing posterior samples of θ using a Gibbs sampler. The posterior conditional distributions for θ_1 and θ_2 are normal with expectations

$$\mathrm{E}[\theta_i \mid y, \theta_j] = \frac{\sigma^2}{\sigma_\varepsilon^2 + \sigma^2} (y - \theta_j), \tag{4.4}$$

and variances

$$\mathrm{Var}(\theta_i \mid y, \theta_j) = \frac{1}{\frac{1}{\sigma_\varepsilon^2} + \frac{1}{\sigma^2}}. \tag{4.5}$$

The small σ_ε induces high correlation between θ_1 and θ_2 in the posterior distributions, and so the Gibbs sampler will mix slowly.

4.4.1 Mistakes in Full Conditionals

Two correctly implemented samplers are considered; one uses random scan of the two coordinates, with the other using systematic scan. Three erroneous samplers, all of which use random scan, are also considered. The first two have mistakes in the conditional expectations and variances respectively; $y - \theta_j$ is replaced with $y + \theta_j$ in (4.4), and in (4.5) the variance terms are replaced with the corresponding standard deviations. The final mistake considered truncates each conditional distribution either to the left or right of its posterior mean. The decision to truncate left or right is random for each distribution.

Table 4.1 presents the results of the power analysis. Each entry records an empirical rejection rate for a given test function(s) and scenario, computed by repeating the test

Table 4.1 Empirical rejection rates from the power analysis described in Section 4.4.1.

Test Function	Correct		Errors		
	Rand. Scan	Sys. Scan	E	Var	Truncated
Sequential exact two-sample test with $\Delta = 2$ *and* $k = 3$.					
θ_1	0.009	0.010	1.000	0.007	0.008
θ_1^2	0.008	0.009	1.000	0.009	0.011
$\theta_1\theta_2$	0.008	0.008	1.000	0.010	0.011
$\pi(\theta)$	0.010	0.011	1.000	0.008	0.009
$p(y \mid \theta)$	0.010	0.009	1.000	1.000	0.007
All [a]	**0.007**	**0.009**	**1.000**	**1.000**	**0.006**
Sequential exact rank test with $\Delta = 2$ *and* $k = 3$.					
θ_1	0.009	0.885	1.000	0.149	0.869
θ_1^2	0.009	0.869	1.000	0.163	0.868
$\theta_1\theta_2$	0.008	0.155	1.000	0.731	1.000
$\pi(\theta)$	0.009	0.158	1.000	0.738	1.000
$p(y \mid \theta)$	0.012	0.012	1.000	1.000	0.010
All [a]	**0.008**	**0.769**	**1.000**	**1.000**	**1.000**
Geweke (2004).					
θ_1	0.310	0.235	1.000	0.278	0.303
θ_1^2	0.322	0.276	1.000	0.119	0.329
$\theta_1\theta_2$	0.105	0.071	1.000	0.101	0.106
$\pi(\theta)$	0.226	0.206	1.000	0.284	0.220
$p(y \mid \theta)$	0.010	0.013	1.000	1.000	0.010
All [a]	**0.523**	**0.441**	**1.000**	**1.000**	**0.516**

NOTES: The exact two-sample tests ran with $L = 5$ and $N_1 = N_2 = 5 \times 10^2$, and KS tests were used to compare the two samples of the test statistic(s). The exact rank tests ran with $L = 5$ and had 5×10^2 simulated rank statistics, using a $\mathcal{X}^2$-test to test the ranks for uniformity. Geweke (2004) used thinning of 5 and 6×10^2 MCMC samples.
[a] Refers to using all aforementioned test functions and a Bonferroni correction for multiple testing.

10^4 times. The nominal false rejection rate of each test was set to $\alpha = 0.01$. Sequential versions of Algorithms 6 and 7 were used because they were found to have higher power than the non-sequential versions. All methods were calibrated to have comparable computational budgets. Please refer to the table for details of all simulation parameters.

As expected, the exact two-sample test (Algorithm 6) achieves the nominal rate of 0.01 for both random scan and systematic scan. The test always detected the wrong expectations and variances. However, only the data likelihood proved able to detect the wrong variance. The variance error only changes the marginal of θ slightly, and so any test using a statistic involving only the parameters will require many samples to detect the error. This illustrates the importance of considering statistics on both data and parameters, rather than just parameters, when using this two-sample test. Notice that the truncation was undetected. Even though all conditional distributions are wrong, the joint distribution of parameter and data is indistinguishable from the correct joint. Therefore, the test could never have higher power than the nominal rejection rate for the truncation error.

Table 4.2 Empirical rejection rates for the power analysis described in Section 4.4.2.

Test	Correct	$\mu = 10$	$\sigma = 5$	$\rho = 0.5$
Seq. Exact Two-Sample	0.007	0.018	0.826	0.049
Seq. Exact Rank	0.011	0.012	1.000	0.551
Exact Two-Sample	0.008	0.012	0.601	0.025
Exact Rank	0.011	0.009	1.000	0.316
Geweke (2004)	0.101	0.909	1.000	0.229
Talts et al. (2020)	1.000	1.000	1.000	1.000

NOTES: Reported results are for multiple testing using all test functions shown in Table 4.1. The seq. two-sample test used $L = 50$ and $N_1 = N_2 = 10^3$, while the seq. rank test used $L = 10$, 5 thinning steps between samples and 10^3 rank statistics. Both versions used $\Delta = 2$ and $k = 3$. The non-sequential versions were adjusted to achieve a similar computational time under the null. Geweke (2004) used 10^3 samples with thinning of 50, and Talts et al. (2020) used 10^2 initial steps to estimate ESS.

The exact rank test, as described in Section 4.2.2, achieved the nominal rate for the random scan Gibbs sampler, however was unable to do so for systematic scan. This is expected as the systematic scan sampler is not reversible. The multiple test always detected the wrong expectation, variance and truncation.

The correlation between the data and parameters poses a problem for the method of Geweke (2004), and the false rejection rate is too high. The test rejects the correct samplers roughly half of the time. Again, the truncation cannot be detected by this method, for the same reason as for the exact two-sample test.

Finally, Algorithm 2 from Talts et al. (2020) was run using 10^3 initial steps in each chain to estimate the effective sample size. Because the posterior correlation is high in this model, the effective sample size was overestimated, and the false rejection rate was entirely uncontrolled. Given that the errors can be detected easily, this method is highly inefficient in the cases considered.

4.4.2 Mistakes in Assumed Prior

The second simulation investigates the power of the tests when mistakes are made in the assumed prior for θ. In all cases considered, the prior is a bivariate normal with common mean μ, standard deviation σ and correlation ρ. As described at the beginning of Section 4.4 the correct version corresponds to $\mu = 0$, $\sigma = 10$ and $\rho = 0$. Three erroneous priors are considered; a mean shift to $\mu = 10$, a variance scale to $\sigma = 5$, and dependency with $\rho = 0.5$. As before, all tests were parameterized to have comparable computational effort and the nominal false rejection rate set to $\alpha = 0.01$. The results are displayed in Table 4.2, which also details the simulation parameters.

Both the exact two-sample and rank tests did well to maintain the nominal rate, and had high power in detecting the scaled variance. They were unable to detect the mean shift because the prior is uninformative and has little effect on the posterior distributions. It seems that the joint distribution tests of Geweke (2004) has a power advantage here

because the marginal distribution of the parameters in the samples will tend to the specified wrong prior as the number of MCMC steps goes to infinity. Nonetheless, the false rejection rate is far above the nominal level in our simulation. The method was also worse than Algorithm 7 at detecting the dependency. It appears that the joint distribution tests of Geweke (2004) can perform comparatively well when errors in individual posterior distributions are subtle, but aggregate in such a way that they are detectable in the simulated joint distribution. The errors in this section are designed such that the Geweke (2004) method will, if run long enough, recover a specified (wrong) joint distribution. In more general cases, it is not clear that the errors will be so easily detectable in the joint. Talts et al. (2020) performed poorly, again due to the autocorrelation in the Gibbs sampler. It rejected every scenario in every test. Obtaining reasonable results using this method would require much more computation than required in the other tests.

Finally, we demonstrate how to improve the power of the exact tests for the above analysis. Recall the extension to the two-sample test where the data is resampled each time θ is updated. The rank test described in Section 4.2.2 was also extended, so that with probability 0.5, the data y rather than θ is updated in line 5 and 7 of Algorithm 7. The power of these generalized methods are estimated under the wrong prior expectation $\mu = 10$ introduced above. The two-sample test was parameterized to used $L = 2 \times 10^3$ and $N_1 = N_2 = 10^3$, while the rank test used $L = 10$, thinning of 200 and 10^3 rank statistics. The empirical ejection rates were 99.2% and 97.2% respectively, computed by replicating each test 10^3 times. The empirical rejection rates for the original tests were 30.8% and 29.7% respectively, when both were parameterized to have similar computation time. The power improves because the generalizations define Gibbs samplers on the joint space, and so they have higher power in detecting the 'aggregated' error in the joint. Nonetheless, for this amount of computation the method of Geweke (2004) achieved the nominal false rejection rate, and had power of 100%.

4.5 Application: Reversible-Jump MCMC for Signal Decomposition

Andrieu and Doucet (1999) propose a Bayesian method to jointly detect and estimate sinusoids in a noisy signal. The number of sinusoids making up the signal is unknown, and the authors propose a reversible-jump MCMC (RJ-MCMC) algorithm to explore the space of models consisting of different numbers of sinusoids. This seminal work precipitated a number of studies applying RJ-MCMC to signal processing problems. Many of these relied on the same Metropolis-Hastings-Green acceptance ratio for 'birth' and 'death' moves that was derived in Andrieu and Doucet (1999), including Andrieu et al. (2001b,a), Larocque and Reilly (2002), Larocque et al. (2002), Ng et al. (2005), Davy et al. (2006), Hong et al. (2010), Schmidt and Mørup (2010), Rubtsov and Griffin (2007).

Roodaki et al. (2013) demonstrate that this ratio is, in fact, erroneous. Through simulation, the authors show that the error leads the sampler to prefer 'sparse' models with fewer sinusoids. In this section, we first briefly outline both the model and the sampler proposed in Andrieu and Doucet (1999). After discussing the error noted by Roodaki et al. (2013), we employ the exact tests introduced in Section 4.2 to show that this error could easily have been detected with our methods. This example demonstrates the utility of routine use of such tests in advance of publishing results that rely on estimation by MCMC.

4.5.1 Model Description

Consider a data vector $y := (y_0, \dots, y_{N-1})^t$, which may aggregate multiple sinusoidal signals in addition to random noise. Andrieu and Doucet (1999) propose a series of competing models to explain such data, which are indexed by the number $k \geq 0$ of latent sinusoids hidden within the noisy signal. The k^{th} $(k > 0)$ model is

$$y_t = \sum_{l=1}^{k} \left(c_{k,l} \cos(w_{k,l} t) + s_{k,l} \sin(w_{k,l} t)\right) + \varepsilon_{k,t},$$

where $\varepsilon_{k,t}$ is white Gaussian noise with variance σ_k^2. The zero$^{\text{th}}$ model corresponds to no latent signal, i.e. $y_t = \varepsilon_{0,t}$. It is convenient, particularly when we discuss the priors, to express these models in matrix-vector form

$$y = \mathrm{D}(w_k) a_k + \varepsilon_k,$$

with radial frequencies $w_k := (w_{k,1}, \dots, w_{k,k})^t$, amplitudes $a_k := (c_{k,1}, s_{k,1}, \dots, c_{k,k}, s_{k,k})^t$, and noise $\varepsilon_k := (\varepsilon_{k,0}, \dots, \varepsilon_{k,N-1})^t$. The $N \times 2k$ design matrix $\mathrm{D}(w_k)$ has odd-column entries $\mathrm{D}(w_k)_{t+1,2l-1} = \cos(w_{k,l} t)$ and even-column entries $\mathrm{D}(w_k)_{t+1,2l} = \sin(w_{k,l} t)$.

The number of latent signals k is given a Poisson prior with rate Λ, truncated to $\{0, \dots, k_{max}\}$, where $k_{max} = \lfloor (N-1)/2 \rfloor$. This upper limit prevents dependence in the columns of D, which would render w_k difficult to identify. Conditional on k, the remaining priors are

$$\begin{aligned} \sigma_k^2 \mid k &\sim \text{InvGamma}(v_0/2, \gamma_0/2) \\ w_k \mid k &\sim \text{Uniform}((0, \pi)^k) \\ a_k \mid k, w_k, \sigma_k^2 &\sim \mathcal{N}(0, \delta^2 \Sigma_k), \end{aligned}$$

where v_0 and γ_0 are shape and scale parameters, and $\Sigma_k = \sigma_k^2 \left(\mathrm{D}^t(w_k)\mathrm{D}(w_k)\right)^{-1}$. The prior on a_k is known as the *g-prior*.

4.5.2 The Sampler and its Error

Andrieu and Doucet (1999) integrate a_k and $\delta := \sigma_k^2$ out of the posterior and target $p(k, w_k|y)$, which is known up to a normalizing constant, and defined on $\Theta := \cup_{k=0}^{k_{max}} \{k\} \times (0, \pi)^k$. Their sampler employs four Markov kernels. Two are *within-model* kernels, designed to update the frequencies w_k while keeping k fixed. The remaining two, namely the 'birth' and 'death' moves, are *between-model* kernels, and propose moves that traverse subspaces of different dimensions; adding and deleting sinusoids respectively.

The within-model kernels are standard Metropolis Hastings updates and alter only one component of w_k at a time. The first perturbs the current state with a symmetric Gaussian proposal with scale σ_{rw}, and is referred to hereon-in as the local frequency kernel (LFK). The other uses a global proposal based on the Fourier coefficients of y. These are computed with the discrete Fourier transform, however y can be padded to length $N_p > N$ to improve interpolation. We refer to this as the global frequency kernel (GFK).

Birth and death kernels allow the chain to transition from dimension k to $k+1$ or $k-1$ respectively. Suppose the current state of the chain is (k, w_k). A birth move proposes a new frequency uniformly on $(0, \pi)$ and appends it to w_k, while a death move attempts to delete a component of w_k randomly. To ensure reversibility with respect to $p(k, w_k|y)$, the authors propose to accept a birth proposal with probability $\min(1, r)$, where

$$r = \left(\frac{\gamma_0 + y^t \mathrm{P}_k y}{\gamma_0 + y^t \mathrm{P}_{k+1} y} \right)^{(N+v_0)/2} \frac{1}{(k+1)(1+\delta^2)}, \tag{4.6}$$

and

$$\mathrm{P}_k = \mathrm{I}_N - \frac{\delta^2}{1+\delta^2} \mathrm{D}(w_k) \left(\mathrm{D}(w_k)^t \mathrm{D}(w_k) \right)^{-1} \mathrm{D}(w_k)^t,$$

for $k \geq 1$ and $\mathrm{P}_0 = \mathrm{I}_N$. Similarly, a death move is accepted with probability $\min(1, r^{-1})$.

Roodaki et al. (2013) prove that (4.6) is erroneous and that the ratio r should be replaced by $(1+k)r$. The authors demonstrate that the error leads the sampler to target the posterior where the prior on k is

$$p(l) \propto \frac{e^{-\Lambda} \Lambda^l}{(l!)^2}, \tag{4.7}$$

for $l \in \{0, \ldots, k_{max}\}$. This is an *accelerated Poisson* distribution, and places greater mass on small values. The implication is that all articles using the erroneous sampler place more emphasis on sparse models than intended.

4.5.3 Testing the Sampler

Our approach is to test the constituent kernels of the sampler individually, insofar as this is possible. We first test LFK and GFK separately on a problem with a known number of sinusoids, i.e. where k is fixed. In the second stage, k is treated as unknown,

and the overall RJ-MCMC sampler is tested. It is difficult to test the birth and death kernels individually because they have a state-dependent selection probability, and are not irreducible by themselves.

Testing the Within-Model Kernels

LFK and GFK are tested separately using the sequential two-sample and rank tests. Throughout, we assume that there is one sinusoid, i.e. $k = 1$, and so the frequency w_1 reduces to a scalar in $(0, \pi)$. This could be extended to $k > 1$ by embedding the kernels in random scan samplers to update all frequency components. To fully define the joint distribution of data and parameters, we set $N = 64$, $v_0 = 10$, $\gamma_0 = 10$, and $\delta = 8$. Ideally, the kernel parameters σ_{rw} and N_p should be set so that the kernels mix fast and the tests have high power. Through experimentation, we determined that $\sigma_{rw} = 1/50$ led to LFK mixing well. For GFK, we use zero padding by letting $N_p = 4N$. This helps to better interpolate the Fourier frequencies, and empirically leads to lower rejection rates.

We use w_1 as the sole test function in all tests. For the two-sample test, we let $N_1 = N_2 = 10^4$, and propagate the direct samples $L = 10^2$ steps with the kernels. The two-sample KS test is used to compare the direct and indirect samples. Given, however, that w_1 is uniform on $(0, \pi)$ under the prior, one could replace the direct samples with the uniform distribution function, and instead employ a one-sided KS test. For the rank tests, we use 10^4 replications, $L = 10$ and thinning of 10. All tests use the default sequential parameters described in Section 4.3; ensuring a false rejection rate of less than 10^{-5}.

The sequential two-sample test applied to LFK gave a p-value of 0.076 in the first iteration, which triggered a second sequential iteration using 4×10^4 iterations. This resulted in a p-value of 0.56 and so no inconsistency was detected. Applying the same test to GFK led to a p-value of 0.36 in the first iteration. The sequential rank test gave p-values of 0.55 and 0.69 for the LFK and GFK kernels respectively in the first iteration. Therefore, in all cases, no error was detected.

Testing the Full Sampler

Having tested the within-model kernels individually, we now test the full sampler that includes LFK, GFK, birth and death moves. For details on the overall algorithm, please refer to Andrieu and Doucet (1999). All tests use the number of sinusoids k as the test function. For the two-sample tests we let $N_1 = N_2 = 10^3$, and for the rank tests we use 10^3 replications. All other parameter values remain the same as in the previous section.

We first test the original, erroneous sampler using the truncated Poisson prior on k with rate $\Lambda = 3$. Both the two-sample and rank tests failed on the first iteration with p-values of 1.4×10^{-6} and 2×10^{-173} respectively, all but proving that the sampler contains an error. The top panel of Figure 4.1 shows the results of these tests. The empirical distribution of k from the indirect samples is skewed to the left, as we expect

given Roodaki et al. (2013)'s finding that the sampler is in effect assuming an accelerated Poisson prior on k. There is also a strong skew in the rank statistics. We performed the same tests using the accelerated Poisson prior on k. As expected, no discrepancy was detected in this case.

Next, we replaced the ratio r with $(1 + k)r$ in the acceptance rate of birth and death moves, as suggested by Roodaki et al. (2013). Neither test detected an error when the correct truncated Poisson prior was used; however, errors were detected when the accelerated Poisson prior was used. All results are shown in Figure 4.1.

4.6 Discussion

This chapter has proposed two tests of MCMC implementations, which are unique in being exact; that is, the false rejection rate can be controlled. This property is leveraged to propose a sequential testing procedure which allows for high power and arbitrarily low false rejection rates, for example 10^{-5}. Such a procedure is useful for unit testing both MCMC and more general Monte Carlo implementations, where one wants to minimize the risk of rejecting a correct sampler.

The performance of the two tests has been tested in a simulation study, and compared to other methods in the literature. The study validates the ability of the tests to achieve the correct nominal level, and generally shows favorable performance of the methods. The exact rank test is shown to have high power over other methods when there are large errors within each conditional distribution, which may not aggregate to an easily detectable error in the joint distribution of data and parameters. On the flip side, we have tested small errors in the conditionals which are detectable in the joint. In this latter case, the Geweke method appears to have a power advantage. However, we have demonstrated extensions to the tests which improve their power.

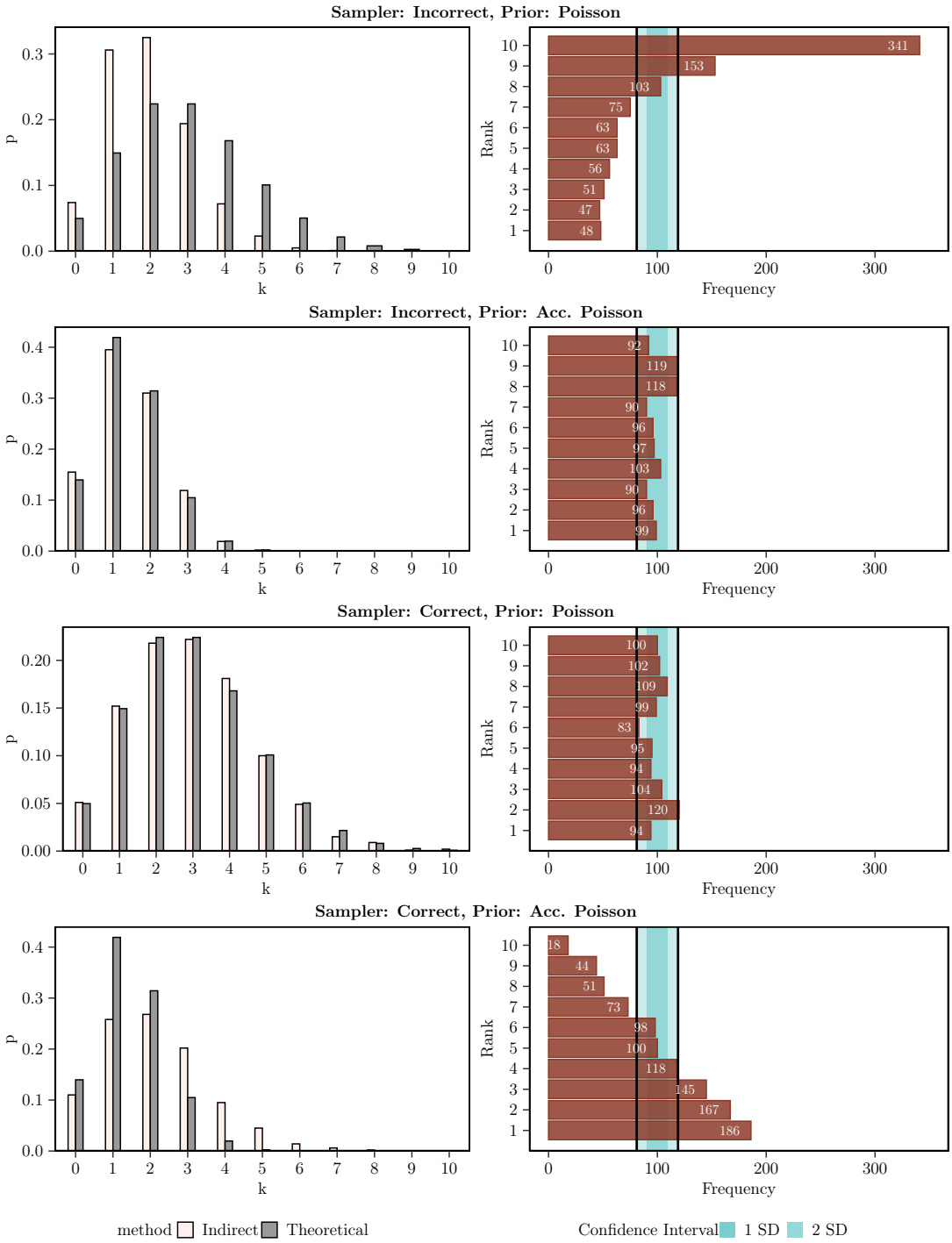


Fig. 4.1 Results of the two-sample and rank tests applied to the full RJ-MCMC sampler.

PART II

The first part of this thesis developed bespoke MCMC algorithms designed to tackle difficult inferential challenges relating to graph sampling. We now move away from the development of novel MCMC samplers, and towards introducing a broad class of epidemic models that leverage MCMC as the workhorse for estimation and fitting. Chapter 5 introduces the modeling framework that was motivated in Section 1.1.2, while Chapter 6 presents an R-package allowing flexible specification and fitting of the models with MCMC.

5

Semi-Mechanistic Bayesian modeling of COVID-19 with Renewal Processes

5.1 Introduction

This chapter presents a general framework for semi-mechanistic Bayesian modeling of infectious diseases using renewal processes. The term semi-mechanistic relates to statistical estimation within some constrained mechanism. Variants of this general model have been used in specific analyses of Covid-19 (Flaxman et al., 2020a, Vollmer et al., 2020, Mellan et al., 2020, Unwin et al., 2020, NYS Press Office, 2020, Olney et al., 2021, The Scottish Government, 2020, Mishra et al., 2020b), and continue to be used in ongoing work to make policy decisions. The present chapter motivates and discusses the key statistical and epidemiological features of this framework, starting from a counting process setup. Various extensions of the basic model are considered, including a latent infection process. We discuss limitations and applications of the modeling framework to stimulate further research.

The model uses a flexible regression-based framework for parameterizing transmission and ascertainment rates. This allows the fitting of multilevel models (Gelman and Hill, 2006, Hox et al., 2010, Kreft and de Leeuw, 2011) for several regions simultaneously. Such partial pooling of parameters has specific advantages in the context of infectious diseases. Suppose we wish to estimate the effect of non-pharmaceutical interventions (NPIs) (Cowling et al., 2020, Flaxman et al., 2020a) or mobility (Badr et al., 2020, Miller et al., 2020) on transmission rates. Estimating these effects separately in different regions could lead to noisy estimates for at least two reasons. There is typically little high-quality data at the early stages of an epidemic. Such data is generally correlated, reducing the information content that can be used to infer such an effect. In addition, NPIs often occur in quick succession and their effects are confounded (SARS Expert Committee, 2003, WHO, 2003). This is exacerbated by the random times between infections (the generation distribution) and between infections and observations, which smooths the observed data, making it more difficult to attribute changes in transmission

rates to a particular NPI. Alternatively, one could pool the effect across all groups. This ignores group-level variations and can lead to poor predictive performance, in particular underestimating variance for previously unmodeled regions. One could augment such a model with group-level indicators, but this results in numerous parameters, which are difficult to estimate and leads to overfitting with classical estimation techniques. Partial pooling provides a natural solution to this.

Sometimes the inferential goal is not to assess the effect of a covariate on outcomes, but rather to infer transmission rates over time. Previous studies have focused on estimating reproduction numbers from case data (Ferguson et al., 2001, Riley et al., 2003, Bettencourt and Ribeiro, 2008, Fraser et al., 2009, Kelly et al., 2010, Cori et al., 2013), sometimes directly substituting observed case counts for the unknown number of infected individuals (Wallinga and Teunis, 2004). However, the emergence of SARS-CoV-2 has highlighted shortcomings of methods that rely on just case data. Limited testing capacity at the early stages of the pandemic led to only a small proportion of infections being detected and reported (Li et al., 2020). Those tested were typically more likely to have been hospitalized or were at higher risk of infection or death. In particular this proportion, referred to as the infection ascertainment rate (IAR) is country-specific and likely to have changed over time due to changes in testing policies and capacity. If unaccounted for, it will lead to biases in the inferred transmission rates.

This highlights the need for more flexible observational models, whereby more varied types of data can be incorporated, and their idiosyncrasies accounted for. Daily death data has been used in Flaxman et al. (2020a) to recover reproduction numbers in the early stages of the SARS-CoV-2 pandemic, and has been seen as more reliable than case data. However, there have been clear variations in definitions and reporting across time and countries. It is therefore important to appropriately model noise within the observational models. Our framework allows for multiple types of data including deaths, cases, hospitalizations, ICU admissions and the results of seroprevalence surveys. This improves robustness of inferred parameters to biases in any one type of data.

The model uses discrete renewal processes to propagate infections within modeled populations. These have been used in a number of previous studies (Fraser, 2007, Cori et al., 2013, Nouvellet et al., 2018, Cauchemez et al., 2008), and are linked to other popular approaches to infectious disease modeling. Champredon et al. (2018) show that the renewal equation leads to identical dynamics as Erlang-Distributed Susceptible-Exposed-Infected-Recovered (SEIR) compartmental models, when a particular form is used for the generation distribution. A special case of this is the standard Susceptible-Infected-Recovered (SIR) model (Kermack et al., 1927). The approach is also connected to counting processes, including the Hawkes process and the Bellman-Harris process (Bellman and Harris, 1948, 1952). Bellman and Harris (1948), Mishra et al. (2020a) derive the renewal equation as the expectation of an age-dependent branching process. Age-dependence allows for more realistic dynamics than age-insensitive processes, like

the Galton-Watson process (Bartoszynski, 1967, Getz and Lloyd-Smith, 2006). More complex branching processes such as the Crump-Mode-Jagers branching process could also be considered. Hawkes processes are also related to renewal processes, with the expectation of the Hawkes intensity function resulting in the renewal equation (Rizoiu et al., 2017).

We describe the general model in detail, and start by considering the bare-bones version in Section 5.2. The motivation for the model lies in continuous-time counting processes, and this connection is discussed in Section 5.3. Sections 5.4 and 5.5 present the infection and observation processes in more detail, and consider important extensions of the basic model. Section 5.6 considers how to use the framework for multilevel modeling. Section 5.7 compares our approach to standard time series models, and outlines the key challenges involved in modeling with our framework. Section 5.8 considers the specific aspect of confounding and causality when estimating the effects of variables on transmission rates. Section 5.9 has a brief discussion.

5.2 Model Overview

We now formulate a basic version of the model for one homogeneous population. The same model can be used for multiple regions or groups jointly. Let $R_t > 0$ be the reproduction number at time $t > 0$. This determines the rate at which infections grow. Infection counts $i_v, \ldots, i_0$ for some $v \leq 0$ are given a prior distribution. For $t > 0$, we let new infections i_t be defined by

$$i_t = R_t \sum_{s<t} i_s g_{t-s}, \tag{5.1}$$

where the generation time, the lag between infections, is given through a probability mass function g, i.e. $g_t \geq 0$ and $\sum_{t=1}^{\infty} g_t = 1$.

Observations occur at certain times $t > 0$. In general, there may be multiple types; case and death counts, for example. Each such type is driven by its own time-varying ascertainment rate $\alpha_t > 0$. The mean of the observations at time t is linked to past infections by

$$y_t = \alpha_t \sum_{s \leq t} i_s \pi_{t-s}, \tag{5.2}$$

where π is a distribution for the lag between an infection and when it gives rise to an observation. The sampling distribution of the observations with these means is typically nonegative and discrete, and may depend on auxiliary parameters. When multiple types are observed, we can superscript the quantities as $y_t^{(l)}, \alpha_t^{(l)}$ and $\pi^{(l)}$ and assign independent data distributions for each type.

Transmission rates R_t and ascertainment rates α_t can be modeled flexibly using Bayesian regression models, and through sharing of parameters, are the means through which we tie together multiple regions or groups using multilevel modeling. One can, for example, model transmission rates as depending on a binary covariate for an NPI, say full lockdown. The coefficient for this can be *partially pooled* between these groups. The effect is to share information between groups, while still permitting between group variation.

5.3 Motivation from continuous time

Our model can be motivated from a continuous time perspective as follows. Infections give rise to additional infections in the future, referred to as offspring. Letting $N^I(t)$ denote the number of infections occurring up to time t, defined by its intensity

$$\lambda(t) = R(t) \int_{s<t} g(t-s) N^I(ds), \quad t > 0, \tag{5.3}$$

where g is the density of a probability distribution on $\mathbb{R}^+$ defining the time between infections, and where $\{R(t) : t > 0\}$ is a non-negative stochastic process. The process can be initialized by assuming values for $N^I(t)$ for t in the seeding period $[v, 0]$.

Equation (5.3) is similar to the Hawkes intensity; however, the *memory kernel* g is scaled by a time-specific factor $R(t)$. The integrand g allows the intensity to increase due to previous infection events, while $R(t)$ tempers the intensity for other time-specific considerations. If $R(t') = R(t)$ for all t' then Equation (5.3) reduces to a Hawkes process. Under this assumption, since g integrates to unity, the expected number of offspring is simply $R(t)$, and so this is the *instantaneous reproduction number* or alternatively the *branching factor* of the Hawkes process. The generation time, defined as the time from an infection to a secondary infection, is distributed according to g and so g is the *generation distribution*.

Observations are precipitated by past infections; a given infection may lead to observation events in the future. Letting $N^Y(t)$ be the count of some observation type over time defined by the intensity

$$\lambda_y(t) = \alpha(t) \int_{s<t} \pi(t-s) N^I(ds), \tag{5.4}$$

for $t > 0$, where $\pi : \mathbb{R}^+ \to \mathbb{R}^+$ is a function and $\{\alpha(t) : t \geq 0\}$ is a non-negative stochastic process. This is similar to Equation (5.3); however, the intensity increases due to past infections, rather than past observations.

Consider the special case where π is a probability density and where $\alpha(t') = \alpha(t)$ for all t'. The average number of observation events attributable to a single infection is then $\alpha(t)$, and so this is an *instantaneous ascertainment rate*. π is then interpreted as the

distribution for the time from an infection to an observation, and therefore we call it the *infection to observation* distribution.

5.4 Infection Process

Starting from the continuous model, we now describe a discrete model, which results in the formulation of Section 5.2. This discrete model is more amenable to inference. Let I_t be the number of new infections at time t; this is the equivalent of $N^I(t) - N^I(t-1)$ in the continuous model. As basic modeling block we use the following discrete version of (5.3):

$$\mathrm{E}[I_t | R_{1:t}, I_{v:t-1}] = R_t L_t, \tag{5.5}$$

where $L_t := \sum_{s<t} I_s g_{t-s}$ is the *case load* or *total infectiousness* by time $t > 0$. Moreover, letting $i_t := \mathrm{E}[I_t | R_{1:t}, I_{v:0}]$ and taking the conditional expectation given reproduction numbers $R_{1:t}$ and seeded infections $I_{v:0}$ on both sides of (5.5) gives

$$i_t = R_t \mathrm{E}[L_t | R_{1:t}, I_{v:0}] = R_t \sum_{s<t} \mathrm{E}[I_s | R_{1:s}, I_{v:0}] g_{t-s} = R_t \sum_{s<t} i_s g_{t-s},$$

which is Equation (5.1). This is a discrete renewal equation, which can alternatively be interpreted as an AR(t)-process with known coefficients g_k. From this point of view, the basic model in Section 5.2 uses i_t as synonymous with actual infections. Since infections are simply a deterministic function of other parameters, there is no need to treat them as unknown latent parameters to sample. This can lead to lower sampling times and faster convergence.

5.4.1 Modeling Latent Infections

The model of Section 5.2 can be extended by replacing each i_t with the actual infections from the process I_t, and then assigning a prior to I_t. Although sampling can be slower, this has certain advantages. When past infection counts are low, significant variance in the offspring distribution can imply that new infections I_t has high variance. This is not explicitly accounted for in the basic model. In addition, this approach cleanly separates infections and observations; the latter being modeled *conditional* on actual infections. The sampling distribution can then focus on idiosyncrasies relating to the observation process.

We assign a prior to I_t conditional on previous infections and current transmission R_t. The expected value for this is given by Equation (5.5). Appendix D.1 shows that assuming the variance of the prior to be a constant proportion d of this mean is equivalent to letting d be the *coefficient of dispersion* for the offspring distribution. $d > 1$ implies overdispersion, and can be used to account for super-spreading events, which has been

shown to be an important aspect for modeling Covid-19. The parameter d can be assigned a prior.

Any two parameter family can be used to match these first two moments. Letting this be continuous rather than discrete allows inference to proceed using Hamiltonian Monte Carlo, whereby new values for I_t are proposed simultaneously with all other parameters. Possible candidates include log-normal, gamma and the Weibull distributions. If an explicit distribution for the offspring distribution is desired, one can show that assuming a Gamma distribution with rate λ for this results in a Gamma distribution for I_t with rate λ. The coefficient of dispersion is then simply $D = \lambda^{-1}$.

5.4.2 Population Adjustments

If R_t remains above unity over time, infections grow exponentially without limit. In practice, infections should be bounded from above by S_0, the initial susceptible population. All else being equal, transmission rates are expected to fall as the susceptible population is diminished.

Consider first the model using I_t, which was described in Section 5.4.1. Equation 5.5 can be replaced with

$$\mathrm{E}[I_t|R_{1:t}, I_{v:t-1}] = (S_0 - I_{t-1})\left(1 - \exp\left(-\frac{R_{u,t}L_t}{S_0}\right)\right), \tag{5.6}$$

where $R_{u,t}$ is an *unadjusted* reproduction number, which does not account for the susceptible population. This satisfies intuitive properties. As the *unadjusted* expected infections $R_{u,t}L_t$ approaches infinity, the *adjusted* expected value approaches the remaining susceptible population. The motivation for and derivation of Equation (5.6) is provided in Appendix D.2. In short, this is the solution to a continuous time model whose intensity is a simplification of Equation (5.3). We must also ensure that the distribution of I_t cannot put positive mass above $S_0 - I_{t-1}$. A simple solution is to use truncated distributions. Of course, this adjusts the mean value from Equation (5.6), however this is unlikely to be significant unless the susceptible population is close to depletion.

In the basic model, one can apply the adjustment to i_t by replacing L_t in Equation (5.6) with

$$\mathrm{E}(L_t|R_{1:t}, I_{v:0}) = \sum_{s<t} i_s g_{t-s}. \tag{5.7}$$

5.5 Observations

Observations are modeled in discrete time, analogous to how we treated infections in Section 5.4. Letting $\pi : \mathbb{N} \to \mathbb{R}^+$ and $Y_t := N_t^Y - N_{t-1}^Y$, the discrete analogue to Equation (5.4) is

$$\mathrm{E}[Y_t|\alpha_t, I_{v:t}] = \alpha_t \sum_{s \le t} I_s \pi_{t-s}. \tag{5.8}$$

Taking the expected value of the above given seeded infections, transmission rates and the current ascertainment rate gives

$$\mathrm{E}[Y_t|\alpha_t, R_{1:t}, I_{v:0}] = \alpha_t \sum_{s \leq t} i_s \pi_{t-s}, \tag{5.9}$$

which is recognizable as Equation (5.2). Thus, we have two possible expressions for the mean of Y_t, one given actual infections, and the other given expected infections i_t. The basic model of Section 5.2 uses the latter, while the extension in Section 5.4.1 uses the former.

We assume that $Y_t \sim \mathcal{F}(y_t, \phi)$, where $\mathcal{F}$ is a non-negative discrete family parameterized by its mean y_t and potentially an auxiliary parameter ϕ. This could be a Poisson distribution, where there is no auxiliary parameter. Using a quasi-Poisson or negative binomial instead allows for overdispersion. This can be useful to capture, for example, day-to-day variation in ascertainment rates when infection counts are low. The mean y_t can be taken to be either (5.8) or (5.9), the latter being used in the basic version of the model. Hidden in this formulation is the assumption that each Y_t is conditionally independent given y_t. Using multiple observation series $Y_t^{(l)}$ can help to improve the model inferences and identifiability of certain parameters. We simply assume that each such series is conditionally independent given the underlying infection process.

5.6 Multilevel Models

Transmission rates can be modeled quite generally within the framework. If the aim is simply to estimate transmission in a single region over time, one approach could be to let $R_t = g^{-1}(\gamma_t)$, where g is a link function and γ_t is some autocorrelation process, for example a random walk. Suppose, however, that transmission is modeled in M regions and the goal is to estimate the effect of a series of NPIs on transmission. Letting $R_t^{(m)}$ denote transmission in region m at time t, we could let

$$R_t^{(m)} = g^{-1}\left(\beta_0^{(m)} + \sum_{l=1}^{p} x_t^{(m)} \beta_k^{(m)}\right), \tag{5.10}$$

where $x_t^{(m)}$ are binary encodings of NPIs, and $\beta_0^{(m)}$ and $\beta_k^{(m)}$ are region-specific intercepts and effects respectively. The intercepts are used to allow regions to have their own baseline transmission rates. Collecting these group-specific parameters into $\beta^{(m)}$, we can partially pool them by letting $\beta^{(m)} \sim \mathcal{N}(0, \Sigma)$, for each group m, and then assigning a prior to the covariance matrix Σ. This could be an inverse-Wishart prior, or alternatively, Σ can be decomposed into variances and a correlation matrix, which are each given separate priors (Tokuda et al., 2011).

One possible option for g is the log-link. This provides easily interpretable effect sizes; a one unit change in a covariate multiplies transmission by a constant factor. However,

this can lead to prior mass on unreasonably high transmission rates. With this in mind, an alternative is to use a generalization of the logit link for which

$$g^{-1}(x) = \frac{K}{1 + e^{-x}}, \tag{5.11}$$

and where K is the maximum possible value for transmission rates. This serves a similar purpose to the carrying capacity in a logistic growth model.

The ascertainment rate α_t can also be modeled with similar considerations to the above. This flexibility is useful, particularly because these quantities are likely to change as an epidemic progresses. This has been clearly seen during the Covid-19 epidemic, where the infection ascertainment rate may have increased over time due to increased testing capacity and improved track and trace systems. Multilevel models can in theory also be specified through α_t.

5.7 Forecasting, epidemiological constants, and seeding

A key benefit of using a semi-mechanistic approach is that forecasts are constrained by plausible epidemiological mechanisms. For example, in the absence of any further interventions or behavioral changes, and looking at a medium term forecast of just incidence (daily new cases/infections), a traditional time series forecasting approach may predict a constant function based on observing broadly constant incidence, but semi-mechanistic approach would expect a monotonic decrease based on a constant rate of transmission and the effect of herd immunity. The performance of epidemiologically constrained models is generally good (Carias et al., 2019); this is perhaps not surprising, as examining the discrete renewal equation shows that these models correspond to autoregressive(n) filters with a convex combination of coefficients specified by the generation interval. However, similar to financial forecasting, the predictive capability of epidemic models are likely to be better interpreted as scenarios rather than actual predictions due to the rapidly adaptive landscape of policy.

A second benefit of epidemic models is to provide a plausible mechanism to explain (non causally) the changes observed in noisy data. For example, in estimating the effect of an intervention on observed death data, we need to consider what that intervention effects, i.e. the rate of transmission or R_t. As we have described above, we can connect the rate of transmission to latent infections to deaths such that we have an epidemiologically motivated mechanism. While we can statistically estimate parameters for how the intervention affects R_t, certain important parameters will be entirely unidentifiable and need to be fixed as constants or with very tight priors. For example, to reliably estimate the number of infections, an infection fatality rate needs to be chosen. A failure to choose an appropriate infection fatality rate can result in a bimodal posterior where changes can either be attributed to herd immunity or to interventions. From a statistical perspective, is it difficult to disentangle which mode of the posterior best represents reality, and

hence it is sensible to first estimate a plausible infection fatality rate and then use this within the semi-mechanistic model. A second example is the onset of symptoms-to-death distribution. Given the lag between transmission, infection and deaths, the effect of an intervention is dependent on the onset of symptoms-to-death distribution.

Infection seeding is a fundamentally challenging aspect of epidemic modeling. Estimating the initial effect of seeding is crucial to understanding a baseline rate of growth (R_0) from which behaviors and interventions can modify. This seeding is heavily confounded by importation and under ascertainment. Both these factors can influence estimates of the initial growth rates, and this in turn can affect the impact of changes in transmission as time progresses. We have proposed heuristic approaches to mitigate issues with early seeding, but principled statistical approaches need to be developed. In particular, Bayesian pair plots show strong correlation between seeding parameters and R_0, which can potentially lead again to a bimodal posterior where initial growth dynamics can be explained through R_0 or via initial seed infections.

Our approach also assumes a known generation interval, that is fixed through time. Similar assumptions are made for the time from infection to an observation (a recorded case or death, for example). These should be set using information from previous studies, however more careful handling of uncertainty over these distributions should be considered. In particular, if the model uses multiple observational types, the assumes infection to observation distributions can lead to conflicting inferences on transmission rates and infections. Modeling these distributions as unknown parameters and assigning them priors is one potential solution to this.

5.8 Confounding and Causality: Estimating the Effect of Interventions

Section 5.6 showed that changes in transmission rates over time can be explained by parameterizing these rates in terms of predictor variables, such as NPIs and mobility. Clarifying the effects of interventions on transmission are important, if only because of their clear economic and human consequences. This is however a significant challenge because the effects are potentially confounded with unobserved behavioral changes, and they are hard to identify. Identification is difficult because interventions are highly correlated when they occur in quick succession. Moreover, the random time between an infection and its recording as a case or death leads to observations being less informative about the effect of any particular intervention.

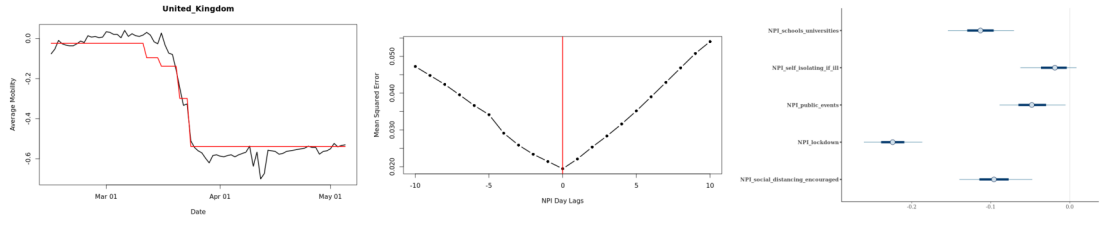
Flaxman et al. (2020a) estimated the effectiveness of NPIs across 11 European countries, and used partial pooling of effect sizes to address the identification problem. At that time, little data existed other than information on deaths and the timing of interventions. NPIs, which were coded as a binary set of mandatory government measures (e.g. school closures, ban on public events, lockdown), could not fully explain the patterns

seen in some countries (e.g. Sweden), and especially at the subnational level. Mobility data became available in April and was used to model the epidemic in Italy, Brazil and the USA (Vollmer et al., 2020, Mellan et al., 2020, Unwin et al., 2020). Such data is useful as it may help account for behavioral changes that confound the effects of NPIs. However, since mobility affects transmission, is linked to the introduction of NPIs and potentially also to voluntary behavioral measures, we expect it to be a confounder.

Section 5.8.2 extends the model in Flaxman et al. (2020a) to investigate further this issue of confounding, and models both NPIs and mobility jointly. This is in keeping with standard practice in regression/ANOVA: expanding a model to take into account more explanatory variables. Nonetheless, NPIs may partially affect transmission *via a path through mobility*. A joint model of mobility and NPIs does not account for this. Therefore, in Section 5.8.3 we take a first step in assessing causal considerations through a simple mediation analysis. We begin however by exploring the relationship between interventions and mobility.

5.8.1 Interventions and Mobility

Regressing average mobility on NPIs in a Bayesian linear model (no intercept or partial pooling) we find a correlation of over 85% with a mean absolute error of 0.1%. Given the mobility data used generally ranges between -1 to 1, this is a good overall fit. Figure 5.1a shows that these fits visually correspond well with changes in average mobility. One could conjecture that mobility and NPIs are lagged, but lagging NPI dates either forwards or backwards in time does not result in a better fitting model (see Figure 5.1b). Indeed, Figure 5.1b) does support a hypothesis that the timing of NPIs and changes in mobility are strongly linked. The coefficient sizes from this regression are entirely consistent with Flaxman et al. (2020a) finding that the NPI with the largest effect size is lockdown (see Figure 5.1c). This simple analysis does not model transmission, but does provide strong evidence that mobility and NPIs do not provide conflicting narratives. We note, to perform this regression as fairly as possible, we used a hierarchical shrinkage prior (Piironen and Vehtari, 2017) that performs both shrinkage and variable selection.



(a) (b) (c)

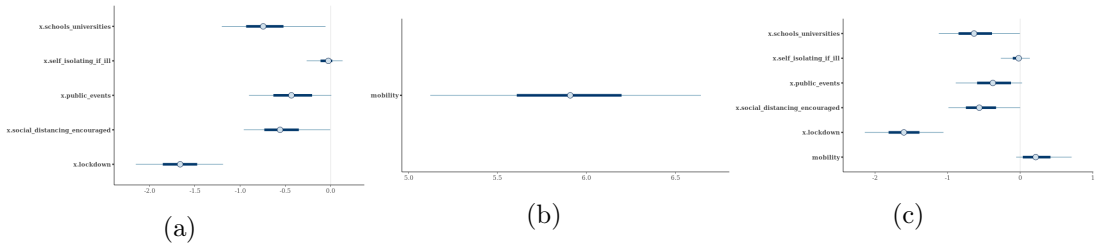
Fig. 5.1 Simple regression of mobility against NPIs. Figure (a) shows the fit for the United Kingdom with mobility in black and the fit for NPIs in red. (b) shows the effect on mean absolute error from lagging the NPIs. (c) shows the coefficient effect sizes from the regression. Y axis are NPIs and X axis the regression effect sizes.

5.8.2 Controlling for Mobility

Section 5.8.1 found a large correlation between interventions and mobility, demonstrating that mobility is a potential confounder. Here, we control for this by jointly modeling NPIs and mobility. This is done using the same 11 European countries, sets of NPIs and death data as used in Flaxman et al. (2020a).

A two-stage approach (Haug et al., 2020) is used, whereby R_t is first estimated using a daily random walk. In the second stage, this is regressed on NPIs and mobility. The random walk can in theory select any arbitrary function of R_t that best describes the data, without any prior information about which interventions happened when or how well they worked. Given these estimates of R_t for all 11 European countries, we run a simple partial pooling model to see if interventions and/or mobility can reproduce the trends in R_t. The model used is a linear regression with country level intercepts (to account for variation in R_0), and both joint and country specific effect sizes for interventions/mobility. As with the earlier analysis, we use a hierarchical shrinkage prior on the coefficients (Piironen and Vehtari, 2017).

Three variations of the model are considered: NPIs only (NPI_only), mobility only (Mobility_only), and both NPIs and mobility (NPI+Mobility). MCMC convergence diagnostics in all cases did not indicate problems. We found the best fitting model to be NPI+Mobility. Relative to this model, the expected log posterior difference ($\pm$ standard error) in WAIC of the model with only NPIs is -5.2 ± 4, ad -565.6 ± 49.2 with only mobility. Therefore, in fits to the estimated R_t, the model with mobility alone is substantially worse than the models with NPIs. Controlling for mobility does not appear to significantly change the estimated effects of NPIs. As in Flaxman et al. (2020a), the largest effect size is attributed to lockdown, as seen in Figure 5.2. This is true with and without the inclusion of the mobility variable. This analysis could be improved using Bayesian leave-one-out cross-validation (Vehtari et al., 2017) to account for the time series nature of the data.



(a) (b) (c)

Fig. 5.2 Regression of NPIs and/or mobility against nonparametric R_t. Figure (a) shows the fit for NPIs only. (b) for mobility only, (c) for NPIs and mobility. Mobility only was not *significantly* preferred by WAIC. Y axis are covariates and X axis the regression effect sizes.

An advantage of the two-stage approach is that it is scalable to many regions. R_t can be estimated in each region in parallel using separate models. Partial pooling can still be leveraged to estimate effects in the second stage. Once R_t has been estimated, any

number of interesting statistical analyses can be conducted. Nonetheless, the estimated R_t is not entirely non-parametric; it is clearly influenced by its random walk form in the first stage. This analysis could be extended by considering a range of alternative priors for R_t. More importantly, however, this approach has not considered causal relationships between NPIs and mobility. This is the focus of the next example.

5.8.3 Causal Mediation

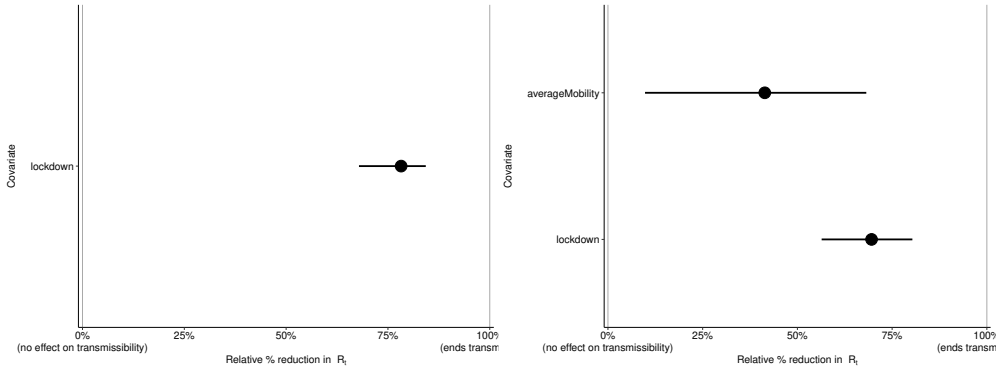
The effect of interventions in the previous analysis holds mobility constant. However, we intuitively expect that part of these effects occur indirectly through changing mobility. We can hypothesize that changes in mobility are both an effect of NPIs and a cause of reductions in transmission. Causal mediation analysis provides a means to disentangle the total effect of a variable into a direct and indirect effect. The indirect effect occurs via some mediator, which in this case is hypothesized to be mobility. Further information about causal mediation can be found here (Pearl, 2009, 2012).

Only lockdown is considered here because performing causal mediation with all NPIs is challenging and lockdown is consistently the NPI with the largest effect size in Section 5.8.1, Section 5.8.2 and in Flaxman et al. (2020a). Briefly, to perform causal mediation, we consider two transmission models

$$R_t^{(m)} = \tilde{R}_m^1 \exp\left(\left(\beta_1^1 + \beta_{1,m}^1\right) L_{t,m} + \varepsilon_{t,m}^1\right), \tag{5.12}$$

$$R_t^{(m)} = \tilde{R}_m^2 \exp\left(\left(\beta_1^2 + \beta_{1,m}^2\right) L_{t,m} + \left(\beta_2^2 + \beta_{2,m}^2\right) M_{t,m} + \varepsilon_{t,m}^2\right), \tag{5.13}$$

where $L_{t,m}$ is a binary indicator for lockdown and $M_{t,m}$ is mobility in country m respectively. $\tilde{R}_m^i$ and $\varepsilon_{t,m}^i$ are country specific parameters modeling baseline transmission and a weekly random walk respectively. All other aspects of both models are the same as in (Flaxman et al., 2020a). Model (5.12) includes effects for lockdown, while (5.13) additionally considers mobility. β_1^1 is the total effect for lockdown, while β_1^2 is the partial effect when controlling for mobility. The mediated effect is therefore $\beta_1^1 - \beta_1^2$. This quantifies the effect of lockdown *via the path through mobility*. We find this mediated effect reduces R_t by 18.3% with a 95% credible interval of $[12.2\%, 44.4\%]$. The posterior probability of the effect being greater than 0 is 89.6% . Individual coefficients are shown in Figure 5.3.



(a) (b)

Fig. 5.3 Mediation analysis with lockdown (a), and both lockdown and mobility (b).

These mediation results suggest a causal link between lockdown and mobility that eventually leads to reduced transmission rates. They also suggest that the mediated effect is far less than the total effect of lockdown, suggesting lockdown will have other causal pathways. Of course, mobility is also mediated through other pathways, and a principled causal analysis is out of the scope of this chapter. The exclusion of other NPIs may introduce omitted variable bias. Nonetheless, this simple analysis with lockdown adds support to causal considerations.

5.9 Discussion

This chapter has discussed a class of statistical models for epidemics such as Covid-19 which can capture key epidemiological mechanisms. The model has appeared in various forms for specific analyses during the Covid-19 crisis and, at the time of writing, continues to be used to inform public policy. By presenting it in a general form and discussing key modeling difficulties, we hope to stimulate discussion around it. One key difficulty within the framework is dealing with confounded variables, particularly those used to explain changes in transmission during the early stages of an epidemic. The analyses in Section 5.8 make a first step in dealing with these, and support the central finding of Flaxman et al. (2020a): that lockdown and other NPIs together served to control the first wave of the epidemic in 11 European countries.

A number of model enhancements have not been discussed here. These include explicitly accounting for importations, and allowing for uncertainty in the generation and infection to observation distributions. The model can be fit using Stan (Stan Development Team, 2018), but the adaptive Hamiltonian Monte Carlo used often has difficulty converging when latent infections are modeled directly, or when multiple regions are jointly modeled. We conjecture that convergence may be improved by carefully choosing initial parameters for the sampler. Future research could explore whether alternative samplers can be developed to fit these models more pragmatically.

6

epidemia: An R Package for Semi-Mechanistic modeling of Infectious Diseases.

6.1 Introduction

This chapter introduces the open-source R (R Core Team, 2021) package **epidemia**, which provides a framework for Bayesian, regression-oriented modeling of the temporal dynamics of infectious diseases. The motivation for these models, and the mathematical framework behind them, was introduced in Chapter 5. The implemented models are typically, but not exclusively, fit to areal time-series; i.e. aggregated event counts for a given population and period. Disease dynamics are described explicitly; observed data are linked to latent infections, which are in turn modeled as a self-exciting process tempered by time-varying reproduction numbers.

Regression models are specified for key objects in the models, which provides users with a high degree of flexibility in defining models. For example, as was the case in Chapter 5, reproduction numbers are expressed as a transformed predictor, which may include both covariates and autoregressive terms. A range of prior distributions can be assigned to unknown parameters by leveraging the functionality of **rstanarm** (Goodrich et al., 2020). Multilevel models are supported by partially pooling covariate effects appearing in the predictor for reproduction numbers between multiple populations.

epidemia's functionality can be used for a number of purposes. A researcher can simulate infection dynamics under assumed parameters by setting tight priors around the assumed values. It is then possible to sample directly from the prior distribution without conditioning on data. This allows *in-silico* experimentation; for example, to assess the effect of varying a single parameter (reproduction numbers, seeded infections, incubation period). Another goal of modeling is to assess whether a simple and parsimonious model of reality can replicate observed phenomena. This helps to isolate processes helpful for explaining the data. Models of varying complexity can be specified within **epidemia**, largely as a result of its regression-oriented framework. Posterior predictive checks can be used to assess model fit. If the model is deemed misspecified, additional features may

be considered. This could be modeling population adjustments, explicit modeling of super-spreader events (Wong and Collins, 2020), alternative and over-dispersed models for the data, or more flexible functional forms for reproduction numbers or ascertainment rates. This can be done rapidly within **epidemia**'s framework.

Forecasting models are critical during an ongoing epidemic as they are used to inform policy decisions under uncertainty. As a sign of their importance, the United States Centers for Disease Control and Prevention (CDC) has run a series of forecasting challenges, including the FluSight seasonal forecasting challenges since 2015 (https://www.cdc.gov/flu/weekly/flusight/) and more recently the Covid-19 Forecast hub (https://covid19forecasthub.org/). Similar challenges have been run by the European Center for Disease Prevention and Control (ECDC) (https://covid19forecasthub.eu/). Long-term forecasts quantify the cost of an unmitigated epidemic, and provide a baseline from which to infer the effects of control measures. Short-term forecasts are crucial in informing decisions on how to distribute resources such as PPE or respirators, or whether hospitals should increase capacity and cancel less urgent procedures. Traditional statistical approaches often give unrealistic long-term forecasts, as they do not explicitly account for population effects. The semi-mechanistic approach of **epidemia** combines the strengths of statistical approaches with plausible infection dynamics, and can thus be used for forecasting at different tenures.

The rest of this chapter is organized as follows. Section 6.1.1 discusses alternative R packages for epidemiology, and highlights the unique features of **epidemia**. Section 6.2 introduces the basic model and various extensions. Sections 6.3 and 6.4 provide installation instructions and introduce some main functions required to specify and fit the models. We proceed in Section 6.5 to demonstrate usage of the package on two examples. The first example considers the task of inferring time-varying reproduction numbers, while the second attempts to infer the effects of control measures using a multilevel model. Finally, we conclude in Section 6.6.

6.1.1 Related packages

The Comprehensive R Archive Network (CRAN) (https://cran.r-project.org/) provides a rich ecosystem of R packages dedicated to epidemiological analysis. The R Epidemics Consortium website (https://www.repidemicsconsortium.org/) lists a number of these. Packages that model infectious disease dynamics vary significantly by the methods used to model transmission. **RLadyBug** (Höhle and Feldmann, 2007) is a R package for parameter estimation and simulation for stochastic compartmental models, including SEIR-type models. Both likelihood-based and Bayesian inference are supported. **amei** (Merl et al., 2010) provides online inference for a stochastic SIR model with a negative-binomial transmission function, however, primary focus is on identifying optimal intervention strategies. See Andersson and Britton (2000) for an introduction to stochastic epidemic modeling.

epinet (Groendyke and Welch, 2018) and **epimodel** (Jenness et al., 2018) provide functionality to simulate compartmental models over contact networks. **epinet** uses the class of dyadic-independent exponential random graph models (ERGMs) to model the network, and perform full Bayesian inference over model parameters. **epimodel** considers instead dynamic networks, inferring only network parameters and assuming epidemic parameters to be known.

Epidemic data often presents in the form of areal data, recording event counts over disjoint groups during discrete time intervals. This is the prototypical data type supported within **epidemia**. Areal data can be modeled using purely statistical methods. The `glm()` function in **stats** can be used to fit simple time-series models to count data. The package **acp** (Vasileios, 2015) allows for fitting autoregressive Poisson regression (ACP) models to count data, with potentially additional covariates. **tscount** (Liboschik et al., 2017) expands on **acp**, and in particular provides more flexible link functions and over-dispersed distributions.

Like **epidemia**, the package **Surveillance** (Meyer et al., 2017) implements regression-oriented modeling of epidemic dynamics. The package offers models for three different spatial and temporal resolutions of epidemic data. For areal data, which is the focus of **epidemia**, the authors implement a multivariate time-series approach (Held et al., 2005, Paul et al., 2008, Paul and Held, 2011, Held and Paul, 2012). This model differs from the semi-mechanistic approach used here in several ways. First, the model has no mechanistic component: neither infections and transmission are explicitly described. The model is similar in form to a vector autoregressive model of order 1 (VAR(1)). The lag 1 assumption implies that each count series is Markovian. In **epidemia**, the infection process has an interpretation as an AR process with both order and coefficients determined by the generation distribution. This can therefore model more flexible temporal dependence in observed data.

EpiEstim (Cori et al., 2013, Cori, 2021) infers time-varying reproduction numbers R_t using case counts over time and an approximation of the disease's generation distribution. Infection incidence is assumed to follow a Poisson process with expectation given by a renewal equation. **R0** (Obadia et al., 2012) implements techniques for estimating both initial and time-varying transmission rates. In particular, the package implements the method of Wallinga and Teunis (2004), which bases estimates off a probabilistic reconstruction of transmission trees. **epidemia** differs from these packages in several ways. First, if infection counts are low then the Poisson assumption may be too restrictive, as super-spreader events can lead to over-dispersion in the infection process. Our framework permits over-dispersed distributions for modeling latent infections. Second, **epidemia** allows flexible prior models for R_t, including the ability to use time-series methods. For example, R_t can be parameterized as a random walk. Finally, infections over time are often unobserved, and subject to under-reporting that is both space and time dependent. We account for this by providing flexible observation models motivated by survival

processes. Several count data series may be used simultaneously within the model to leverage additional information on R_t.

The probabilistic programming language Stan (Stan Development Team, 2018) has been used extensively to specify and fit Bayesian models for disease transmission during the Covid-19 pandemic. Example analyses include Flaxman et al. (2020a), Hauser et al. (2020) and van Doremalen et al. (2020). For tutorials on implementing such models, see for example Grinsztajn et al. (2020) or Chatzilena et al. (2019). **epidemia** uses the framework offered by Stan to both specify and fit models. User-specified models are internally translated into data that is passed to a precompiled Stan program. The models are fit using sampling methods from **rstan** (Stan Development Team, 2020).

6.2 Model Description

Here, we present the modeling framework implemented by the package. Section 6.2.1 outlines the bare-bones version of the model, which is elaborated on in Sections 6.2.2, 6.2.3 and 6.2.4. Section 6.2.5 extends the model and introduces multilevel modeling, treating infections as parameters, and accounting for population effects.

6.2.1 Basic Model

We now formulate the basic version of the model for one homogeneous population. The same model can be used for multiple regions or groups jointly. Suppose we observe a non-negative time series of count data $Y = (Y_1, \dots Y_n)$ for a single population. This could for example be daily death or case incidence. Y_t is modeled as deriving from past new infections i_s, $s < t$, and some parameter $\alpha_t > 0$, a multiplier, which in most contexts represents an instantaneous *ascertainment rate*. The general model can be expressed as

$$Y_t \sim p(y_t, \phi), \tag{6.1}$$

$$y_t = \alpha_t \sum_{s<t} i_s \pi_{t-s}, \tag{6.2}$$

where y_t is the expected value of the data distribution and ϕ is an auxiliary parameter. π_k is typically the time distribution from an infection to an observation, which we refer to as the *infection to observation* distribution. More generally, however, π_k can be used to obtain any linear combination of past infections. New infections i_t at times $t > 0$ are modeled through a renewal equation, and are tempered by a non-negative parameter R_t which represents the reproduction number at time t. Formally

$$i_t = R_t \sum_{s<t} i_s g_{t-s}, \tag{6.3}$$

where g_k is a probability mass function for the time between infections. The recursion is initialized with *seeded* infections $i_{v:0}$, $v < 0$, which are treated as unknown parameters.

All parameters are assigned priors, i.e.

$$i_{v:0}, R, \phi, \alpha \sim p(\cdot), \tag{6.4}$$

where $R = (R_1, \ldots, R_n)$ and $\alpha = (\alpha_1, \ldots, \alpha_n)$. The posterior distribution is then proportional to prior and likelihood, i.e.

$$p(i_{v:0}, R, \phi, \alpha \mid Y) \propto p(i_{v:0})p(R)p(\phi)p(\alpha) \prod_t p(Y_t \mid y_t, \phi). \tag{6.5}$$

This posterior distribution is represented in a Stan program, and an adaptive Hamiltonian Monte Carlo sampler (Hoffman and Gelman, 2014) is used to approximately draw samples from it. These samples allow for inference on the parameters, in addition to simulating data from the posterior predictive distribution.

Reproduction numbers R and multipliers α can be modeled flexibly with Bayesian regression models, and by sharing parameters, are the means by which multiple regions or groups are tied together through multilevel models. One can, for example, model R as depending on a binary covariate for a control measure, say full lockdown. The coefficient for this can be *partially pooled* between multiple populations. The effect is to share information between groups, while still permitting between group variation.

6.2.2 Observations

As mentioned, Y_t is usually a count of some event type occurring at time t. These events are precipitated by past infections. Prototypical examples include daily cases or deaths. α_t is a multiplier, and when modeling count data, it typically is interpreted as an *ascertainment rate*, i.e. the proportion of events at time t that are recorded in the data. For case or death data, this would be the infection ascertainment rate (IAR) or the infection fatality rate (IFR) respectively.

The multiplier α plays a similar role for observations as R does for infections; tempering expected observations for time-specific considerations. As such, **epidemia** treats α in a similar manner to reproductions number, and allows the user to specify a regression model for it. Section 6.2.4 discusses this in detail in the context of reproduction numbers, and this discussion is not repeated here. Figure E.1 in Appendix E.3 details the model for α, as well as for observational models in general.

The sampling distribution $p(y_t, \phi)$ (Equation (6.1)) should generally be informed by parts of the data generating mechanism not captured by the mean y_t: i.e. any mechanisms which may induce additional variation around y_t. Options for $p(y_t, \phi)$ include the Poisson, quasi-Poisson and negative-binomial families. The Poisson family has no auxiliary parameter ϕ, while for the latter two families this represents a non-negative *dispersion parameter* which is assigned a prior.

epidemia allows simultaneous modeling of multiple observation vectors. In this case, we simply superscript $Y_t^{(l)}$, $\alpha_t^{(l)}$ and $\pi^{(l)}$, and assign independent sampling distributions for each type. Separate regression models are then specified for each multiplier $\alpha_t^{(l)}$. Leveraging multiple observation types can often enhance a model. For example, high-quality death data existed during the first wave of the Covid-19 pandemic in Europe. Case data gradually increased in reliability over time, and has the advantage of picking up changes in transmission dynamics much quicker than death data.

6.2.3 Infections

Infections i_t propagate over time through the discrete renewal equation (6.3). This is *self-exciting*: past infections give rise to new infections. The theoretical motivation for this lies in counting processes and is explained in more detail in Chapter 5. The equation is connected to Hawkes processes and the Bellman Harris branching process (Bellman and Harris, 1948, 1952, Mishra et al., 2020a). Such processes have been used in numerous previous studies (Fraser, 2007, Cori et al., 2013, Nouvellet et al., 2018, Cauchemez et al., 2008), and are also connected to compartmental models such as the SEIR model (Champredon et al., 2018).

Equation (6.3) implies that infections i_t, $t > 0$ are deterministic given R and seeded infections $i_{v:0}$. **epidemia** sets a prior on $i_{v:0}$ by first assuming that daily seeds are constant over the seeding period. Formally, $i_k = i$ for each $k \in \{v, \dots 0\}$. The parameter i can be assigned a range of prior distributions. One option is to model it hierarchically; for example, as

$$i \sim \text{Exp}(\tau^{-1}), \tag{6.6}$$

$$\tau \sim \text{Exp}(\lambda_0), \tag{6.7}$$

where $\lambda_0 > 0$ is a rate hyperparameter. This prior is uninformative, and allows seeds to be largely determined by initial transmission rates and the chosen start date of the epidemic.

Several extensions to the infection model are possible in **epidemia**, including extending (6.3) to better capture dynamics such as super-spreading events, and also adjusting the process for the size of the remaining susceptible population. These extensions are discussed in Section 6.2.5 and 6.2.5 respectively. The basic infection model is shown in Figure E.2 in Appendix E.3.

6.2.4 Transmission

Reproduction numbers are modeled flexibly. One can form a linear predictor consisting of fixed effects, random effects and autocorrelation terms, which is then transformed via

a suitable link function. Formally

$$R = g^{-1}(\eta), \tag{6.8}$$

where g is a link function and η is a linear predictor. In full generality, η can be expressed as

$$\eta = \beta_0 + X\beta + Zb + Q\gamma, \tag{6.9}$$

where X is an $n \times p$ model matrix, Z is an $n \times q$ model matrix for the q-vector of group-specific parameters b. Q is an $n \times r$ model matrix for the r-vector of autocorrelation terms. The columns of X are predictors explaining changes in transmission. These could, for example, be binary vectors encoding non-pharmaceutical interventions, as in Flaxman et al. (2020a). A number of families can be used for the prior on β, including normal, Cauchy, and hierarchical shrinkage families. The parameters b are modeled hierarchically as

$$b \sim N(0, \Sigma), \tag{6.10}$$

where Σ is a covariance matrix that is itself assigned a prior. The particular form for Σ, as well as its prior, is discussed in more detail in Appendix E.1.4. These partially pooled parameters are particularly useful when multiple regions are being modeled simultaneously. In this case, they allow information on transmission rates to be shared between groups.

Q is a binary matrix specifying which of the autocorrelation terms in γ to include for each period t. Currently, **epidemia** supports only random walk processes. However, multiple such processes can be included, and can have increments that occur at a different timescale to R; for example weekly increments can be used.

Link Functions

Choosing an appropriate link function g is difficult. R_t is non-negative, but clearly cannot grow exponentially: regardless of the value of the linear predictor η_t, one expects R_t to be bounded by some maximum value K. In other words, R_t has some *carrying capacity*. One of the simplest options for g is the log-link. This satisfies non-negativity, and also allows for easily interpretable effect sizes; a one unit change in a predictor scales R_t by a constant factor. Nonetheless, it does not respect the carry capacity K, often placing too much prior mass on large values of R_t. With this in mind, **epidemia** offers an alternative link function satisfying

$$g^{-1}(x) = \frac{K}{1 + e^{-x}}. \tag{6.11}$$

This is a generalization of the logit-link, and we refer to it as the *scaled-logit*.

6.2.5 Extensions

Various extensions to the basic model just presented are possible, including multilevel modeling, adding variation to the infection process, and explicitly accounting for population effects. These are discussed in turn.

Joint Modeling of Multiple Populations

Consider modeling the evolution of an epidemic across multiple regions or populations. Of course, separate models can be specified for each group. This approach is fast as each model can be fit in parallel. Nonetheless, often there is little high-quality data for some groups, particularly in the early stages of an epidemic. A joint model can benefit from improved parameter estimation by *sharing signal across groups*. This can be done by partially or fully pooling effects underlying reproduction numbers R.

We give an example for concreteness. Suppose the task is to infer the effect of a series of p control measures on transmission rates. Letting $R^{(m)}$ be the vector of reproduction numbers for the m^{th} group, one could write

$$R^{(m)} = g^{-1}\left(\beta_0 + b_0^{(m)} + X^{(m)}(\beta + b^{(m)})\right), \tag{6.12}$$

where $X^{(m)}$ is a $n \times p$ matrix whose rows are binary vectors indicating which of the p measures have been implemented in the m^{th} group at that point in time. The parameters $b_0^{(m)}$ allow each region to have its own initial reproduction number R_0, while $b^{(m)}$ allow for region-specific policy effects. These parameters can be partially pooled by letting

$$(b_0^{(m)}, b^{(m)}) \sim N(0, \tilde{\Sigma}), \tag{6.13}$$

for each m, and assigning a hyper-prior to the covariance matrix $\tilde{\Sigma}$.

In addition to hierarchical modeling of parameters making up R, seeded infections are also modeled hierarchically. Equations (6.6) and (6.7) are replaced with

$$i^{(m)} \sim \text{Exp}(\tau^{-1}), \tag{6.14}$$

$$\tau \sim \text{Exp}(\lambda_0), \tag{6.15}$$

where $i^{(m)}$ is the daily seeded infections for the m^{th} group.

Infections as Parameters

Recall the renewal equation (Equation (6.3)) which describes how infections propagate in the basic model. Infections i_t for $t > 0$ are a deterministic function of seeds $i_{v:0}$ and reproduction numbers R. If infections counts are large, then this process may be realistic enough. However, when infection counts are low, there could variation in day-to-day infections caused by a heavy tailed offspring distribution and super-spreader events.

This may cause actual infections to deviate from those implied by the renewal equation. Although the *expected* number of offspring of any given infection is driven by R, in practice the actual number of offspring can exhibit considerable variation around this. To capture this randomness, replace Equation (6.3) with

$$i_t \sim p(i'_t, d), \tag{6.16}$$

$$i'_t = R_t \sum_{s<t} i_s g_{t-s}. \tag{6.17}$$

This treats i_t as latent parameters which must be sampled. Instead, the *mean value* is described by the renewal equation. $p(i'_t, d)$ is parameterized by the mean and the coefficient of dispersion d, which is assigned a prior. This extension can be motivated formally through counting processes. Please see Chapter 5 for more details.

Depletion of the Susceptible Population

Nothing in Equation (6.3) prevents cumulative infections from exceeding the total population size P. In particular, if $R_t > 1$ then infections can grow exponentially over time. This does not always present a problem for modeling. Indeed, the posterior distribution usually constrains past infections to reasonable values. Nonetheless, forecasting in the basic model will be unrealistic if projected infections grow too large. As the susceptible population diminishes, the transmission rate is expected to fall.

epidemia can apply a simple transformation to ensure that cumulative infections remain bounded by P, and that transmission rates are adjusted for changes in the susceptible population. Let $S_t \in [0, P]$ be the number of susceptible individuals in the population at time t. Just like infections, this is treated as a continuous quantity. S_t consists of those who have not been infected by time t, and have not been removed from the susceptible class by other means; i.e. vaccination.

Let i'_t denote *unadjusted infections* from the model. This is given by (6.3) in the basic model, or by (6.16) if the extension of Section 6.2.5 is applied. These are interpreted as the number of infections if the entire population were susceptible. These are adjusted with

$$i_t = S_{t-1}\left(1 - \exp\left(-\frac{i'_t}{P}\right)\right). \tag{6.18}$$

The motivation for this is provided in Chapter 5. Equation (6.18) satisfies intuitive properties: if $i'_t = 0$ then $i_t = 0$, and as $i'_t \to \infty$ we have that $i_t \to S_{t-1}$. All infections at time t are then removed from the susceptible population so that

$$S_t = S_{t-1} - i_t \tag{6.19}$$

We are left to define S_{v-1}, the susceptible population the day before modeling begins. If this is the start of an epidemic, it is natural to take $S_{v-1} = P$. Nonetheless, it is often of interest to begin modeling later, when a degree of immunity already exists within the population. In this case, **epidemia** allows the user to assign a prior distribution to S_{v-1}/P. This must lie between 0 and 1.

Accounting for Vaccinations Previous infection is one avenue through which individuals are removed from the susceptible population. Immunity can also be incurred through vaccination. **epidemia** provides a basic way to incorporate such effects.

Let v_t be the proportion of the susceptible population at time t who are removed through some means other than infection. These are individuals who have never been infected but may have been previously vaccinated, and their immunity is assumed to have developed at time t.

epidemia requires v_t to be supplied by the user. Then (6.19) is replaced with

$$S_t = (S_{t-1} - i_t)\,(1 - v_t)\,. \tag{6.20}$$

Of course, v_t is a difficult quantity to estimate. It requires the user to estimate the time-lag for a jab to become effective, and to also adjust for potentially different efficacies of jabs and doses. Recognizing this, we allow the update

$$S_t = (S_{t-1} - i_t)\,(1 - v_t\xi)\,, \tag{6.21}$$

where ξ is a noise term that is assigned a prior distribution. ξ helps to account for potentially systematic biases in calculating vaccine efficacy.

6.3 Installation

epidemia requires R v3.5.0 or above. The package can be installed directly from GitHub. However, this requires you to have a working C++ tool chain. To ensure that this is working, please first install **rstan** by following these installation instructions.

After installing **rstan**, running

```
R> #install.packages("devtools")
R> devtools::install_github("ImperialCollegeLondon/epidemia")
```

will install the latest development version of **epidemia**. If using windows, you can alternatively install the binary. Vignettes are not currently included in the package because they are computationally demanding, and are best viewed online.

6.4 Model Implementation

Here, we give a high-level overview of the workflow required for defining and fitting a model with **epidemia**. The primary model fitting function is `epim()`. This takes a model description and additional arguments relating to the fitting algorithm, and proceeds to fit the model using a precompiled Stan program. This is similar to the workflow for fitting Bayesian regression models with **rstanarm**. A key difference, however, is that the models fit by **epidemia** are generally complex, and are therefore inherently more difficult to specify. We simplify this process by taking a modular approach; models are defined through three distinct parts: transmission, infections and observations. These components of the model are defined with the functions `epirt()`, `epiinf()` and `epiobs()` respectively.

The package contains an example dataset `EuropeCovid` which contains data on daily death counts from Covid-19 in 11 European Countries from February through May 2020, and a set of binary indicators of non-pharmaceutical interventions. This is used as an example throughout.

```
R> library(dplyr)
R> library(epidemia)
R> library(rstanarm)
R> data("EuropeCovid")
```

We begin by describing `epim()` in more detail, and then proceed to discuss the three modeling functions.

6.4.1 Model Fitting

`epim()` is the only model fitting function in **epidemia**. It has arguments `rt`, `inf`, and `obs` which expect a description of the transmission model, infection model and all observational models respectively. Together, these fully define the joint distribution of data and parameters. Each of these model components are described in terms of variables that are expected to live in a single data frame, `data`. This data frame must be compatible with the model components, in the sense that *it holds all variables defined in these models*. For our example, these variables are the following.

```
R> data <- EuropeCovid$data
R> colnames(data)
```

```
[1] "country"
[2] "date"
[3] "schools_universities"
[4] "self_isolating_if_ill"
[5] "public_events"
```

```
[6] "lockdown"
[7] "social_distancing_encouraged"
[8] "deaths"
[9] "pop"
```

The `data` argument is described in more detail in Section 6.4.5.

In addition to taking a model description and a data frame, `epim()` has various additional arguments which specify how the model should be fit. If `algorithm = "sampling"` then the model will be fit using Stan's adaptive Hamiltonian Monte Carlo sampler (Hoffman and Gelman, 2014). This is done internally by calling `sampling()` from **rstan**. If instead this is `"meanfield"` or `"fullrank"`, then Stan's Variational Bayes algorithms (Kucukelbir et al., 2015, 2017) are employed by calling `vb()` from **rstan**. Any unnamed arguments in the call to `epim()` are passed directly onto the **rstan** sampling function. `epim()` returns a fitted model object of class `epimodel`, which contains posterior samples from the model along with other useful objects.

In general, Hamiltonian Monte Carlo should be used for final inference. Nonetheless, this is often computationally demanding, and Variational Bayes can often be used fruitful for quickly iterating models. All arguments for `epim()` are described in Table 6.1.

6.4.2 Transmission

`epirt()` defines the model for time-varying reproduction numbers, which was described in Section 6.2.4. Recall that these are modeled as a transformed linear predictor. `epirt()` has a `formula` argument which defines the linear predictor η, an argument `link` defining the link function `g`, and additional arguments to specify priors on parameters making up η.

A general R formula gives a symbolic description of a model. It takes the form `y ~ model`, where `y` is the response and `model` is a collection of terms separated by the `+` operator. `model` fully defines a linear predictor used to predict `y`. In this case, the "response" being modeled are reproduction numbers which are unobserved. `epirt()` therefore requires that the left hand side of the formula takes the form `R(group, date)`, where `group` and `date` refer to variables representing the modeled populations and dates respectively. The right hand side can consist of fixed effects, random effects, and autocorrelation terms. For our example, a viable call to `epirt()` is the following.

```
R> rt <- epirt(formula = R(country, date) ~ 1 + lockdown + public_events,
+              link = scaled_logit(7))
```

Here, two fixed effects are included which represent the effects of implementing lockdown and banning public events. These effects are assumed constant across countries. They could alternatively be partially pooled by using the term `(lockdown + public_events | country)`. For information on how to interpret such terms, please

Table 6.1 Formal arguments for the model fitting function `epim()`. The first three arguments listed below define the model to be fitted.

Argument	Description
`rt`	An object of class `epirt`, resulting from a call to `epirt()` (Section 6.4.2). This defines the model for time-varying reproduction numbers R. See Section 6.4.2 for more details.
`inf`	An object of class `epiinf`, resulting from a call to `epiinf()` (Section 6.4.3). This entirely defines the model for infections i_t.
`obs`	Either an object of class `epiobs`, or a list of such objects. Each of these define a model for an observation vector in `data`, and result from a call to `epiobs()` (Section 6.4.4).Each element of the list defines a model for an observed variable.
`data`	A dataframe with all data required for fitting the model. This includes all observations and covariates specified in the model. See Section 6.4.5 for more details.
`algorithm`	One of `"sampling"`, `"meanfield"` or `"fullrank"`. This determines the **rstan** sampling function to use for fitting the model. `"sampling"` corresponds to HMC, while `"meanfield"` and `"fullrank"` are Variational Bayes algorithms.
`group_subset`	If specified, a character vector naming a subset of groups/populations to include in the model.
`prior_PD`	If `TRUE`, parameters are sampled from their prior distributions. This is useful for prior predictive checks. Defaults to `FALSE`.
`...`	Additional arguments to pass to the **rstan** function used to fit the model. If `algorithm = "sampling"`, then this function is `sampling()`. Otherwise `vb()` is used.

read Appendix E.2. Using `link = scaled_logit(7)` lets the link function be the scaled logit link described by Equation (6.11), where $K = 7$ is the maximum possible value for reproduction numbers. For simplicity, we have omitted any prior arguments, however these should generally be specified explicitly. Please see Appendix E.1 for detailed information on how to use priors. All arguments for `epirt()` are listed in Table 6.1.

6.4.3 Infections

The infection model is represented by `epiinf()`. In the most basic version, this defines the distribution of the generation time of the disease, the number of days for which to seed infections, and the prior distribution on seeded infections. These three parameters are controlled by the arguments `gen`, `seed_days` and `prior_seeds` respectively. A possible model is the following.

```
R> inf <- epiinf(gen = EuropeCovid$si, seed_days = 6L,
+                prior_seeds = hexp(exponential(0.02)))
```

Table 6.2 Formal arguments for `epirt()`, which defines the model for R_t.

Argument	Description
`formula`	An object of class `formula` which determines the linear predictor η for R. The left hand side must take the form `R(group, date)`, where `group` must be a factor vector indicating group membership (i.e. country, state, age cohort), and `date` must be a vector of class `Date`. This is syntactic sugar for the reproduction number in the given group at the give date.
`link`	The link function g. Can be `"log"`, `"identity"` or a call to `scaled_logit()`. Defaults to `"log"`.
`center`	If `TRUE`, covariates specified in `formula` are centered to have mean zero. All priors should then be interpreted as priors on the centered covariates.
`prior`	Same as in `stan_glm()` from **rstanarm**. Defines the prior on fixed effects β. Priors provided by **rstanarm** can be used, and additionally `shifted_gamma`. **Note**: if `autoscale = TRUE` in the call to the prior function, then automatic rescaling takes place.
`prior_intercept`	Same as in `stan_glm()` from **rstanarm**. Prior for the regression intercept β_0 (if it exists).
`prior_covariance`	Same as in `stan_glmer()` from `rstanarm`. Defines the prior on the covariance matrix Σ. Only use if the `formula` has one or more terms of the form `(x | y)`, in which case there are parameters to partially pool, i.e. b has positive length.
`...`	Additional arguments to pass to `model.frame()` from **stats**.

`EuropeCovid$si` is a numeric vector representing the distribution for the serial interval of Covid-19. There is an implicit assumption that the generation time can be approximated well by the serial interval. Seeds are modeled hierarchically, and are described by (6.6) and (6.7). τ has been assigned an exponential prior with a mean of 50. Seeded infections are assumed to occur over a period of 6 days.

`epiinf()` has additional arguments that allow the user to extend the basic model. Using `latent = TRUE` replaces the renewal equation (6.3) with Equation (6.16). Daily infections are then treated as latent parameters that are sampled along with other parameters. The `family` argument specifies the distribution $p(i'_t, d)$, while `prior_aux` defines the prior on the coefficient of dispersion d.

Recall from Section 6.2.5 that the infection process may be modified to explicitly account for changes in infection rates as the remaining susceptible population is depleted. In particular, the adjustment ensures that cumulative infections never breaches the population size. It can be employed by setting `pop_adjust = TRUE` and using the `pop` argument to point towards a static variable in the data frame giving the population size. All argument to `epiinf()` are described in Table 6.3.

Table 6.3 Formal arguments for `epiinf()`, which defines the infection model.

Argument	Description
`gen`	A numeric vector giving the probability mass function g_k for the generation time of the disease (must be a probability vector).
`seed_days`	An integer giving the number of days $v+1$ for which to seed infections. Defaults to 6L.
`prior_seeds`	Prior distribution on the seed parameter i. Defaults to `hexp(prior_aux = rstanarm::exponential(0.03))`.
`latent`	If `TRUE`, treat infections as latent parameters using the extensions described in Section 6.2.5.
`family`	Specifies the family for the infection distribution $p(i'_t, d)$. Only used if `latent = TRUE`, and defaults to `"normal"`.
`prior_aux`	Prior on the auxiliary variable d of $p(i'_t, d)$. This is either the variance-to-mean ratio or the coefficient of variation, depending on the value of `fixed_vtm`. Only used if `latent = TRUE`.
`fixed_vtm`	If `TRUE`, then $p(i'_t, d)$ has a fixed variance-to-mean ratio, i.e. variance is $\sigma^2 = di'_t$; In this case, d refers to the *variance-to-mean ratio*. Id `FALSE` then instead standard deviation is assumed proportional to the mean, in which case d is the *coefficient of variation*. Only used if `latent = TRUE`.
`pop_adjust`	If `TRUE`, applies the population adjustment (6.18) to the infection process.
`pops`	A character vector giving the population variable. Only used if `pop_adjust = TRUE`.
`prior_susc`	Prior on S_{v-1}/P, the initial susceptible population as a proportion of the population size. If `NULL`, this is assumed to be equal to 1 (i.e. everyone is initially susceptible). Otherwise, can be a call to `normal()` from **rstanarm**, which assigns a normal prior truncated to $[0, 1]$. Only used if `pop_adjust = TRUE`.
`rm`	A character vector giving the variable corresponding to v_t, i.e. the proportion of S_t to remove at time t. Only used if `pop_adjust = TRUE`.
`prior_rm_noise`	Prior on the parameter ξ, which controls noise around v_t. If `NULL`, no noise is added. Only used if `pop_adjust = TRUE`.

6.4.4 Observations

An observational model is defined by a call to `epiobs()`. In particular, this must also make explicit the model for the multipliers α_t, and must also specify the coefficients π_k. `epiobs()` has a `formula` argument. The left hand side must indicate the observation vector to be modeled, while the right hand side defines a linear predictor for α_t. The argument `i2o` plays a similar role to the `gen` argument in `epiinf()`, however it instead corresponds the vector π in Equation (6.2).

Take for example the task of modeling daily `deaths`, which as we saw is a variable in `data`. A possible model is the following.

```
R> deaths <- epiobs(formula = deaths ~ 1, i2o = EuropeCovid$inf2death,
+                   link = scaled_logit(0.02))
```

Here, α_t corresponds to the infection fatality rate (IFR), and is modeled as an intercept transformed by the scaled-logit link. This implies that the IFR is constant over time and its value lies somewhere between 0% and 2%. If the prior on the intercept (specified by the `prior_intercept` argument) is chosen to be symmetric around zero, then the prior mean for the IFR is 1%. `EuropeCovid$inf2death` is a numeric simplex vector that gives the same delay distribution as used in Flaxman et al. (2020a). This is a density function for a discretized mixture of Gamma random variables.

Additional arguments include `family`, which specifies the sampling distribution $p(y_t, \phi)$. There are also arguments allowing the user to control prior distributions for effects in the linear predictor for α_t, and the prior on the auxiliary variable ϕ. All arguments to `epiobs()` are shown in Table 6.4.

Table 6.4 Formal arguments for `epiobs()`. This defines a single observation model. Multiple such models can be used and passed to `epim()` in a list.

Argument	Description
`formula`	An object of class `"formula"` which determines the linear predictor for the ascertainment rate. The left hand side must define the response that is being modeled (i.e. the actual observations, not the latent ascertainments)
`i2o`	A numeric (probability) vector defining the probability mass function π_k of the time from an infection to an observation.
`family`	A string representing the family of the sampling distribution $p(y_t, \phi)$. Can be one of `"poisson"`, `"neg_binom"`, `"quasi_poisson"`, `"normal"` or `"log_normal"`.
`link`	A string representing the link function used to transform the linear predictor. Can be one of `"logit"`, `"probit"`, `"cauchit"`, `"cloglog"`, `"identity"`. Defaults to `"logit"`.
`center`, `prior`, `prior_intercept`	same as in `epirt()`, described above.
`prior_aux`	The prior distribution for the auxiliary parameter ϕ, if it exists. Only used if family is `"neg_binom"` (reciprocal dispersion), `"quasi_poisson"` (dispersion), `"normal"` (standard deviation) or `"log_normal"` (sigma parameter).
`...`	Additional arguments for `model.frame()` from **stats**.

6.4.5 Data

Before fitting our first model in Section 6.4.6, we elaborate on the `data` argument to `epim()`. Recall that this must contain all variables used in the transmission and infection models, and in all observational models. For our example, `data` looks like

```
R> head(data)
```

```
# A tibble: 6 x 9
# Groups:   country [1]
  country date       schools_universiti~ self_isolating_if_~
  <fct>   <date>                   <int>               <int>
1 Austria 2020-02-22                   0                   0
2 Austria 2020-02-23                   0                   0
3 Austria 2020-02-24                   0                   0
4 Austria 2020-02-25                   0                   0
5 Austria 2020-02-26                   0                   0
6 Austria 2020-02-27                   0                   0
# ... with 5 more variables: public_events <int>,
#   lockdown <int>, social_distancing_encouraged <int>,
#   deaths <int>, pop <int>
```

The columns `country` and `date` define the region and time period corresponding to each of the remaining variables. `epim()` assumes that the first seeding day (i.e. the start of the epidemic) in each region is the first date found in the data frame. The last data found for each region is the final data at which the epidemic is simulated. It is up to the user to appropriately choose these dates. For our example, the first and last dates for each group can be seen as follows.

```
R> dates <- summarise(data, start = min(date), end = max(date))
R> head(dates)
```

```
# A tibble: 6 x 3
  country start      end
  <fct>   <date>     <date>
1 Austria 2020-02-22 2020-05-05
2 Belgium 2020-02-18 2020-05-05
3 Denmark 2020-02-21 2020-05-05
4 France  2020-02-07 2020-05-05
5 Germany 2020-02-15 2020-05-05
6 Italy   2020-01-27 2020-05-05
```

Here, the start dates have been heuristically chosen to be 30 days prior to observing 10 cumulative deaths in each country.

6.4.6 A First Fit

We are now ready to fit our first model. For this we return to the model fitting function `epim()`. The following command is used to instruct **epidemia** to run Markov chains in parallel, rather than sequentially, if multiple cores are detected.

```
R>  options(mc.cores = parallel::detectCores())
```

Our call to `epim()` is as follows. We use `refresh = 0` to suppress printing output in this chapter, however, this should not generally be used as such output is useful.

```
R> fm <- epim(rt = rt, inf = inf, obs = deaths, data = data,
+             group_subset = "France", algorithm = "sampling", iter = 1e3,
+             seed = 12345, refresh = 0)
```

The print method for `epimodel` objects prints summary statistics for model parameters. These are obtained from the sampled posterior distribution. Parameter are displayed according to which part of the model they belong to (transmission, observations, infections). An estimate of the standard deviation, labeled `MAD_SD` is displayed. This is the median absolute deviation from the median, and is more robust than naive estimates of the standard deviation for long-tailed distributions.

```
R> print(fm)
```

```
Rt regression parameters:
==========
coefficients:
                Median MAD_SD
R|(Intercept)    0.7    0.2
R|lockdown      -2.4    0.3
R|public_events -0.4    0.3

 deaths  regression parameters:
==========
coefficients:
                             Median MAD_SD
deaths|(Intercept)            0.0    0.2
deaths|reciprocal dispersion 10.4    0.4

Infection model parameters:
==========
              Median MAD_SD
seeds[France] 15.2    5.2
seeds_aux     27.3   22.0
```

Alternatively, the summary method can be used. This gives quantiles of the posterior draws, and also displays some MCMC diagnostics.

```
R> summary(fm)
```

```
Estimates:
                                mean   sd   10%   50%   90%
R|(Intercept)                   0.7    0.2  0.5   0.7   0.9
R|lockdown                     -2.4    0.3 -2.8  -2.4  -2.1
R|public_events                -0.4    0.3 -0.8  -0.4   0.0
deaths|(Intercept)              0.0    0.2 -0.3   0.0   0.2
seeds[France]                  16.2    5.9  9.6  15.2  24.0
seeds_aux                      39.7   38.0  8.7  27.3  84.8
deaths|reciprocal dispersion   10.5    0.5 10.1  10.4  11.1

MCMC diagnostics
                               mcse Rhat n_eff
R|(Intercept)                  0.0  1.0  1110
R|lockdown                     0.0  1.0  1119
R|public_events                0.0  1.0   921
deaths|(Intercept)             0.0  1.0  1422
seeds[France]                  0.2  1.0  1297
seeds_aux                      1.2  1.0  1061
deaths|reciprocal dispersion   0.0  1.0  2238
log-posterior                  0.1  1.0   738
```

6.5 Examples

6.5.1 Spanish Flu in Baltimore

Our first example infers R_t during the H1N1 pandemic in Baltimore in 1918, using only case counts and a serial interval. This is, relatively speaking, a simple setting for several reasons. Only a single population (that of Baltimore) and observational model (case data) are considered. R_t will follow a daily random walk with no additional covariates. Of course, **epidemia** is capable of more complex modeling, and Section 6.5.2 takes a step in this direction.

In addition to inferring R_t, this example demonstrates how to undertake posterior predictive checks to graphically assess model fit. The basic model outlined above is then extended to add variation to the infection process, as was outlined in Section 6.2.5. This is particularly useful for this example because infection counts are low. We will also see that the extended model appears to have a computational advantage in this setting.

The case data is provided by the R package **EpiEstim**.

```
R> library(EpiEstim)
R> data("Flu1918")
R> print(Flu1918)

$incidence
 [1]   5   1   6  15   2   3   8   7   2  15   4  17   4  10
[15]  31  11  13  36  13  33  17  15  32  27  70  58  32  69
[29]  54  80 405 192 243 204 280 229 304 265 196 372 158 222
[43] 141 172 553 148  95 144  85 143  87  73  70  62 116  44
[57]  38  60  45  60  27  51  34  22  16  11  18  11  10   8
[71]  13   3   3   6   6  13   5   6   6   5   5   1   2   2
[85]   3   8   4   1   2   3   1   0

$si_distr
 [1] 0.000 0.233 0.359 0.198 0.103 0.053 0.027 0.014 0.007
[10] 0.003 0.002 0.001
```

Data

First form the `data` argument, which will eventually be passed to the model fitting function `epim()`. Recall that this must be a data frame containing all observations and covariates used to fit the model. Therefore, we require a column giving cases over time. In this example, no covariates are required. R_t follows a daily random walk, with no additional covariates. In addition, the case ascertainment rate will be assumed at 100%, and so no covariates are used for this model either.

```
R> date <- as.Date("1918-01-01") + seq(0, along.with = c(NA, Flu1918$incidence))
R> data <- data.frame(city = "Baltimore", cases = c(NA, Flu1918$incidence),
+                     date = date)
```

The variable `date` has been constructed so that the first cases are seen on the second day of the epidemic rather than the first. This ensures that the first observation can be explained by past infections.

Transmission

Recall that we wish to model R_t by a daily random walk. This is specified by a call to `epirt()`. The `formula` argument defines the linear predictor which is then transformed by the link function. A random walk can be added to the predictor using the `rw()` function. This has an optional `time` argument which allows the random walk increments to occur at a different frequency to the `date` column. This can be employed, for example, to define a weekly random walk. If unspecified, the increments are daily. The increments are modeled as half-normal with a scale hyperparameter. The value of this is set using the `prior_scale` argument. This is used in the snippet below.

```
R> rt <- epirt(formula = R(city, date) ~ 1 + rw(prior_scale = 0.01),
+              prior_intercept = normal(log(2), 0.2), link = 'log')
```

The prior on the intercept gives the initial reproduction number R_0 a prior mean of roughly 2.

Observations

Multiple observational models can be collected into a list and passed to `epim()` as the `obs` argument. In this case, only case data is used and so there is only one such model.

```
R> obs <-  epiobs(formula = cases ~ 0 + offset(rep(1,93)), link = "identity",
+                 i2o = rep(.25,4))
```

For the purpose of this exercise, we have assumed that all infections will eventually manifest as a case. The above snippet implies *full* ascertainment, i.e. $\alpha_t = 1$ for all t. This is achieved using `offset()`, which allows vectors to be added to the linear predictor without multiplication by an unknown parameter.

The `i2o` argument implies that cases are recorded with equal probability in any of the four days after infection.

Infections

Two infection models are considered. The first uses the renewal equation (Equation 6.3) to propagate infections. The extended model adds variance to this process, and can be applied by using `latent = TRUE` in the call to `epiinf()`.

```
R> inf <- epiinf(gen = Flu1918$si_distr)
R> inf_ext <-  epiinf(gen = Flu1918$si_distr, latent = TRUE,
+                     prior_aux = normal(10,2))
```

The argument `gen` takes a discrete generation distribution. Here we have used the serial interval provided by **EpiEstim**. As in Section 6.4.3, this makes the implicit assumption that the serial interval approximates the generation time. `prior_aux` sets the prior on the coefficient of dispersion d. This prior assumes that infections have conditional variance around 10 times the conditional mean.

Fitting the Model

We are left to collect all remaining arguments required for `epim()`. This is done as follows.

```
R> args <- list(rt = rt, obs = obs, inf = inf, data = data, iter = 2e3,
+               seed = 12345)
R> args_ext <- args; args_ext$inf <- inf_ext
```

The arguments `iter` and `seed` set the number of MCMC iterations and seeds respectively, and are passed directly on to the `sampling()` function from **rstan**.

We wrap the calls to `epim()` in `system.time` in order to assess the computational cost of fitting the models. The snippet below fits both versions of the model. `fm1` and `fm2` are the fitted basic model and extended model respectively.

```
R> system.time(fm1 <- do.call(epim, args))

   user  system elapsed
341.488   1.550  93.214

R> system.time(fm2 <- do.call(epim, args_ext))

   user  system elapsed
 55.377   0.895  17.583
```

Note the stark difference in running time. The extended model appears to fit faster even though there are 87 additional parameters being sampled (daily infections after the seeding period, and the coefficient of dispersion). We conjecture that the additional variance around infections adds slack to the model, and leads to a posterior distribution that is easier to sample.

The results of both models are shown in Figure 6.1. This figure has been produced using **epidemia**'s plotting functions. The key difference stems from the infection process. In the basic model, infections are deterministic given R_t and seeds. However, when infection counts are low, we generally expect high variance in the infection process. Since this variance is unaccounted for, the model appears to place too much confidence in R_t in this setting. The extended model, on the other hand, has much wider credible intervals for R_t when infections are low. This is intuitive: when counts are low, changes in infections could be explained by either the variance in the offspring distribution of those who are infected, or by changes in the R_t value. This model captures this intuition.

Posterior predictive checks are shown in the bottom panel of Figure 6.1, and show that both models can fit the data well.

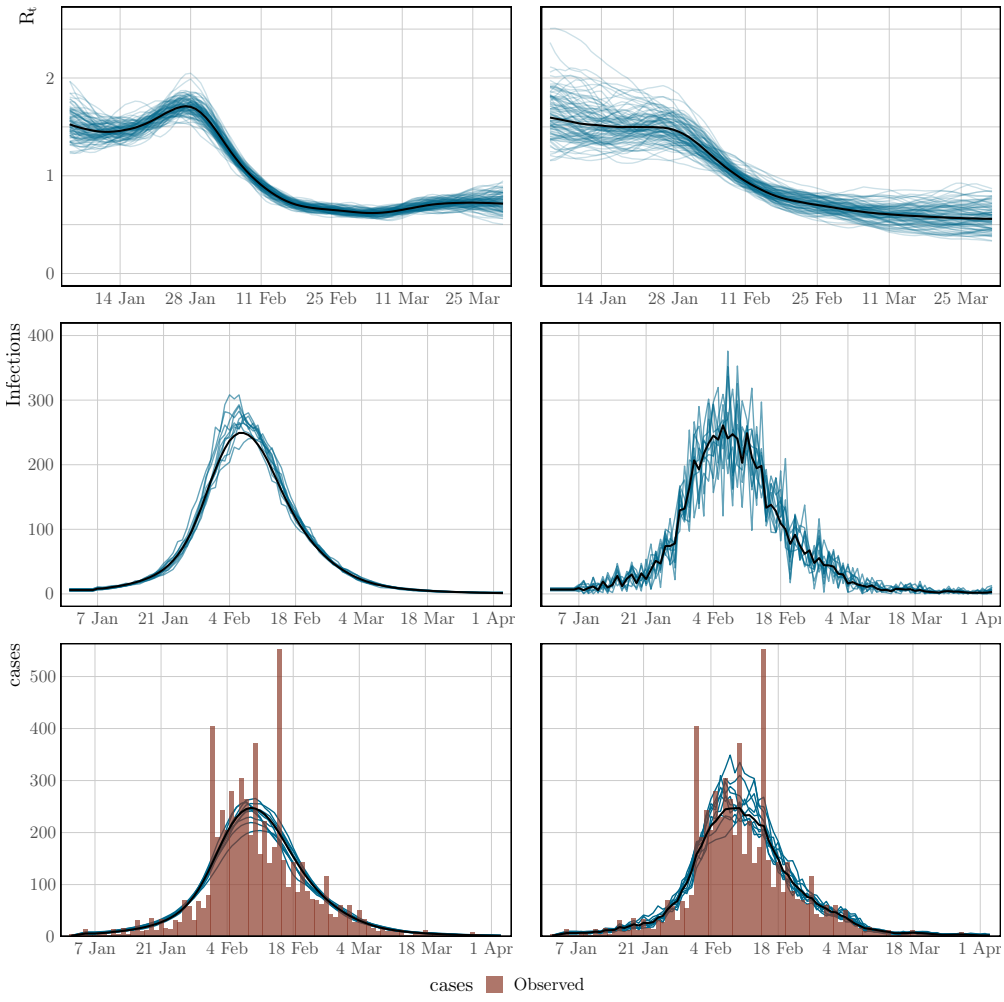


Fig. 6.1 Spaghetti plots showing the median (black) and sample paths (blue) from the posterior distribution. These result from the model fits in Section 6.5.1. Left corresponds to the basic model, and the right panel is for the extended version. Top: Inferred time-varying reproduction numbers, which have been smoothed over 7 days for illustration. Middle: Inferred latent infections. Bottom: Observed cases, and cases simulated from the posterior. These align closely, and so do not flag problems with the model fit.

6.5.2 Assessing the Effects of Interventions on COVID-19

The Spanish flu example (Section 6.5.1) considered inferring the instantaneous reproduction number over time in a single population. Here, we demonstrate some of the more advanced modeling capabilities of the package.

Consider modeling the evolution of an epidemic in multiple distinct regions. As discussed in Section 6.2.5, one can always approach this by modeling each group separately. It was argued that this approach is fast, because models may be fit independently. Nonetheless, often there is little high quality data for some groups, and the data does little to inform parameter estimates. This is particularly true in the early stages of an epidemic. Joining regions together through hierarchical models allows information to

be shared between regions in a natural way, improving parameter estimates while still permitting between group variation.

In this section, we use a hierarchical model to estimate the effect of non-pharmaceutical interventions (NPIs) on the transmissibility of Covid-19. We consider the same setup as Flaxman et al. (2020a): attempting to estimate the effect of a set of measures that were implemented in March 2020 in 11 European countries during the first wave of Covid-19. This will be done by fitting the model to daily death data. The same set of measures and countries that were used in Flaxman et al. (2020a) are also used here. Flaxman et al. (2020b) considered a version of this model that used partial pooling for all NPI effects. Here, we consider a model that uses the same approach.

This example is not intended to be a fully rigorous statistical analysis. Rather, the intention is to demonstrate partial pooling of parameters in **epidemia** and how to infer their effect sizes. We also show how to forecast observations into the future, and how to undertake counterfactual analyses.

Data

We use a data set `EuropeCovid2`, which is provided by **epidemia**. This contains daily death and case data in the 11 countries concerned up until the 1st July 2020. The data derives from the WHO COVID-19 explorer as of the 5th of January 2021. This differs from the data used in Flaxman et al. (2020a), because case and death counts have been adjusted retrospectively as new information came to light. **epidemia** also has a data set `EuropeCovid` which contains the same data as that in Flaxman et al. (2020a), and this could alternatively be used for this exercise.

`EuropeCovid2` also contains binary series representing the set of five mitigation measures considered in Flaxman et al. (2020a). These correspond to the closing of schools and universities, the banning of public events, encouraging social distancing, requiring self-isolation if ill, and finally the implementation of full lockdown. The dates at which these policies were enacted are the same as those used in Flaxman et al. (2020a).

Load the data set as follows.

```
R> data("EuropeCovid2")
R> data <- EuropeCovid2$data
R> head(data)
```

```
# A tibble: 6 x 11
# Groups:   country [1]
  id    country date       cases deaths schools_universities
  <chr> <chr>   <date>     <int>  <int>                <int>
1 AT    Austria 2020-01-03     0      0                    0
2 AT    Austria 2020-01-04     0      0                    0
3 AT    Austria 2020-01-05     0      0                    0
4 AT    Austria 2020-01-06     0      0                    0
```

```
5 AT    Austria 2020-01-07     0      0                    0
6 AT    Austria 2020-01-08     0      0                    0
# ... with 5 more variables: self_isolating_if_ill <int>,
#   public_events <int>, lockdown <int>,
#   social_distancing_encouraged <int>, pop <int>
```

Recall that for each country, **epidemia** will use the earliest date in `data` as the first date to begin seeding infections. Therefore, we must choose an appropriate start date for each group. One option is to use the same rule as in Flaxman et al. (2020a), and assume that seeding begins in each country 30 days prior to observing 10 cumulative deaths. To do this, we filter the data frame as follows.

```
R> data <- filter(data, date > date[which(cumsum(deaths) > 10)[1] - 30])
```

This leaves the following assumed start dates.

```
R> dates <- summarise(data, start = min(date), end = max(date))
R> head(dates)
```

```
# A tibble: 6 x 3
  country start      end
  <chr>   <date>     <date>
1 Austria 2020-02-23 2020-06-30
2 Belgium 2020-02-15 2020-06-30
3 Denmark 2020-02-22 2020-06-30
4 France  2020-02-09 2020-06-30
5 Germany 2020-02-16 2020-06-30
6 Italy   2020-01-28 2020-06-30
```

Although `data` contains observations up until the end of June, we fit the model using a subset of the data. We hold out the rest to demonstrate forecasting out-of-sample. Following Flaxman et al. (2020a), the final date considered is the 5$^{\text{th}}$ May.

```
R> data <- filter(data, date < as.Date("2020-05-05"))
```

Model Components

We have seen several times now that **epidemia** require the user to specify three model components: transmission, infections, and observations. These are now considered in turn.

Transmission Country-specific reproduction numbers $R_t^{(m)}$ are expressed in terms of the control measures. Since the measures are encoded as binary policy indicators, reproduction rates must follow a step function. They are constant between policies, and either increase or decrease as policies come into play. The implicit assumption, of course,

is that only control measures may affect transmission, and that these effects are fully realized instantaneously.

Let $t_k^{(m)} \geq 0$, $k \in \{1, \ldots, 5\}$ be the set of integer times at which the k^{th} control measure was enacted in the m^{th} country. Accordingly, we let $I_k^{(m)}$, $k \in \{1, \ldots, 5\}$ be a set of corresponding binary vectors such that

$$I_{k,t}^{(m)} = \begin{cases} 0, & \text{if } t < t_k^{(m)} \\ 1. & \text{if } t \geq t_k^{(m)} \end{cases} \tag{6.22}$$

Reproduction numbers are mathematically expressed as

$$R_t^{(m)} = R' g^{-1} \left(b_0^{(m)} + \sum_{k=1}^{5} \left(\beta_k + b_k^{(m)} \right) I_{k,t}^{(m)} \right), \tag{6.23}$$

where $R' = 3.25$ and g is the logit-link. Parameters $b_0^{(m)}$ are country-specific intercepts, and each $b_k^{(m)}$ is a country effect for the k^{th} measure. The intercepts allow each country to have its own initial reproduction number, and hence accounts for possible variation in the inherent transmissibility of Covid-19 in each population. β_k is a fixed effect for the k^{th} policy. This quantity corresponds to the average effect of a measure across all countries considered.

Control measures were implemented in quick succession in most countries. For some countries, a subset of the measures were in fact enacted simultaneously. For example, Germany banned public events at the same time as implementing lockdown. The upshot of this is that policy effects are *highly colinear* and may prove difficult to infer with uninformative priors.

One potential remedy is to use domain knowledge to incorporate information into the priors. In particular, it seems a priori unlikely that the measures served to increase transmission rates significantly. It is plausible, however, that each had a significant effect on reducing transmission. A symmetric prior like the Gaussian does not capture this intuition and increases the difficulty in inferring effects, because they are more able to offset each other. This motivated the prior used in Flaxman et al. (2020a), which was a Gamma distribution shifted to have support other than zero.

We use the same prior in our example. Denoting the distribution of a Gamma random variable with shape a and scale b by $\text{Gamma}(a, b)$, this prior is

$$-\beta_k - \frac{\log(1.05)}{6} \sim \text{Gamma}(1/6, 1). \tag{6.24}$$

The shift allows the measures to increase transmission slightly.

All country-specific parameters are partially pooled by letting

$$b_k^{(m)} \sim N(0, \sigma_k), \tag{6.25}$$

where σ_k are standard deviations, $\sigma_0 \sim \text{Gamma}(2, 0.25)$ and $\sigma_k \sim \text{Gamma}(0.5, 0.25)$ for all $k > 0$. This gives the intercept terms more variability under the prior.

The transmission model described above is expressed programmatically as follows.

```
R> rt <- epirt(formula = R(country, date) ~ 0 + (1 + public_events +
+                schools_universities + self_isolating_if_ill +
+                social_distancing_encouraged + lockdown || country) +
+                public_events + schools_universities + self_isolating_if_ill +
+                social_distancing_encouraged + lockdown,
+              prior = shifted_gamma(shape = 1/6, scale = 1, shift = log(1.05)/6),
+              prior_covariance = decov(shape = c(2, rep(0.5, 5)), scale = 0.25),
+              link = scaled_logit(6.5))
```

The operator `||` is used rather than `|` for random effects. This ensures that all effects for a given country are independent, as was assumed in the model described above. Using `|` would alternatively give a prior on the full covariance matrix, rather than on the individual σ_i terms. The argument `prior` reflects Equation (6.24). Since country effects are assumed independent, the `decov` prior reduces to assigning Gamma priors to each σ_i. By using a vector rather than a scalar for the `shape` argument, we are able to give the prior on the intercepts a larger shape parameter.

Infections Infections are kept simple here by using the basic version of the model. That is to say that infections are taken to be a deterministic function of seeds and reproduction numbers, propagated by the renewal process. Extensions to modeling infections as parameters and adjustments for the susceptible population are not considered. The model is defined as follows.

```
R> inf <- epiinf(gen = EuropeCovid$si, seed_days = 6)
```

`EuropeCovid$si` is a numeric vector giving the serial interval used in Flaxman et al. (2020a). As in that work, we make no distinction between the generation distribution and serial interval here.

Observations In order to infer the effects of control measures on transmission, we must fit the model to data. Here, daily deaths are used. In theory, additional types of data can be included in the model, but such extension are not considered here. A simple intercept model is used for the infection fatality rate (IFR). This makes the assumption that the IFR is constant over time. The model can be written as follows.

```
R> deaths <- epiobs(formula = deaths ~ 1, i2o = EuropeCovid2$inf2death,
+                   prior_intercept = normal(0,0.2), link = scaled_logit(0.02))
```

By using `link = scaled_logit(0.02)`, we let the IFR range between 0% and 2%. With the symmetric prior on the intercept, this gives the IFR a prior mean of 1%. `EuropeCovid2$inf2death` is a numeric vector giving the same distribution for the time from infection to death as that used in Flaxman et al. (2020a).

Model Fitting

In general, **epidemia**'s models should be fit using Hamiltonian Monte Carlo. For this example, however, we use Variational Bayes (VB) as opposed to full MCMC sampling. This is because full MCMC sampling of a joint model of this size is computationally demanding, due in part to renewal equation having to be evaluated for each region and for each evaluation of the likelihood and its derivatives. Nonetheless, VB allows rapid iteration of models and may lead to reasonable estimates of effect sizes. For this example, we have also run full MCMC, and the inferences reported here are not substantially different.

Prior Check Section 6.5.1 gave an example of using posterior predictive checks. It is also useful to do prior predictive checks as these allow the user to catch obvious mistakes that can occur when specifying the model, and can also help to affirm that the prior is in fact reasonable.

In **epidemia** we can do this by using the `priorPD = TRUE` flag in `epim()`. This discards the likelihood component of the posterior, leaving just the prior. We use Hamiltonian Monte Carlo over VB for the prior check, partly because sampling from the prior is quick (it is the likelihood that is expensive to evaluate). In addition, we have defined Gamma priors on some coefficients, which are generally poorly approximated by VB.

```
R> args <- list(rt = rt, inf = inf, obs = deaths, data = data, seed = 12345,
+               refresh = 0)
R> pr_args <- c(args, list(algorithm = "sampling", iter = 1e3, prior_PD = TRUE))
R> fm_prior <- do.call(epim, pr_args)
```

Figure 6.2 shows approximate samples of $R_{t,m}$ from the prior distribution. This confirms that reproduction numbers follow a step function, and that rates can both increase and decrease as measures come into play.

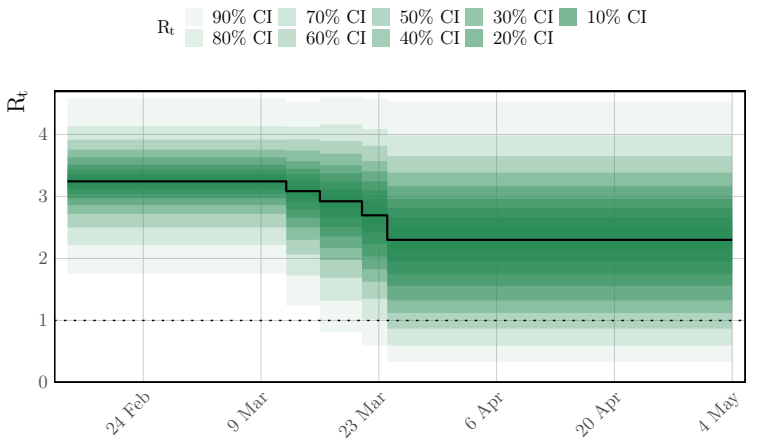


Fig. 6.2 A prior predictive check for reproduction numbers R_t in the multilevel model of Section 6.5.2. Only results for the United Kingdom are presented. The prior median is shown in black, with credible intervals shown in various shades of green. The check appears to confirm that R_t follows a step-function, as we expect given the definition in Section 6.5.2.

Approximating the Posterior The model will be fit using Variational Bayes by using `algorithm = "fullrank"` in the call to `epim()`. This is generally preferable to `"meanfield"` for these models, largely because `"meanfield"` ignores posterior correlations. We decrease the parameter `tol_rel_obj` from its default value, and increase the number of iterations to aid convergence.

```
R> args$algorithm <- "fullrank"; args$iter <- 5e4; args$tol_rel_obj <- 1e-3
R> fm <- do.call(epim, args)
```

A first step in evaluating the model fit is to perform posterior predictive checks. This is to confirm that the model adequately explains the observed daily deaths in each region. This can be done using the command `plot_obs(fm, type = "deaths", levels = c(50, 95))`. The plot is shown in Figure 6.3.

Figure 6.3 suggest that the epidemic was bought under control in each group considered. Indeed, one would expect that the posterior distribution for reproduction numbers lies largely below one in each region. Figure 6.4 is the result of `plot_rt(fm, step = T, levels = c(50,95))`, and confirms this.

Effect Sizes

In **epidemia**, estimated effect sizes can be visualized using the `plot.epimodel` method. This serves a similar purpose to `plot.stanreg` in **rstanarm**, providing an interface to the **bayesplot** package. The models in **epidemia** often have many parameters, some of which pertain to a particular part of the model (i.e. transmission), and some which pertain to

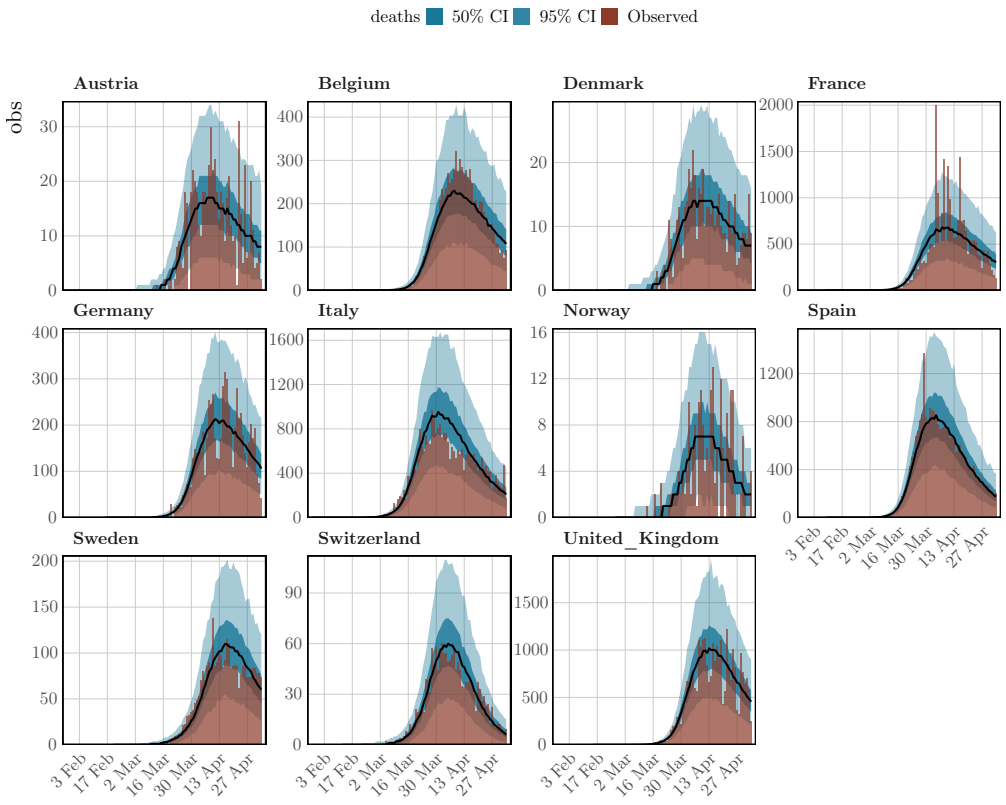


Fig. 6.3 Posterior predictive checks for the multilevel model. Observed daily deaths (red) is plotted as a bar plot. Credible intervals from the posterior are plotted in shades of blue, in addition to the posterior median in black.

particular groups (i.e., country-specific terms). Therefore `plot.epimodel` has arguments `par_models`, `par_types` and `par_groups`, which restrict the parameters considered to particular parts of the model.

As an example, credible intervals for the global coefficients β_i can be plotted using the command `plot(fm, par_models = "R", par_types = "fixed")`. This leads to Figure 6.5.

Figure 6.5 shows a large negative coefficient for lockdown, suggesting that this is on average the most effective intervention. The effect of banning public events is the next largest, while the other policy effects appear closer to zero. Note that the left plot in 6.5 shows only global coefficients, and does not show inferred effects in any given country. To assess the latter, one must instead consider the quantities $\beta_i + b_i^{(m)}$. We do this by extracting the underlying draws using `as.matrix.epimodel`, as is done below for Italy.

```
R> beta <- as.matrix(fm, par_models = "R", par_types = "fixed")
R> b <- as.matrix(fm, regex_pars = "^R\\|b", par_groups = "Italy")
R> mat <- cbind(b[,1], beta + b[,2:6])
R> labels <- c("Events", "Schools", "Isolating", "Distancing", "Lockdown")
R> colnames(mat) <- c("Intercept", labels)
```

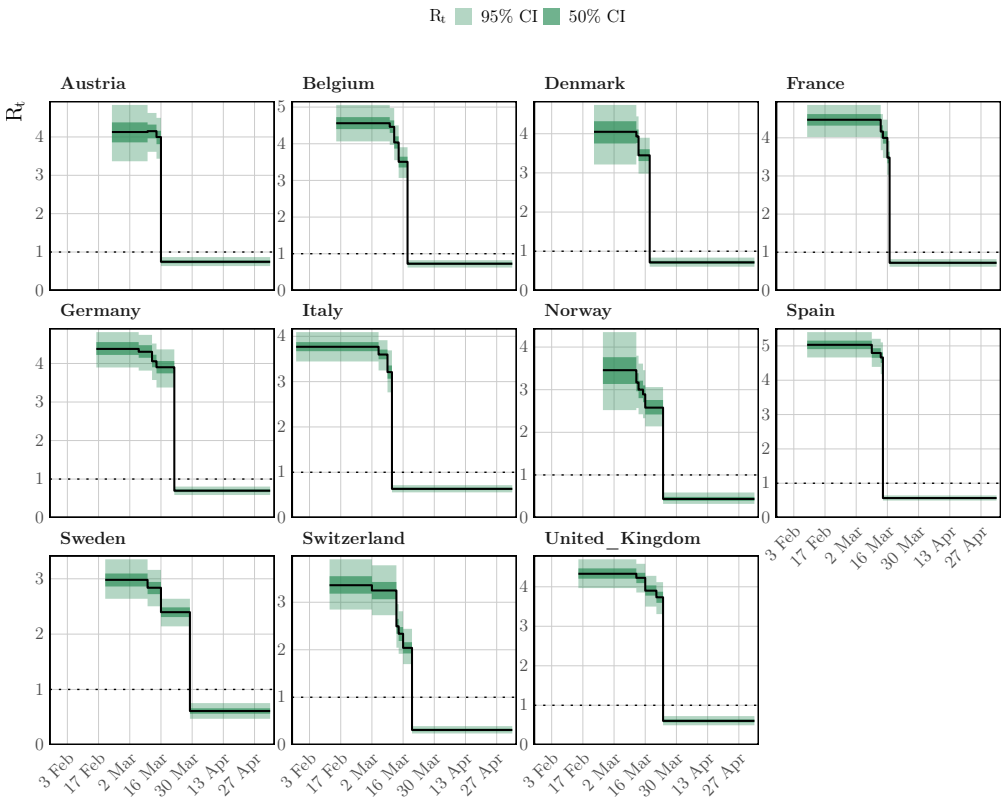


Fig. 6.4 Inferred reproduction numbers in each country. Credible intervals from the posterior are plotted in shades of green, in addition to the posterior median in black.

Calling `bayesplot::mcmc_intervals(mat)` leads to the results shown in the right panel of Figure 6.5.

Figure 6.5 has relatively narrow intervals for many of the effect sizes. This appears to be an artifact of using Variational Bayes. In particular, when repeating this analysis with full MCMC, we observe that the intervals for all policies other than lockdown overlap with zero.

Consider now the role of partial pooling in this analysis. Figure 6.4 shows that Sweden did enough to reduce R below one. However, it did so without a full lockdown. Given the small effect sizes for other measures, the model must explain Sweden using the country-specific terms. Figure 6.6 shows estimated seeds, intercepts and the effects of banning public events for each country. Sweden has a lower intercept than other terms which in turn suggests a lower R_0 - giving the effects less to do to explain Sweden. There is greater variability in seeding, because the magnitude of future infections becomes less sensitive to initial conditions when the rate of growth is lower. Figure 6.6 shows that the model estimates a large negative coefficient for public events in Sweden. This is significantly larger than the effects for other policies - which are not reported here. However, the idiosyncrasies relating to Sweden must be explained in this model by at least one of the covariates, and the large effect for public policy in Sweden is most probably an

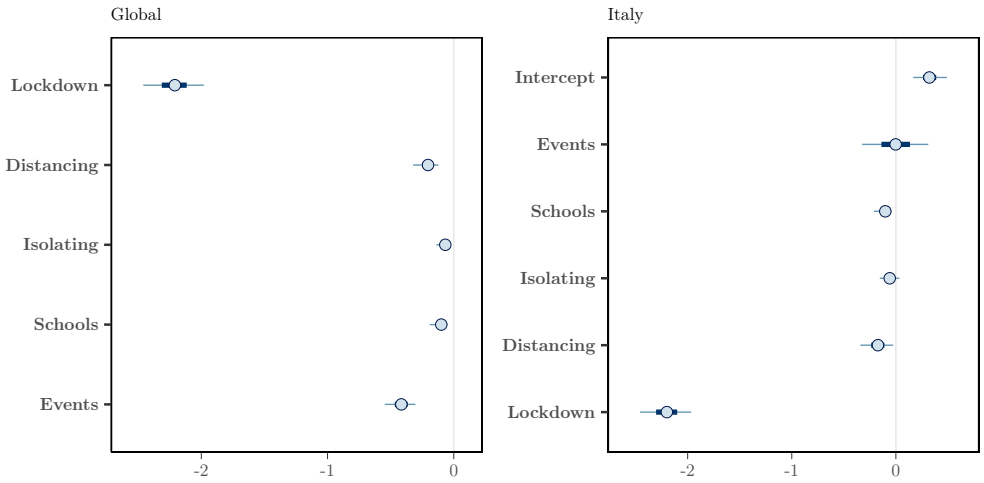


Fig. 6.5 Left: Global Effect sizes for the five policy measures considered. Right: Effect sizes specific to Italy. The global and country-specific effects may differ because the effects are partially pooled.

artifact of this. Nonetheless, the use of partial pooling is essential for explaining difference between countries. If full pooling were used, effect sizes would be overly influenced by outliers like Sweden. This argument is made in more detail in Flaxman et al. (2020b).

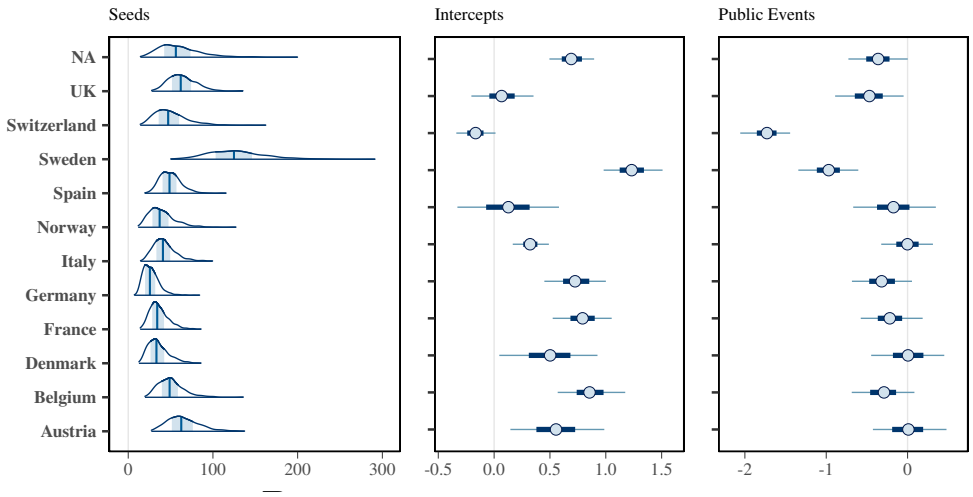


Fig. 6.6 Left: Inferred daily seeded infections in each country. These have been assumed to occur over a period of 6 days. Middle: Estimated Intercepts in the linear predictor for reproduction numbers. Right: Country-specific effect sizes corresponding to the banning of public events.

Forecasting

Forecasting within **epidemia** is straightforward, and consists of constructing a new data frame which is used in place of the original data frame. This could, for example, change

the values of covariates, or alternatively include new observations in order to check the out-of-sample performance of the fitted model.

Recall that `EuropeCovid2` holds daily death data up until the end of June 2020, however we only fitted the model up until the 5th May. The following constructs a data frame `newdata` which contains the additional observations. Note that we are careful to select the same start dates as in the original data frame.

```
R> newdata <- EuropeCovid2$data
R> newdata <- filter(newdata, date > date[which(cumsum(deaths) > 10)[1] - 30])
```

This data frame can be passed to plotting functions `plot_rt()`, `plot_obs()`, `plot_infections()` and `plot_infectious()`. If the raw samples are desired, we can also pass as an argument to `posterior_rt()`, `posterior_predict()` etc. The top panel of Figure 6.7 is the result of using the command `plot_obs(fm, type = "deaths", newdata = newdata, groups = "Italy")`. This plots the out of sample observations with credible intervals from the forecast.

Counterfactuals

Counterfactual scenarios are also easy. Again, one simply has to modify the data frame used. In this case we shift all policy measures back three days.

```
R> shift <- function(x, k) c(x[-(1:k)], rep(1,k))
R> days <- 3
R>
R> newdata <- mutate(newdata,
+   lockdown = shift(lockdown, days),
+   public_events = shift(public_events, days),
+   social_distancing_encouraged = shift(social_distancing_encouraged, days),
+   self_isolating_if_ill = shift(self_isolating_if_ill, days),
+   schools_universities = shift(schools_universities, days)
+ )
```

The bottom panel of Figure 6.7 visualizes the counterfactual scenario of all policies being implemented in the UK three days earlier. Deaths are projected over both the in-sample period, and the out-of-sample period. The left plot is obtained using `plot_obs(fm, type = "deaths", newdata = newdata, groups = "United_Kingdom")`, while the right plot adds the `cumulative = TRUE` argument. We reiterate that these results are not intended to be fully rigorous: they are simply there to illustrate usage of **epidemia**.

6.5.3 Tracking SARS-CoV-2 In England

Previous examples conditioned on a single observation series (case and death counts respectively). Simple models were used for describing ascertainment rates. Section 6.5.1

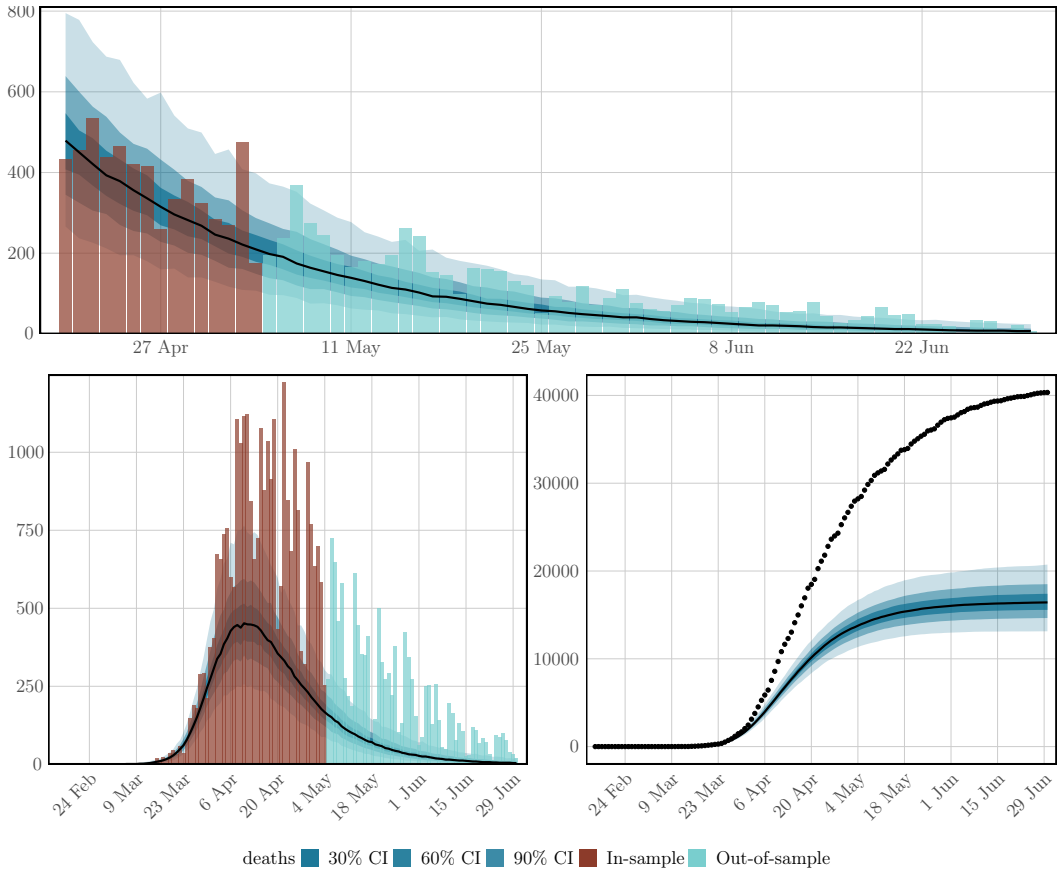


Fig. 6.7 Forecasts and counterfactual scenarios. All results pertain to the United Kingdom. Top: An out-of-sample forecast for daily deaths. Below: Results corresponding to a counterfactual whereby all policies were implemented 3 days earlier. Left: Credible intervals for daily deaths under this scenario. Right: Cumulative deaths. The black dotted line shows observed cumulative deaths.

assumed *full ascertainment* of infections, while Section 6.5.2 took the IFR to be *constant but unknown*. Population adjustments were not considered, and both examples modeled the entire history of the epidemic from the first observed cases.

Here we model the trajectory of SARS-CoV-2 in England using data over a two month period from late March 2021 through to the end of May. In doing so, we relax the aforementioned modeling assumptions and demonstrate further capabilities of the package. Population adjustments are applied in order to explicitly account for pre-existing immunity in the population. The model is conditioned on two observation series: daily case counts, and positivity from seroprevalence surveys.

Fitting the model to case counts requires a model for the case ascertainment rates (IAR). Unlike in previous examples, we do not assume that this is constant. While the IFR of Sars-CoV-2 may be broadly stable over several weeks or months, the IAR could vary frequently, not least due to changes in testing capacity, surge testing, and the general prevalence of the disease. Therefore we opt instead to *infer the time-varying IAR from positivity data*, which should provide an "anchor" for the absolute size of the latent

infection process. We allow IAR dynamics to be inferred by modeling it as a random walk.

Our goal is to infer recent infection dynamics, and also to project future infection and case counts. These goals do not necessitate a detailed understanding of the fully history of disease dynamics, and is primarily why we limit ourselves to a two month period. Other reasons include that modeling the entire history is computationally demanding and requires more intricate modeling to explain the full trajectory.

Data

epidemia has a data set `EnglandNewCases` that contains daily counts of confirmed SARS-CoV-2 infections in England from the 1st January 2020 up to and including the 30th May 2021. The data was downloaded from Public Health England (2020) on the 4nd of June 2021.

```
R> data("EnglandNewCases")
R> data <- EnglandNewCases
```

The model will be fit to the most recent two months (60 days) of case count data, starting on the 1st April. We take the first modeled date to be 20 days prior to this initial observation, which is the 12th March. This initial 20 days will be the seeding period.

```
R> data <- filter(data, date > max(date) - 80)
R> data$cases[1:20] <- NA
```

Transmission Model

As in Section 6.5.1, we model R_t as a transformed random walk with daily updates. In later sections, however, we will make projections that assume R_t remains at its current value. For this reason, we provide a new column in `data` to control the random walk increments. This will allow us to "stop" the walk updating at dates after the 30th May.

```
R> data <- data %>% mutate(dt = replace(date, date < as.Date("2020-04-01"), NA))
```

The model for R_t is

$$R_t = Kg^{-1}\left(\beta_0 + w_t\right),$$

where $K = 7$ and g is the logit link. The term w_t is a random walk satisfying the recursion $w_t = w_{t-1} + \varepsilon_t$ and initial condition $w_{-1} = 0$. The daily updates ε_t are given a prior $\varepsilon_t \sim \mathcal{N}(0, \sigma)$, and the scale hyperparameter follows $\sigma \sim \mathcal{N}^+(0, 0.05)$. The intercept β_0 is assigned a normal prior, so that $\beta_0 \sim \mathcal{N}(-1, 1)$. This prior has been chosen to reflect our belief that the epidemic was under control at start of March. We have, however, provided a large scale to reflect uncertainty over the initial value. The full transmission model is represented as follows.

```
R> rt <- epirt(formula = R(region, date) ~ 1 + rw(time = dt, prior_scale=0.05),
+              link = scaled_logit(7), prior_intercept = normal(-1,1))
```

Infection Model

Daily recorded cases are consistently above 1000 over the period considered. Given the large infection counts, we do not consider extending the infection model to add variation around the renewal equation. Such extensions would be useful for modeling the epidemic at a finer scale, say at the local authority level. Please refer to Section 6.5.1 for an example of how this is done.

We account for pre-existing immunity through population adjustments, which were described in Section 6.2.5. The first modeled date is the 12$^{\text{th}}$ of March 2021, over a year before SARS-CoV-2 was first detected in England. It is likely that prior infection and vaccination will have induced a degree of immunity within the population. Accounting for this helps to constrain long-term forecasts for the size of the susceptible population.

Mathematically, our model describes infections i_t for $t > 0$ by

$$
\begin{aligned}
i'_t &= R_t \sum_{s<t} i_s \pi_{t-s}, \\
i_t &= S_t \left(1 - \exp\left(-\frac{i'_t}{P}\right)\right),
\end{aligned}
$$

where P is the population size and S_t is the time-varying susceptible population. We use the same generation distribution π_k as in Section 6.5.2. The initial susceptible population S_v is given a prior $S_v \sim \mathcal{N}(0.48, 0.10)$. This is motivated by an ONS antibody survey (ONS, 2021a), which estimates that 51% of the population in England would have tested positive for SARS-CoV-2 antibodies between the 8$^{\text{nd}}$ March and the 14$^{\text{th}}$ March. It is important however to distinguish the presence of antibodies from immunity. An individual's level of antibodies may be below the required threshold to test positive; however, have protection through T-cells. Similarly, the presence of antibodies does not imply immunity. For these reasons, and due to wide credible intervals for the ONS estimate, we have used a large standard deviation on the prior immune population.

The population of England was estimated to be 56,286,961 as of mid-2019 (ONS, 2021c). This is added to `data` as a static variable.

```
R> data$pop <- pop <-  56286961
```

The infection model is expressed as follows.

```
R> inf <- epiinf(gen = EuropeCovid$si, seed_days=20L, pop_adjust = TRUE, pops = pop,
+                prior_susc = normal(0.49, 0.1), prior_seeds = normal(15e3, 2e3))
```

Observation Models

Models are defined for our two data series: case counts and positivity rates derived from a seroprevalence study. These are discussed in turn.

Case Counts Letting $Y_t^{(1)}$ denote daily case counts, we model $Y_t^{(1)} \sim \text{QP}(y_t^{(1)}, d)$ and

$$y_t = \alpha_t^{(1)} \left(\frac{1}{7} \sum_{s=1}^{7} i_{t-s-3} \right),$$

where QP denotes the Quasi-Poisson distribution with mean $y_t^{(1)}$ and variance-to-mean ratio d, which is given the prior $d \sim \mathcal{N}^+(10, 5)$. The assumption is that infections over the last three days are undetected, and those occurring over the week before are equally likely to be detected.

The IAR $\alpha_t^{(1)}$ is modeled as a random walk with additional dummy day-of-week variables to account for a clear weekly pattern in the data (see Figure 6.10). We take

$$\alpha_t^{(1)} = g^{-1} \left(\beta_0^{(1)} + w_t^{(1)} + \sum_{k=1}^{6} \gamma_k D_{k,t} \right),$$

where g is the logit-link, $\beta_0^{(1)} \sim \mathcal{N}(0, 1.5)$ is an intercept, and $w_t^{(1)}$ is a random walk following the same model as the walk for R_t, however starting on the first observation date (1st April) rather than immediately after the seeding period. The prior on $\beta_0^{(1)}$ has a large scale to reflect our prior uncertainty over the IAR, allowing the posterior to be largely driven by the positivity data. The terms $D_{k,t}$ are dummy indicators for the day of the week, and $\gamma_k \sim \mathcal{N}(0, 0.5)$ are associated weekday effects. These terms allow modeling weekly seasonality. First add the day-of-week factors to `data`.

```
R> data$day <- lubridate::wday(data$date, label=T)
```

The aforementioned model is defined as follows.

```
R> cases <- epiobs(formula = cases ~ 1 + factor(day, ordered = FALSE) +
+                    rw(time = dt, prior_scale = 0.05),
+                  i2o =  c(rep(0,3), rep(1/7,7)), link = "logit",
+                  prior_intercept = normal(0,1.5), prior = normal(0,0.5),
+                  family = "quasi_poisson", prior_aux = normal(10,5))
```

Seroprevalence Data The ONS Infection survey (ONS, 2021b) provides weekly estimates of positivity rates among the English population. These numbers are based on repeated RT-PCR tests on a representative sample of the population. We leverage this data for the week beginning the 28th March, and for the subsequent 6 weeks. We arbitrarily use Thursday of each week as the representative date for the positivity rates.

```
R> ons <- data.frame(date = as.Date("2021-04-01") + 7 * (0:6),
+                    positivity = c(0.21, 0.17, 0.1, 0.08, 0.07, 0.09, 0.09))
R> data <- left_join(data, ons)
```

Let $Y_t^{(2)}$ be the observed positivity rate at time t. This is an estimate of the proportion of the population who, at that point in time, would test positive on an RT-PCR test. We model this as $Y_t^{(2)} \sim \mathcal{N}(y_t^{(2)}, \sigma^2)$, where

$$y_t^{(2)} = \frac{\sum_{s=1}^{14} i_{t-s-3}}{P}. \tag{6.26}$$

Equation (6.26) assumes that those infected in the previous three days will definitively test negative, and that everyone infected within the two weeks before this will definitively test positive. All infections occurring before this will not be detected. This is, of course, a simplification. A careful model requires an understanding of the likelihood of testing positive in an RT-PCR test k days after infection, and also accounting for the sensitivity/specificity of the tests. We have made the above assumption in the absence of such information.

Equation (6.26) does not fit within our usual framework for observations. In particular, there is no explicit ascertainment rate, and the infection weights π_k do not form a probability distribution. Nonetheless, infections are still weighted linearly, and this model can be represented as follows.

```
R> data$off <- 1
R> ons <- epiobs(formula = positivity ~ 0 + offset(off),
+                i2o = c(rep(0,3), rep(1,14)) * 100 / pop, link = "identity",
+                family = "normal", prior_aux = normal(0.01, 2.5e-3))
```

Here we have assigned a prior $\sigma \sim \mathcal{N}^+(0.01, 0.0025)$. This is motivated by the credible intervals in the ONS study.

Fitting and Analysis

The model is fit with the NUTS sampler. In order to ensure a large enough effective sample size, we increase the maximum tree depth for the algorithm and use 4,000 iterations.

```
R> fm <- epim(rt = rt, inf = inf, obs = list(cases, ons), data = data,
+             iter = 4e3, control = list(max_treedepth = 12), seed=12345)
```

Figure 6.8 plots credible intervals for R_t over the two month period for which we have data. The model appears confident that R_t has risen above 1 over this time, and indeed daily cases appear to be rising as of the end of May. The credible intervals widen towards 30$^{\text{th}}$ May because there is a lag between reproduction numbers and recording cases.

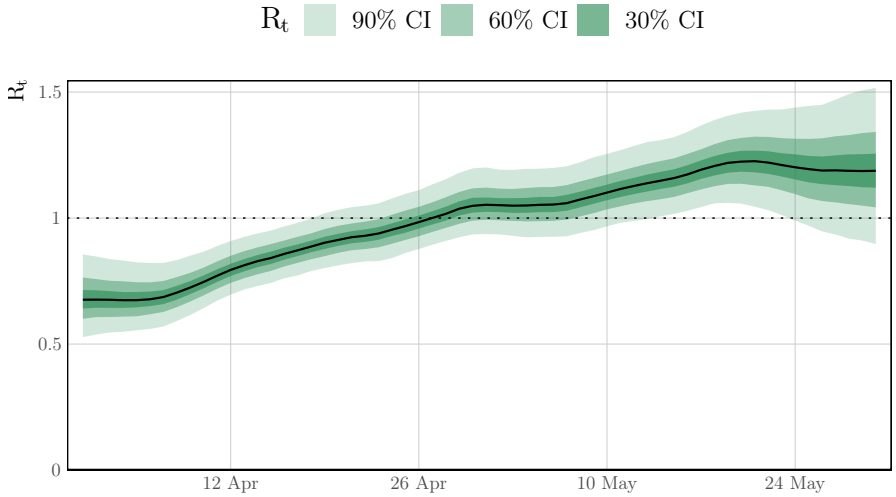


Fig. 6.8 R_t in England over the period starting on the 1^{April} 2021 and ending the 30^{th} May 2021.

The role of the population adjustment in this model is best understood by looking at long term forecasts. As was first shown in Section 6.5.2, to make forecasts we have to construct a new data frame. This can be passed as the `newdata` argument to **epidemia**'s plotting functions.

```
R> fut <- tibble(date = max(data$date) + seq_len(150), region = "England",
+                cases = -1, dt = max(data$date), positivity = -1, off = 1,
+                pop = data$pop[1])
R> fut$day <- lubridate::wday(fut$date, label=T)
R> newdata <- rbind(data, fut)
```

For all dates after the modeled period, the `dt` column is equal to the final modeled day (30^{th} May). In effect, this fixes the random walks for R_t and $\alpha_t^{(2)}$ at their most recent value. All forecasts made here are conditional on this assumption.

Figure 6.9 forecasts both R_t and cumulative infections. The leftmost panel shows the *unadjusted* reproduction number, which we denote by R_t^u. This is the value of R_t if there were an infinitely large susceptible population, and is the quantity modeled by `epirt()`. The *adjusted* R_t is then taken to be

$$R_t = \frac{S_{t-1}}{P} R_t^{\text{u}}.$$

Figure 6.9 shows that R_t^u remains pathwise constant over the projected period. R_t on the other hand, falls smoothly as the susceptible population is diminished. Cumulative infections also taper out as the reduction in S_t causes R_t to fall below 1. The population adjustment serves to *constrain* long-term forecasts for the total size of the population. Without this adjustment, infections could continue to rise exponentially over time.

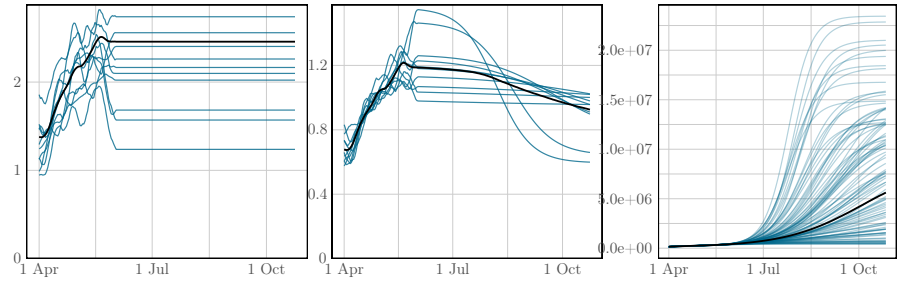


Fig. 6.9 Long-term forecasts for R_t and infections, continuing until the 27$^{\text{th}}$ October 2021. The median is plotted in black, while sample paths are in blue. Left and center show unadjusted and adjusted R_t respectively. **Right**: projected infections over time. These plots are produced with **epidemia's** spaghetti plot functions.

The top panel of Figure 6.10 shows the IAR over time. The IAR exhibits strong weekday effects, and appears to increase during the first half of April. The pattern repeats after the 30$^{\text{th}}$ May because we have stopped the random walk. The bottom two plots demonstrate that the model can fit the observed data well. These plots have been generated with `plot_linpred` and `plot_obs`.

The primary purpose of this example is to demonstrate advanced modeling in **epidemia**. The model itself has many limitations, These include the quite naive specification of the prior on S_t, and not accounting for vaccinations during the modeled period. In addition, the `i2o` argument used in the observational models is not motivated by data from previous studies.

6.6 Conclusions

This chapter has presented **epidemia**, an R package for modeling the temporal dynamics of infectious diseases. This is done in a Bayesian framework, and is regression-oriented, allowing the user flexibility over model specification. **epidemia** can be used for a number of inferential tasks. In particular, the examples of Section 6.5 have demonstrated how to estimate time-varying reproduction numbers, and to infer the effect of interventions on disease transmission. We have not been able to demonstrate all features of **epidemia**. Most notably, we have not given examples of applying population adjustments, using multiple observation vectors, and of starting modeling at some point after the beginning of an epidemic.

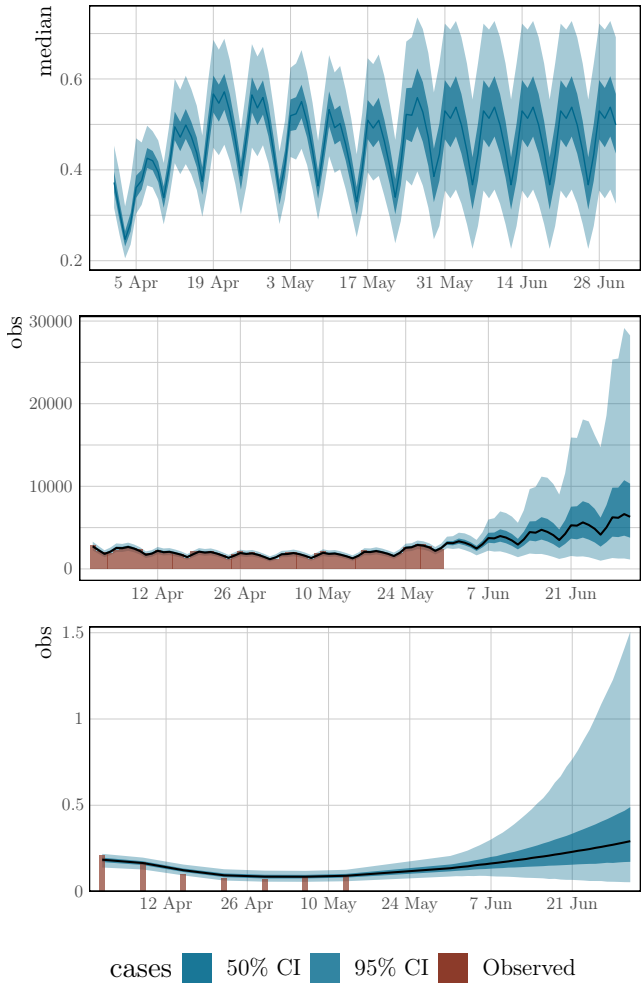


Fig. 6.10 Historical estimates for IAR, daily cases, and positivity rates. Also included are 1 month forecasts for each quantity. Top panel displays IAR over time. For illustration, the daily effects have been excluded from this. Center shows daily case counts, and bottom displays positivity rate.

7 Conclusions and Future Research

7.1 Contributions

The main contributions of this thesis are summarized as follows.

Chapter 2 developed new methods for sampling unweighted graphs conditional on node degrees, and integer-weighted graphs conditional on node strengths. These Markov chains are unique in their ability to traverse the state space while fixing an arbitrary set of edges/non-edges; a property that was illustrated using a real ecological network. Alternate methods that rely on computing a Markov basis, or those employing sequential importance sampling, were computationally infeasible for this example. Another innovation in this chapter was to develop a framework for carefully selecting local moves, termed state-dependent kernel selection, which was employed for efficient sampling in both sparse and dense graphs.

Chapter 3 continued the theme of conditional graph sampling. A null model was suggested for testing the significance of patterns in graphs with real-valued edge weights. The model fixes node strengths and approximately fixes node degrees to within ± 1 of the values of an observed network. A new MCMC sampler (motivated by those in Chapter 2) was developed to sample from the null model. It was shown empirically that the sampler is capable of rapidly randomizing large and sparse networks. A power study was performed to compare the performance of the null model to alternatives. The model compared favorably and appears capable of detecting subtle patterns, while also effectively controlling for nodal heterogeneity.

Chapter 4 considered the problem of empirically testing whether a given MCMC sampler has a desired invariant distribution. This topic was motivated by the previous two chapters, where the graph samplers rely on derivations and proofs to justify invariance. The main innovation was to propose new tests that bound the probability of falsely rejecting a valid sampler. By embedding the tests into a sequential framework, we were able to improve power, while keeping this false rejection probability arbitrarily small. An extensive simulation study was performed, which validated the ability of the tests to achieve the correct nominal level, and generally showed favorable performance of the

methods. The tests were applied to an erroneous RJ-MCMC sampler proposed for signal decomposition problems.

Part II of the thesis moved away from the development of new MCMC samplers, and towards building a general modeling framework that leverages MCMC, and in particular Stan's (Stan Development Team, 2018) implementation of the No-U-Turn sampler (Hoffman and Gelman, 2014), as the underlying inference engine. More concretely, this part of the thesis introduced a Bayesian framework for modeling the dynamics of infectious diseases.

Chapter 5 described the general model and its key statistical and epidemiological features. The chapter motivated the model from continuous-time counting processes. Various extensions to the basic model were discussed. This included explicitly modeling latent infections as unknown parameters, rather than treating them as a deterministic function of previous infections and the current reproduction number. Another enhancement considered was accounting for depletion of the susceptible population. Key limitations of the proposed methodology were discussed, in particular that of dealing with confounded variables within the framework An analysis was conducted to explore this issue, using data on interventions put in place during the first wave of SARS-COV-2 in Europe.

Chapter 6 introduced **epidemia**, an R package implementing the modeling framework of Chapter 5. We showed how the package can be used to flexibly specify and fit Bayesian, regression-oriented models for infectious diseases. The implemented models define a likelihood for all observed data while also explicitly modeling transmission dynamics: an approach often termed as semi-mechanistic. Multiple regions can be modeled simultaneously with multilevel models. Key epidemiological quantities, including reproduction numbers and latent infections, may be estimated within the framework. Our examples showed how the models can be used to evaluate the determinants of changes in transmission rates, including the effects of control measures. Epidemic dynamics may be simulated either from a fitted model or a "prior" model; allowing for prior/posterior predictive checks, experimentation, and forecasting.

7.2 Directions for Future Research

The work on graph sampling can be extended in a number of ways. Chapter 3 only considered directed graphs, however the methods could in principle be extended to undirected graphs. From initial work in this direction, it appears that the undirected equivalent of k-cycles is not sufficient for maintaining irreducibility of the sampler. This challenge would need to be overcome. Another open question is whether the sampler is capable of reaching all possible graph topologies in the general case of conditioning on degrees $\pm l$ for small $l > 0$. We were only able to prove irreducibility when conditioning on strengths. Nonetheless, the simulation study presented showed that the sampler

is capable of traversing different graph topologies, even when conditioning tightly on degrees. The graph sampler introduced in Chapter 3 could also be used for Bayesian reconstruction of financial networks (see, for example, Gandy and Veraart (2016)). Since in sparse networks, our sampler is considerably more efficient than the method used in Gandy and Veraart (2016), we expect our method to be useful in this field.

The modeling framework presented in Part II of the thesis can be extended in numerous directions. Currently, reproduction numbers can be modeled as a random walk, however additional autocorrelated processes such as ARMA processes could be considered. Importations between populations are not currently modeled. One approach would be to add additional additive terms to the renewal equation (Equation 6.3). More flexible prior distributions for seeded infections that go beyond the hierarchical model presented here could be included. Certain epidemiological quantities, such as the generation distribution, are assumed to be known. Uncertainty could be incorporated by, for example, assigning the generation distribution a Dirichlet prior. Finally, the question of efficient and robust fitting of these models is not yet fully resolved. We conjecture that cleverly selecting starting values for sampling may help prevent the chains becoming trapped in local modes. This would be an interesting avenue for future research.

References

Acemoglu, D., Ozdaglar, A., and Tahbaz-Salehi, A. (2013). Systemic Risk and Stability in Financial Networks. Technical report, National Bureau of Economic Research. Issue: 18727.

Agresti, A. (1992). A Survey of Exact Inference for Contingency Tables. *Statistical Science*, 7(1):131–153.

Agresti, A. (2013). *Categorical Data Analysis*. John Wiley & Sons.

Aicher, C., Jacobs, A. Z., and Clauset, A. (2015). Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248.

Aldecoa, R. and Marín, I. (2011). Deciphering Network Community Structure by Surprise. *PLOS ONE*, 6(9):e24195.

Allatta, J. T. (2003). Structural Analysis of Communities of Practice: An Investigation of Job title, Location, and Management Intention. In *Communities and Technologies*, pages 23–42. Springer Netherlands, Dordrecht.

Andersson, H. and Britton, T. (2000). *Stochastic Epidemic Models and Their Statistical Analysis*, volume 151. Springer New York, New York, NY.

Andrieu, C., Barat, E., and Doucet, A. (2001a). Bayesian deconvolution of noisy filtered point processes. *IEEE Transactions on Signal Processing*, 49(1):134–146.

Andrieu, C., De Freitas, N., and Doucet, A. (2001b). Robust Full Bayesian Learning for Radial Basis Networks. *Neural Computation*, 13(10):2359–2407.

Andrieu, C. and Doucet, A. (1999). Joint Bayesian model selection and estimation of noisy sinusoids via reversible jump MCMC. *IEEE Transactions on Signal Processing*, 47(10):2667–2676.

Aoki, S. and Takemura, A. (2005). Markov Chain Monte Carlo Exact Tests for Incomplete two-way Contingency Tables. *Journal of Statistical Computation and Simulation*, 75(10):787–812.

Artzy-Randrup, Y. and Stone, L. (2005). Generating uniformly distributed random networks. *Physical Review E*, 72(5):056708.

Badr, H. S., Du, H., Marshall, M., Dong, E., Squire, M. M., and Gardner, L. M. (2020). Association between mobility patterns and COVID-19 transmission in the USA: a mathematical modelling study. *The Lancet. Infectious Diseases*, 20(11):1247–1254.

Baird, D. and Ulanowicz, R. E. (1989). The Seasonal Dynamics of the Chesapeake Bay Ecosystem. *Ecological Monographs*, 59(4):329–364.

Bartoszynski, R. (1967). Branching processes and the theory of epidemics. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 4:259–269.

Bates, D., Mächler, M., Bolker, B. M., and Walker, S. C. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48. Publisher: American Statistical Association.

Bayati, M., Kim, J. H., and Saberi, A. (2010). A Sequential Algorithm for Generating Random Graphs. *Algorithmica*, 58(4):860–910.

Bellman, R. and Harris, T. (1952). On Age-Dependent Binary Branching Processes. *Annals of Mathematics*, 55(2):280–295.

Bellman, R. and Harris, T. E. (1948). On the Theory of Age-Dependent Stochastic Branching Processes. *Proceedings of the National Academy of Sciences*, 34(12):601–604.

Bender, E. A. and Canfield, E. R. (1978). The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3):296–307.

Besag, J. and Clifford, P. (1989). Generalized Monte Carlo significance tests. *Biometrika*, 76(4):633–642.

Bettencourt, L. M. A. and Ribeiro, R. M. (2008). Real Time Bayesian Estimation of the Epidemic Potential of Emerging Infectious Diseases. *PLOS ONE*, 3(5):e2185.

Bezáková, I., Sinclair, A., Štefankovič, D., and Vigoda, E. (2012). Negative Examples for Sequential Importance Sampling of Binary Contingency Tables. *Algorithmica*, 64(4):606–620.

Bhatt, S., Ferguson, N., Flaxman, S., Gandy, A., Mishra, S., and Scott, J. A. (2020). Semi-Mechanistic Bayesian Modeling of COVID-19 with Renewal Processes. *arXiv preprint arXiv:2012.00394*.

Bishop, Y. M. M. and Fienberg, S. E. (1969). Incomplete Two-Dimensional Contingency Tables. *Biometrics*, 25(1):119–128.

Blitzstein, J. and Diaconis, P. (2011). A Sequential Importance Sampling Algorithm for Generating Random Graphs with Prescribed Degrees. *Internet Mathematics*, 6(4):489–522.

Bollobás, B. (1980). A Probabilistic Proof of an Asymptotic Formula for the Number of Labelled Regular Graphs. *European Journal of Combinatorics*, 1(4):311–316.

Box, G. E. P. and Jenkins, G. M. (1962). Some Statistical Aspects of Adaptive Optimization and Control. *Journal of the Royal Statistical Society: Series B (Methodological)*, 24(2):297–331.

Carias, C., O'Hagan, J. J., Gambhir, M., Kahn, E. B., Swerdlow, D. L., and Meltzer, M. I. (2019). Forecasting the 2014 West African Ebola Outbreak. *Epidemiologic Reviews*, 41(1):34–50.

Castro, R., Coates, M., Liang, G., Nowak, R., and Yu, B. (2004). Network Tomography: Recent Developments. *Statistical Science*, 19(3):499–517.

Cauchemez, S., Valleron, A.-J., Boëlle, P.-Y., Flahault, A., and Ferguson, N. M. (2008). Estimating the impact of school closure on influenza transmission from Sentinel data. *Nature*, 452(7188):750–754.

Champredon, D., Dushoff, J., and Earn, D. J. D. (2018). Equivalence of the Erlang-Distributed SEIR Epidemic Model and the Renewal Equation. *SIAM Journal on Applied Mathematics*, 78(6):3258–3278.

Chatterjee, S., Diaconis, P., and Sly, A. (2011). Random graphs with a given degree sequence. *The Annals of Applied Probability*, 21(4):1400–1435.

Chatzilena, A., van Leeuwen, E., Ratmann, O., Baguelin, M., and Demiris, N. (2019). Contemporary statistical inference for infectious disease models using Stan. *Epidemics*, 29:100367.

Chen, Y. (2007). Conditional Inference on Tables With Structural Zeros. *Journal of Computational and Graphical Statistics*, 16(2):445–467.

Chen, Y., Diaconis, P., Holmes, S. P., and Liu, J. S. (2005). Sequential Monte Carlo Methods for Statistical Analysis of Tables. *Journal of the American Statistical Association*, 100(469):109–120.

Chung, F. and Lu, L. (2002a). The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences of the United States of America*, 99(25):15879–15882.

Chung, F. and Lu, L. (2002b). Connected Components in Random Graphs with Given Expected Degree Sequences. *Annals of Combinatorics*, 6(2):125–145.

Connor, E. F. and Simberloff, D. (1979). The Assembly of Species Communities: Chance or Competition? *Ecology*, 60(6):1132–1140.

Cook, S. R., Gelman, A., and Rubin, D. B. (2006). Validation of Software for Bayesian Models Using Posterior Quantiles. *Journal of Computational and Graphical Statistics*, 15(3):675–692.

Cori, A. (2021). EpiEstim: Estimate Time Varying Reproduction Numbers from Epidemic Curves. R package version 2.2-4.

Cori, A., Ferguson, N. M., Fraser, C., and Cauchemez, S. (2013). A New Framework and Software to Estimate Time-Varying Reproduction Numbers During Epidemics. *American Journal of Epidemiology*, 178(9):1505–1512.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to algorithms*. MIT press.

Cowles, M. K. and Carlin, B. P. (1996). Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review. *Journal of the American Statistical Association*, 91:883–904.

Cowling, B. J., Ali, S. T., Ng, T. W. Y., Tsang, T. K., Li, J. C. M., Fong, M. W., Liao, Q., Kwan, M. Y., Lee, S. L., Chiu, S. S., Wu, J. T., Wu, P., and Leung, G. M. (2020). Impact assessment of non-pharmaceutical interventions against coronavirus disease 2019 and influenza in Hong Kong: an observational study. *The Lancet Public Health*, 5(5):e279–e288.

Davy, M., Godsill, S., and Idier, J. (2006). Bayesian analysis of polyphonic western tonal music. *The Journal of the Acoustical Society of America*, 119(4):2498–2517.

Diaconis, P. and Gangolli, A. (1995). Rectangular Arrays with Fixed Margins. In Aldous, D., Diaconis, P., Spencer, J., and Steele, editors, *Discrete Probability and Algorithms*, volume 72 of *The IMA Volumes in Mathematics and its Applications*, pages 15–41. Springer New York.

Diaconis, P. and Sturmfels, B. (1998). Algebraic Algorithms for Sampling from Conditional Distributions. *Annals of Statistics*, 26(1):363–397.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Eisinger, R. D. and Chen, Y. (2017). Sampling for Conditional Inference on Contingency Tables. *Journal of Computational and Graphical Statistics*, 26(1):79–87.

Elliott, M., Golub, B., and Jackson, M. O. (2014). Financial Networks and Contagion. *American Economic Review*, 104(10):3115–3153.

Ferguson, N. M., Donnelly, C. A., and Anderson, R. M. (2001). Transmission intensity and impact of control policies on the foot and mouth epidemic in Great Britain. *Nature*, 413(6855):542–548.

Fienberg, S. E. (1969). Preliminary Graphical Analysis and Quasi-Independence for two-way Contingency Tables. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 18(2):153–168.

Flaxman, S., Mishra, S., Gandy, A., Unwin, H. J. T., Mellan, T. A., Coupland, H., Whittaker, C., Zhu, H., Berah, T., Eaton, J. W., Monod, M., Ghani, A. C., Donnelly, C. A., Riley, S., Vollmer, M. A. C., Ferguson, N. M., Okell, L. C., and Bhatt, S. (2020a). Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe. *Nature*, 584(7820):257–261.

Flaxman, S., Mishra, S., Scott, J., Ferguson, N., Gandy, A., and Bhatt, S. (2020b). Reply to: The effect of interventions on COVID-19. *Nature*, 588(7839):E29–E32.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75–174.

Fortunato, S. and Barthélemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41.

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

Fraser, C. (2007). Estimating Individual and Household Reproduction Numbers in an Emerging Epidemic. *PLOS ONE*, 2(8):e758.

Fraser, C., Donnelly, C. A., Cauchemez, S., Hanage, W. P., Kerkhove, M. D. V., Hollingsworth, T. D., Griffin, J., Baggaley, R. F., Jenkins, H. E., Lyons, E. J., Jombart, T., Hinsley, W. R., Grassly, N. C., Balloux, F., Ghani, A. C., Ferguson, N. M., Rambaut, A., Pybus, O. G., Lopez-Gatell, H., Alpuche-Aranda, C. M., Chapela, I. B., Zavala, E. P., Guevara, D. M. E., Checchi, F., Garcia, E., Hugonnet, S., Roth, C., and Collaboration, T. W. R. P. A. (2009). Pandemic Potential of a Strain of Influenza A (H1N1): Early Findings. *Science*, 324(5934):1557–1561.

Gabrielli, A., Mastrandrea, R., Caldarelli, G., and Cimini, G. (2019). Grand canonical ensemble of weighted networks. *Physical Review E*, 99(3):030301.

Gai, P. and Kapadia, S. (2010). Contagion in financial networks. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 466(2120):2401–2423.

Galeano, J., Pastor, J. M., and Iriondo, J. M. (2009). Weighted-Interaction Nestedness Estimator (WINE): A New Estimator to Calculate over Frequency Matrices. *Environmental Modelling and Software*, 24(11):1342–1346.

Gandy, A. (2009). Sequential Implementation of Monte Carlo Tests With Uniformly Bounded Resampling Risk. *Journal of the American Statistical Association*, 104(488):1504–1511.

Gandy, A. and Veraart, L. A. M. (2016). A Bayesian Methodology for Systemic Risk Assessment in Financial Networks. *Management Science*, 63(12):4428–4446.

Gandy, A. and Veraart, L. A. M. (2019). Adjustable network reconstruction with applications to CDS exposures. *Journal of Multivariate Analysis*, 172:193–209.

Gelman, A. (2017). Correction to: Validation of Software for Bayesian Models using Posterior Quantiles. *Journal of Computational and Graphical Statistics*, 26(4):940.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2015). *Bayesian Data Analysis*. Chapman and Hall/CRC, New York, 3 edition.

Gelman, A. and Hill, J. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press, Cambridge.

Gelman, A., Jakulin, A., Pittau, M. G., and Su, Y.-S. (2008). A weakly informative default prior distribution for logistic and other regression models. *The Annals of Applied Statistics*, 2(4):1360–1383.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Getz, W. M. and Lloyd-Smith, J. O. (2006). Basic methods for modeling the invasion and spread of contagious diseases. In Feng, Z., Dieckmann, U., and Levin, S. A., editors, *Disease Evolution: Models, Concepts, and Data Analyses, Proceedings of a DIMACS Workshop*, volume 71 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 87–109. DIMACS/AMS.

Geweke, J. (2004). Getting It Right. *Journal of the American Statistical Association*, 99(467):799–804.

Gibbons, C. L., Mangen, M.-J. J., Plass, D., Havelaar, A. H., Brooke, R. J., Kramarz, P., Peterson, K. L., Stuurman, A. L., Cassini, A., Fèvre, E. M., and Kretzschmar, M. E. (2014). Measuring underreporting and under-ascertainment in infectious disease datasets: a comparison of methods. *BMC Public Health*, 14(1).

Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214.

Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826.

Glasserman, P. and Young, H. P. (2016). Contagion in Financial Networks. *Journal of Economic Literature*, 54(3):779–831.

Goodman, L. A. (1963). Statistical Methods for the Preliminary Analysis of Transaction Flows. *Econometrica*, 31(1):197–208.

Goodman, L. A. (1968). The Analysis of Cross-Classified Data: Independence, Quasi-Independence, and Interactions in Contingency Tables with or without Missing Entries. *Journal of the American Statistical Association*, 63(324):1091–1131. Publisher: Taylor & Francis Group.

Goodrich, B., Gabry, J., Ali, I., and Brilleman, S. (2020). rstanarm: Bayesian applied regression modeling via {Stan}. R package version 2.21.1.

Gotelli, N. and Entsminger, G. (2001). Swap and Fill Algorithms in Null Model Analysis: Rethinking the Knight's Tour. *Oecologia*, 129(2):281–291.

Grinsztajn, L., Semenova, E., Margossian, C. C., and Riou, J. (2020). Bayesian workflow for disease transmission modeling in Stan. *arXiv preprint arXiv:2006.02985*.

Groendyke, C. and Welch, D. (2018). epinet : An R Package to Analyze Epidemics Spread across Contact Networks. *Journal of Statistical Software*, 83(11).

Guimerà, R., Sales-Pardo, M., and Amaral, L. A. N. (2004). Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101.

Guimerà, R., Stouffer, D. B., Sales-Pardo, M., Leicht, E. A., Newman, M. E. J., and Amaral, L. A. N. (2010). Origin of Compartmentalization in Food Webs. *Ecology*, 91(10):2941–2951.

Gustafsson, J.-E. (1980). A Solution of the Conditional Estimation Problem for Long Tests in the Rasch model for Dichotomous Items. *Educational and Psychological Measurement*, 40(2):377–385.

Hakimi, S. L. (1962). On Realizability of a Set of Integers as Degrees of the Vertices of a Linear Graph. I. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506.

Haldane, A. G. and May, R. M. (2011). Systemic risk in banking ecosystems. *Nature*, 469:351–355.

Hastie, T., Tibshirani, R., and Wainwright, M. (2019). *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC.

Haug, N., Geyrhofer, L., Londei, A., Dervic, E., Desvars-Larrive, A., Loreto, V., Pinior, B., Thurner, S., and Klimek, P. (2020). Ranking the effectiveness of worldwide COVID-19 government interventions. *Nature Human Behaviour*, 4(12):1303–1312.

Hauser, A., Counotte, M. J., Margossian, C. C., Konstantinoudis, G., Low, N., Althaus, C. L., and Riou, J. (2020). Estimation of SARS-CoV-2 mortality during the early stages of an epidemic: A modeling study in Hubei, China, and six regions in Europe. *PLOS Medicine*, 17(7):1–17.

He, Z., Liang, H., Chen, Z., Zhao, C., and Liu, Y. (2020). Computing exact P-values for community detection. *Data Mining and Knowledge Discovery*, 34(3):833–869.

Held, L., Höhle, M., and Hofmann, M. (2005). A statistical framework for the analysis of multivariate infectious disease surveillance counts. *Statistical Modelling*, 5(3):187–199.

Held, L. and Paul, M. (2012). Modeling seasonality in space-time infectious disease surveillance data. *Biometrical Journal*, 54(6):824–43.

Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.

Holland, P. W. and Leinhardt, S. (1981). An Exponential Family of Probability Distributions for Directed Graphs. *Journal of the American Statistical Association*, 76(373):33–50.

Hong, M., Bugallo, M. F., and Djuric, P. M. (2010). Joint Model Selection and Parameter Estimation by Population Monte Carlo Simulation. *IEEE Journal of Selected Topics in Signal Processing*, 4(3):526–539.

Hox, J. J., Moerbeek, M., and de Schoot, R. (2010). *Multilevel analysis: Techniques and applications*. Routledge.

Höhle, M. and Feldmann, U. (2007). RLadyBug—An R package for stochastic epidemic models. *Computational Statistics & Data Analysis*, 52(2):680–686.

Jenness, S. M., Goodreau, S. M., and Morris, M. (2018). EpiModel: An R Package for Mathematical Modeling of Infectious Disease over Networks. *Journal of Statistical Software, Articles*, 84(8):1–47.

Kahle, D., Garcia-Puente, L., and Yoshida, R. (2017). latte: LattE and 4ti2 in R. R package version 0.2.0.

Karrer, B. and Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107.

Kelly, H. A., Mercer, G. N., Fielding, J. E., Dowse, G. K., Glass, K., Carcione, D., Grant, K. A., Effler, P. V., and Lester, R. A. (2010). Pandemic (H1N1) 2009 Influenza Community Transmission Was Established in One Australian State When the Virus Was First Identified in North America. *PLOS ONE*, 5(6):e11341.

Kermack, William Ogilvy and McKendrick, A. G. (1932). Contributions to the mathematical theory of epidemics. II. —The problem of endemicity. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 138(834).

Kermack, William Ogilvy and McKendrick, A. G. (1933). Contributions to the mathematical theory of epidemics. III.—Further studies of the problem of endemicity. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 141(843).

Kermack, W. O., McKendrick, A. G., and Walker, G. T. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721.

Krause, A. E., Frank, K. A., Mason, D. M., Ulanowicz, R. E., and Taylor, W. W. (2003). Compartments Revealed in Food-Web Structure. *Nature*, 426(6964):282–285.

Kreft, I. and de Leeuw, J. (2011). *Introducing Multilevel Modeling*. SAGE Publications Ltd.

Kucukelbir, A., Ranganath, R., Gelman, A., and Blei, D. M. (2015). Automatic Variational Inference in Stan. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, volume 1 of *NIPS'15*, pages 568–576, Cambridge, MA, USA. MIT Press.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1):430–474.

Kumpula, J. M., Saramäki, J., Kaski, K., and Kertész, J. (2007). Limited resolution in complex network community detection with Potts model approach. *European Physical Journal B*.

Lancichinetti, A. and Fortunato, S. (2009). Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118.

Lancichinetti, A., Fortunato, S., and Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110.

Larocque, J.-R. and Reilly, J. (2002). Reversible jump MCMC for joint detection and estimation of sources in colored noise. *IEEE Transactions on Signal Processing*, 50(2):231–240.

Larocque, J.-R., Reilly, J., and Ng, W. (2002). Particle filters for tracking an unknown number of sources. *IEEE Transactions on Signal Processing*, 50(12):2926–2937.

Lehmann, E. L. and Romano, J. P. (2006). *Testing statistical hypotheses*. Springer Science & Business Media.

Li, R., Pei, S., Chen, B., Song, Y., Zhang, T., Yang, W., and Shaman, J. (2020). Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV-2). *Science*, 368(6490):489–493.

Liboschik, T., Fokianos, K., and Fried, R. (2017). tscount: An R Package for Analysis of Count Time Series Following Generalized Linear Models. *Journal of Statistical Software*, 82(5):1–51.

Maslov, S., Sneppen, K., and Zaliznyak, A. (2004). Detection of topological patterns in complex networks: correlation profile of the internet. *Physica A: Statistical Mechanics and its Applications*, 333:529–540.

Mastrandrea, R., Squartini, T., Fagiolo, G., and Garlaschelli, D. (2014a). Enhanced reconstruction of weighted networks from strengths and degrees. *New Journal of Physics*, 16(4):43022.

Mastrandrea, R., Squartini, T., Fagiolo, G., and Garlaschelli, D. (2014b). Enhanced reconstruction of weighted networks from strengths and degrees. *New Journal of Physics*, 16(4):043022.

May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459–467.

McDonald, J. W., Smith, P. W. F., and Forster, J. J. (2007). Markov chain Monte Carlo Exact Inference for Social Networks. *Social Networks*, 29(1):127–136.

Mellan, T. A., Hoeltgebaum, H. H., Mishra, S., Whittaker, C., Schnekenberg, R. P., Gandy, A., Unwin, H. J. T., Vollmer, M. A., Coupland, H., Hawryluk, I., Faria, N. R., Vesga, J., Zhu, H., Hutchinson, M., Ratmann, O., Monod, M., Ainslie, K. E., Baguelin, M., Bhatia, S., Boonyasiri, A., Brazeau, N., Charles, G., Cucunuba, Z., Cuomo-Dannenburg, G., Dighe, A., Eaton, J., van Elsland, S. L., Gaythorpe, K. A., Green, W., Knock, E., Laydon, D., Lees, J. A., Mousa, A., Nedjati-Gilani, G., Nouvellet, P., Parag, K. V., Thompson, H. A., Verity, R., Walters, C. E., Wang, H., Wang, Y.,

Watson, O. J., Whittles, L., Xi, X., Dorigatti, I., Walker, P., Ghani, A. C., Riley, S., Ferguson, N. M., Donnelly, C. A., Flaxman, S., and Bhatt, S. (2020). Subnational analysis of the COVID-19 epidemic in Brazil. *medRxiv*.

Merl, D., Johnson, L. R., Gramacy, R. B., and Mangel, M. (2010). **amei** : An *R* Package for the Adaptive Management of Epidemiological Interventions. *Journal of Statistical Software*, 36(6).

Meyer, S., Held, L., and Höhle, M. (2017). Spatio-Temporal Analysis of Epidemic Phenomena Using the R Package surveillance. *Journal of Statistical Software, Articles*, 77(11):1–55.

Meyn, S., Tweedie, R. L., and Glynn, P. W. (2009). *Markov Chains and Stochastic Stability*. Cambridge University Press.

Miller, A. C., Foti, N. J., Lewnard, J. A., Jewell, N. P., Guestrin, C., and Fox, E. B. (2020). Mobility trends provide a leading indicator of changes in SARS-CoV-2 transmission. *medRxiv*.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827.

Mishra, S., Berah, T., Mellan, T. A., Unwin, H. J. T., Vollmer, M. A., Parag, K. V., Gandy, A., Flaxman, S., and Bhatt, S. (2020a). On the derivation of the renewal equation from an age-dependent branching process: an epidemic modelling perspective. *arXiv preprint arXiv:2006.16487*.

Mishra, S., Scott, J., Zhu, H., Ferguson, N. M., Bhatt, S., Flaxman, S., and Gandy, A. (2020b). A COVID-19 Model for Local Authorities of the United Kingdom. *medRxiv*. Publisher: Cold Spring Harbor Laboratory Press _eprint: https://www.medrxiv.org/content/early/2020/11/27/2020.11.24.20236661.full.pdf.

Miyauchi, A. and Kawase, Y. (2016). Z-Score-Based Modularity for Community Detection in Networks. *PLOS ONE*, 11(1):1–17.

Mizruchi, M. S. (1983). Who Controls Whom? An Examination of the Relation between Management and Boards of Directors in Large American Corporations. *The Academy of Management Review*, 8(3):426–435.

Myers, M. F., Rogers, D. J., Cox, J., Flahault, A., and Hay, S. I. (2000). Forecasting disease risk for increased epidemic preparedness in public health. *Advances in Parasitology*, 47:309–330.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133.

Newman, M. E. J., Strogatz, S. H., and Watts, D. J. (2001). Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118.

Ng, W., Reilly, J., Kirubarajan, T., and Larocque, J.-R. (2005). Wideband array signal processing using MCMC methods. *IEEE Transactions on Signal Processing*, 53(2):411–426.

Nouvellet, P., Cori, A., Garske, T., Blake, I. M., Dorigatti, I., Hinsley, W., Jombart, T., Mills, H. L., Nedjati-Gilani, G., Van Kerkhove, M. D., Fraser, C., Donnelly, C. A., Ferguson, N. M., and Riley, S. (2018). A simple approach to measure transmissibility and forecast incidence. *Epidemics*, 22:29–35.

Nowicki, K. and Snijders, T. A. B. (2001). Estimation and Prediction for Stochastic Blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087.

NYS Press Office (2020). Amid Ongoing COVID-19 Pandemic, Governor Cuomo Announces State is Bringing in International Experts to Help Advise the State's Reopening Plan.

Obadia, T., Haneef, R., and Boëlle, P.-Y. (2012). The R0 package: a toolbox to estimate reproduction numbers for epidemic outbreaks. *BMC Medical Informatics and Decision Making*, 12(1).

Olney, A. M., Smith, J., Sen, S., Thomas, F., and Unwin, H. J. T. (2021). Estimating the Effect of Social Distancing Interventions on COVID-19 in the United States. *American Journal of Epidemiology*, 190(8):1504–1509.

ONS (2021a). Coronavirus (COVID-19) antibody and vaccination data for the UK - Office for National Statistics.

ONS (2021b). Coronavirus (COVID-19) Infection Survey: England - Office for National Statistics.

ONS (2021c). Population estimates for the UK, England and Wales, Scotland and Northern Ireland - Office for National Statistics.

Opsahl, T. (2013). Triadic Closure in Two-Mode Networks: Redefining the Global and Local Clustering Coefficients. *Social Networks*, 35(2):159–167.

Palowitch, J., Bhamidi, S., and Nobel, A. B. (2018). Significance-based community detection in weighted networks. *Journal of Machine Learning Research*, 18(188):1–48.

Park, J. and Newman, M. E. J. (2004). Statistical mechanics of networks. *Physical Review E*, 70(6):66117.

Paul, M. and Held, L. (2011). Predictive assessment of a non-linear random effects model for multivariate time series of infectious disease counts. *Statistics in Medicine*, 30(10):1118–1136.

Paul, M., Held, L., and Toschke, A. M. (2008). Multivariate modelling of infectious disease surveillance data. *Statistics in Medicine*, 27(29):6250–6267.

Pearl, J. (2009). Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146.

Pearl, J. (2012). The causal mediation formula–a guide to the assessment of pathways and mechanisms. *Prevention Science: The Official Journal of the Society for Prevention Research*, 13(4):426–436.

Piironen, J. and Vehtari, A. (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2):5018–5051.

Pimm, S. L. and Lawton, J. H. (1980). Are Food Webs Divided into Compartments? *The Journal of Animal Ecology*, 49(3):879–898.

Pinsky, M. and Karlin, S. (2010). *An introduction to stochastic modeling*. Academic press.

Pons, P. and Latapy, M. (2005). Computing Communities in Large Networks Using Random Walks. In Yolum, p., Güngör, T., Gürgen, F., and Özturan, C., editors, *Computer and Information Sciences - ISCIS 2005*, pages 284–293, Berlin, Heidelberg. Springer Berlin Heidelberg.

Public Health England (2020). Official UK Coronavirus Dashboard.

R Core Team (2021). R: A Language and Environment for Statistical Computing.

Rao, A. R., Jana, R., and Bandyopadhyay, S. (1996). A Markov Chain Monte Carlo Method for Generating Random (0, 1) Matrices with Given Marginals. *Sankhya: The Indian Journal of Statistics, Series A (1961-2002)*, 58(2):225–242.

Rapallo, F. (2006). Markov Bases and Structural Zeros. *Journal of Symbolic Computation*, 41(2):164–172.

Rasch, G. (1960). *Probabilistic Models for some Intelligence and Achievement Tests*. University of Chicago Press.

Reichardt, J. and Bornholdt, S. (2006). When are networks truly modular? *Physica D: Nonlinear Phenomena*, 224(1):20–26.

Rezende, E. L., Albert, E. M., Fortuna, M. A., and Bascompte, J. (2009). Compartments in a Marine Food Web Associated with Phylogeny, Body Mass, and Habitat Structure. *Ecology Letters*, 12(8):779–788.

Riley, S., Fraser, C., Donnelly, C. A., Ghani, A. C., Abu-Raddad, L. J., Hedley, A. J., Leung, G. M., Ho, L.-M., Lam, T.-H., Thach, T. Q., Chau, P., Chan, K.-P., Lo, S.-V., Leung, P.-Y., Tsang, T., Ho, W., Lee, K.-H., Lau, E. M. C., Ferguson, N. M., and Anderson, R. M. (2003). Transmission Dynamics of the Etiological Agent of SARS in Hong Kong: Impact of Public Health Interventions. *Science*, 300(5627):1961–1966.

Rizoiu, M.-A., Xie, L., Sanner, S., Cebrian, M., Yu, H., and Van Hentenryck, P. (2017). Expecting to be HIP: Hawkes Intensity Processes for Social Media Popularity. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 735–744, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Roberts, G. O. and Stramer, O. (2002). Langevin Diffusions and Metropolis-Hastings Algorithms. *Methodology And Computing In Applied Probability*, 4(4):337–357.

Roberts, J. M. (2000). Simple Methods for Simulating Sociomatrices with Given Marginal Totals. *Social Networks*, 22(3):273–283.

Roodaki, A., Bect, J., and Fleury, G. (2013). Comments on "Joint Bayesian Model Selection and Estimation of Noisy Sinusoids Via Reversible Jump MCMC". *IEEE Transactions on Signal Processing*, 61(14):3653–3655.

Roosa, K. and Chowell, G. (2019). Assessing parameter identifiability in compartmental dynamic models using a computational approach: application to infectious disease transmission models. *Theoretical Biology and Medical Modelling*, 16(1).

Rubtsov, D. V. and Griffin, J. L. (2007). Time-domain Bayesian detection and estimation of noisy damped sinusoidal signals applied to NMR spectroscopy. *Journal of Magnetic Resonance*, 188(2):367–379.

Runeson, P. (2006). A survey of unit testing practices. *IEEE software*, 23(4):22–29.

Ryser, H. J. (1963). *Combinatorial Mathematics*. Mathematical Association of America.

SARS Expert Committee (2003). Chronology of the SARS epidemic in Hong Kong. Technical report, SARS Expert Committee of HKSAR, Hong Kong.

Schmidt, M. N. and Mørup, M. (2010). Infinite non-negative matrix factorization. In *18th European Signal Processing Conference*, pages 905–909.

Serrano, M. A. and Boguñá, M. (2005). Weighted Configuration Model. *AIP Conference Proceedings*, 776(1):101–107.

Serrano, M. A., Boguñá, M., and Pastor-Satorras, R. (2006). Correlations in weighted networks. *Physical Review E*, 74(5):055101.

Snijders, T. A. B. (1991). Enumeration and Simulation Methods for 0-1 Matrices with Given Marginals. *Psychometrika*, 56(3):397–417.

Sokal, A. (1997). Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms. In DeWitt-Morette, C., Cartier, P., and Folacci, A., editors, *Functional Integration: Basics and Applications*, NATO ASI Series, pages 131–192. Springer US, Boston, MA.

Squartini, T., Caldarelli, G., Cimini, G., Gabrielli, A., and Garlaschelli, D. (2018). Reconstruction methods for networks: The case of economic and financial systems. *Physics Reports*, 757:1–47.

Squartini, T., Cimini, G., Gabrielli, A., and Garlaschelli, D. (2017). Network reconstruction via density sampling. *Applied Network Science*, 2(1):3.

Squartini, T., Fagiolo, G., and Garlaschelli, D. (2011). Randomizing world trade. I. A binary network analysis. *Physical Review E*, 84(4):046117. arXiv: 1103.1243.

Squartini, T., Picciolo, F., Ruzzenenti, F., and Garlaschelli, D. (2013). Reciprocity of weighted networks. *Scientific Reports*, 3(1):2729.

Stan Development Team (2018). The Stan Core Library.

Stan Development Team (2020). RStan: the R interface to Stan.

Staum, J., Feng, M., and Liu, M. (2016). Systemic risk components in a network model of contagion. *IIE Transactions*, 48(6):501–510.

Stone, L. and Roberts, A. (1990). The checkerboard score and species distributions. *Oecologia*, 85(1):74–79.

Stouffer, D. B., Camacho, J., Jiang, W., and Nunes Amaral, L. A. (2007). Evidence for the existence of a robust pattern of prey selection in food webs. *Proceedings of the Royal Society B: Biological Sciences*, 274(1621):1931–1940.

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., and Gelman, A. (2020). Validating Bayesian Inference Algorithms with Simulation-Based Calibration. *arXiv:1804.06788 [stat]*.

Taylor, J. and Tibshirani, R. J. (2015). Statistical learning and selective inference. *Proceedings of the National Academy of Sciences*, 112(25):7629–7634.

Tebaldi, C. and West, M. (1998). Bayesian Inference on Network Traffic Using Link Count Data. *Journal of the American Statistical Association*, 93(442):557–573.

The Scottish Government (2020). Coronavirus (COVID-19): modelling the epidemic.

Tokuda, T., Goodrich, B., Van Mechelen, I., Gelman, A., and Tuerlinckx, F. (2011). Visualizing distributions of covariance matrices. *Columbia Univ., New York, USA, Tech. Rep.*

Traag, V. A., Krings, G., and Van Dooren, P. (2013). Significant Scales in Community Structure. *Scientific Reports*, 3(1):2930.

Ulrich, W. and Gotelli, N. J. (2007). Null Model Analysis of Species Nestedness Patterns. *Ecology*, 88(7):1824–1831.

Unwin, H. J. T., Mishra, S., Bradley, V. C., Gandy, A., Mellan, T. A., Coupland, H., Ish-Horowicz, J., Vollmer, M. A. C., Whittaker, C., Filippi, S. L., Xi, X., Monod, M., Ratmann, O., Hutchinson, M., Valka, F., Zhu, H., Hawryluk, I., Milton, P., Ainslie, K. E. C., Baguelin, M., Boonyasiri, A., Brazeau, N. F., Cattarino, L., Cucunuba, Z., Cuomo-Dannenburg, G., Dorigatti, I., Eales, O. D., Eaton, J. W., van Elsland, S. L., FitzJohn, R. G., Gaythorpe, K. A. M., Green, W., Hinsley, W., Jeffrey, B., Knock, E., Laydon, D. J., Lees, J., Nedjati-Gilani, G., Nouvellet, P., Okell, L., Parag, K. V., Siveroni, I., Thompson, H. A., Walker, P., Walters, C. E., Watson, O. J., Whittles, L. K., Ghani, A. C., Ferguson, N. M., Riley, S., Donnelly, C. A., Bhatt, S., and Flaxman, S. (2020). State-level tracking of COVID-19 in the United States. *Nature Communications*, 11(1):6189.

van Doremalen, N., Bushmaker, T., Morris, D. H., Holbrook, M. G., Gamble, A., Williamson, B. N., Tamin, A., Harcourt, J. L., Thornburg, N. J., Gerber, S. I., Lloyd-Smith, J. O., de Wit, E., and Munster, V. J. (2020). Aerosol and Surface Stability of SARS-CoV-2 as Compared with SARS-CoV-1. *New England Journal of Medicine*, 382(16).

Vasileios, S. (2015). acp: Autoregressive Conditional Poisson. R package version 2.1.

Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5):1413–1432.

Verhelst, N. D. (2008). An Efficient MCMC Algorithm to Sample Binary Matrices with Fixed Marginals. *Psychometrika*, 73(4):705–728.

Vollmer, M. A., Mishra, S., T Unwin, H. J., Gandy, A., Mellan, T. A., Bradley, V., Zhu, H., Coupland, H., Hawryluk, I., Hutchinson, M., Ratmann, O., Monod, M., Walker, P., Whittaker, C., Cattarino, L., Ciavarella, C., Cilloni, L., Ainslie, K., Baguelin, M., Bhatia, S., Boonyasiri, A., Brazeau, N., Charles, G., Cooper, L. V., Cucunuba, Z., Cuomo-Dannenburg, G., Dighe, A., Djaafara, B., Eaton, J., van Elsland, S. L., FitzJohn, R., Fraser, K., Gaythorpe, K., Green, W., Hayes, S., Imai, N., Jeffrey, B., Knock, E., Laydon, D., Lees, J., Mangal, T., Mousa, A., Nedjati-Gilani, G., Nouvellet, P., Olivera, D., Parag, K. V., Pickles, M., Thompson, H. A., Verity, R., Walters, C., Wang, H., Wang, Y., Watson, O. J., Whittles, L., Xi, X., Ghani, A., Riley, S. M., Okell, L., Donnelly, C. A., Ferguson, N. M., Dorigatti, I., Flaxman, S., and Bhatt, S. (2020). Report 20: Using mobility to estimate the transmission intensity of COVID-19 in Italy: A subnational analysis with future scenarios. Publication Title: medRxiv.

Wallinga, J. and Teunis, P. (2004). Different Epidemic Curves for Severe Acute Respiratory Syndrome Reveal Similar Impacts of Control Measures. *American Journal of Epidemiology*, 160(6):509–516.

Wasserman, S. and Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press.

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

WHO (2003). SARS: chronology of a serial killer. Technical report, World Health Organization.

Wickham, H. (2011). testthat: Get Started with Testing. *The R Journal*, 3:5–10.

Wikipedia contributors (2019). List of unit testing frameworks — Wikipedia, The Free Encyclopedia.

Williams, R. J. and Martinez, N. D. (2004). Limits to Trophic Levels and Omnivory in Complex Food Webs: Theory and Data. *The American Naturalist*, 163(3):458–468.

Wong, F. and Collins, J. J. (2020). Evidence that coronavirus superspreading is fat-tailed. *Proceedings of the National Academy of Sciences*, 117(47):29416–29418.

Zhang, J. and Chen, Y. (2013). Sampling for Conditional Inference on Network Data. *Journal of the American Statistical Association*, 108(504):1295–1307.

A

Appendix to Chapter 2

A.1 Proofs

Proof of Lemma 2.2.2. Without loss of generality (and for notational simplicity) assume π and w are dominated by one-dimensional Lebesgue measure. We show

$$\int_A \pi(x)Q(x,B)\,\mathrm{d}x = \int_B \pi(x)Q(x,A)\,\mathrm{d}x \quad \text{for all } A,B \in \mathcal{B}.$$

Fix any A and B in $\mathcal{B}$, and define the densities $\{f_x\}$ as in Definition 2.2.1. Then

$$\begin{aligned}\int_A \pi(x)Q(x,B)\,\mathrm{d}x &= \int_A \int_{\mathcal{Z}} \int_B \pi(x)K_z(x,y)f_x(z)\,\mathrm{d}y\,\mathrm{d}z\,\mathrm{d}x \\ &= \int_A \int_{\mathcal{Z}} \int_B \pi(y)K_z(y,x)f_y(z)\,\mathrm{d}y\,\mathrm{d}z\,\mathrm{d}x = \int_B \pi(x)Q(x,A)\,\mathrm{d}x\end{aligned}$$

as required. In the first step we have expressed the integral using densities. In the second, we use π-reversibility of each K_z, and the fact that for each $z \in \mathcal{Z}$ and $x \in \mathcal{X}$, $f_x(z) = f_y(z)$ for $K_z(x,\cdot)$- a.e. $y \in B$. A simple change of variables then yields the result. □

Proof of Proposition 2.3.1. Fix G in $\mathcal{G}$. Define the second-order Markov chain $(Y_n)_{n\geq 0}$, where $Y_n := (a_{n-1}, a_n, G_n)$ and G_n is defined as follows. Let $G_0 = G$, otherwise if n is odd, let G_n be the graph obtained after $a_n a_{n-1}$ is removed from $E(G_{n-1})$. If n is even, G_n is the graph obtained after $a_{n-1}a_n$ is added to $E(G_{n-1})$. Define $\mathcal{Y}$ as the set of points reachable from $(*, a_0, G)$ for some a_0 in $\{v : N_G(v) \neq \emptyset\}$. Let $D := (\mathcal{Y}, E)$ be the digraph underlying this chain and define A as the subset of points (v, u, G') in $\mathcal{Y}$ for which $G' \in \mathcal{G}$. Let $T := \inf\{n \geq 1 \colon Y_n \in A\}$ be the first passage time of A. Proposition 2.3.1 is equivalent to showing $\mathbb{E}(T) < \infty$. The following holds true, and will be shown at the end of this proof.

$$\text{From any } (v,u,G') \in \mathcal{Y}, \text{ there exists a simple path to } A. \tag{A.1}$$

We can bound the probability of traversing each edge in D from below by some constant $p > 0$. Let N denote the size of $\mathcal{Y}$. Suppose the chain is at some state $y \notin A$. By (A.1), this implies the probability of hitting A within the next N steps is bounded from below by p^N. Hence,

$$\mathbb{E}(T) = \sum_{n=1}^{\infty} nP[T = n] \leq N \sum_{k=1}^{\infty} kP[k-1 < T/N \leq k] \leq N \sum_{k=1}^{\infty} kp^N(1-p^N)^{k-1} = Np^{-N}.$$

It remains to show (A.1). Note that for any $(v_1, u_1, G_1)(v_2, u_2, G_2) \in E(D)$:

$$(v^*, u_2, G_2)(u_2, u_1, G_1) \in E(D) \text{ if and only if } (v^*, u_2, G_2) \in \mathcal{Y} \text{ and } v^* \neq v_2. \tag{A.2}$$

By definition, there exists a point $y_0 := (*, u_0, G)$ and a walk $y_0 ... y_k$ in D such that $y_k = (v, u, G')$. Given that $V(D)$ is finite, continuing an arbitrary walk along D from y_k implies we must eventually either return to A, or visit a graph already seen along the walk. Denote the vertex visited at the lth step of this walk by $y_l = (u_{l-1}, u_l, G_l)$. If we revisit A we are done, otherwise define

$$n := \inf\{l > k : G_l = G_m \text{ for some } m < l\}.$$

The condition $G_n = G_m$ implies that $u_n = u_m$. Additionally $u_{n-1} \neq u_{m-1}$, otherwise this would imply $G_{n-1} = G_{m-1}$, which contradicts the definition of n. By (A.2), $(u_{m-2}, u_{m-1}, G_{m-1})(u_{m-1}, u_m, G_m) \in E(D)$ implies that $(u_{n-1}, u_n, G_n)(u_m, a_{m-1}, G_{m-1}) \in E(D)$. Thus we can traverse to (u_m, u_{m-1}, G_{m-1}). Iteratively applying (A.2) (which we can do as $u_{l+1} \neq u_{l-1}$ for all $l \geq 0$) implies we can reach a state with graph $G = G_0$, which must be in A, completing the proof of (A.1). □

Proof of Lemma 2.3.2. We first show that (K, w) is a symmetric decomposition. Fix any $G \in \mathcal{G}$ and any $z \in \mathcal{Z}$, and let a be a representative of z. Let p refer to the statement '$w_G(z) = w_{G^*}(z)$ for all G^* for which $K_z(G, G^*) > 0$'. It suffices to show that p is true.

Consider a Markov chain with kernel K_z and current state G. Suppose the chain remains unchanged after one iteration of Algorithm 1. Then p is true trivially. Without loss of generality, suppose the swaps corresponding to a are viable, and the chain moves to some $G^* \in \mathcal{G}$. Remark 1 implies swaps corresponding to a^r are not viable. Since the swaps corresponding to a sampled vertex sequence must be viable, $w_G(z)$ is the probability of sampling a given the chain is at G.

At G^*, the swaps corresponding to a^r are viable. By an analogous argument, it follows that $w_{G^*}(z)$ is the probability of sampling a^r given the chain is at G^*. One can deduce from Algorithm 1 that the probability of sampling w given the chain is at G is equal to the probability of sampling a^r given the chain is at G^*. This holds because the degree sequence is the same for either state.

We now show that each $K_z \in K$ is reversible with respect to the uniform distribution. This is implied by detailed balance. Specifically, for each $K_z \in K$ we show

$$K_z(G, G^*) = K_z(G^*, G) \quad \text{for all } G, G^* \in \mathcal{G}.$$

Fix any G and G^*. $K_z(G, G^*) = 1$ if and only if $K_z(G^*, G) = 1$, because applying two iterations of a Markov chain with kernel K_z from some current state G', returns G'. The result follows by additionally observing that $K_z(G, G^*)$ can only be zero or one. □

Proof of Proposition 2.3.3. Lemma 2.3.2 and connectedness of the chain suffice. Fix any $G, G' \in \mathcal{G}$, and suppose the current state of the chain is G. Form a digraph H as follows. For each vertex pair uv, if $uv \in E(G)$ and $uv \notin E(G')$, add a *red* edge uv to $E(H)$. If $uv \notin E(G)$ and $uv \in E(G')$, add a *blue* edge uv to $E(H)$. Define an *alternating* cycle as a cycle whose edges are alternately red and blue. G and G' are equivalent if and only if H has no edges.

Then H is the union of a finite sequence of edge-disjoint alternating cycles. Fix any such cycle $v_0v_1...v_kv_0$, ordered so that the v_0v_1 is red. The chain can sample $v_0v_1...v_kv_0$ with positive probability, yielding a new graph G'', whilst removing all edges in H corresponding to this cycle. Iterate until H has no more edges. Hence the chain is connected. □

Proof of Proposition 2.3.4. For a given $\mathcal{F}$, the map from $\mathcal{G}$ to $B_{\mathcal{G}}$ is injective, so the sampler can be thought of as a Markov chain ergodic with respect to the uniform distribution on $B_{\mathcal{G}}$.

We briefly describe how to view the Markov chain as operating on $B_{\mathcal{G}}$. An initial vertex v_j is sampled from V. The chain now samples u_i from the out-neighborhood of v_j and replaces the edge v_ju_i with u_iv_j. If G is undirected, additionally switch v_iu_j with u_jv_i. Continue walking along the vertices of the graph in this manner until the sampler returns to the initial vertex for the first time.

Without loss of generality, suppose G is directed. Partition B's vertex set into strongly connected components $S_1, ..., S_K$. Fix $u_i \in S_k$ and $v_j \in S_l$. If no edge is incident to u_i and v_j then $ij \in \mathcal{F} \subseteq \tilde{\mathcal{F}}$. Otherwise if $k \neq l$, edges between S_k and S_l are uniformly in one direction; say from S_k to S_l. Suppose the chain on B traverses u_iv_j, replacing it with v_ju_i. Returning to the initial vertex requires traversal of v_ju_i. Hence, u_iv_j can be flipped only an even number of times, and the direction is unchanged. By Lemma 2.3.3, $ij \in \tilde{\mathcal{F}}$. If $k = l$, u_iv_j can be switched odd number of times, so $ij \notin \tilde{\mathcal{F}}$. The undirected case holds analogously. □

Proof of Proposition 2.4.1. It suffices to show connectedness. Define

$$d(G, G') := \sum_{u \in V} \sum_{v \in V} |c_G(uv) - c_{G'}(uv)| \quad \text{for all } G, G' \in \mathcal{G}.$$

Then $(\mathcal{G}, d)$ is a metric space. Fix any two distinct graphs $G, G' \in \mathcal{G}$, and suppose the current state of the chain is G. It suffices to show that one can construct a sampling step yielding a new graph *strictly* closer to G' in this metric space.

Let $n_{uv} := c_G(uv) - c_{G'}(uv)$ for each vertex pair uv. We form a multi-graph H as follows. If $n_{uv} > 0$, add n_{uv} *red* copies of the *direction reversed* edge vu to $E(H)$, while if $n_{uv} < 0$, add $-n_{uv}$ *blue* copies of uv to $E(H)$. The graphs G and G' are equivalent if and only if H has no edges. Define an *alternating* cycle in H as a cycle whose edges are alternately *red* and *blue*.

H can be expressed as the union of a finite number of edge-disjoint alternating cycles. Fix any such alternating cycle $v_0v_1...v_lv_0$ in H. Order the cycle so that v_0v_1 is red. Letting $\mathbb{O}_n$ denote the set of odd natural numbers less than or equal to n, we define

$$k := \inf\{n \in \mathbb{O}_{l-2} : v_nv_{n+1} \notin \mathcal{F}\}$$

where we let $\inf \emptyset := l$.

Under Algorithm 2, there is a positive probability of sampling the vertex sequence $v_0v_1...v_kv_0$ given the chain is at G. Sampling $\Delta = -1$ along this vertex sequence returns a new graph G'', removing at least three edges from H whilst adding at most one. Hence $d(G'', G') \leq d(G, G') - 2$. □

B

Appendix to Chapter 3

B.1 Proof of Proposition 3.5.2

We start by defining several objects that are used in the proof. In section 3.4.1, we introduced the f as the density of P with respect to (3.8). This implies that Q has unnormalised density f with respect to the measure

$$\sum_{b\in\{0,1\}^{2k}} \lambda_b, \tag{B.1}$$

where λ_b is $\|b\|_0$-dimensional Lebesgue measure on $V_b := \{x \in \Omega_{2k} : x_i^0 = b_i\}$ and where as usual $0^0 = 0$. We parameterise V_b with $\gamma_b : (0,\infty)^d \to V_b$, where $d := \|b\|_0$. Letting $(\sigma_1, \ldots, \sigma_d)$ be the ordered vector formed from $\{i : b_i = 1\}$, we let

$$\gamma_b(c) := c_1 e_{\sigma_1} + c_2 e_{\sigma_2} + \ldots + c_d e_{\sigma_d}, \tag{B.2}$$

where e_i is the i^{th} standard basis vector for $\mathbb{R}^{2k}$. Finally, we we will also use the inclusion map $\iota_b : V_b \hookrightarrow \Omega_{2k}$.

Our strategy is to partition Ω_{2k} into sets over which (3.11) can be verified. To do this, first let O and E be the set of all binary vectors of length $2k$ with at least one zero on the odd/even index, and no zeros on the even/odd index respectively. For example, $a \in O$ if and only if $a_{2i} = 1$ for all i, and $a_{2i+1} = 0$ for some i. Fix any $a \in O$ and $a' \in E$ and define

$$\Omega_{a,a'} := \{x \in \Omega_{2k} : x_i \leq \min_j\{x_{2j+1}\} \text{ iff } a_i = 0 \text{ and } x_i \leq \min_j\{x_{2j}\} \text{ iff } a'_i = 0\}. \tag{B.3}$$

This set consists of vectors whose smallest odd elements coincide with the indices at which a is zero, and whose smallest even elements coincide with the indices at which a' is zero. A little thought shows that the sets (B.3) form a partition of Ω_{2k}. We also define the pushforward of these sets by $\mathcal{T}_{a,a'} := T(\Omega_{a,a'})$.

Lemma B.1.1 shows that (3.11) holds when applied to these subsets. Most of our work will be in proving this lemma.

Lemma B.1.1. *Fix any set $\Omega_{a,a'}$ as in (B.3). Then if $g : \Omega_{2k} \to \mathbb{R}$ is non-negative and measurable then for each $t \in \mathcal{T}_{a,a'}$, we have that $t \mapsto \int g(w)Q_t(\mathrm{d}w)$ is measurable and*

$$\int_{\Omega_{a,a'}} g(x)Q(\mathrm{d}x) = \int_{\mathcal{T}_{a,a'}} \int g(x)Q_t(\mathrm{d}x)TQ(\mathrm{d}t).$$

Lemma B.1.1 makes the proof of Proposition 3.5.2 straightforward. We first prove the proposition assuming the lemma, and then proceed to prove the lemma.

Proof of Proposition 3.5.2. First recall that the sets (B.3) form a measurable partition of Ω_{2k}. Moreover the sets $\mathcal{T}_{a,a'}$ are disjoint. To see this, first fix some $t \in \mathcal{T}_{a,a'}$. Then there exists $x \in \Omega_{a,a'}$ such that $T(x) = t$. Any other point in $\{T = t\}$ takes the form $L_t(\Delta)$ for some Δ as defined in (3.12). Such points must also lie in $\Omega_{a,a'}$. The overall result then simply follows from additivity of measure over disjoint measurable sets, and applying Lemma B.1.1.

$$\begin{aligned}\int g(x)Q(\mathrm{d}x) &= \sum_{a\in O}\sum_{a'\in E}\int_{\Omega_{a,a'}} g(x)Q(\mathrm{d}x) \\ &= \sum_{a\in O}\sum_{a'\in E}\int_{\mathcal{T}_{a,a'}}\int g(x)Q_t(\mathrm{d}x)TQ(\mathrm{d}t) \\ &= \int\int g(x)Q_t(\mathrm{d}x)TQ(\mathrm{d}t).\end{aligned}$$

□

Proof of Lemma B.1.1. Recall that $\Omega_{a,a'}$ consists of all $x \in \Omega_{2k}$ whose smallest odd elements coincide with the indices at which a is zero, and whose smallest even elements coincide with the indices at which a' is zero. If both the smallest odd and even element of x are zero, then $x \in V_{a''}$, where $a'' := a \odot a'$ and where the operator $\odot$ denotes component-wise multiplication. If the smallest odd (even) element is zero, but smallest even (odd) is positive then $x \in V_a$ ($x \in V_{a'}$). If all elements are positive then $x \in V_1$, where 1 is the unit vector of length $2k$. This shows that $\Omega_{a,a'}$ is partitioned by its intersection with V_a, $V_{a'}$, $V_{a''}$ and V_1. For notational convenience, we let $\Omega := \Omega_{a,a'}$ and $\mathcal{T} := \mathcal{T}_{a,a'}$ for what follows.

In particular, it is easy to see that $V_{a''} \subset \Omega_{a,a'}$. Letting $\mathcal{T}_1 = T(V_{a''})$, we first demonstrate that

$$\int_{V_{a''}} g(x)Q(\mathrm{d}x) = \int_{\mathcal{T}_1}\int g(x)Q_t(\mathrm{d}x)TQ(\mathrm{d}t), \tag{B.4}$$

and then

$$\int_{\Omega\setminus V_{a''}} g(x)Q(\mathrm{d}x) = \int_{\mathcal{T}\setminus\mathcal{T}_1}\int g(x)Q_t(\mathrm{d}x)TQ(\mathrm{d}t). \tag{B.5}$$

The lemma then follows trivially from summing both sides of (B.4) and (B.5).

To verify (B.4), observe that a'' must have at least one zero on both its odd and even side. Fix any $x \in V_{a''}$. The parameterization (3.12) shows that the level set $\{T = T(x)\}$ consists only of x itself, implying that $x = u_{T(x)} = v_{T(x)}$ for any $x \in V_{a''}$. Therefore

$$\begin{aligned}\int_{V_{a''}} g(x)Q(\mathrm{d}x) &= \int_{V_{a''}} g(u_{T(x)})Q(\mathrm{d}x)\\ &= \int_{\mathcal{T}_1} g(x_l)TQ(\mathrm{d}t)\\ &= \int_{\mathcal{T}_1}\int g(x)Q_t(\mathrm{d}x)TQ(\mathrm{d}t),\end{aligned}$$

where the second step uses a change of variables $t = T(x)$ and the third uses the definition of Q_t in (3.14).

Proving (B.5) is slightly more involved. First recall that points in $\Omega \setminus V_{a''}$ must lie in exactly one of V_a, V_a' or V_1. Also recall that Q has density f with respect to (B.1). Therefore

$$\begin{aligned}\int_{\Omega\setminus V_{a''}} g(x)Q(\mathrm{d}x) = &\int_{\Omega\cap V_a} g(x)f(x)\lambda_a(\mathrm{d}x) + \int_{\Omega\cap V_{a'}} g(x)f(x)\lambda_{a'}(\mathrm{d}x)\\ &+ \int_{\Omega\cap V_1} g(x)f(x)\lambda_1(\mathrm{d}x),\end{aligned} \tag{B.6}$$

where we have removed null integrals that result from distributing (B.1). Define $f_b := f\circ\gamma_b$ and $g_b := g\circ\gamma_b$ for an arbitrary vector $b \in \{0,1\}^{2k}$. This allows us to write

$$\int_{\Omega\setminus V_{a''}} g(x)Q(\mathrm{d}x) = \int_{U_a} g_a(p)f_a(p)\,\mathrm{d}p + \int_{U_{a'}} g_{a'}(q)f_{a'}(q)\,\mathrm{d}q + \int_{\Omega\cap V_1} g(x)f(x)\,\mathrm{d}x, \tag{B.7}$$

where $U_a := \{p : \gamma_a(p) \in \Omega\}$ and $U_{a'} := \{q : \gamma_{a'}(q) \in \Omega\}$. We must be careful in interpreting each integral on the right hand side of (B.7). The vector p for example is of length $\|a\|_0$, while q is of length $\|a'\|_0$.

We split the proof into three cases.

Case 1, $\|\mathbf{a}\|_\mathbf{0} = \|\mathbf{a}'\|_\mathbf{0} = \mathbf{2k-1}$: This implies that $a_i = 0$ for exactly one odd i. Consider the function $T_a : \mathbb{R}^{2k-1} \to \mathbb{R}^{2k-1}$ given by

$$T_a(p) := (p_1 + p_2, \ldots, p_{i-2} + p_{i-1}, p_{i-1}, p_i, p_i + p_{i+1}, \ldots, p_{2k-2} + p_{2k-1})^t.$$

This function satisfies $T_a(p) = T\circ\iota_a\circ\gamma_a(p)$ for all $p \in (0,\infty)^{2k-1}$. It is linear, non-singular and its Jacobian J has determinant one. To see this, observe that J is block diagonal with matrices J_1 and J_2 where J_1 has dimension $i-1$. J_1 is upper triangular and J_2 is

lower triangular, and both matrices have ones along the diagonal. Therefore

$$\det(J) = \det(J_1) \times \det(J_2) = 1 \times 1 = 1.$$

Applying the change of variables formula to the integral gives

$$\begin{aligned}\int_{U_a} g_a(p) f_a(p)\,\mathrm{d}p &= \int_{T(\Omega\cap V_a)} g(\gamma_a(T_a^{-1}(t)))f(\gamma_a(T_a^{-1}(t)))\,\mathrm{d}t \\ &= \int_{T(\Omega\cap V_a)} g(x_l)f(x_l)\,\mathrm{d}t.\end{aligned}$$

Precisely the same argument gives an analogous form for the second integral on the right of (B.7), but with x_u in the integrand rather than x_l, and with $T(\Omega \cap V_{a'})$ as the integration range.

We deal with the final integral in (B.7) using a transformation $T_1 : \mathbb{R}^{2k} \to \mathbb{R}^{2k}$ defined by

$$T_1(x) := (x_1, x_1 + x_2, x_2 + x_3, \ldots, x_{2k-1} + x_{2k})^t.$$

This is again linear and non-singular. The Jacobian is lower triangular with ones along the diagonal, and so the determinant is one. Using change of variables

$$\begin{aligned}\int_{\Omega\cap V_1} g(x)f(x)\,\mathrm{d}x &= \int_{T(\Omega\cap V_1)} \left(\int g(T_1^{-1}(x_1,t))f(T_1^{-1}(x_1,t))\,\mathrm{d}x_1\right)\mathrm{d}t \\ &= \int_{T(\Omega\cap V_1)} \left(\frac{1}{\sqrt{2k}}\int_{L_t} g(x)f(x)\,\mathrm{d}s\right)\mathrm{d}t \\ &= \int_{T(\Omega\cap V_1)} \left(\frac{1}{\sqrt{2k}}\int_{L_t} f(x)\,\mathrm{d}s \frac{\int_{L_t} g(x)f(x)\,\mathrm{d}s}{\int_{L_t} f(x)\,\mathrm{d}s}\right)\mathrm{d}t \\ &= \int_{T(\Omega\cap V_1)} \left(\alpha_t \int g(x)\mu_t(\mathrm{d}x)\right)\mathrm{d}t.\end{aligned}$$

In the second step the inner integral is rewritten as a line integral over L_t. Then, it is written in a form that allows application of the definition of μ_t in (3.5.2).

Putting this all together and grouping the integrals gives

$$\int_{\Omega_{a,a'}\setminus V_{a''}} g(x)P(\mathrm{d}x) \tag{B.8}$$

$$= \int_{\mathcal{T}^1_{a,a'}} \left(g(x_l)f(x_l) + g(x_u)f(x_u) + \alpha_t \int g(x)\mu_t(\mathrm{d}x) \right) \mathrm{d}t \tag{B.9}$$

$$= \int_{\mathcal{T}^1_{a,a'}} \int g(x)P_t(\mathrm{d}x)\left(f(x_l) + f(x_u) + \alpha_t\right) \mathrm{d}t \tag{B.10}$$

$$= \int_{\mathcal{T}^1_{a,a'}} \int g(x)P_t(\mathrm{d}x)\left(f(x_l) + f(x_u) + \int f(x_1,t)\,\mathrm{d}x_1 \right) \mathrm{d}t \tag{B.11}$$

$$= \int_{\mathcal{T}^1_{a,a'}} \int g(x)P_t(\mathrm{d}x)\left(TP_a + TP_{a'} + TP_1\right)(\mathrm{d}t) \tag{B.12}$$

$$= \int_{\mathcal{T}^1_{a,a'}} \int g(x)P_t(\mathrm{d}x)TP(\mathrm{d}t). \tag{B.13}$$

(B.10) is obtained from (B.9) by evaluating the integral of $g(x)$ with respect to P_t, where P_t is defined in Proposition 3.5.2.

Case 2, $\|\mathbf{a}\|_\mathbf{0} = \|\mathbf{a}'\|_\mathbf{0} < \mathbf{2k-1}$: Fix some $x \in \Omega \setminus V_{a''}$. Either the smallest even element, the smallest odd element, or both, are positive. Since the smallest element appears at more than one index, there must be ties between positive elements. Therefore, $\Omega \setminus V_{a''}$ must be a null set under Q, and so Q_t may be defined arbitrarily on $\mathcal{T} \setminus \mathcal{T}_1$.

Case 3, $\|\mathbf{a}\|_\mathbf{0} \neq \|\mathbf{a}'\|_\mathbf{0}$: Now suppose that $\|a\|_0 < \|a'\|_0$ and assume, for the moment, that the second and third integrals on the right hand side of (B.7) are null over Ω. Now

$$\begin{aligned}
\int_{\Omega_{a,a'}} g(x)P(\mathrm{d}x) &= \int_{\Omega_{a,a'}} g(u_{T(x)})P(\mathrm{d}x) && \text{(using that } x = u_{T(x)} \text{ on } V_a\text{)} \\
&= \int_{\mathcal{T}_{a,a'}} g(x_l)TP(\mathrm{d}t) && \text{(change of variables)} \\
&= \int_{\mathcal{T}_{a,a'}} \int g(x)P_t(\mathrm{d}x)TP(\mathrm{d}t) && \text{(definition of } P_t\text{)},
\end{aligned}$$

as required. The analogous result is established for $\|a'\|_{l_0} < \|a\|_{l_0}$ using identical reasoning.

□

B.2 Proof of Proposition 3.6.2

Proof of Proposition 3.6.2. Begin by verifying the first statement; that the set of graphs producing inadmissible data is Λ-negligible. Fix some data (d, s) and substitute

the definition of s_u^+ and s_v^+ in (3.20) to get

$$\sum_{u\in U_i}\sum_{v\in N} w_{uv} = \sum_{u\in N}\sum_{v\in V_i} w_{uv} \tag{B.14}$$

$$\sum_{u\in U_i}\sum_{v\in N\setminus V_i} w_{uv} = \sum_{u\in N\setminus U_i}\sum_{v\in V_i} w_{uv}, \tag{B.15}$$

for each $i \in I$ and any $G \in \mathcal{G}_m(d,s)$. In the second step, we have simply removed summands common to both sides. If (B.15) is positive on either side then it implies a disjoint sum of edge weights exactly equate. This event is Λ-negligible, and so it suffices to show that any graph producing inadmissible data must have (B.15) positive for some $i \in I$.

First suppose that condition 1 of Definition 3.6.1 is violated. Then $(U_i)_{i\in I}$ and $(V_i)_{i\in I}$ can be labeled so that $U_1 \cap U_2 \neq \emptyset$ and/or $V_1 \cap V_2 \neq \emptyset$. We show by contradiction that (B.15) must be positive for some $i \in \{1,2\}$. Suppose first that (B.15) is zero for $i \in \{1,2\}$. Then (B.15) holds for $U' = U_1 \setminus U_2$ and $V' = V_1 \setminus V_2$. We show this by expanding the left hand side of (B.15)

$$\sum_{u\in U_1}\sum_{v\in N\setminus V_1} w_{uv} = \sum_{u\in U_1\setminus U_2}\sum_{v\in N\setminus V_1} w_{uv} \tag{B.16}$$

$$= \sum_{u\in U'}\left(\sum_{v\in N\setminus V'} w_{uv} - \sum_{v\in V_2\cap V_1} w_{uv}\right) \tag{B.17}$$

$$= \sum_{u\in U'}\sum_{v\in N\setminus V'} w_{uv} - \sum_{u\in U'}\sum_{v\in V_2\cap V_1} w_{uv} \tag{B.18}$$

$$= \sum_{u\in U'}\sum_{v\in N\setminus V'} w_{uv} = 0. \tag{B.19}$$

In (B.17) we have used that $N \setminus V' = (V_1 \cap V_2) \cup (N \setminus V_1)$. To see how we remove the final summation in (B.18), observe that if $u \in U'$ then $u \in N \setminus U_2$. Also if $v \in V_1 \cap V_2$ then $v \in V_2$. Therefore because (B.15) holds for $i = 2$ and by assumption is equal to zero, this summation must also be zero. The same reasoning as above can be applied to the right hand side of (B.15) to show that

$$\sum_{u\in U'}\sum_{v\in N\setminus V'} w_{uv} = \sum_{u\in N\setminus U'}\sum_{v\in V'} w_{uv} = 0,$$

which implies that U' and V' satisfy (3.20). Since $U' \times V' \subset U_1 \times V_1$ this contradicts the definition of U_1 and V_1, and establishes that graphs for such data have positive ties, and thus lie in a Λ-negligible set.

Now suppose that the second condition of Definition 3.6.1 is violated, i.e. $\tilde{\mathcal{G}}_m(d,s)$ is empty. This could be because the reference set (3.6) is empty, in which case no graph in $\mathcal{G}$ aligns with the data (d,s). Suppose instead that (3.6) is non-empty. Any graph in

(3.6) has some edge uv such that

$$uv \notin \bigcup_{i \in I} (U_i \times V_i)\,.$$

This implies that (B.15) is positive for some $i^* \in I$ and that the graph must have positive ties in its weight matrix. Therefore the set of such graphs is P-null. Moreover, this argument verifies the last statement in the proposition, which is that the set of graphs not in $\tilde{\mathcal{G}}_m(d,s)$ for some data (d,s) is P-null. □

B.3 Proof of Proposition 3.6.4

Our strategy is to first establish open set irreducibility with respect to some topology on $\mathcal{G}_m(d,s)$. This result is stated in Lemma B.3.3. This is linked to ψ-irreducibility, which then establishes Proposition 3.6.4.

First we define objects used in the proofs. Fix some admissible data (d,s), as in the proposition. Associate each graph $G \in \mathcal{G}_m(d,s)$ with a weighted, bipartite and undirected graph $B(G) := (R, C, W)$, with vertex sets $R := \{r_u : u \in N\}$ and $C := \{c_u : u \in N\}$, and with weight matrix defined through $w_{r_u c_v}(B) := w_{uv}(G)$. Recall the vertex sets $\{U_i\}_{i \in I}$ and $\{V_i\}_{i \in I}$ introduced in Section 3.6, which are associated with the data (d,s). We define sets

$$E_i := \{p_u : u \in U_i\} \cup \{q_v : v \in V_i\}, \tag{B.20}$$

for each $i \in I$. By the definition of admissibility (Definition 3.6.1) it is clear that these sets form a partition of $P \cup Q$.

We begin by stating and proving two lemmas which will help establish Lemma B.3.3.

Lemma B.3.1. *Fix some $i \in I$. The vertex set E_i is connected in $B(G)$ for all $G \in \mathcal{G}_m(d,s)$.*

Proof. Fix any $G \in \mathcal{G}_m(d,s)$ and let C be a connected component of $B(G)$. It is easy to see that

$$\sum_{u \in \{u : p_u \in C\}} s_u^- = \sum_{v \in \{v : q_v \in C\}} s_v^+,$$

and so, by the definition of admissibility, C must be the union of sets of the form (B.20). This implies that E_i must wholly lie within a connected component for all graphs in the reference set. □

The next lemma establishes that the Markov chain can move between different graph topologies. In particular, it shows that an edge can be added without altering the rest of the graph's topology. It will be used in the proof of Lemma B.3.3.

Lemma B.3.2. *Let $uv \in U_i \times V_i$ for some $i \in I$, and $uv \notin \mathcal{F}$. Suppose the current state of the chain is G and $uv \notin E(G)$. Fixing $\varepsilon > 0$, these is positive probability of reaching some G^* satisfying $W_{uv}(G^*) \in (0, \varepsilon]$ and $E(G^*) = E(G) \cup \{uv\}$ in one iteration.*

Proof of Lemma B.3.2. Since $uv \in U_i \times V_i$, p_u and q_v must belong to E_i. By Lemma B.3.1, p_u and q_v must be connected by a simple path in $B(G)$. Because $B(G)$ is bipartite, the path must have odd length, and when including $p_u q_v$, defines a k-cycle with one zero entry and no forced edges. The k-cycle selection strategy defined in Algorithm 4 gives positive probability to all such k-cycles. Sampling this k-cycle and $\Delta \in (0, \varepsilon]$ yields a graph with the required properties. Sampling Δ in this range has positive probability because the density f is assumed to be positive everywhere. □

Lemma B.3.3 shows that the Markov chain is open set irreducible, where the open sets are those induced by the metric

$$d(G_1, G_2) := \max_{uv \in N^2} \mid w_{uv}(G_1) - w_{uv}(G_2) \mid + \rho(G_1, G_2), \tag{B.21}$$

where $\rho(G_1, G_2) = 1$ if G_1 and G_2 hae differing topologies, and is otherwise zero. It is easy to check that this really is a metric. Before stating the lemma, we recall the definition of $\tilde{\mathcal{G}}_m(d, s)$ from Section 3.6.

Lemma B.3.3. *Let the current state of the chain be $G \in \mathcal{G}_m(d, s)$, and fix any $G' \in \tilde{\mathcal{G}}_m(d, s)$, and $\varepsilon > 0$. There exists some integer n for which there is positive probability of reaching an ε-neighbourhood (under (B.21)) of G' within n steps.*

Proof of Lemma B.3.3. Form a signed graph $H := (N, D)$, where $D = (d_{uv})$ is a matrix of possibly negative weights that satisfy $d_{uv} := w_{uv}(G) - w_{uv}(G')$. Label uv *red* if $d_{uv} > 0$ and *blue* if $d_{uv} < 0$. G and G' are equal if and only if $E(H)$ is empty. Red edges must be in $E(G)$, however blue edges may not be in $E(G)$. Therefore, begin by using Lemma B.3.2 repeatedly to adjust G so that all blue edges in H are in $E(G)$. This Lemma can be applied because we have assumed that $G' \in \tilde{\mathcal{G}}_m(d, s)$. This implies that if uv is blue then it must belong to $U_i \times V_i$ for some $i \in I$, because otherwise $w_{uv}(G') = 0$, which contradicts the requirement that $w_{uv}(G) < w_{uv}(G')$ for blue edges.

Call a k-cycle *alternating* if its vertex pairs (3.9) alternate between red and blue edges , when interpreted as part of H. As long as $E(H)$ is non-empty, one can always form an alternating k-cycle. To see this, note that the in- and out-strengths of vertices in H are uniformly zero. Therefore, if uv is red ($d_u v > 0$), then there must exist blue wv for which $d_{wv} < 0$. Hence walk along alternating edges until returning to a vertex for the first time, forming an alternating k-cycle.

Fix one such k-cycle z_1 ordered so that the first edge is *red*. All edges in the cycle are positive and do not belong to $\mathcal{F}$. Let

$$\Delta_1' := - \min_{uv \in z_1} \{\|D_{uv}\|\}$$

along the cycle. If we sampled $\Delta = \Delta_1$ exactly along z, this would remove an edge from $E(H)$. Repeating the process at most d times, where d is the size of $E(h)$, yields k-cycles $z_1, \dots, z_d$ and $\Delta'_1, \dots, \Delta'_d$, after which we reach G'. Therefore there must exist some $\varepsilon_0 > 0$ such that sampling $\Delta_i \in \Delta'_i + [-\varepsilon_0, \varepsilon_0]$ sequentially along these cycles gives a graph in an ε-neighbourhood of G'. □

To link open set irreducibility (Lemma B.3.3) with ψ-irreducibility, it is helpful to view the reference set $\mathcal{G}$ as a subset of a vector space. This will provide a geometric interpretation to the k-cycles that form the basis of our Markov chain. This will motivate a measure φ on $\mathcal{G}$ for which it is easy to demonstrate φ-irreducibility of the chain.

Let $\mathbb{R}^{N\times N}$ be the vector space of real-valued $N \times N$ matrices. Equip this with the metric defined by (B.21). We will assume that if $uv \notin U_i \times V_i$ for some $i \in \mathcal{I}$ then $uv \in \mathcal{F}$. Let V be the affine subspace of $\mathbb{R}^{N\times N}$ with row and column margins s^- and s^+ respectively and additionally respecting the forced zeros implied by $\mathcal{F}$. This has some dimension

$$d \geq N^2 + 1 - 2N - |\mathcal{F}|.$$

Then the reference set satisfies

$$\mathcal{G} = V \cap \Omega_{N\times N},$$

where $\Omega_{N\times N}$ is the $N \times N$-dimensional non-negative orthant.

Fixing $W \in \mathcal{G}$, we see that a k-cycle is equivalent to sampling from a line in V. The 'direction' of this line is given by an $N \times N$ matrix M, where $M_{uv} = 1$ if uv is on the odd side of the cycle, $M_{uv} = -1$ if on the even side, and with all other entries being zero. Fix any W^* in V. The arguments used in the proof of B.3.3 can easily be extended to show that there exists $(M_1^*, \dots, M_L^*)$ and a real-valued vector $(\Delta_1^*, \dots, \Delta_L^*)$ for which

$$W^* = W + \sum_{l=1}^{L} M_l^* \Delta_l^*,$$

and each M_l^* corresponds to a k-cycle. This in turn implies that there exists a set of such matrices labelled $(M_1, \dots, M_d)$ which form an affine basis for V. Therefore

$$f_W(\Delta) := W + \sum_{i=1}^{d} M_i \Delta_i, \tag{B.22}$$

where $\Delta := (\Delta_1, \dots, \Delta_d)$ parameterizes V. Equation (B.22) is a homeomorphism between $\mathbb{R}^d$ and V. We are now ready to prove Proposition 3.6.4.

Proof of Proposition 3.6.4. Fix any $W \in V$ for which $W_{uv} > 0$ if $uv \notin \mathcal{F}$. The existence of such a W is implied by Lemma B.3.2. Consider the parameterization of V defined

by (B.22). We use this to define a measure φ on $\mathcal{G}$, and show that the Markov chain is irreducible with respect to φ.

Let $\varepsilon_0 := \min_{uv \notin \mathcal{F}} \mid w_{uv} \mid /d$, and define $N_{\varepsilon_0} := (-\varepsilon_0, \varepsilon_0)^d \subset \mathbb{R}^d$. Let μ be Lebesgue measure restricted to N_{ε_0}. Define φ on $(\mathcal{G}, \mathcal{B})$ as the pushforward of μ under f_w, so that

$$\varphi(A) = \lambda^d \left(f^{-1}(A) \cap N_{\varepsilon_0} \right),$$

for measurable A.

We now show φ-irreducibility. Fix any measurable A for which $\varphi(A) > 0$. Letting $E := f_w^{-1}(A) \cap N_{\varepsilon_0}$, it is clear that E must be Lebesgue positive in $\mathbb{R}^d$. Consider a Markov chain X_n with kernel Λ. Define $\Phi_n := f_w^{-1}(X_n)$ and suppose $\Phi_n \in N_{\varepsilon_0}$. Each of the basis matrices $m_1, \ldots, m_d$ corresponds to a k-cycle. There is positive probability that the chain selects the cycle corresponding to m_k at the $n + k^{\text{th}}$ step. Conditional on this, φ_{n+d} has positive density everywhere on N_{ε_0}. This implies that if $X_n \in f_w(N_{\varepsilon_0})$ then $Q^d(X_n, A) > 0$.

It remains to show that for each x, $Q^n(x, f_w(N_{\varepsilon_0})) > 0$ for some n. Since f_w maps open sets, $f_w(N_{\varepsilon_0})$ is open in $\mathcal{G}$. Therefore this result follows from Lemma B.3.3. □

C

Appendix to Chapter 4

C.1 Proofs

Proof of Lemma 4.2.2. For $k \in 1, \ldots, N$,

$$\begin{aligned} \mathrm{P}\{R_M = k\} &= \sum_{m=1}^{N} \mathrm{P}\{R_M = k \mid M = m\}\mathrm{P}\{M = m\} \\ &= \frac{1}{N}\sum_{m=1}^{N} \mathrm{P}\{R_m = k\} = \frac{1}{N}, \end{aligned}$$

where the last equality holds because exactly one element of $(R_1, ..., R_N)$ must equal k. In other words, the events $\{R_m = k\}$ partition the sample space. □

Proof of Proposition 4.2.3. Consider the variables involved in one evaluation of Algorithm 7. The proposition assumes that the kernel K_y is $\pi(\theta \mid y)$-reversible. Letting f denote the joint distribution of $\theta_{1:N}$ then for $m > 1$,

$$\begin{aligned} f(\theta_{1:N} \mid M = m, y) &= \pi(\theta_m \mid y)\left(\prod_{i=1}^{m-1} K_y(\theta_{m-i+1}, \theta_{m-i})\right)\left(\prod_{j=m+1}^{N} K_y(\theta_{j-1}, \theta_j)\right) \\ &= \pi(\theta_m \mid y)K(\theta_m, \theta_{m-1})\left(\prod_{i=1}^{m-2} K_y(\theta_{m-1-i+1}, \theta_{m-1+i})\right)\left(\prod_{j=m+1}^{N} K_y(\theta_{j-1}, \theta_j)\right) \\ &= \pi(\theta_{m-1} \mid y)K(\theta_{m-1}, \theta_m)\left(\prod_{i=1}^{m-2} K_y(\theta_{m-1-i+1}, \theta_{m-1+i})\right)\left(\prod_{j=m+1}^{N} K_y(\theta_{j-1}, \theta_j)\right) \\ &= \pi(\theta_{m-1} \mid y)\left(\prod_{i=1}^{m-2} K_y(\theta_{m-1-i+1}, \theta_{m-1+i})\right)\left(\prod_{j=m}^{N} K_y(\theta_{j-1}, \theta_j)\right) \\ &= f(\theta_{1:N} \mid M = m-1, y). \end{aligned}$$

This implies that the distribution of $\theta_{1:N}$ is independent of M. Since additionally R is also assumed independent of M, both $R(\theta_{1:L})$ and M satisfy the conditions of Lemma 4.2.2 and so $R_M(\theta_{1:L})$ is uniformly distributed. □

Proof of Theorem 4.3.1. We use induction from $j = k$ to $j = 1$ to show that

$$\mathrm{P}\{\text{fail} \mid \text{step } j \text{ reached}\} \leq (k + 1 - j)\beta_j, \tag{C.1}$$

where $\beta_j = \beta/\gamma^{j-1}$ is as defined in Algorithm 8.

First, by the usual arguments for the Bonferroni correction, $\mathrm{P}\{q_i \leq p\} \leq p$ for all $p \in [0, 1]$ and for $i = 1, \ldots, k$. This, immediately shows that (C.1) holds for $j = k$.

To show that (C.1) holds for $j = i \in \{1, \ldots, k-1\}$ given that it holds for $j = i+1$, we argue as follows. Let $A_i = \{\beta_i < q_i \leq \gamma + \beta_i\}$ and $B_i = \{q_i \leq \beta_i\}$. Using the arguments for the Bonferroni correction again gives $\mathrm{P}\{B_i\} \leq \beta_i$ and $\mathrm{P}\{A_i\} \leq \gamma + \beta_i - \mathrm{P}\{B_i\}$. Then

$$\begin{aligned}
&\mathrm{P}\{\text{fail} \mid \text{step } i \text{ reached}\} = \mathrm{P}\{B_i\} + \mathrm{P}\{A_i, \text{fail} \mid \text{step } i \text{ reached}\} \\
&\quad = \mathrm{P}\{B_i\} + \mathrm{P}\{A_i\}\,\mathrm{P}\{\text{fail} \mid \text{step i+1 reached}\} \\
&\quad \leq \mathrm{P}\{B_i\} + (\gamma + \beta_i - \mathrm{P}\{B_i\})\,\mathrm{P}\{\text{fail} \mid \text{step i+1 reached}\} \\
&\quad = \gamma\,\mathrm{P}\{\text{fail} \mid \text{step i+1 reached}\} + \beta_i\,\mathrm{P}\{\text{fail} \mid \text{step i+1 reached}\} + \mathrm{P}\{B_i\}(1 - \mathrm{P}\{\text{fail} \mid \text{step i+1 reached}\}) \\
&\quad \leq \gamma\,\mathrm{P}\{\text{fail} \mid \text{step } i+1 \text{ reached}\} + \beta_i \\
&\quad \leq \gamma(k + 1 - (i + 1))\beta_{i+1} + \beta_i = (k + 1 - i)\beta_i
\end{aligned}$$

Thus using (C.1) for $i = 1$ gives $\mathrm{P}\{\text{fail}\} \leq k\beta_1 = k\beta = \alpha$. □

C.2 Tuning Sequential Parameters

We use a simulation study to propose default parameters for the sequential tests. The classical goodness-of-fit setting is considered: independent and identically distributed samples from can be generated and the task is to test if the samples derive from a standard normal distribution. The two-sided Kolmogorov-Smirnov test is used to test this.

The sample size for $k = 1$ and $\Delta = 1$ (i.e. the non-sequential setting) was chosen to be 10^4. Sample sizes for other settings were adjusted using (4.2) so that under the null the computational effort were identical. α is set at 10^{-5} to replicate the situation where we only want very few rejections.

Results are in Table C.1, based on 10^4 repeated tests. The first two columns are under the null i.e. we would only expect the nominal number of rejections. This seems to be roughly the case.

Table C.2 is similar to Table C.1, with the exception that the sample size for $k = 1$ and $\Delta = 1$ (i.e. the non-sequential setting) was chosen to be 10^3 and that some different alternative have been considered.

Table C.1 Power of the sequential procedure using a KS test on iid data. Null is $\mathcal{N}(0,1)$.

	N(0,1),α=0.01	N(0,1)	N(0.05,1)	N(0.03,1)	N(0.02,1)	N(0,0.95^2)	N(0,0.97^2)
k=1,Δ=1	0.011	0.000	0.415	0.028	0.003	0.007	0.000
k=3,Δ=1	0.010	0.000	0.939	0.202	0.016	0.380	0.004
k=3,Δ=2	0.009	0.000	0.967	0.448	0.061	0.681	0.025
k=3,Δ=4	0.009	0.000	0.960	0.529	0.156	0.658	0.109
k=5,Δ=1	0.009	0.000	0.974	0.285	0.026	0.615	0.008
k=5,Δ=2	0.010	0.000	0.986	0.583	0.100	0.861	0.080
k=5,Δ=4	0.009	0.000	0.979	0.632	0.233	0.808	0.227
k=7,Δ=1	0.009	0.000	0.988	0.392	0.035	0.876	0.035
k=7,Δ=2	0.010	0.000	0.990	0.692	0.141	0.959	0.231
k=7,Δ=4	0.011	0.000	0.975	0.702	0.286	0.887	0.408
k=9,Δ=1	0.008	0.000	0.987	0.384	0.035	0.916	0.054
k=9,Δ=2	0.009	0.000	0.990	0.673	0.124	0.963	0.266
k=9,Δ=4	0.010	0.000	0.965	0.693	0.252	0.892	0.430
k=11,Δ=1	0.010	0.000	0.985	0.364	0.027	0.919	0.063
k=11,Δ=2	0.010	0.000	0.988	0.636	0.105	0.966	0.267
k=11,Δ=4	0.008	0.000	0.949	0.674	0.198	0.891	0.420

$n = 10^4$ for the non-sequential test ($k = 1, \Delta = 1$); other n adjusted to give same expected effort under null, $\alpha = 10^{-5}$, unless otherwise indicated.

Table C.2 Power of the sequential procedure using a KS test on iid data. Null is $\mathcal{N}(0,1)$.

	N(0,1),α=0.01	N(0,1)	N(0.15,1)	N(0.1,1)	N(0.05,1)	N(0,0.85^2)	N(0,0.9^2)
k=1,Δ=1	0.009	0.000	0.324	0.039	0.001	0.006	0.000
k=3,Δ=1	0.009	0.000	0.902	0.249	0.004	0.349	0.006
k=3,Δ=2	0.009	0.000	0.937	0.522	0.014	0.668	0.058
k=3,Δ=4	0.009	0.000	0.939	0.576	0.050	0.658	0.162
k=5,Δ=1	0.008	0.000	0.948	0.348	0.006	0.604	0.019
k=5,Δ=2	0.010	0.000	0.971	0.655	0.022	0.848	0.145
k=5,Δ=4	0.009	0.000	0.959	0.677	0.085	0.817	0.295
k=7,Δ=1	0.011	0.000	0.973	0.481	0.006	0.887	0.077
k=7,Δ=2	0.011	0.000	0.983	0.759	0.031	0.952	0.372
k=7,Δ=4	0.009	0.000	0.958	0.744	0.095	0.890	0.487
k=9,Δ=1	0.009	0.000	0.972	0.468	0.005	0.917	0.112
k=9,Δ=2	0.009	0.000	0.985	0.738	0.025	0.962	0.412
k=9,Δ=4	0.009	0.000	0.949	0.725	0.072	0.883	0.531
k=11,Δ=1	0.008	0.000	0.968	0.441	0.004	0.921	0.127
k=11,Δ=2	0.009	0.000	0.977	0.712	0.019	0.967	0.418
k=11,Δ=4	0.009	0.000	0.936	0.722	0.047	0.883	0.525

$n = 10^3$ for the non-sequential test ($k = 1, \Delta = 1$), $\alpha = 10^{-5}$, unless otherwise indicated.

For the alternatives there is a very substantial increase in terms of power compared to the non-sequential approach ($k = 1, \Delta = 1$). Increasing the sample size at the second step seems beneficial - $\Delta = 2$ and $\Delta = 4$ seem to be doing better than $\Delta = 1$ in the simulation results. Furthermore, the number of sequential steps should be large (at least $k \geq 5$).

An over all good performance seems to be achieved by using $k = 7$ and $\Delta = 4$. Therefore, these are the default settings used in our R-package.

D

Appendix to Chapter 5

D.1 Offspring Dispersion

Define the offspring distribution of any given infection to be the distribution of the random number of offspring attributable to that infection. We show that assuming the variance of these distributions are a constant proportion of the mean implies, under suitable independence assumptions, the same result for new infections I_t for all time points.

Assume some ordering over infections at each period, and let $O_t^{(i)}$ denote the number of offspring of the i^{th} infection at time t. This can be decomposed as

$$O_t^{(i)} = \sum_{s=t+1}^{\infty} O_{ts}^{(i)}, \tag{D.1}$$

where $O_{ts}^{(i)}$ are the number of offspring of i birthed at time s. The branching process behind Equation (5.5) implies that $O_{ts}^{(i)}$ has mean $R_s g_{s-t}$. Assume that $\{O_{ts}^{(i)} : s \geq t\}$ are mutually independent and have variance which is a fixed proportion d of the mean. By Equation (D.1), this implies the same variance relationship for $O_t^{(i)}$. In particular, if $R_s = R_t$ for $s > t$ then $O_t^{(i)}$ has mean R_t and variance dR_t. New infections at time t can be expressed as

$$I_t = \sum_{s=1}^{t-1} \sum_{i=1}^{I_s} O_{st}^{(i)}. \tag{D.2}$$

Assume that all $O_{st}^{(i)}$ appearing in Equation (D.2) are mutually independent conditional on everything occurring up to time $t-1$, the result clearly follows by taking the variance of both sides of Equation D.2 given R_t and $I_{v:t-1}$.

D.2 Population Adjustment

Here we motivate Equation (5.6), which is used to adjust transmission rates for the size of the infectable population. The most obvious starting point for such an adjustment would be to let

$$\mathrm{E}[I_t|R_t, I_{v:t-1}] = \left(\frac{S_0 - I_{t-1}}{S_0}\right) R_{u,t} L_t, \tag{D.3}$$

where R_{u_t} is defined as in Section 5.4.2. This is similar in form to a *discrete logistic growth model*. Such models are well known as examples of simple models that exhibit chaotic dynamics ((May, 1976)). In particular, it is possible that the expected value on the left hand side exceeds the remaining susceptible population. Intuitively, this issue occurs because multiple infections can occur simultaneously in the discrete model. We therefore propose solving this by using a population adjustment motivated by the solution to a continuous time model whose intensity is a simplification of Equation (5.3).

Suppose we observe $I_{v:t-1}$ and current transmission R_t. We evolve infections from time $t-1$ to t continuously, and hence avoid overshooting. Define a continuous time counting $\tilde{I}(s)$ process starting at time $t-1$ by the intensity

$$\tilde{\lambda}(s) = \left(\frac{S_0 - \tilde{I}(s)}{S_0}\right) R_{u,t} L_t, \tag{D.4}$$

for $s \geq t-1$, and with initial condition $\tilde{I}(t-1) = I_{t-1}$. Supplementary D.3 shows that

$$\mathrm{E}[\tilde{I}(t)] = I_{t-1} + (S_0 - I_{t-1})\left(1 - \exp\left(-\frac{R_{u,t} L_t}{S_0}\right)\right), \tag{D.5}$$

which is the motivation for Equation (5.6).

D.3 Proof of Equation (D.5)

Without loss of generality, we prove the result for time $t = 1$. The argument remains the same for all $t > 1$.

From ((Pinsky and Karlin, 2010, Lemma 5.5)), we have

$$\mathrm{E}[\tilde{I}(s)] = \tilde{I}(0) + \int_0^s \mathrm{E}[\tilde{\lambda}(l)]dl \text{ for } s \geq 0.$$

The following lemma derives an expression for the the expected intensity on the right hand side.

Lemma D.3.1. *The expected intensity takes the form*

$$E[\tilde{\lambda}(s)] = \tilde{\lambda}(0) \exp\left(-\frac{R_{u,1}L_1}{S_0}s\right),$$

for all $s \geq 0$.

Proof of Lemma D.3.1. Fix $s \geq 0$, some small $\Delta > 0$ and let $h(s) := \mathrm{E}[\tilde{\lambda}(s)]$. We have from Equation (D.4) that

$$h(s+\Delta) = \left(\frac{S_0 - \mathrm{E}[\tilde{I}(s+\Delta)]}{S_0}\right) R_{u,1}L_1. \tag{D.6}$$

We can write

$$\mathrm{E}[\tilde{I}(s+\Delta)|\tilde{\lambda}(s)] = \mathrm{E}[\tilde{I}(s)|\tilde{\lambda}(s)] + \tilde{\lambda}(s)\Delta + \mathcal{O}(\Delta),$$

and taking expectations on both sides,

$$\mathrm{E}[\tilde{I}(s+\Delta)] = \mathrm{E}[\tilde{I}(s)] + h(s)\Delta + \mathcal{O}(\Delta).$$

Substituting this into (D.6) and rearranging gives

$$\begin{aligned} h(s+\Delta) &= \left(\frac{S_0 - \mathrm{E}[\tilde{I}(s+\Delta)]}{S_0}\right) R_{u,1}L_1 - \frac{R_{u,1}L_1}{S_0}\left(h(s)\Delta + \mathcal{O}(\Delta)\right), \\ &= h(s) - \frac{R_{u,1}L_1}{S_0}\left(h(s)\Delta + \mathcal{O}(\Delta)\right). \end{aligned}$$

Rearranging gives

$$\frac{h(s+\Delta) - h(s)}{\Delta} = -\frac{R_{u,1}L_1}{S_0}\left(h(s) + \frac{\mathcal{O}(\Delta)}{\Delta}\right).$$

Taking the limit as $\Delta \to 0$ and rearranging gives the differential equation

$$\frac{h'(s)}{h(s)} = -\frac{R_{u,1}L_1}{S_0}.$$

Integrating both sides gives

$$\log(h(s)) = -\frac{R_{u,1}L_1}{S_0}s + C.$$

Using that $h(0) = \tilde{\lambda}(0)$ gives the constant $C = \log(\tilde{\lambda}(0))$. Plugging in yields the required result. □

Hence,

$$\begin{aligned}
\mathrm{E}[\tilde{I}(s)] &= I_0 + \tilde{\lambda}(0)\int_0^s \exp\left(-\frac{R_{u,1}L_1}{S_0}l\right) dl \\
&= I_0 + \tilde{\lambda}(0)\frac{S_0}{R_{u,1}L_1}\left(1 - \exp\left(-\frac{R_{u,1}L_1}{S_0}s\right)\right) \\
&= I_0 + (S_0 - \tilde{I}(s))\left(1 - \exp\left(-\frac{R_{u,1}L_1}{S_0}s\right)\right).
\end{aligned}$$

Letting $s = 1$ gives the required result.

E

Appendix to Chapter 6

E.1 Priors on Model Parameters

epidemia aims to give the user a high degree of control over setting prior distributions. It does this by leveraging the functionality provided by **rstanarm**, which provides functions representing a number of different prior families. These include for example student-t, Laplace, and hierarchical shrinkage families. In this appendix, we provide a brief introduction to the available families, and discuss some important quirks to be aware of when defining priors. We use the same mathematical notation as in Section 6.2.

*Please do not rely on the default priors in **epidemia**. Although these have been designed to be weakly informative, they are not guaranteed to be appropriate for your particular model. Please adjust prior distributions as required.*

Priors must be defined for all parameters in each of the three model components: transmission, infection, and observations. In the transmission model, priors must be set for all effects appearing in the linear predictor η. In the infection model, a prior must be set on τ, but also on the dispersion parameter d in the extended version of the model. In each observational model, priors must be set for effects defining the multipliers α_t, but also for the auxiliary parameter for the sampling distribution, ϕ.

In general, primitive model parameters can be classified as are either intercepts, fixed effects, a covariance matrix, an auxiliary parameter, or the error term in a random walk. We discuss each in turn, in particular highlighting where they appear in the model, and what distributions are available for them.

E.1.1 Priors on Intercepts

Intercepts can appear in the linear predictor η for the reproduction numbers R and in the linear predictors for multipliers α. The prior distribution is specified using an argument `prior_intercept`. This appears in both `epirt()` and `epiobs()`. `prior_intercept` must be a call to an **rstanarm** function that represents a student-t family: i.e. one of `normal()`, `student_t()` or `cauchy()` from **rstanarm**. `prior_intercept` is of course

only used if the formula specifies an intercept. Please note that the interpretation of `prior_intercept` depends on the `center` argument to `epirt()` and `epiobs()`. Please see Section E.1.6 for more details.

E.1.2 Priors on Regression Coefficients

In addition to intercepts, the predictors for R and α may also contain fixed effects. In the regression for R this corresponds to the parameter vector β. The prior distribution is set using the `prior` argument, which, similarly to `prior_intercept`, appears in both `epirt()` and `epiobs()`. Note that this *does not* set the prior for the group-specific effects b, which are instead controlled by `prior_covariance`.

`prior` can be a call to one of **rstanarm**'s prior functions. These can be broadly grouped into four families: student-t, hierarchical shrinkage, Laplace and the product normal family. Note that *all effects must follow the same family*; for example, it is not possible for β_1 to have a normal prior while β_2 has a Cauchy prior. Nonetheless, different hyperparameters can be set for each effect.

As an example, suppose the following formula is used to model R, where `cov1` and `cov2` are some covariates.

```
R> R(group, date) ~ 1 + cov1 + cov2
```

Consider the following two prior specifications in the call to `epirt()`.

- `prior = rstanarm::normal(location=0,scale=1)` gives a standard normal prior to both covariate effects.
- `prior = rstanarm::normal(location=c(0,1),scale=c(1,2))` sets priors $\beta_1 \sim N(0,1)$ and $\beta_2 \sim N(1,2)$, where β_1 and β_2 are the effects for `cov1` and `cov2` respectively. To give different prior locations and or scales for each covariate, we simply pass numeric vectors instead of scalars.

The interpretation of `prior` depends on whether covariates are being centered, and whether automatic scale adjustments are occurring. Please see Section E.1.6 for more details.

Additional Priors

In addition to **rstanarm**'s prior functions, **epidemia** offers additional prior families for regression coefficients. Currently the only additional prior available is `shifted_gamma`. This represents a gamma distribution that can be shifted to have support other than on $[0,\infty)$. Specifically,

$$\beta_i \sim \text{Gamma}(\alpha_i, \theta_i) - a_i, \tag{E.1}$$

where α_i and θ_i are shape and scale parameters, and a_i is a shift. This prior is used in Flaxman et al. ((2020a)) to model the prior effect of control measures on Covid-19

transmission. Intuitively, it is unlikely that a measure designed to reduce transmission rates ends up increasing transmission significantly. This implies that a symmetric prior may not be appropriate for these effects: it makes sense to put low mass on large positive effect sizes. In addition, this prior can help to improve identifiability when multiple measures occur in quick succession - as is often the case during the early stages of an epidemic.

E.1.3 Priors on Auxiliary Parameters

Auxiliary parameters can appear in the sampling distributions for observations. This corresponds to the parameter ϕ introduced in Section 6.2.1. The interpretation of this parameter depends on the chosen distribution. The Poisson distribution has no auxiliary parameter as it is fully defined by its mean. For the negative binomial distribution (specified by using `family = "neg_binom"` in the call to `epiobs()`), ϕ represents the reciprocal dispersion. An auxiliary parameter d also exists in the extended version of the infection model (when using `latent = TRUE` in the call to `epiinf()`). See Section 6.2.5 for more information on this parameter. This represents the *coefficient of dispersion* of the offspring distribution. Auxiliary parameters are always non-negative in **epidemia**.

Priors for auxiliary parameters are set using the `prior_aux` argument in the `epiobs()` and `epiinf()` modeling functions. It is not used when `family = "poisson"` in the call to `epiobs()` or when `latent = FALSE` in the call to `epiinf()`. `prior_aux` can be a call to one of `normal()`, `student_t()`, `cauchy()` or `exponential()` from **rstanarm**.

E.1.4 Priors on Covariance Matrices

Recall that partial pooling can be used in the regression for R_t. The partially pooled parameters b are characterized as zero mean multivariate normal with an unknown covariance matrix, which must itself be assigned a prior. The precise model for these parameters is described in detail in Appendix E.2. The prior on the covariance matrix can be set using the `prior_covariance` argument in `epirt()`.

Although the Inverse-Wishart prior is a popular prior for covariance matrices, it does not cleanly separate shape and scale ((Tokuda et al., 2011)). A general approach is to decompose the prior on the covariance matrix into a prior on the correlation matrix and a vector of variances. This is the approach taken by **rstanarm**, which has functions `decov()` and `lkj()` which represent priors for covariance matrices. These are also used by **epidemia** for the same purpose.

We briefly describe **rstanarm**'s decov prior, as it applies to partially pooled parameters in the regression for R_t. Suppose the formula for R_t contains a term of the form `(expr | factor)`, and that `expr` evaluates to a model matrix with p columns, and `factor` has L levels. Let θ_l denote the p-vector of parameters for the l^{th} group. From Appendix E.2 this is modeled as

$$\theta_l \sim N(0, \Sigma), \tag{E.2}$$

where Σ is a $p \times p$ covariance matrix. The decov prior decomposes Σ into a vector of variances $(\sigma_1^2, \dots \sigma_p^2)$ and a correlation matrix Ω, which is given an LKJ prior. The variance vector is decomposed into the product of a simplex vector s and the trace of Ω, which is just the sum of the individual variances. Specifically,

$$\sigma_i^2 = s_i \text{tr}\left(\Sigma\right). \tag{E.3}$$

The simplex vector is given a symmetric Dirichlet prior, while the trace is decomposed into $tr(\Sigma) = p\kappa^2$, where p is the order of the matrix (i.e. the number of correlated effects), and κ is a parameter which is assigned a scale invariant prior; specifically a Gamma with given shape and scale hyperparameters. When $p = 1$, for example with `(1 | factor)`, the prior simplifies considerably. Σ simply reduces to κ^2, which has a Gamma prior.

E.1.5 Priors on Random Walks

Section 6.2.4 described how the linear predictor for R_t can include autocorrelation terms. Currently, **epidemia** supports random walk terms. The random walk errors are given a zero-mean normal prior, with an unknown scale. This scale is itself assigned a half-normal hyperprior with a known scale.

Consider a very simple random walk parameterization of R_t, whereby `formula = R(country, date) ~ rw(prior_scale=0.05)` is used in the call to `epirt()`. Assuming only one population is being considered, this implies a functional form of

$$R_t = g^{-1}\left(\beta_0 + W_t\right)$$

for reproduction numbers. Here W_t is a random walk satisfying $W_t = W_{t-1} + \gamma_t$ for $t > 0$ and with initial condition $W_0 = 0$. Under the prior, the error terms γ_t follow $\gamma_t \sim \mathcal{N}(0, \sigma)$ with $\sigma \sim \mathcal{N}^+(0, 0.05)$.

E.1.6 Caveats

There are several important caveats to be aware of when using prior distributions in **epidemia**.

Covariate Centering

By default, covariates in the regressions for R_t and α_t are not centered automatically by **epidemia**. This can, however, be done by using `center = TRUE` in the call to `epirt()` and `epiobs()` respectively. It is important to note that if `center = TRUE`, the arguments `prior_intercept` and `prior` set the priors on the intercept and coefficients *after centering the covariates*.

Covariates are not centered automatically because often the intercept has an intuitive interpretation in the model. For example, if all covariates are zero at the beginning of the

epidemic, then the intercept can be seen as specifying the initial reproduction number R_0 of the disease. If `center = TRUE`, then the intercept no longer has an easily intuited interpretation.

Autoscaling

rstanarm's prior functions have an argument called `autoscale`. If `autoscale = TRUE`, then **epidemia** automatically adjusts the prior scale to account for the scale of the covariates. This only applies to priors on fixed effects, and not to the intercepts. **epidemia** rescales according to the following rules.

- If a predictor has only one unique value, no rescaling occurs.
- If it has two unique values, the original scale is divided by the range of the values.
- For more than two unique values, the original scale is divided by the standard deviation of the predictor.

If you are unsure whether rescaling has occurred, call `prior_summary` on a fitted model object. This gives details on the original priors specified, and the priors that were actually used after rescaling.

E.2 Partial Pooling in epidemia

We describe how to partially pool parameters underlying the reproduction numbers. This is done using a special operator in the formula passed to `epirt()`. If you have previously used any of the **lme4**, **nlmer**, **gamm4**, **glmer** or **rstanarm** packages then this syntax will be familiar.

A general R formula is written as `y ~ model`, where `y` is the response that is modeled as some function of the linear predictor which is symbolically represented by `model`. `model` is made up of a series of terms separated by `+`. In **epidemia**, as in many other packages, parameters can be partially pooled by using terms of the form `(expr | factor)`, where both `expr` and `factor` are R expressions. `expr` is a standard linear model (i.e. treated the same as `model`), and is parsed to produce a model matrix. The syntax `(expr | factor)` makes explicit that columns in this model matrix have separate effects for different levels of the factor variable.

Of course, separate effects can also be specified using the standard interaction operator `:`. This however corresponds to *no pooling*, in that parameters at different levels are given separate priors. The `|` operator, on the other hand, ensures that effects for different levels are given a common prior. This common prior itself has parameters which are given hyperpriors. This allows information to be shared between different levels of the factor. To be concrete, suppose that the model matrix parsed from `expr` has p columns, and that `factor` has L levels. The p-dimensional parameter vector for the l^{th} group can

be denoted by θ_l. In **epidemia**, this vector is modeled as multivariate normal with an unknown covariance matrix. Specifically,

$$\theta_l \sim N(0, \Sigma), \tag{E.4}$$

where the covariance Σ is given a prior. **epidemia** offers the same priors for covariance matrices as **rstanarm**; in particular the `decov()` and `lkj()` priors from **rstanarm** can be used. Note that Σ is not assumed diagonal, i.e. the effects within each level may be correlated.

If independence is desired for parameters in θ_l, we can simply replace `(expr | factor)` with `(expr || factor)`. This latter term effectively expands into p terms of the form `(expr_1 | factor)`, ..., `(expr_p | factor)`, where `expr_1` produces the first column of the model matrix given by `expr`, and so on. From the above discussion, the effects are independent across terms, and essentially Σ is replaced by p one-dimensional covariance matrices (i.e. variances).

E.2.1 Example Formulas.

The easiest way to become familiar with how the `|` operator works is to see a multitude of examples. Here, we give many examples, their interpretations, and where possible we compare the models to the no pooling and full pooling equivalents. For a comprehensive reference on mixed model formulas, please see Bates et al. ((2015)).

There are many possible ways to specify intercepts. Table E.1 demonstrates some of these, including fully pooled, partially pooled and unpooled. Effects may also be partially pooled. This is shown in Table E.2.

Table E.1 Different intercept specifications. The intercept often has an interpretation as setting R_0 in each region. The left hand side of each formula is assumed to take the form `R(region, date)`.

Formula R.H.S.	Interpretation
`1 + ...`	Full pooling, common intercept for all regions.
`region + ...`	Separate intercepts for each region, not pooled.
`(1 | region) + ...`	Separate intercepts for each region which are partially pooled.
`(1 | continent) + ...`	Separate intercepts based on a factor other than `region`, partially pooled.

The final example in Table E.2 shows that it is important to remember that to parse the term `(expr | factor)`, `epim()` first parses `expr` into a model matrix in the same way as functions like `lm()` and `glm()` parse models. In this case, the intercept term is implicit. Therefore, if this is to be avoided, we must explicitly use either `(0 + npi | region)` or `(-1 + npi | region)`.

Table E.2 Different covariate specifications. Here NPI refers to some non-pharamceutical intervention. The left hand side of each formula is assumed to take the form `R(region, date)`.

Formula R.H.S.	Interpretation
`1 + npi + ...`	Full pooling. Effect of NPI the same across all regions.
`1 + npi:region + ...`	No pooling. Separate effect in each region.
`1 + (0 + npi|region) + ...`	Partial pooling. Separate effects in each region.
`1 + (npi|region) + ...`	Right hand side expands to `1 + (1 + npi|region)`, and so both the intercept and effect are partially pooled.

Independent Effects

By default, the vector of partially pooled intercepts and slopes for each region are correlated. The `||` operator can be used to specify independence. For example, consider a formula of the form

```
R> R(region, date) ~ npi + (npi || region) + ...
```

The right hand side expands to `1 + npi + (1 | region) + (npi | region) + ...`. Separate intercepts and effects for each region which are partially pooled. The intercept and NPI effect are assumed independent within regions.

Nested Groupings

Often groupings that are nested. For example, suppose we wish to model an epidemic at quite a fine scale, say at the level of local districts. Often there will be little data for any given district, and so no pooling will give highly variable estimates of reproduction numbers. Nonetheless, pooling at a broad scale, say at the country level may hide region specific variations.

If we have another variable, say `county`, which denotes the county to which each district belongs, we can in theory use a formula of the form

```
R> R(district, date) ~ (1 | county / district) + ...
```

The right hand side expands to `(1 | county) + (1 | county:district)`. There is a county level intercept, which is partially pooled across different counties. There are also district intercepts which are partially pooled *within* each county.

E.3 Model Schematic

We provide schematics for different parts of the model introduced in Section 6.2. These are useful because they clarify how different model objects, including data and parameters, are related to one another.

Figures E.1 illustrates a complete observational model, and in particular details the model for multipliers α_t. Figure E.2 presents the basic infection model, and also shows the GLM-style model for reproduction numbers R_t. Finally Figure E.3 shows extensions of the basic infection model, including treating latent infections as parameters and including population adjustments.

All mathematical notation shown in the figures corresponds to that used in Section 6.2. Each node is outlined in a color corresponding to the type of object considered. These are interpreted as follows.

- **Grey**: A user provided object or quantity that is assumed to be known.
- **Green**: A model parameter that is, generally speaking, directly sampled. Occasionally **epidemia** will sample a transformation of this parameter for efficiency purposes.
- **Red**: A transformed parameter. This is a quantity that is a deterministic function of other model parameters.
- **Orange**: A quantity that is either a parameter or transformed parameter, depending on the context.
- **Blue**: An observation.

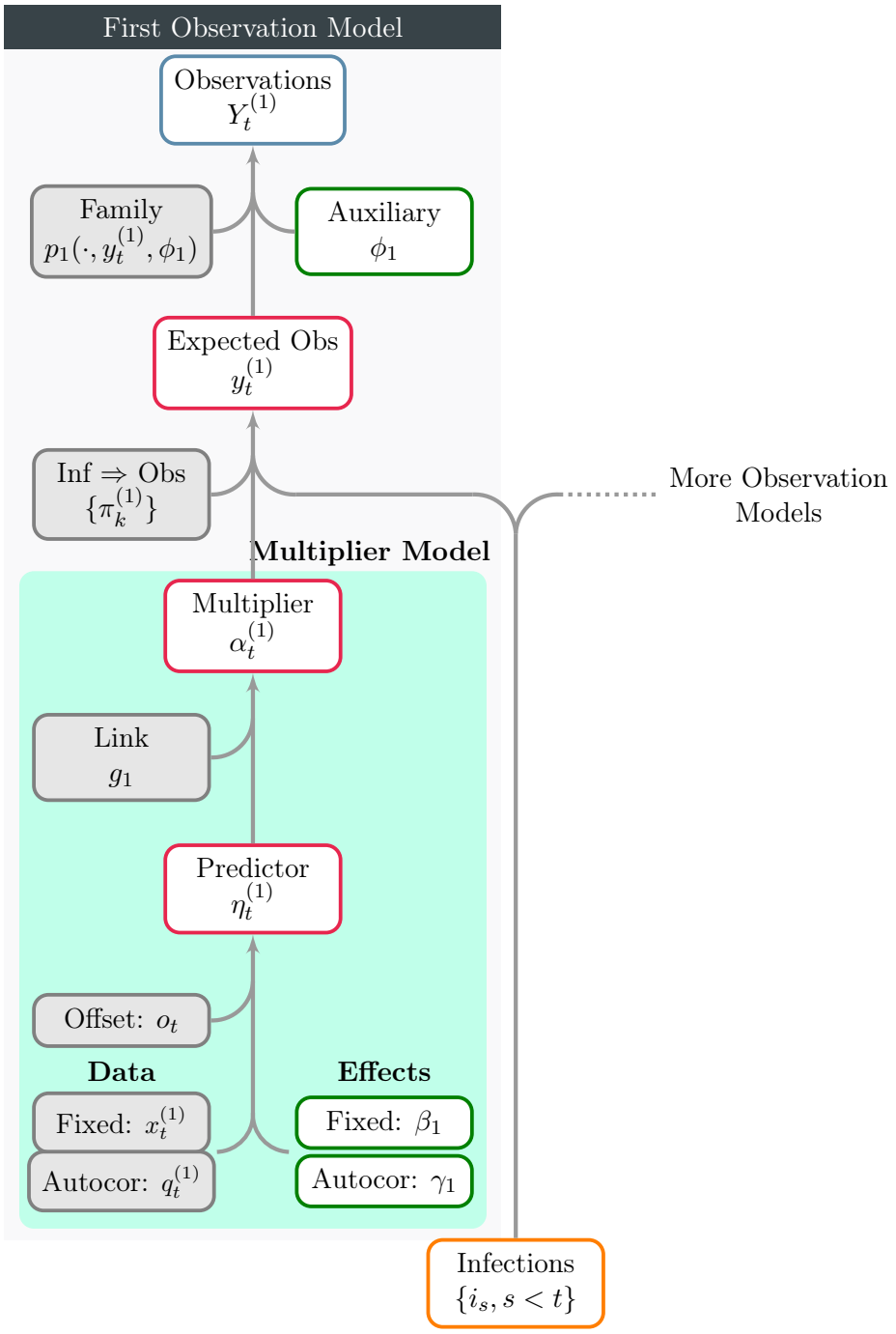


Fig. E.1 A schematic for observational models. Only one observational model is shown here, however the figure makes clear that additional models may be included. The model for the multiplier α_t is shown in the shaded green region. This is very similar in form to the transmission model shown in Figure E.2. Infections shown at the bottom may be directly from either the basic infection model, or from an extended model (as described in Section 6.2.

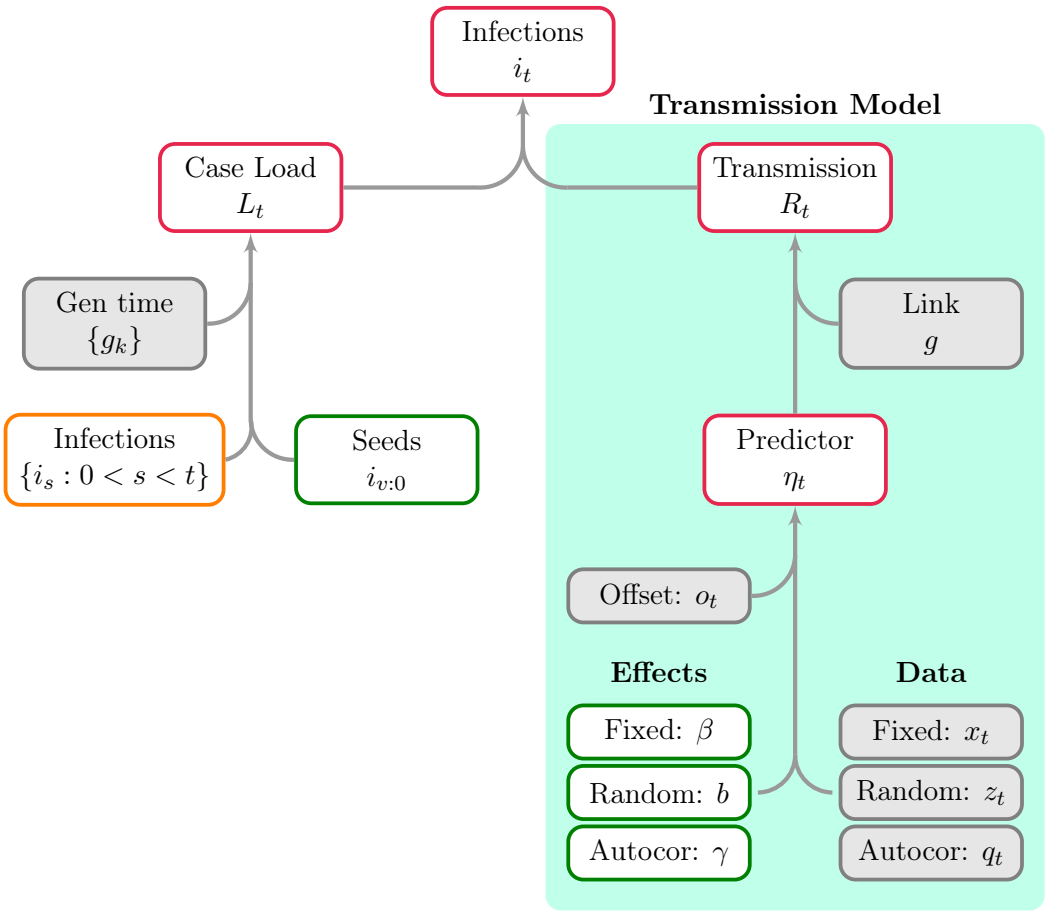


Fig. E.2 A schematic showing both the basic infection model and the transmission model (the green region). Here infections are a transformed parameter, and are recursively linked to previous infections. The model for R_t is similar to a GLM, however autocorrelation terms can be included. η_t is the predictor for the reproduction number at time t, and is one element of the predictor η introduced in Section 6.2

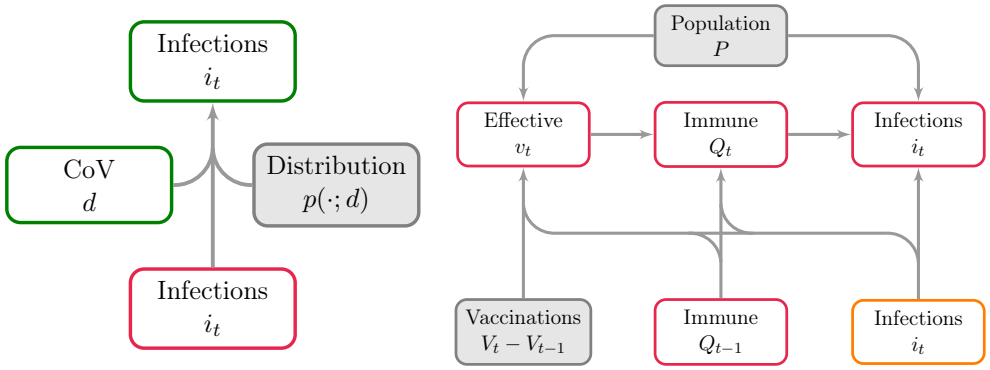


Fig. E.3 Possible extensions to the infection process. **Left** corresponds to the extension of Section 6.2.5, while **right** shows the extension of Section 5.4.2. The population adjustment, shown in the right figure, may be applied to either the infections shown at the bottom of the left figure (basic model), or those at the top of the left figure.