Bachelor Mathematics

Bachelor thesis

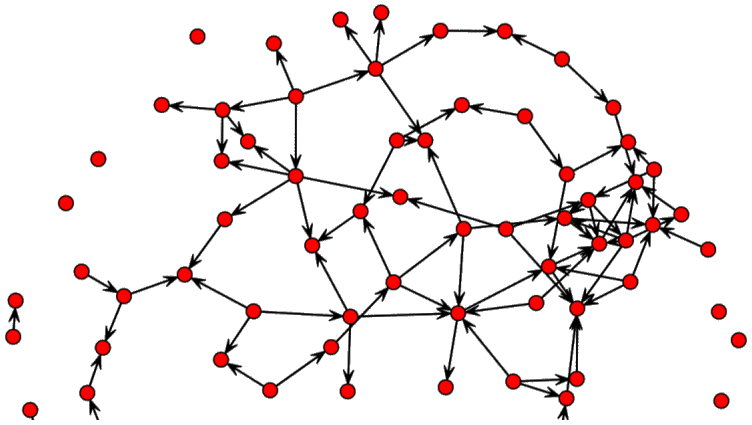
Bayesian Inference for Epidemic Transmission Across Contact Networks and its Application to Social Media Analysis

by

Danna Shao

May 13, 2024

Supervisor: prof.dr. Meulen, F.H. van der



Department of Mathematics
Faculty of Sciences

VU UNIVERSITY AMSTERDAM

Abstract

This project studies a set of statistical methods and analyzing tools created by Groendyke et al. that estimate the parameters of the spread of epidemics through social networks, and applies this method to another situation where information is spread through a social media platform. The transmission of disease or information is modelled as a stochastic susceptible-exposed-infectious-removed (SEIR) process, whereas the social network structure is described by dyadic independence exponential-family random graph models (ERGMs). The parameters of the model are estimated using Bayesian inference and Markov chain Monte Carlo (MCMC) simulation. The method is applied to analyzing a measles pandemic and a dissemination of a piece of help-seeking information across an internet social media platform. The analysis is performed using the R package `epinet`.

Title: Bayesian Inference for Epidemic Transmission Across Contact Networks and its Application to Social Media Analysis
Author: Danna Shao, d.shao@student.vu.nl, 2663369
Supervisor: prof.dr. Meulen, F.H. van der
Date: May 13, 2024

Department of Mathematics
VU University Amsterdam
de Boelelaan 1081, 1081 HV Amsterdam
`http://www.math.vu.nl/`

Contents

1. Introduction

Social media has gradually evolved into an important platform for people to share information as Internet technology has advanced. In contrast to traditional media, a large amount of content on social media is created and published by ordinary users rather than professional journalists. The information distribution of the latter is centralized, that is, the majority of audiences obtain information directly from those official platforms. On the contrary, the former, the spread of information is gradually amplified, which is often referred to as "going viral". As the phrase suggests, such information dissemination is quite comparable to the spread of a pandemic. We already have a wide variety of established techniques and models to analyze the behaviour of epidemics. Therefore, it is potentially feasible to apply such methods to social media analysis.

Compared with the spread of epidemics among the population, the spread of information on social media is more heterogeneous and clustered, since with greater autonomy to decide what they want to see, people on social media are more likely to only be exposed to the information they are interested in or are related to (such as policies, offline activities, natural disasters alarms, etc.) Meanwhile, for most social media sites, the relationships between users are always follower-following, and much of the information that a user can obtain is from the users they actively follow. As a result, the structure of the social network has a stronger impact on the spread of information than the spread of disease. The structure of such social networks is influenced by the similarity of the individuals, such as their geographical location, interests, gender, frequency of usage, and so on.

The set of infectious disease analysis methods developed by Groendyke et al. is ideally suited to analyze this scenario. This approach integrates the ERGM network model with the SEIR compartmental model of infectious diseases. The former assumes that the structure of social networks is controlled by attribute differences between individuals, such as gender or geographic distance, whereas the latter assumes that during a pandemic, each individual may go through the four stages including susceptible, exposure, infectious and removal. Under the Bayesian framework, the joint posterior distribution of the model parameters is estimated by Markov chain Monte Carlo (MCMC) simulation. They also implemented this method in the `R` package `epinet`, which is open source available on the Comprehensive `R` Archive Network (CRAN).

This project reviews the mathematical basis of this methodology, introduces and tests the `epinet` package, and reproduces the results of the 1861 Hagelloch Measles outbreak studied by Groendyke et al. using this approach. Based on this, social media platform data were collected and inferred by this method. The results show that this method can indeed provide meaningful results and new insights for the information dissemination process without prior biological or pathological facts rather than epidemics.

The remainder of this thesis is structured as follows: Chapter 2 introduces the mathematical models of ERGM and SEIR respectively. Chapter 3 describes the Bayesian inference process and the MCMC scheme for the model. Chapter 4 is the introduction and test of the `epinet` package and the result reproduction of the 1861 Hagelloch Measles. Chapter 5 introduces in detail how to apply this method to social media analysis and provides the results. Chapter 6 discusses the results and several issues encountered during the analysis, and gives the general conclusion of this research project.

2. Models and notation

2.1. Social network model

A time-invariant social network consisting of N individuals is modelled as an undirected graph $\mathcal{G}$ with N nodes in total. Each individual is seen as a node in $\mathcal{G}$. If two distinct individuals i, j have a connection in the social network, then $\mathcal{G}_{\{i,j\}} = 1$, otherwise $\mathcal{G}_{\{i,j\}} = 0$. Let $G = (\mathcal{G}_{\{1,2\}}, \mathcal{G}_{\{1,3\}}, ...)$ be the vector indicating the existence of edges in $\mathcal{G}$, then G is in $\mathbb{R}^{\binom{N}{2}}$.

Since actual social connections are not easy to be observed and recorded directly, the network structure is estimated based on a natural assumption that people having higher similarities, such as living closer or attending the same class, are more likely to be connected. Dyadic independent exponential family random graph (ERGM) is an effective tool to approximate the contact network under this rule. An ERGM is defined by its probability mass function $f(\mathcal{G}) = \exp(\boldsymbol{\eta}^\intercal G)/\kappa(\boldsymbol{\eta})$. $\boldsymbol{\eta} = \{\eta_1, \eta_2, ..., \eta_s\}$ is the parameter vector where η_i corresponds to the i-th network statistic, that is, the nodes of the graph has s different independent features that affect the probability of their connection state. $\kappa(\boldsymbol{\eta})$ is a normalizing function.

If the ERGM is dyadic independent, the probability of the existence of an edge between node i and node j is

$$p_{\{i,j\}} = \psi\left(\sum_{k=1}^{s} \mathbf{X}_{\{i,j\},k}\eta_k\right) \tag{2.1}$$

where $\psi(x)$ is the Sigmoid function $\psi(x) = (1 + e^{-x})^{-1}$ and $\mathbf{X}$ is a $\binom{N}{2} \times k$ matrix of dyadic covariates. Each row of $\mathbf{X}$ corresponds to each of the $\binom{N}{2}$ dyads (i.e. a pair of two nodes) and each column corresponds to each of the covariates (i.e. their relationship or similarity) in the model.

The probability mass function will then be

$$f(\mathcal{G}) = \prod_{\mathcal{G}_{\{i,j\}}} p_{\{i,j\}}^{\mathcal{G}_{\{i,j\}}}(1 - p_{\{i,j\}})^{1-\mathcal{G}_{\{i,j\}}} \propto \prod_{\mathcal{G}_{\{i,j\}}} \exp\Bigg(\mathcal{G}_{\{i,j\}} \underbrace{\log \frac{p_{\{i,j\}}}{1 - p_{\{i,j\}}}}_{\sum_{k=1}^{s} \mathbf{X}_{\{i,j\},k}\eta_k}\Bigg) \tag{2.2}$$

The advantage of this subclass of ERGM is that it is easy to be understood, interpreted and computed [2]. Each coefficient can be interpreted as the incremental log odds of the probability of an edge regarding its effect, and the dyadic-independence property makes it possible to implement the MCMC algorithm through the dyads. However, this property also brings the major shortcoming of this model. In a real-world social

network, friends of friends are more likely to be friends and thus is not dyadic independent. Nevertheless, taking such dyadic-dependent attributes into account will make the normalization constant $\kappa(\eta)$ unmanageable as the formula for calculating the exist probability of an edge will be different on each network statistic. It will also leads to difficulties in model inference.

2.2. Transmission model

The SEIR compartmental model is one of the most frequently used mathematical models for infectious diseases. In this model, individuals are separated into four groups: susceptible, exposed, infectious and removed according to their current conditions. Susceptible individuals are those who have not been infected yet but may be infected in the future. After an infectious contact, a susceptible individual will be moved to the exposed group where they are already infected but not yet infectious. After a latency period, the individuals move to the infectious group. Finally, they step into the removed stage when they are no longer infectious nor susceptible to the disease. This process is unidirectional.

There are many variants of the model describing different situations. For example, the SIS model assumes that the infectious individuals gain no immunity and would become susceptible to the disease again once recovered. This can be a proper way to model the transmission of common cold or influenza. In this project, re-infection corresponds to an individual broadcasting the same information twice and is very unlikely to happen. Therefore, individuals will be moved to the removed group eventually once infected.

Let $\mathbf{T} = (\mathbf{E}, \mathbf{I}, \mathbf{R})$, where $\mathbf{E} = (E_1, ..., E_N)$, $\mathbf{I} = (I_1, ..., I_N)$, $\mathbf{R} = (R_1, ...R_N)$ be the set that records the time when each of the N individuals arrives the corresponding stages. For convenience, denote m as the number of individuals that have been infected ultimately. If an individual is not infected throughout the whole outbreak, these values are assigned ∞. To model those time periods, the waiting time for a transmission across a particular edge is modelled by an exponential random variable $Exp(\beta)$. The time spent in the exposed and infectious stages is modelled as gamma random variables considering the flexibility of the model: $E \sim Gamma(k_E, \theta_E)$ and $I \sim Gamma(k_I, \theta_I)$. Note that in this project, all of the notation of Gamma distribution $Gamma(k, \theta)$ is using the characterization with shape k and scale θ (i.e. the mean is $k\theta$ and the variance is $k\theta^2$).

To apply this transmission model to a social network structure, the epidemic can be considered as a directed subgraph of $\mathcal{G}$ which includes all nodes that are infected during the epidemic, and its edges are where transmission occurred. As the process is unidirectional, this subgraph would be a tree. Denote the transmission tree as $\mathcal{P}$ and its root as r who is the initial infected individual. Together with $\mathbf{T}$, we can now represent the complete process of an epidemic spreading over a fixed social network. A mini demonstration of this process is shown in Figure B.1.

3. Bayesian Inference and the MCMC scheme

3.1. Likelihood calculation and prior distribution selection

As explained in detail by Groendyke et al. (2011) [1], the likelihood function of the above-described model can be obtained by summing over all possible $\mathcal{G}$ and $\mathcal{P}$, yet that requires excessive computation. Conditioning on given $\mathcal{G}$ and $\mathcal{P}$, the likelihood function will be [3]

$$f(\mathbf{T} \mid \beta, k_E, \theta_E, k_I, \theta_I, \boldsymbol{\eta}) = \sum_{\mathcal{G}} \sum_{\mathcal{P}} f\left(\mathbf{T} \mid \beta, k_E, \theta_E, k_I, \theta_I, \boldsymbol{\eta}, \mathcal{G}, \mathcal{P}\right) f(\mathcal{P} \mid \mathcal{G}) f(\mathcal{G} \mid \boldsymbol{\eta}) \tag{3.1}$$

Since the likelihood depends on $\boldsymbol{\eta}$ only through $\mathcal{G}$, the first term of the above function can be written as $L\left(\mathbf{T} \mid \beta, k_E, \theta_E, k_I, \theta_I, \mathcal{G}, \mathcal{P}\right)$, and four parts contribute to this term. Write L_1 as the contribution from contacts on the edges that the epidemic transmitted through ($\mathcal{P}$), L_2 as those on edges without transmission ($\mathcal{G} \setminus \mathcal{P}$). L_3 and L_4 are the contributions from the transition processes from exposed to infectious and from infectious to removed respectively. The likelihood can be calculated by multiplying the assumed distributions $L_1 L_2 L_3 L_4$.

From the model assumption we have

$$L_1 = \beta^{m-1} \exp\left[-\beta \sum_{(a,b)\in\mathcal{P}} (E_b - I_a)\right],$$

$$L_2 = \exp\left[-\beta \sum_{(a,b)\in\mathcal{G}\setminus\mathcal{P}} [\{(E_b \wedge R_a) - I_a\} \vee 0]\right].$$

Therefore $L_1 L_2 = \beta^{m-1} \exp[-\beta A]$, where

$$A = \sum_{(a,b)\in\mathcal{P}} (E_b - I_a) + \sum_{(a,b)\in\mathcal{G}\setminus\mathcal{P}} [\{(E_b \wedge R_a) - I_a\} \vee 0] \tag{3.2}$$

A is the total time spent by all infectious contact, that is, the contact between a pair of infectious and susceptible individuals.

$$L_3 = \left[\prod_{i=1}^{m} (I_i - E_i)\right]^{k_E - 1} \theta_E^{-mk_E} \mathrm{e}^{-B/\theta_E} \Big/ \left[\Gamma\left(k_E\right)\right]^m,$$
$$L_4 = \left[\prod_{i=1}^{m} (R_i - I_i)\right]^{k_I - 1} \theta_I^{-mk_I} \mathrm{e}^{-Cl\theta_I} \Big/ \left[\Gamma\left(k_I\right)\right]^m.$$

where

$$B = \sum_{i=1}^{m} (I_i - E_i), \quad C = \sum_{i=1}^{m} (R_i - I_i) \tag{3.3}$$

are the total time all individuals have spent in the exposed and infectious states respectively.

The remaining term $f(\mathcal{P} \mid \mathcal{G})$ is simply the uniform distribution assigning equal probability to every possible $\mathcal{P}$ with given $\mathcal{G}$. Meanwhile, with the dyadic independent property stated in Section 2.1, $f(\mathcal{G} \mid \boldsymbol{\eta})$ can be directly calculated by multiplying all probabilities of the existence of an edge, which is $p_{i,j}^{\mathcal{G}_{\{i,j\}}}(1 - p_{i,j})^{1-\mathcal{G}_{\{i,j\}}}$, where $p_{i,j}$ is calculated with Equation (2.1).

The choice of prior distributions for the parameters depends on several factors. Conjugate priors are always more preferable regarding computational efficiency [1]. For this particular model, one conjugate prior choice for parameters θ_E and θ_I is the inverse Gamma distribution (denoted with IG), and the Gamma distribution for parameter β. Gamma or uniform distribution can be used for k_E and k_I. When inference for certain parameters is necessary, uninformative (uniform) priors are assigned.

The parameter choices for those prior distributions are normally based on the pathological facts of a certain type of disease. However, in the present project, there is no reference to how the transmission of the social media posts behaves.

3.2. MCMC scheme

The MCMC algorithm is used to produce samples from the posterior distributions of the parameters $\{\beta, k_E, \theta_E, k_I, \theta_I, \mathbf{I}, \mathbf{E}, \mathcal{P}, \mathcal{G}, \boldsymbol{\eta}\}$. Each parameter is updated on each iteration in turn.

Parameter $k_E, k_I, \beta, \theta_E, \theta_I$ are updated using a standard Metropolis–Hastings step. Assigning the previously described priors $\pi_\beta(\beta) \sim \text{Gamma}\,(a_\beta, b_\beta)$, $\pi_{\theta_I}(\theta_I) \sim \text{IG}\,(a_I, b_I)$, $\pi_{\theta_E}(\theta_E) \sim \text{IG}\,(a_E, b_E)$, the corresponding full conditional distributions of these parameters calculated via likelihood are:

$$\pi_\beta(\beta \mid \mathbf{T}) \sim \text{Gamma}\left(m + a_\beta - 1, \frac{1}{A + 1/b_\beta}\right),$$
$$\pi_{\theta_E}(\theta_E \mid \mathbf{T}) \sim \text{IG}\left(mk_E + a_E, \frac{1}{B + 1/b_E}\right),$$
$$\pi_{\theta_I}(\theta_I \mid \mathbf{T}) \sim \text{IG}\left(mk_I + a_I, \frac{1}{C + 1/b_I}\right).$$

with A, B, C defined by Equation (3.2) and (3.3).

Updating the transmission tree $\mathcal{P}$ is to locate the parent of each infected node (i.e. the node infected them). Write $\mathcal{P}_j$ as the parent node of node j, $\pi_{\mathcal{P}_j}(r)$ as the prior for node r being the parent node of node j. The candidate parent nodes for node j are all the nodes i that are connected with j and are infectious when j gets infected. That is, $\mathcal{G}_{\{i,j\}} = 1$ and $I_i \leq E_j \leq R_i$. Denote those nodes by $i_1, \ldots, i_k$, then the probability of a specific candidate node i_p being the parent node of j would simply be $\pi_{\mathcal{P}_j}(i_p) \sum_{n=1}^{k} / \pi_{\mathcal{P}_j}(i_n)$. That is to say, the probability only depends on prior assumptions. When no extra information is presented, the prior is selected as a uniform distribution, indicating that every candidate node has the same probability of being the parent node. A sample for the full conditional distribution of $\mathcal{P}$ will be generated by repeating the process on all the infected nodes without the root node.

Parameter $\boldsymbol{\eta}$ is updated by a Metropolis–Hastings step that the next value of $\boldsymbol{\eta}$ is sampled from a multivariate normal distribution centered at the current $\boldsymbol{\eta}$. G is updated by cycling through each of the independent dyads [3].

4. Implementation and tests

4.1. The R pacakge epinet

All the above-described algorithms have been implemented by Groendyke and Welch in a publicly available R package epinet [4]. The package provides functions for both generation of a simulated transmission on a fixed social network, and inference given epidemic data using the MCMC algorithm stated above. The latter is implemented in C and interfaced in R through the epinet() function, therefore is hard to modify. In this study, the major functions used in this package are:

- BuildX: converts a nodal covariate matrix into a dyadic covariate matrix.
- epinet: uses epidemic data to perform Bayesian inference on a given contact network.
- SimulateDyadicLinearERGM: uses a given dyadic covariate matrix to generate a random ERGM network.
- SEIR.simulator: simulates an epidemic on a given undirected contact network.

The complete description of all the functions in the package is stated in its official documentation.

4.2. Simulated example

This example is reproduced from Groendyke et al. (2018) [2] using the R code in appendix A.1. With a population of 50 individuals with their corresponding node ID from 1 to 50, consider only two covariates, the overall edge density (a constant covariate of 1) and the Euclidean distance between each pair of individuals. Each node is assigned a random two-dimensional coordinate. Taking the baseline log-odds probability of an edge to be 0 and the incremental log-odds of an edge to decrease by 7 for each unit of distance between the nodes in a dyad, a random ERGM network can then be generated.

An SEIR epidemic can be simulated across this network. In this example, we assume that the length of time spent in the exposed state follows a gamma distribution. The parameters used for this simulation are $\beta = 1, k_I = 3, \theta_I = 7, k_E = 3, \theta_E = 7$. The MCMC procedure is unnecessary to perform in this example since all the values of exposure and infectious times are known.

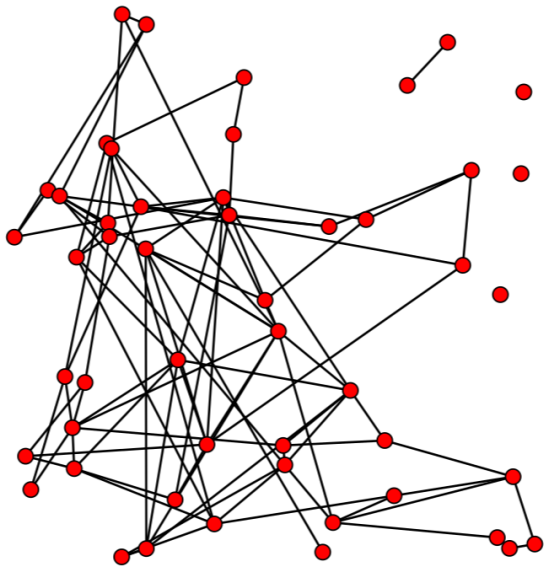


Figure 4.1.: The generated dyadic-independent ERGM network representing the social connection between 50 individuals considering their positions plotted by library `network`. All of the nodes are drawn at their actual assigned coordinates.

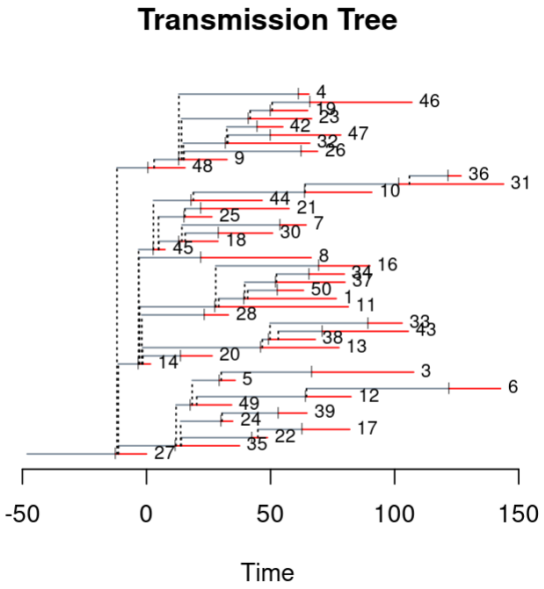


Figure 4.2.: Simulated pandemic across this network plotted with `plot` method of the `epidemic` class. Vertical dashed lines show the transmission paths. Each horizontal line represents the total infected time interval (grey for the exposed and red for the infectious, separated by a small vertical bar) for each corresponding individual. Time $t = 0$ is set as the first removal observation.

4.3. Result reproduction of the 1861 Hagelloch Measles

In this section, we reproduce the result from Groendyke et al. (2012) [3] analyzing the 1861 Hagelloch Measles outbreak in order to demonstrate the complete procedure of the analysis and to compare the results with the social media news transmission in the following chapters as well.

A measles outbreak infected 188 children in Hagelloch, Germany in the year 1861. The town doctor, Pfeilsticker, made a detailed record of the information of all the infected children including their personal features such as their living locations, classrooms they belong to, age, gender, etc. as well as their clinical records such as the date when they showed symptoms. This dataset is widely used for testing new methods for its completeness and small scale. The complete dataset is also included in the package `epinet`. It can be accessed using R command `View(HagellochTimes)` for the transmission time

matrix and `View(HagellochDyadCov)` for the corresponding dyadic covariate matrix.

The assigned priors are: $\beta \sim Uniform(0, 4)$, $\pi_{k_E} \sim Uniform(8, 20)$, $\pi_{\theta_E} \sim Uniform(0.25, 1)$, $\pi_{k_I} \sim Uniform(15, 25)$, $\pi_{\theta_I} \sim Uniform(0.25, 0.75)$. The flat, uninformative uniform prior is assigned since the exposure and infectious times are necessary to infer, and the value of the hyperparameters are based on the virological fact of measles. For the network parameter $\boldsymbol{\eta}$, each η_i were assigned independent normal $N(0, 3)$ prior.

Other input arguments are: parameters for `MCMCcontrol` to control the number of iterations, burn-in, thinning, etc.; parameters for the main function `epinet` including a formula deciding which parameters to utilize and how to do that. For example, in the further research of this dataset, this formula is set as `~‘Classroom 1‘ + ‘Classroom 2‘ + ‘House Distance‘` since significant classroom and household effects had been found. Here we are just using all parameters to reproduce the outcome. An example of the complete input to run the algorithm is stated in appendix A.2.

After the MCMC program completes, a convergence test will be performed. This can be done by calculating the Geweke diagnostic statistics for the model parameters and inspecting the trace plot.

```
Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

     (Intercept)     ‘Classroom 1‘     ‘Classroom 2‘ ‘House Distance‘
        -0.52627          -0.05657           0.44814           0.61257
```

The test statistics for other epidemic parameters range from -0.3713 to 0.6372. These values are not large enough to question the convergence. The trace plot also shows that the chain should have converged.

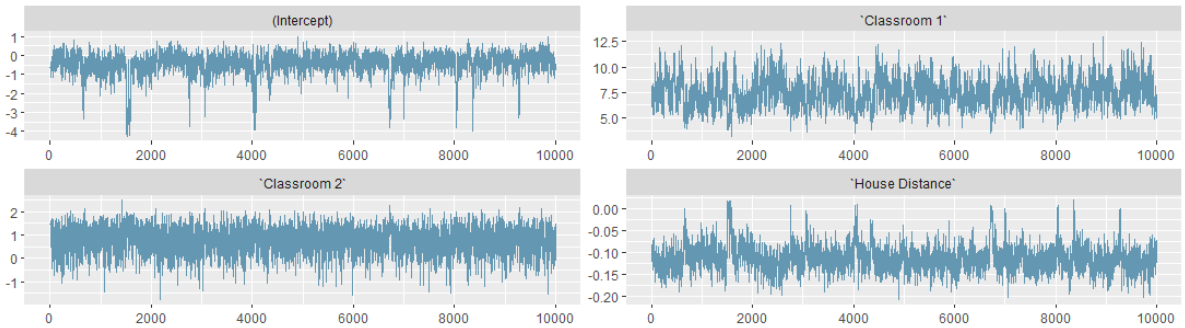


Figure 4.3.: Trace plot for the parameter $\boldsymbol{\eta}$. The Markov chain has a length of 10,000,000 and was thinned every 1000 iterations. The first 2,000,000 samples were considered as burn-in.

We can then look into the sampled posterior distribution of the parameters, shown in Figure 4.4. All those results are consistent with what was discovered by Groendyke et al., and the biological facts of the measles disease.

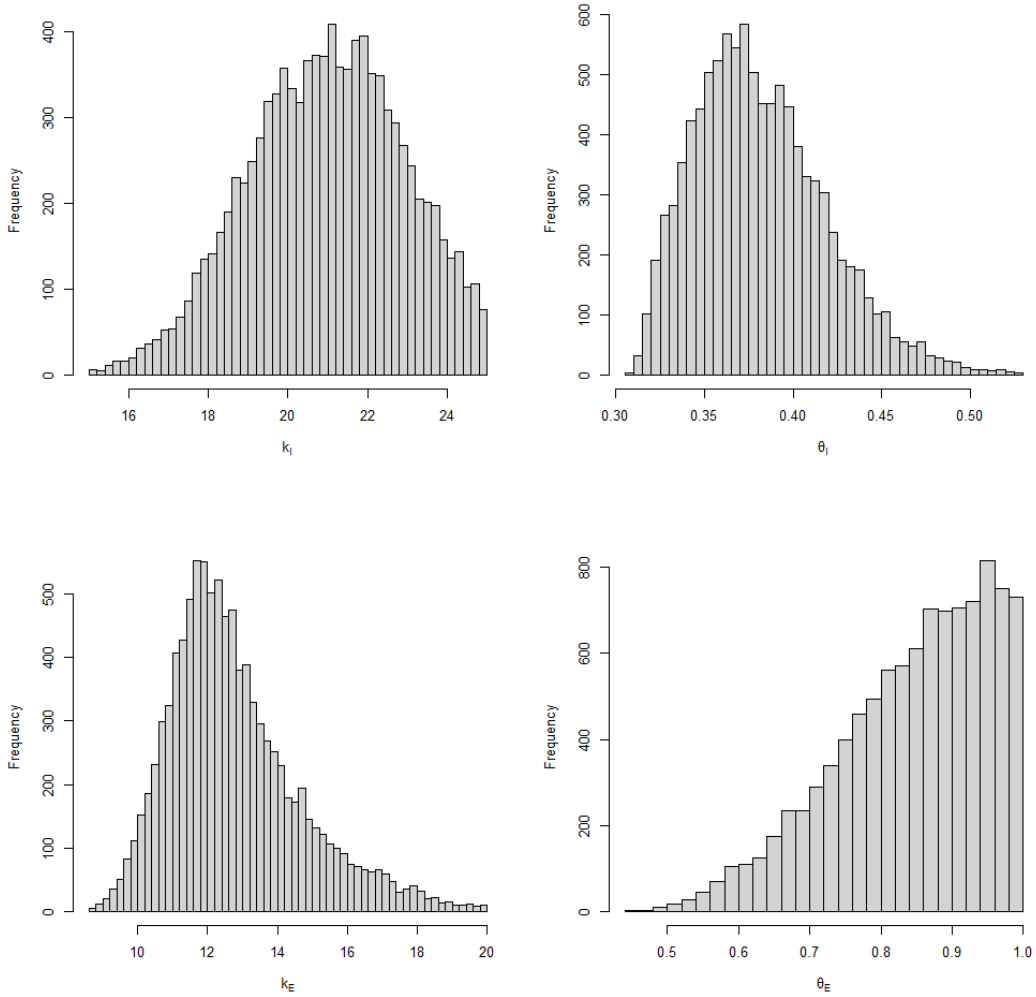


Figure 4.4.: Estimated posterior densities for epidemic parameters k_I, θ_I, k_E, and θ_E with previously described priors for the 1861 Hagelloch measles data.

5. Inferring the information dissemination behavior on a social media platform

5.1. Dataset description

To perform the analysis stated above using the `epinet` package, the social media data needs to be separated into two segments: a covariate matrix representing the relationship or similarity between two nodes; and a time-series matrix $\mathbf{T} = (\mathbf{E}, \mathbf{I}, \mathbf{R})$. The concrete meaning of T_i will be explained in the following section.

The desired data is gathered using a web crawler from a Chinese social media platform, *Weibo*. The reason for this platform selection is that it preserves the complete content forward (repost) path, which can be seen as a transmission tree with the forward time of the post by a user corresponding to the time when an individual becomes infectious. For example, the platform will store the posts in the form "C:repost//@B:repost//@A:original" if C reposts B's repost from A, which is directly the transmission path of user C. On the contrary, Twitter and most of the other mainstream platforms will not show representations of the intermediary reposts when it comes to multi-level retweeting.

To better fit the model, a help-seeking message posted by ordinary users rather than a verified news platform with a huge amount of followers is selected because the former behaves more similarly to the decentralized human-to-human transmission during an epidemic. The chosen post was about a cat trapped in the house and need feeding during the Covid-19 lockdown. The complete data set consists of 105 nodes.

5.1.1. Social network data for ERGM model

This 105 rows $\times$ 9 columns matrix consists of user attributes, including their sex, city, number of posts and followers/followings, user level and check-mark type. The following table is a part of the covariate data.

NodeID	sex	location	posts	followers	followings	level	vip_level	checkmark
1	0	0	633	8	149	0	0	-1
2	1	1	7430	580	1305	37	1	220
3	1	2	34	10	276	4	0	-1
...	...	...	...	...	...	...	...	...
104	1	38	5389	207	694	29	1	-1
105	1	20	3008	93	511	14	1	-1

Here, each distinct location (it could be a district, city or country) is assigned a categorical value. A dyadic covariate matrix could be generated from this matrix.

5.1.2. Transmission time data for SEIR model

Setting the user who first posted cat rescue post as the root node and the timestamp of the post as 0, the web crawler scrapes all forwards. From the scarped data, a transmission tree can be constructed. The time that a user i had forwarded the post is considered as the infectious time I_i in the SEIR model, since that is the time when other users could see and repost it again. All of the removal times are estimated since there is no certain data that can show this directly. Since the transmission is over as the information is spread 2 years ago and is very unlikely to have new forwards, it is reasonable to assume such time exists as the definition of removal is when an infected individual becomes no longer susceptible nor infectious anymore. Therefore, for all the parent nodes, their removal time R_i is assumed to be the infectious time I_j of the lastly infected user j among all of the successors they have, as they no longer infect other nodes after that. However, there are still many nodes with no successors. Nevertheless, the known data of time intervals from the infected to the removal state can already fit an exponential distribution that can be used to generate random values for those nodes. The following table is a part of the 105 rows $\times$ 5 columns transmission data matrix and the unit of time is minutes.

NodeID	Parent	Etime	Itime	estRtime
1	Na	Na	0	1584
2	1	Na	6	90
3	1	Na	9	73
...	...	...	...	...
104	1	Na	1583	2816
105	104	Na	2815	2841

5.1.3. Dealing with missing values

There are missing values due to the fact that some accounts no longer existed when the data were extracted. That may be caused by the users deleting their accounts themselves or being banned from the platform. Such accounts' existences are confirmed through the forward path, but all the attribute parameters and their exact forwarding times are lost. The Missing node IDs are: 9, 17, 33, 48, 63, 100. To solve this, for the social network data, categorical values are assigned `sex=2, location=40` (there are 38 distinct known locations) and other numerical parameters are filled with their medians. For the transmission time data, `Itime` is randomly assigned a reasonable value (that is, a uniform random value which is no later than the `estRtime` of its parent node and the `Itime` of all its child nodes, and no earlier than the `Itime` of its parent node), and `estRtime` is estimated by the posting time of their parent and child nodes.

The reason why those data needed to be filled is that firstly the `R` package requires all attribute data from every node, otherwise it is not able to generate an ERGM network,

so the first matrix must be completed. Secondly, there is a bug in the package that wrongly treats all the following nodes of the first node whose `Rtime` is missing. If an unknown `Rtime` data entry (`Na`) appears in the middle, it will wrongly treat all the other known data entries following by as susceptibles, that is, assuming they are never infected during the entire outbreak even if their `Itime` and `Rtime` are provided. Details of which piece of code is causing this bug are further analyzed in appendix C, yet fixing it is hard and out of the scope of this project, therefore the above method is used to avoid this issue.

5.2. Assumptions, priors and input

5.2.1. Prior for the transmission parameter $\beta, k_I, \theta_I, k_E, \theta_E$

For this particular situation where all those parameters need to be inferred, as described in the previous sections, uninformative (i.e. uniform) priors are assigned. Unlike the previous example, there is no biological data for reference. Therefore, a series of different hyperparameters are tested to obtain a proper prior range, under which the shape of the posterior sample is not truncated and the range is also not too large. The priors thus determined is $\beta \sim Uniform(0, 1)$, $\pi_{k_E} \sim Uniform(0, 20)$, $\pi_{\theta_E} \sim Uniform(0, 4500)$, $\pi_{k_I} \sim Uniform(0, 10)$, $\pi_{\theta_I} \sim Uniform(0, 300)$.

5.2.2. prior for the network parameter η

Parameter $\boldsymbol{\eta}$ controls how the ERGM social network is constructed. Despite it seems natural to obtain the network directly from the social media platform by the follower-following relationship of the users, there are several reasons why the network is estimated. First, much of the information is spread over topics rather than users on this public platform. That is, users can visit "news square" to view the posts under certain topics marked by hashtags (for example, #BLM) without following the users who have posted them. Therefore, it makes more sense to simulate their social connection by their similarities since it grants a higher possibility for them to engage in the same range of topics. The second reason is the practical issue. The follower list of this social media platform is not fully publicly accessible.

As no scientific or empirical references for this scenario exists, the $\boldsymbol{\eta}$ parameter is estimated manually by visualizing different networks generated by different η_i values and see if it looks similar to the real-world connection. This might be considered an intuitive step, as the criterion is only that the network looks reasonable regarding the data. The key parameter controlling the shape of the random graph is `follower_count`, which is reasonable since the more followers an account has the more connection it might gain. The estimated $\boldsymbol{\eta}$ has mean value $(0, -0.02, -0.5, -0.001, -0.0003, -0.001, -0.005, -0.002, -0.001)$ in the order of the covariate data set described in section 5.1.1.

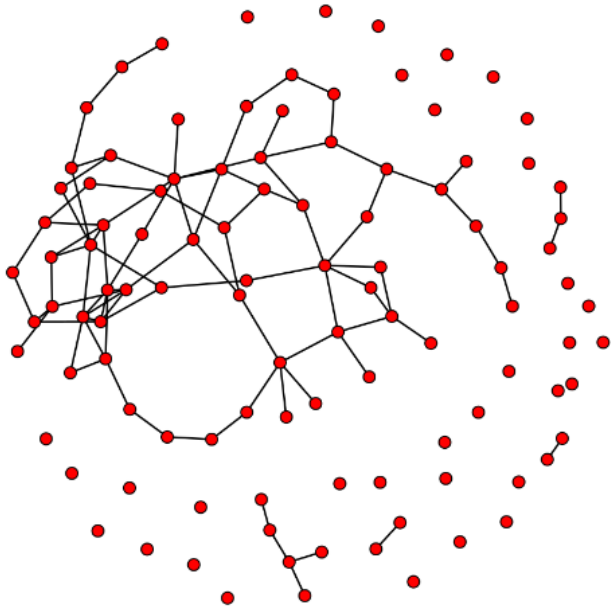


Figure 5.1.: The visualization of the social network under $\boldsymbol{\eta}_e$. According to the social media data, there are about half of the users who reposted the message do not have a direct follow relationship with their parent node. Those users are considered isolated from other users in this social network. Meanwhile, a few users have many followers, resulting in many connections. The rest of the reposts are in a chained manner, that is, the message is passed from one to another. The shown network satisfies these features.

5.2.3. Model selection

As mentioned above, not every attribute of the users is taken into consideration. This is also the case for the Hagelloch Measles research, where only three attributes were used in the utilize formula as presented in section 4.3. Groendyke et al. (2012) [3] used several techniques such as Reverse jump MCMC algorithm to confirm that this model outperforms all other models. As those methods are out of the scope of the present project, the model selection is based on some natural assumptions. That is, among the nine attributes collected in the dataset, considering the content of this post (cat rescue information on a very specific location), the selected three attributes are location matching, the absolute difference of post, and follower number where the underlying assumptions are: the more followers a user has, the more likely that other users see the information from them; the more post a user have posted implies that this user browse the social media more often thus is also more likely to spread the information. In other words, the calling formula for the `epinet` function will be `~location.match` + `follower_count.diff` + `post_count.diff`. Together with $\boldsymbol{\eta}$ prior, the actual input would be `etaprior = c(0, 1, -0.5, 1, -0.0003, 1, -0.001, 1))`. That is, mean -0.5, -0.0003, -0.001 and variance 1. The mean is from section 5.2.2 and a more diffuse prior (i.e. larger variance) did not significantly alter the posteriors.

5.3. Results

The algorithm was run 10,000,000 iterations and was thinned every 1000 iterations. The complete code for generating this result can be found at appendix A.3.

5.3.1. Convergence test

The Geweke diagnostic statistics are as follows:

```
Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

(Intercept)      location.match follower_count.diff      post_count.diff
     4.3804              1.6989             -0.6442              -0.2487
```

and that of the epidemic parameters range from -0.942 to 0.841. The corresponding trace plots are shown in appendix B. However, from the trace plots, it is doubtful whether the location matching η parameter has really converged. This might be caused by that there are too many different locations (39 in total) and the `.match` method does not work well for such a condition.

5.3.2. Posterior samples of the parameters

The sampled posterior distributions of the parameters are shown in Figure 5.2. The estimated mean of the exposure duration ($k_E \times \theta_E$) is around 410 minutes and that of the infectious duration ($k_I \times \theta_I$) is around 51 minutes.

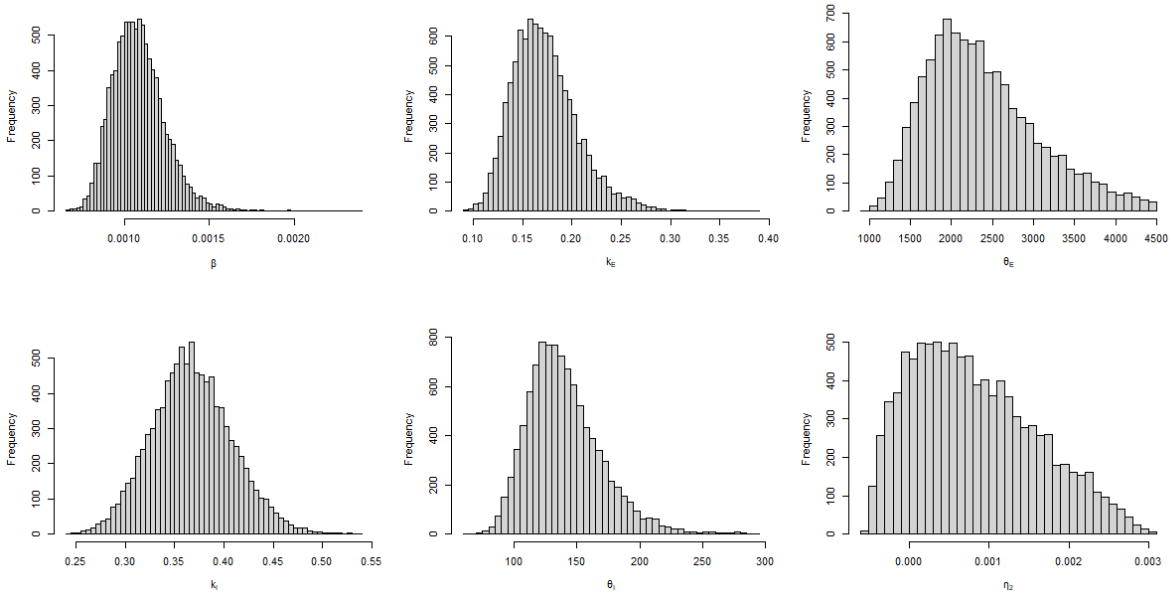


Figure 5.2.: Estimated posterior densities for epidemic parameters β, k_I, θ_I, k_E, θ_E and η_2 where η_2 represents the absolute difference of follower number.

5.3.3. The transmission tree

The transmission tree provides an intuitive perception of the spreading process. The transmission tree is updated in each iteration of the MCMC algorithm just as other parameters. The following figure is one such example. From the figure we can see that the reposting is highly clustered, that is, there are few parent nodes and those nodes have many successors.

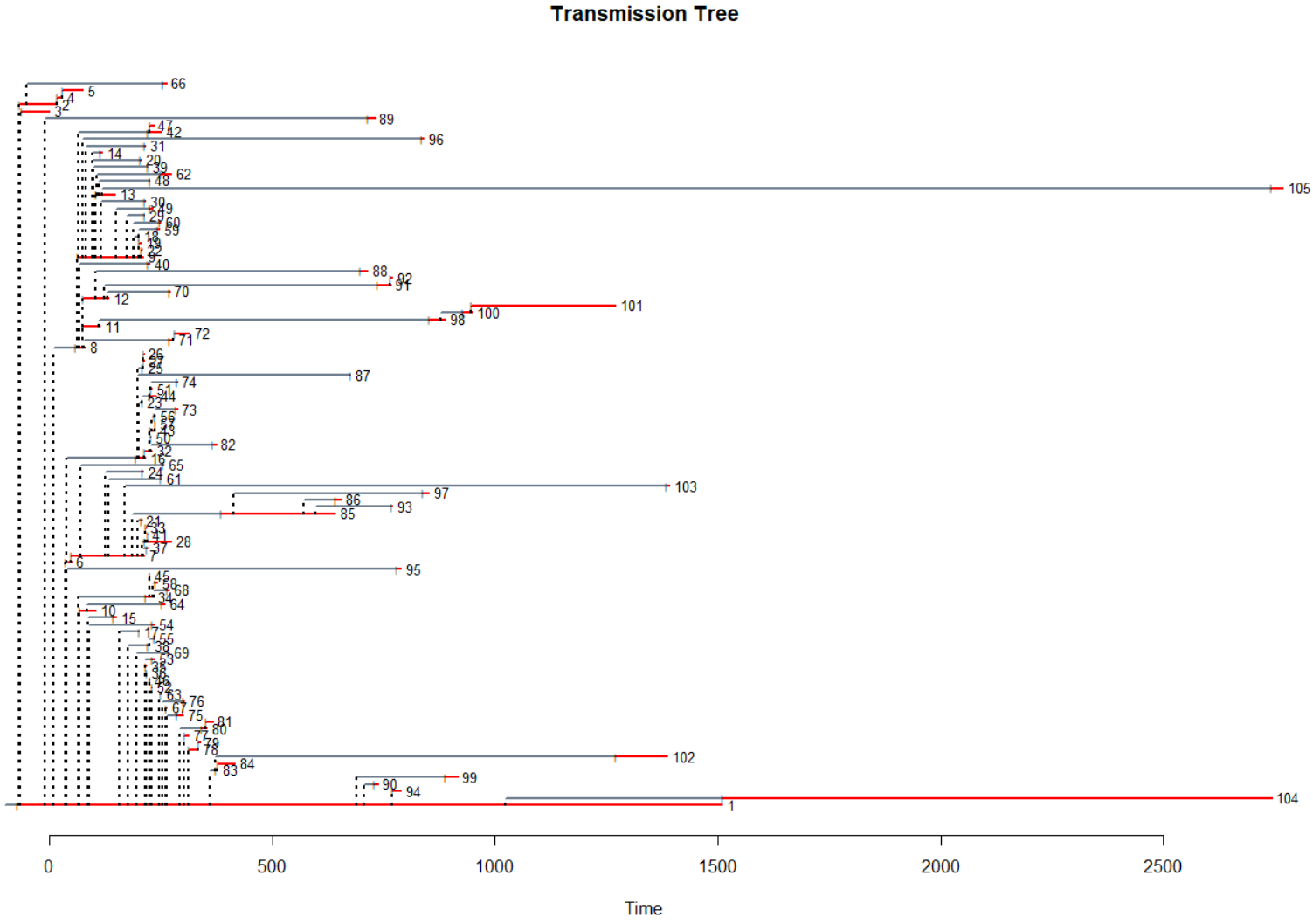


Figure 5.3.: An example of the (truncated) sampled transmission tree. The exposure time of the root node, which is a huge negative number, was cut off from the figure in order to show the detail of the transmission across other nodes. The full untruncated figure is shown in Figure B.3. The meanings of the lines and the numbers in this figure are the same as Figure 4.2. It is important to emphasize that this figure is only a single sample from the chain. It does not represent the posterior distribution of the transmission tree, and thus should not be over-interpreted.

6. Discussion and conclusion

6.1. Interpretations of the social media results

We are interested in the parameters for the exposure and infectious time. Further interpreting the model in the context of information dissemination on social media platforms, the exposure time can be seen as the consideration or hesitation time before a user presses the forward button and broadcasts the message to someone else. There are many possible causes that can extend this time period. For example, the content may be long or sophisticated and reading it can be time-consuming; the users may question the credibility of the content and spend some time doing a fact check; or they may worry if reposting the message is suitable as some of their followers may dislike it.

The distribution of the exposure time of the spread of a post among a certain group of people can therefore be considered the property of the content and the social network. For instance, if some highly unrealistic conspiracy theory content has a short exposure time in a user group, then we should perhaps take some targeted measures to prevent further radicalization of this group.

Meanwhile, the infectious time can be regarded as the lifespan of the content i.e. the duration from posted to outdated of the information. This is more likely to be controlled by the user group and the algorithm of the social media platform. If the users are posting more frequently, then a particular post will soon be supplanted by other contents.

Unfortunately, the above results show that the current model might not be a good representation of social media content if we do not assign a smaller prior for the θ_E parameter, which will result in the estimated posterior distribution being truncated. This is because the exposure duration should not be too long in reality, as users are very unlikely to repost something after hours of hesitation. Large hyperparameter for π_{θ_E} also causes an unexpected issue in the transmission tree, which will be discussed in the next section.

6.2. Abnormal behaviour for the exposure time of the root node

As described in section 5.3.3, the demonstrated transmission tree is not complete. The actual transmission tree is shown in Figure B.3. From the figure, we can see that the root node has an unreasonably long exposure time. Since the root node is the original poster in the context of social media, this exposure time should not be long.

This value seems to be controlled directly by the prior of θ_E, as shown in the following

Figure 6.1. The larger the range of this prior, the longer this initial exposure value would be. However, large hyperparameters for the prior are required for the posterior sample of θ_E is not truncated ($Uniform(0, 4500)$ in this case). Observing the output, one will observe that this value is stored in vector `initexp` and will only vary if π_{θ_E} is not too big. So in the current situation, this abnormal value is never updated. It is still not known what is causing this problem.

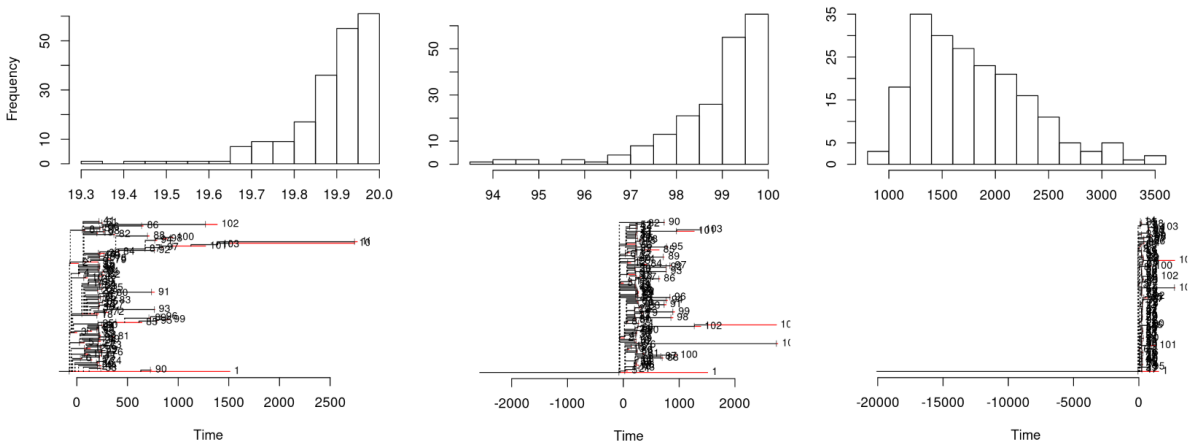


Figure 6.1.: Posterior densities of θ_E with prior $\mathcal{U}(0, 20)$, $\mathcal{U}(0, 100)$, $\mathcal{U}(0, 4000)$ and the corresponding transmission tree. The black line at the bottom is the exposure time for the root node.

6.3. Algorithm scaling with respect to the size of the network

It can be observed from the numerical experiments that the scale of the data (i.e. how many nodes it has in total) strongly influences the run time of the MCMC algorithm. Besides the 1861 Hagelloch dataset which has 188 nodes and the presented social media dataset which has 105 nodes, another social media dataset containing 377 nodes and having similar content (no exposure time and full parent, infectious and removal time) was also created for testing. However, with the same formula, the algorithm runs about 100 times slower than the 105-node social media dataset and is therefore unrealistic to run on a personal computer. The algorithm runs about 3 to 5 times slower on the 188-node Hagelloch dataset (with proper proposal distributions for $\boldsymbol{\eta}$ parameter that result in a similar acceptance rate). This is not surprising since the algorithm calculates the probability of the existence of edge between each pair of nodes. With the growth of the number of total nodes N, the possible edges $\binom{N}{2}$ grows quadratically.

Due to the iterative update nature of the MCMC algorithm, further optimization such as parallelization of the program seems impossible. Since the time consumption shows an exponential explosion trend with the growth of the network scale, it is doubtful whether this framework can be used for larger-scale analysis.

6.4. Conclusion

In this thesis we studied a set of statistical methods that are designed for analyzing the transmission dynamics of infectious diseases over social networks and explored the feasibility of using such methods for social media analysis. It turns out that while such an application is possible, we still need some finer tuning of the model. Specifically, we may need to limit the prior of the exposure time, which may reduce the interpretability of the model in the absence of evidence from other scientific studies.

Nevertheless, there is still great potential for research that aggregates network analysis and infectious disease models and applies them to the field of communication. For this application, possible future research includes optimizing the efficiency of the algorithm so that it can be applied to a larger network and introducing dynamic changes in the network itself. It will also be a great improvement to this model if the dyadic independent requirement is loosen a little bit so that we can better simulate the "friends of friends are more likely to be friends" feature of the real life social networks.

Bibliography

[1] Groendyke, C., Welch, D. and Hunter, D.R. (2011). *Bayesian Inference for Contact Networks Given Epidemic Data*. https://doi.org/10.1111/j.1467-9469.2010.00721.x

[2] Groendyke, C., & Welch, D. (2018). *epinet: An R Package to Analyze Epidemics Spread across Contact Networks*. https://doi.org/10.18637/jss.v083.i11

[3] Groendyke, C., Welch, D. and Hunter, D.R. (2012), *A Network-based Analysis of the 1861 Hagelloch Measles Data*. https://doi.org/10.1111/j.1541-0420.2012.01748.x

[4] Groendyke C, Welch D (2018). *epinet: Epidemic/Network-Related Tools.* `R` package version 2.1.8, https://cran.r-project.org/web/packages/epinet/index.html

A. Related R code

A.1. Simulation

```
## SIMULATE CONTACT NETWORK
N <- 50 # Total number of nodes
# Random 2-dimensional position for each node
simCov <- data.frame(id = 1:N, xpos = runif(N), ypos = runif(N))
# Consider only two covariates, the overall density = 1 and Euclidean distance
simDyadCov <- BuildX(simCov, binaryCol = list(c(2,3)), binaryFunc = "euclidean")
simeta <- c(0,-7)
# incremental log-odds of an edge decrease by 7 for each unit distance
simnet <- SimulateDyadicLinearERGM(N = N, dyadiccovmat = simDyadCov, eta = simeta)
simNet <- as.network(simnet)
simCoord <- data.matrix(data.frame(simCov[,2], simCov[,3]))
plot(simNet, usearrows = 0, coord =simCoord)

## SIMULATE EPIDEMIC using beta=1, k_I=3, theta_I=7, k_E=3, theta_E=7
simepi <- SEIR.simulator(M = simnet, N = N, beta = 1, ki = 3,
                         thetai = 7, ke = 3, latencydist = "gamma")
plot(simepi, e.col = "slategrey", i.col = "red")
```

A.2. Input example

```
priors <- priorcontrol(etaprior = c(0, 0, 0, 0, 0, 0, 0, 0),
                       bprior = c(0, 4), tiprior = c(0.25, 0.75),
                       teprior = c(0.25, 1), keprior = c(8, 20),
                       kiprior = c(15, 25), priordists = "uniform")

mcmcinput <- MCMCcontrol(nsamp = 200000, thinning = 100,
                         extrathinning = 10, burnin = 10000, seed = 1,
                         etapropsd = c(rep(0.05, times = 3), 0.005))

output <- epinet(~ `location.match` + `follower_count.diff` + `post_count.diff`,
                 epidata = Times, dyadiccovmat = mydyadCov,
                 mcmcinput = mcmcinput, priors = priors)
```

A.3. Working code that produces all the social media results

```
# PRIOR
priorF <- priorcontrol(etaprior = c(0, 1, -0.5, 1, -0.0003, 1, -0.001, 1),
                       bprior = c(0, 1), tiprior = c(0, 300), teprior = c(0, 5000),
                       keprior = c(0, 20), kiprior = c(0, 10),
                       priordists = "uniform")

# MCMC CONTROL
Fmcmcinput <- MCMCcontrol(nsamp = 10000000, thinning = 1000,
                                  extrathinning = 10, burnin = 200000, seed = 2,
                                  etapropsd = 0.00005*c(rep(1, times = 3)))

# FUNCTION CALL
myoutF <- epinet(~ `follower_count.diff` + `post_count.diff`, epidata = Times,
                 dyadiccovmat = mydyadCov, mcmcinput = Fmcmcinput, priors = priorF)

# CONVERGENCE TEST
varA <- myoutF  # for convenience
library("coda")
geweke.diag(as.mcmc(varA$eta))   # SAME CODE FOR OTHER PARAMETERS
library("bayesplot")
mcmc_trace(varA[["eta"]],facet_args = list(nrow=3,ncol=1))
plot(varA[["beta"]],type="l",xlab="")  # SAME CODE AGAIN

# TRANSMISSION TREE
varA[["exptimes"]][1,] <- c(rep(-100,times=100))  # truncating the initial exposed
plot(varA, e.col = "slategrey", i.col = "red", leaf.cex=0.5, lwd=1)

# POSTERIOR SAMPLES
hist(varA[["beta"]],breaks="FD") # SAME CODE FOR OTHER PARAMETERS
```

B. Additional plots

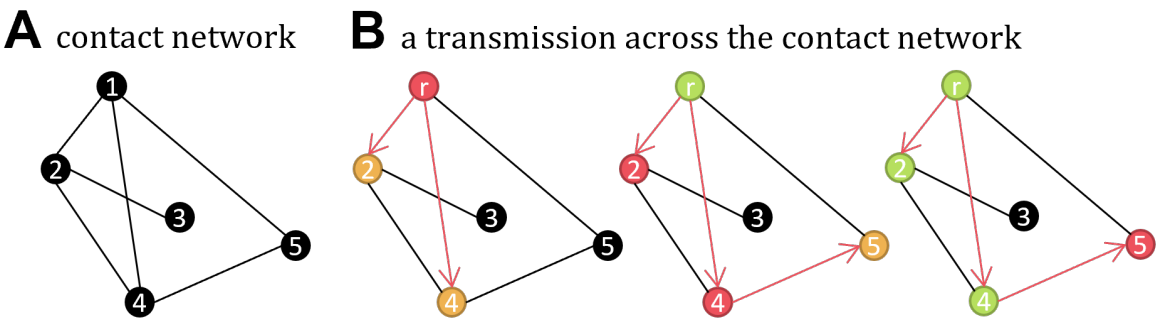


Figure B.1.: (A) An example contact network. (B) An example of an (three steps) SEIR pandemic transmission across the network. Black, yellow, red and green nodes represent the current susceptible, exposed, infectious and removed individuals correspondingly. The red arrows show the transmission path ($\mathcal{P}$) across the network and the black lines are the edges that the epidemic did not spread ($\mathcal{G}/\mathcal{P}$). The node r is the root node.

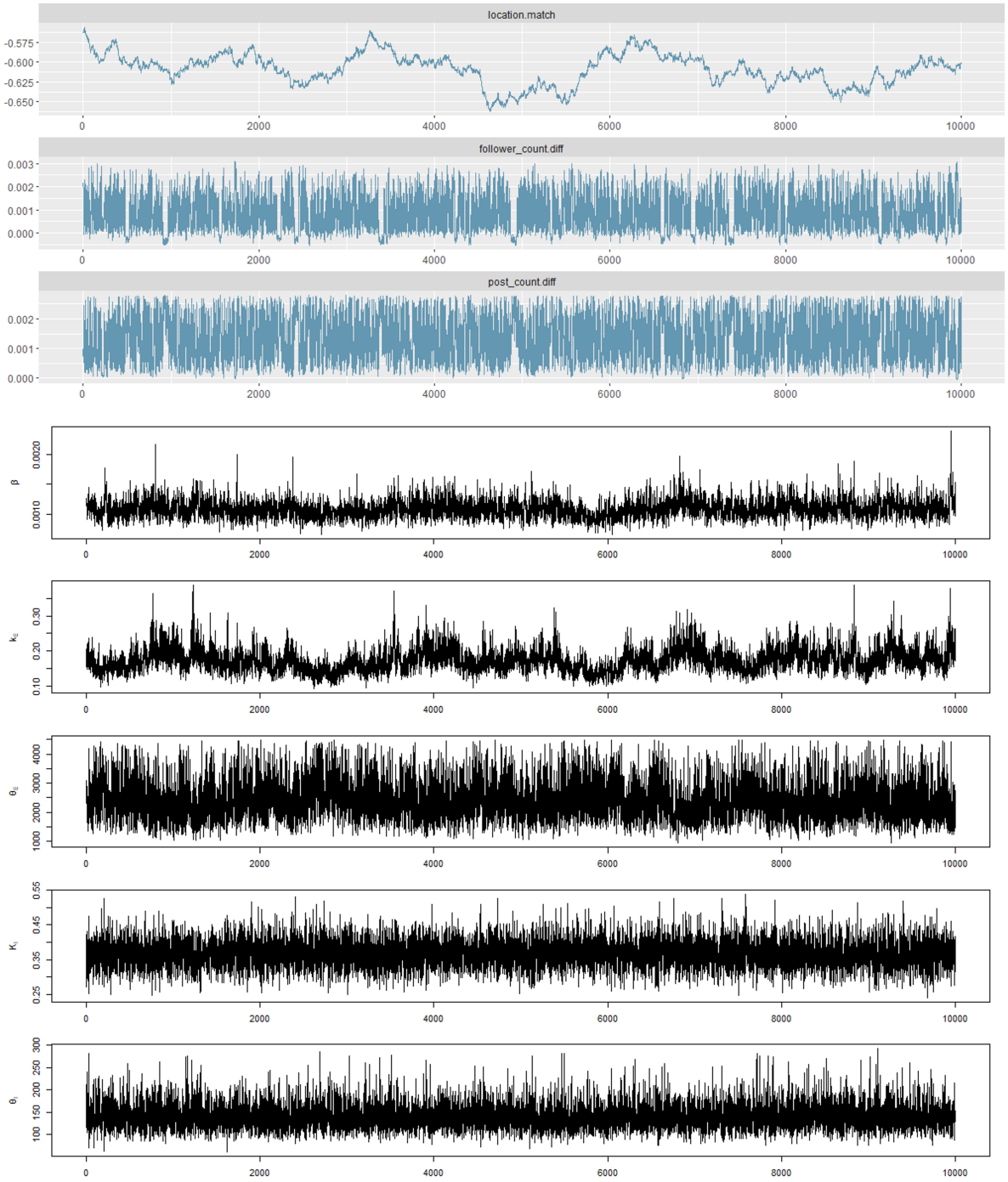


Figure B.2.: Trace plot for all the parameters. The Markov chain has a length of 10,000,000 and was thinned every 1000 iterations. The first 2,000,000 samples is set as burn-in.

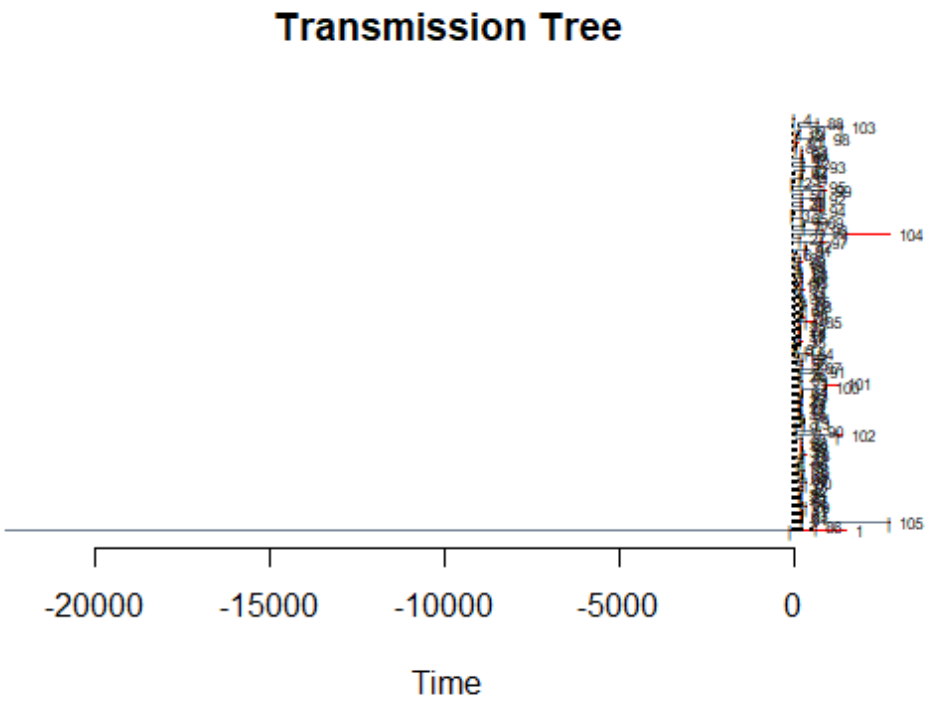


Figure B.3.: The actual transmission tree without truncating the initial exposure time.

C. The bug of the package

Following is the ninth line of the function `epinet`:

```
ninf <- min(which(is.na(epidata[, 5])), dim(epidata)[1] + 1) - 1
```

This line also exists in the line 62 of function `epibayesmcmc` that controls `C` to perform the actual MCMC calculation. After this line, the function writes

```
if (verbose)
  cat("Epidemic␣data:␣", ninf, "infecteds,", N - ninf,
    "susceptibles,␣", N, "total␣individuals.␣\n␣\n")
```

From this line, we can see `ninf` is the number of infected individuals, and this is calculated by locating the smallest `Na` in the fifth column (the column storing the removal time) with the first line of code written above. This is a wrong way to calculate it. As a result, the first node, say node j whose `Rtime` is `Na` is mistaken as the starting of all the susceptibles, and the total amount of the infected is mistaken as $j - 1$.

For example, if the only missing `Rtime` is that of the sixth node, then the program wrongly assumes there are only `6-1=5` infecteds, and the remaining `N-ninf` are all susceptibles. Following is the test run code reproducing this bug:

```
bugtestTimes <- Times
bugtestTimes[6,5] <- NA
```

Then call the `epinet` using the same argument in appendix A.2 except `epidata=bugtestTimes`. The program will then show the following output before the MCMC starts.

```
Epidemic data:  5 infecteds, 100 susceptibles,  105 total individuals.
```

Where there should only be one susceptible (i.e. the sixth individual.)

As a mini example, suppose that there are only 5 nodes in the network and they have all been infected during the transmission, and the removal time of the third node is unknown. If the corresponding `Rtime` column of the transmission matrix is `(2,3,NA,4,5)`, the package will read it as `(2,3,NA,NA,NA)` and assume node 3,4,5 all remain susceptible (i.e. had never been infected) throughout the epidemic.