Disease Modeling

1 Intro

In this thesis we perform inference on an extension of a model presented by Held et al. (2006) for modeling parameters of infectious disease counts in a Bayesian framework. Their method models surveillance data of infectious disease counts as a branching Poisson process with a cyclical endemic parameter. The cyclical endemic parameter models "seasonality" of disease counts (e.g. seasonal flu patterns) and the Poisson branching process occasional "outbreaks" (e.g. swine flu, H1N1). Their method models a single time series of count data from a specific location or region. We attempt to extend their model to multiple time series of count data that are spatially related.

Specifically the dataset of interest contains data from 412 administrative districts in Germany. Each district has surveillance data which contains the weekly case counts of severe gastroenteritis or stomach flu. The data spans May 2 and July 26 2011 when a strain of *Escherichia coli* (*E. coli*) caused an outbreak of severe illness in Germany. Over 3,950 people were affected compared with the typically seen ~200 cases a year typically seen. Many developed severe complications and over 50 people died (*EHEC O104* 2011 , pp. 2-3, 11).

We attempt to extend their model to incorporate a graph. The graph represents a transmission network between the different spatial regions (e.g., the administration district) and allow dependencies between count data in those regions. The end goal will be to be able to estimate covariates that affect the probability of transmission between regions (e.g., the shipping network that transmitted the *E. coli* contaminated food).

Statistical methods to model infectious disease data typically use mechanistic models that is models that use expert knowledge of the underyling process (Pawitan 2001, ch. 1). One such model is the Susceptible-(Exposed)-Infectious-Recovery (SIR/SEIR) (Keeling and Rohani 2008 , pp. 41-43) model and other variants. In the SEIR model, the entire closed population of individuals is in one of the four states. susceptible individuals can become exposed when in contact with an infectious invidual, exposed individuals will eventually become infectious and infectious individuals will eventually be recovered and cannot be re-exposed or infect susceptibles (e.g. by immunity). Given partial data on the susceptible, exposed, infectious and removed times parameters surrounding the epidemic of interest can be estimated. For example, in Groendyke et al. (2011), the time spent in each state is modeled with a Gamma random variables and the parameters of the random variables can be estimated from the data. Other parameters of interest such as the intial infected can also be estimated. They also estimate contact network between individuals, covariates affecting the formation of the contact network and a corresponding transmission tree. While effective, this sort of mechanistic model requires data at a granularity not typical of surveillance data, (such as information on the susceptibles) and often contains other problems such as underreporting (Diggle et al. 2003 , pp. 233-266)

2 Intro to Two-Component Model

Held et al. (2006) presents a stochastic model for the statistical analysis of infectious disease counts that serves as the basis of the theextended graph model.

The two components of the model are a simple Poisson branching process with autoregressive parameter λ and a seasonal component fit with a Fourier series. These components are described as the "epidemic" and "endemic" components respectively. Additionally, the two-component model allows for the λ to change over time allowing for the disease to change infectivity over time.

A branching process is a model that is used to model the evolution (Grimmett and Stirzaker 2004 , pp. 171-175, 243-255) of population over time over time. Suppose there's currently one parent alive. Then in a branching process, this individual would give birth to a random number of offspring and then immediately die. Then at the next step, each of those offspring become parents who individually give birth to a random number of offspring. If the random number of offspring comes from a Poisson distribution, then the process is known as a Poisson branching process. It's in this sense that λ is an autoregressive parameter since it describes the relationship between the individuals in the previous time step to the current one. In thise cas we assume that the λ value is constant across individuals and over time (e.g. λ is some biological reproduction rate). In the case of the Two-Component model, is fixed for individuals at a given time, but is allowed to vary over time. This allows the model to capture the dynamics of disease outbreaks.

2.1 Two-Component Model Notation

Let $Z = (Z_0, Z_1, ..., Z_n)$ be the infectious disease counts at each time step t. The model is then specified through $Z_t|Z_{t-1}$.

Each Z_t is determined as $Z_t = Y_t + X_t,\ t \in \{1, \dots, N\}$

Where Y_t is the epidemic component and X_t is the endemic component.

2.2 Epidemic Component

The epidemic component is given by:

$$Y_t|Z_{t-1} \sim \text{Pois}(\lambda_t Z_{t-1})$$

Where λ_t is the time varing infectivity parameter and Z_{t-1} is the the infected count in the previous time step.

We can think of λ_t as the infectivity of the disease at time t with an infected person causing new infections as $\text{Pois}(\lambda_t)$. Since each infected at time Z_{t-1} generates new infected i.i.d $\text{Pois}(\lambda_t)$, then Z_t is the sum of those random variables which itself is Poisson; $\sum_1^{Z_{n-1}} \text{Pois}(\lambda_t) = \text{Pois}(\lambda_t Z_{n-1})$.

In this model, the λ_t is allowed to vary over time. The parameter $\lambda = (\lambda_1, \cdots, \lambda_n)$ is a piecewise constant function with unknown number of changepoints K and unknown location of changepoints $\theta_1 < \cdots < \theta_K$ with $\theta \in \{1, ..., n-1\}$. If $K = 0$ there is no changepoint and the λ parameter is constant throughout.

2.3 TK add references below

For a Poisson process, when $\lambda > 1$ an outbreak occurs (Held et al. 2006). When $\lambda < 1$ then the process "goes extinct" or reaches and remains at 0 with probability 1 (Grimmett and Stirzaker 2004 , pp. 245). Once the process reaches a point where $Z_t = 0$, it remains there as there are no more infected to create new infected at the next time step. When $\lambda_t > 1$ an "outbreak" occurs shown as the spike in the graph between. When $\lambda_t < 1$ the outbreak ends.

Allowing λ_t to vary captures many scenarios, for example a particulary infectious strain of the flu could cause λ_t to increase above 1 and cause an outbreak. Later, better or new vaccines and quarentine procedures can cause the overall infectivity to decrease below 1.

2.4 Endemic Component

The endemic component in the model plays two roles. It allows capture of cyclical behaviors in disease counts (e.g. seasonal flu) and it also prevents the branching process from going extinct. The endemic count is modeled as

$$X_t \sim \text{Pois}(\nu_t)$$
$$\log \nu_t = \gamma_0 + \sum_{l=1}^{L} (\gamma_{2l-1} \sin(\rho l t) + \gamma_{2l} cos(\rho l t))$$

That is, $\log \nu_t$ is fit with a Fourier series which can approximate any function arbitrarily closely. Following, Held et al. (2006) the series is computed with $L = 1$ since it was determined higher order frequencies were insignificant. That is the truncated series is still flexible enough to model cylical patterns seen in disease counts.

The parameter can then be fit with a linear regression $\log \nu_t = s_t \gamma^T$

where $s_t = \langle 1, sin(\rho l t), \gamma_{2l} cos(\rho l t) \rangle$

2.5 Likelihood

Then the probability of the full time series Z given the initial starting count Z_0 can be factored as a product of the probabilities of each Z_t given the previous count Z_{t-1}

$$P(Z|Z_0, \theta, K, \lambda^{(1)}, \ldots, \lambda^{(K+1)}, \gamma_0, \gamma_1, \gamma_2) = \prod_{t=1}^{b} P(Z_t|Z_t{-}1, \theta, K, \lambda^{(1)}, \ldots, \lambda^{(K+1)}, \gamma_0, \gamma_1, \gamma_2)$$

Where Z_t is distributed as sum of the Poisson endemic and epidemic components

$$Z_t|Z_{t-1}, \theta, K, \lambda^{(1)}, \ldots, \lambda^{(K+1)}, \gamma_0, \gamma_1, \gamma_2 \sim \text{Pois}(\nu_t + \lambda_t Z_{t-1})$$

where the endemic component ν_t is described as

$$\log \nu_t = \gamma_0 + \gamma_1 \sin(\rho l t) + \gamma_2 \cos(\rho l t)$$

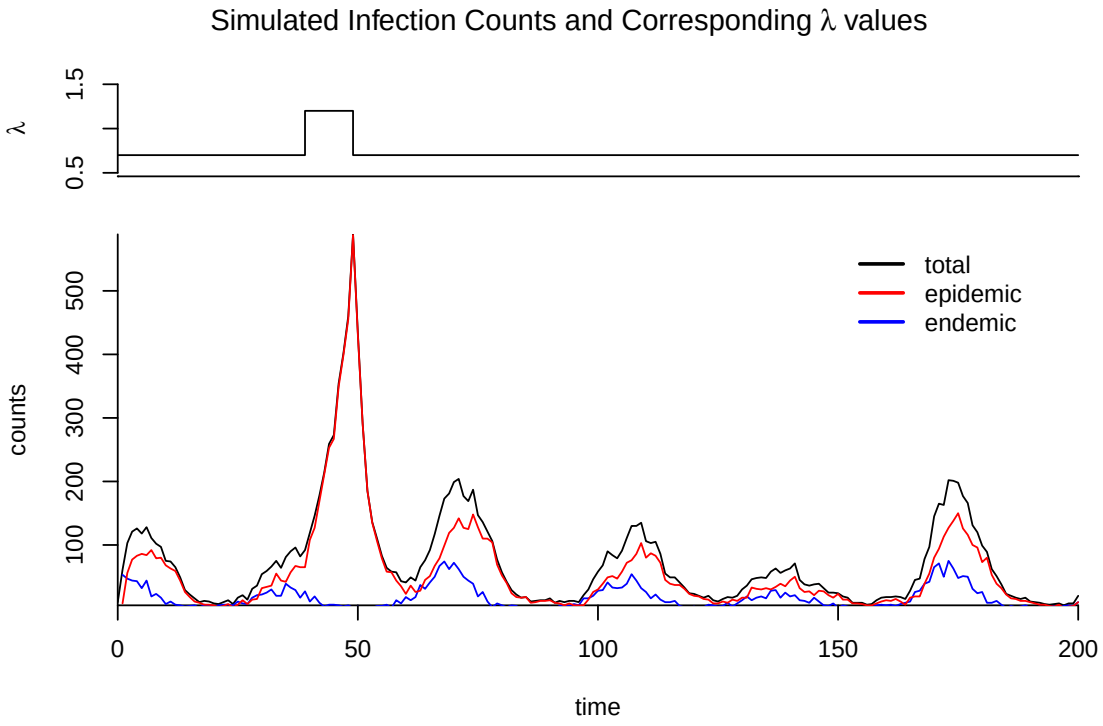


Figure 2.1: plotting example

and λ_t piecewise function is described by

$$\lambda_t = \begin{cases} \lambda^{(1)} & t < \theta_0 \\ \lambda^{(k)}, & \theta_k \leq t < \theta_{(k-1)} \\ \lambda^{(K+1)}, & t \geq \theta_K \end{cases}$$

3 Multiple locations connected by a graph

We would like to extend our data from a univariate time series of counts Z_t to a multiple time series of counts $Z_{i,t}$ where i now indexes separate time series. In this case we will say the i indexes individual "cities" where Z_i represents the infectious disease counts in city i. Now, a city i's epidemic disease count at time t is modelled as a function of both it's counts $Z_{i,t-1}$ and possibly other cities counts as well.

This dependency between cities is represented with a fixed graph G where N_v, the number of vertices in graph is fixed and equal to the number of cities and N_e is the number of edges in the graph, representing dependencies between cities' disease counts. We call $V = \{1, \ldots, N_v\}$ the vertex set of the graph where N_v is the number of cities/individual time series of disease counts, and $E(G)$ is a set of unordered pairs of vertices $\{i, j\}$ where $i, j \in V$ and $i \neq j$ that describes the edges present in graph G. We represent the edge $\{i, j\}$ as e_{ij} and the indicator $1[e_{ij} = 1]$ if $\{i, j\} \in E(G)$ and 0 otherwise.

We then model the disease count of city i at time t as a function of $Z_{i,t-1}$ (as before, it's own counts at time $t-1$) as well all the counts of the cities j it is connected to, $Z_{j,t-1}$.

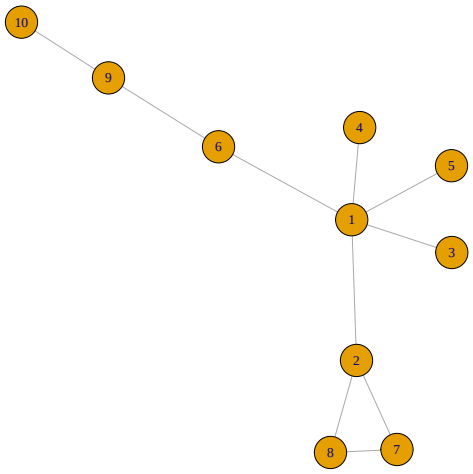


Figure 3.1: A graph configuration. The vertices $i \in \{1, \ldots, 10\}$ represent cities each with their own disease counts $Z_{i,t}$. The edges between the graph represent whether the counts between the cities can affect each other. In this example city 1's disease counts at time t are influenced by both it's own counts and cities 2, 4, 5 and 6's (i.e. every city connected to it) disease counts. City 10's disease counts are only it's own and city 9's.

Then the counts at city i at time t are modeled as $Z_t|Z_{t-1}, G = X_{i,t} + Y_{i,t}|G$ where

$$X_{i,t} : \text{infected count in city } i \text{ at time step } t, \text{ due to endemic factors}$$

$$Y_{i,t}|G : \text{infected count in city } i \text{ at time step } t, \text{ due to epidemic factors}$$

$$Z_{i,t}|G = X_{i,t} + Y_{i,t}|G : \text{infected count in city } i \text{ at time step } t$$

As before we have the epidemic component as $X_{i,t} \sim \text{Pois}(\nu_t)$. For the epidemic component we now include the additional counts from connected cities as

$$Y_{i,t}|G \sim \text{Pois}(\lambda_t * \sum_{j \neq i}^{N_v} Z_{j,t-1} * 1[e_{ij} = 1] + \lambda_t Z_{i,t-1})|G$$

Where $\lambda_t * \sum_{j \neq i}^{N_v} Z_{j,t} * 1[e_{ij} = 1]$ are the counts from cities connected to city i.

That is the epidemic component for city i is the sum of all counts in every city j connected to i in addition to the counts in city i.

3.1 Migration

One issue with this model is that connecting two isolated cities essentially doubles the infectivity parameter, since we include the counts from both cities. In this formulation a connection between two cities is equivalent to treating them a single city. To make the model more realistic, a migration parameter $m \in (0, 1)$ is introduced.

The migration parameter enters into the likelihood as

$$Y_{i,t}|G \sim \text{ Pois}(m * \lambda_t * \sum_{j \neq i}^{N_v} Z_{j,t-1} * 1[e_{ij} = 1] + \lambda_t Z_{i,t-1})|G$$

The migration parameter m is then the fraction of infected in city j that can cause infections in city i, where $m = 1$ is equivalent to the previous model and all infected in city j are counted and $m = 0$ is no infected are counted and is equivalent to the cities i, j not being connected in graph G (i.e, $e_{ij} \notin G$)

3.1.1 Future Extensions

While this thesis only covers a fixed migration rate, the end goal would be to model the migration rate as a function of city specific attributes, e.g. size of the cities, distance between the cities, if the cities are near large highways or have ports for shipping. Then m can be modelled as a logistic function of these city parameters $m \sim logit(\beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \dots)$. This would allow estimation of which city specific attributes cause the highest migration.

3.2 Graph Likelihood

We model the graph G modeled as an Erdos-Renyi random graph. An Erdos-Renyi random graph has a fixed vertex set $V(G) = \{1, \dots, N_v\}$ and is parameterized by $p \in (0, 1)$. Then an edge e_{ij} is in the edge set $E(G)$ with probability p independent of every other edge. That is an ER graph $G \sim ER(p)$ and the likelihood of a graph G given probability p on an edge is

$$\begin{aligned} G|p &= \prod_{i,j \in V(G), i \neq j} p^{1[e_{ij}=1]}(1-p)^{1-1[e_{ij}=1]} \\ &= p^{N_e}(1-p)^{\binom{N_v}{2}-N_e} \end{aligned}$$

where N_e is the number of edges.

4 Bayesian Inference

In Bayesian analysis, we aim to estimate the distribution of the parameters on interest given the data via Bayes' theorem.

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

$P(\theta|D)$ is the posterior distribution of the parameters, θ given the data D. $P(D|\theta)$ is the likelihood of the data D given the parameter value θ. $P(\theta)$ is the prior distribution of θ and represents the belief that the parameters will take certain values. The prior distribution can be specified to by "uninformative" in the sense that the contribution to the posterior likelihood is much smaller than the likelihood. $P(D)$ is the probability of the data marginalized over the parameter space. One issue is the computation of $P(\text{data})$ which is given by $\int P(\text{data}|\text{parameters})P(\text{parameters})$ over the entire parameter space.

With many parameters this integral it typically analytically intractable. To handle this Markov chain Monte-Carlo methods are used. We now specify the priors of the Two-Component and graph portions of the model.

4.1 Two-Component Priors

The prior values for the γ parameters are Multivariate-Normal distributed with variance $3I_3$ to describe the uncertainty.

$$\gamma_i \sim N(0, 3I_3), i \in \{0, 1, 2\}$$

Since the λ values parameterize a Poisson distribution we set the prior to be Gamma(1, 1), it's conjugate distribution. If Gamma is interpreted as the sum of exponentials, then the shape = 1, rate = 1 parameterization represents seeing a single occurence in 1 unit of time and represents a vague prior.

$$\lambda^{(k)} \sim Gamma(1, 1),\ k \in \{1, \ldots, K+1\}$$

The number of changepoints K takes values in $\{1, \ldots, N\}$ where N is the total number of time points of counts collected. In the Held et al. (2006) paper the number of changepoint is uniformly distributed $P(K = k) = 1/N$ representing uncertainty in the number of changepoints. This is changed to

$$K \sim \text{Pois}(2)$$

representing that idea that the disease count data is already of interest due to a potential change in the infectivity of the disease. $K \sim \text{Pois}(2)$ places the highest mass on $K = 2$ changepoints which can capture a spike in disease counts (as seen in the simulated data) before returning to a baseline endemic rate. It also places mass on $K = 1$ (e.g. capturing a long term decrease in infectivity due to intervention) and $K = 3$ changepoints. It also acts as a regularizer to help reduce overfitting the count time series where each time point is given a unique λ_t value.

The probability of a specfic location for a change point given the number of changepoints is uniformly distributed among all the possible changepoints

$$P(\theta | K = k) = \binom{N}{k}^{-1}$$

Then the unnormallized posterior is then the product of the likelihood and the priors

$$P(\theta, K, \lambda^{(1)}, \ldots, \lambda^{(K+1)}, \gamma_0, \gamma_1, \gamma_2 | Z) \propto$$

$$\prod_{t=1}^{N} P(Z_t | Z_{t-1}, \theta, K, \lambda^{(1)}, \ldots, \lambda^{(K+1)}, \gamma_0, \gamma_1, \gamma_2) * P(\theta | K) * P(K) * P(\prod_{k=1}^{K+1} \lambda^{(k)}) * P(\prod_{i=0}^{2} \gamma_i)$$

4.2 Graph Prior

We then place a prior on p as $p \sim Unif(0,1)$ representing lack of knowledge of the sparseness of the graph. Now computing the marginal probability of a graph we find

$$\begin{aligned} P(G) &= \int_0^1 P(G,p)dp = \int_0^1 P(G|p) * P(p)dp \\ &= \int_0^1 p^{N_e}(1-p)^{\binom{N_v}{2}-N_e} * 1 * dp = \frac{1}{(\binom{N_v}{2}+1) * \binom{\binom{N_v}{2}}{N_e}} \end{aligned}$$

That is the probability of graph is proportional to the number of edges $|N_e|$ in the graph. In essence the probability of the number of edges in the graph is uniform with $P(N_e) = 1/\binom{N_v}{2}$. However a graph with 2 edges compared with a graph with 1 edge is

$$\begin{aligned} \frac{P(G_{N_e=2})}{P(G_{N_e=1})} &= \frac{1/(\binom{N_v}{2}+1)\binom{N_v}{2}}{1/(\binom{N_v}{2}+1)\binom{N_v}{1}} \\ &= \frac{\binom{N_v}{1}}{\binom{N_v}{2}} \\ &= \frac{N_v - 1}{2} \end{aligned}$$

more likely. Then graphs with $N_e = \binom{N_v}{2}/2$ are more likely compared to all graphs with $|N_e| \in (0, N_v)$ and empty and complete graphs have the highest probability overall. We instead would like the probability of a graph to be uniform amongst all possible graphs. To do so we set a prior on the number of edges so that $P(G_{N_e}) = \binom{\binom{N_v}{2}}{N_e}$. It may be desirable to place a $Beta$ prior in p with higher mass on lower probabilities.

4.3 Posterior

To summarize posterior is proportional to the product of the likelihood, graph and non-graph priors

$$\begin{aligned} &P(\vec{\theta}, K, \lambda^{(1)}, \ldots, \lambda^{(K+1)}, \gamma_0, \gamma_1, \gamma_2, G, p|Z) \propto \\ &\prod_{i=1}^{N_v}\prod_{t=1}^{N} P(Z_{i,t}|Z_{t-1}, \theta, K, \lambda^{(1)}, \ldots, \lambda^{(K+1)}, \gamma_0, \gamma_1, \gamma_2, G, p) * \\ &\qquad P(\theta|K)P(K)P(\prod_{k=1}^{K+1}\lambda^{(k)})P(\prod_{i=0}^{2}\gamma_i)P(G|p)P(p) \end{aligned}$$

Where $\vec{\theta}$ is a K dimensional vector of locations of change points (and takes values in $\{1, \ldots, N\}$ where n is end of the time series), $\lambda^{(1)}, \ldots, \lambda^{(K+1)}$ parameterize the epidemic component at the corresponding change points, $\gamma_0, \gamma_1, \gamma_2$ parameterize the Fourier series that drives the endemic component, G is Erdos-Renyi (ER) random graph that represents the dependencies between cities and $p \in (0,1)$ parameterizes the ER graphs and is the probability of an edge being in the graph.

4.4 Metropolis-Hastings Algorithm

To compute samples from the posterior distribution, the Metropolis-Hastings algorithm (and its variant, the Metropolis-Hastings-Green algorithm) is used. The algorithm begins with an Markov chain with some arbitrary transition probabilities q, whose state space is the parameter space of the model, θ. After initializing the Markov chain in some initial state θ_0, the algorithm modifies the transition probabilities in such a way that the new transition probabilities has a stationary distribution that is the target posterior. That is the Markov chain will eventually enter states in proportion to the posterior distribution.

The algorithm is as follows:

1. Initialize the Markov chain at some state θ_0
2. From the current state θ at time n propose a new state j according to q. The probability of proposing a transition $\theta \to \theta^*$ is $q(\theta^*|\theta)$
3. Compute the acceptance probability
$$\alpha(\theta^*|\theta) = \min\{1, \frac{P(D|\theta^*)P(\theta^*)}{P(D|\theta)P(\theta)} \frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}\}$$
4. Generate $U \sim Unif(0, 1)$
5. If $U < \alpha(\theta|\theta^*)$ then accept the move and the parameter value at time $n + 1$ is $\theta_{n+1} = \theta^*$. If $U \geq \alpha(\theta|\theta^*)$ reject the move. Then the chain remains in the same state at time $n + 1$ and $\theta_{n+1} = \theta$.
6. Repeat steps 1-5 for a large number of iterations.

The fraction $\frac{q(\theta^*|\theta)}{q(\theta|\theta^*)}$ is known as the Hastings ratio and is a function of the proposal distribution. It is typically chosen to be symmetric so that the ratio is always 1. Then the transition probability is ratio of the posterior distributions whose denominators, $P(D)$, cancels leaving $\frac{P(D|\theta^*)P(\theta^*)}{P(D|\theta)P(\theta)}$. The proposal ratio becomes important in asymmetric proposal distributions which occurs in some of the graph proposals below. For example if two parameter θ, θ^* values are equally likely, $P(\theta^*|D) = P(\theta|D)$ then the Markov chain should enter those states with the same frequency. However if the proposal distribution proposes entering θ^* twice as frequently as θ^*, then without the Hastings ratio there would be twice the frequency of θ^*.

A similar issue is also seen when jumping between parameter spaces. To handle this the Metropolis-Hastings-Green algorithm is introduced which adds an additional Jacobian term $|J|$ to handle the change in dimension. This is described further below.

4.5 Model Checking

An important step in all statistical modelling is checking the model.

4.5.1 Prior Checks

The priors can be checked by setting likelihood to 1. Then the posterior is only a function of the priors

$$\begin{aligned} P(\theta|D) &\propto P(D|\theta)P(\theta) \\ &\propto 1 * P(\theta) \end{aligned}$$

that is the samples returned by the MCMC should be drawn from the prior distribution. This helps shows that the prior distributions are properly implemented in the model as well as show potentially unintended assumptions about the priors.

Other prior analysis include prior sensitivity analysis where the effect of different priors on the posterior distribution are examined. This is less important when the datasets are large sense the likelihood portion generally dominates the posterior calculation.

4.5.2 Well-Calibrated

A goal of frequentist statistical inference is to obtain confidence intervals (CI), where a 95% CI for a parameter would capture the true parameter in 95% of replications.

A credible interval plays a similar role in Bayesian inference. We would like a 95% credible interval for a parameter to capture the "true" parameter value 95% of the time. If this is the case, then the model is considered "well-calibrated". To do this for Bayesian models, we draw samples of the estimated parameters from their prior distributions $\theta_{sample_1} \sim P(\theta)$ and then using the sampled parameters we simulate data according to the likelihood function $D_1 \sim P(D|\theta_{sample_1})$. The model is then used to compute 95% credible intervals as if it were real data. This process is repeated for $(\theta_{sample_2}, D_2), \ldots, (\theta_{sample_N}, D_N)$ and their corresponding credible intervals are collected. We can then compare to see if the 95% credible intervals from the N parameter samples, covers the true sampled parameter 95% of the time (Carpenter 2017; Cook et al. 2006).

4.5.3 Posterior Predictive Checking

If the model is a good fit for the data, then simulating data by sampling according the posterior distribution, should result in simulated data that looks similar to the true data. If this is not the case, the model potentially inadequate for capturing important features of the data. This can be done in a qualitative manner where obviously poor models can be investigated (Gelman et al. 2013,pp. 141–159; Kruschke 2014,pp. 130–131)

5 Methods/ Implementation

Here I describe the Metropolis Hastings algorithms used to draw samples from the posterior distributions of the parameters of interest. The algorithms are implemented in base R (R Core Team 2019) with some elements of the likelihood computation implemented in Rcpp (Eddelbuettel and François 2011). The operators generate and evaluate proposals. Each operator is a R function that accepts a list of parameters that represent the current state of

the Markov chain, then proposes an new state for a given parameter (or set of parameters). The function then calculates the log acceptance ratio and determines whether to accepts it's proposal or reject it. It then returns the proposed parameters or returns the original parameters (in the case of rejecting).

5.1 Likelihood Computation

Each operator receives the log-likelihood of its proposal by passing the proposal to either `compute_log_like()` function or the `compute_log_like_rat()` function. The `compute_log_like()` function returns the unnormalized log-likelihood of the proposed parameters. This is computationally faster since computing the normalized likelihood of a Poisson distribution requires the evalution of $Z_t!$ for each data point. The naive method of computing the entire log-likelihood was used as it was computationally fast enough on the simulated dataset (without the graph).

This differs from Held et al. (2006) and Green (1995) who perform separate likelihood computations for each three operators: birth (adding a changepoint), death (removing a changepoint) and changing a λ value. Each of these operators changes only a portion of the total likelihood computation and as such, it is computationally more efficient to only compute the ratio of the changes. For example, let $\boldsymbol{\theta}^*$ be the proposed changepoint vector where $K^* = K + 1$ (i.e. a new changepoint is added). Let m be the index of the proposed changepoint and the rest of the changepoints remain the same. Then the only part of the likelihood computation that changes is in the interval between timepoints $[\theta_{m-1}, \theta_{m+1})$ and since its the ratio between the likelihoods of the proposed vs the current parameter set is important, the parts that remain identical have a likelihood of 1 (or log-likelihood of 0) and all that remains of the likelihood computation is

Similar reductions in comptutation can be done following changes in the graph and. If an edge e^*_{ik} is proposed then only $Y_{i,t}$ and $Y_{k,t}$ and the corresponding Z's are affected. As such only those two cities need to be updated.

$$Y^*_{i,t}|G \sim \text{Pois}\big(m\lambda_t Z_{k,t-1} + m\lambda_t \sum_{j=1}^{N_v} Z_{j,t-1} 1[e_{ij} \in E(G)] + \lambda_t Z_{i,t-1}\big)$$

$$Y^*_{k,t}|G \sim \text{Pois}\big(m\lambda_t Z_{i,t-1} + m\lambda_t \sum_{j=1}^{N_v} Z_{j,t-1} 1[e_{ij} \in E(G)] + \lambda_t Z_{i,t-1}\big)$$

$$\begin{aligned} Y_{j,t}|G &\sim \text{Pois}\big(m\lambda_t \sum_{l=1}^{N_v} Z_{j,t-1} 1[e_{jl} \in E(G^*)] + \lambda_t Z_{i,t-1}\big) \\ &\sim \text{Pois}\big(m\lambda_t \sum_{l=1}^{N_v} Z_{j,t-1} 1[e_{jl} \in E(G)] + \lambda_t Z_{i,t-1}\big) \end{aligned}$$

Then the likelihood ratio can be computed as (dropping conditioning notation for clarity)

$$\frac{P(Z^*_{i,t})P(Z^*_{k,t})}{P(Z_{i,t})P(Z_k,t)} \frac{\prod_j P(Z_j)}{\prod_j P(Z_j)} = \frac{P(Z^*_{i,t})P(Z^*_{k,t})}{P(Z_{i,t})P(Z_k,t)}$$

This is currently implemented for the `add_edge_op()` and `del_edge_op()` operators.

5.2 `change_gamma()`, `change_lambda()`

The gamma and lambda parameters are updated in blocks via a standard Metropolis-Hastings step (Gelman et al. 2013,p. 280). The gamma and lambda proposals are each drawn from Multivariate Normal (MVN) distributions centered at the current parameter values. That is the proposed parameter vectors γ^* and λ^* are drawn from $MVN(\gamma, \sigma_\gamma I_3)$ and $MVN(\lambda, \sigma_\lambda I_{K+1})$ where I_n is the identity matrix of dimension n. The variance of the distributions is scaled as σ_γ and σ_λ which are hand selected to improve mixing.

Since the Normal distribution is symmetric, the Hastings ratio is 1 so the acceptance ratio is the ratio of the log-likelihoods between the current and proposed steps.

One potential issue is that gamma and lambda are used to compute the parameter of a Poisson distribution, but the Normal distribution proposals could potentially propose invalid parameter values. To remedy this, the proposal operator checks to see whether the proposed parameter value is valid (in this case positive) and automatically rejects the proposed state (staying in the current state). This can lead to inefficient mixing as a step is "wasted". This inefficiency can be resolved by using a non-symmetric proposal distribution such as log-normal or a truncated normal and an adjustment of the Hastings ratio to compensate for the asymmetry. However this does not seem to be an issue in the simulated cases, the initial values are positive and the jump (σ) size is small enough not to propose negative values.

5.3 `change_theta()`

This operator proposes a new location for one of the current change points $\theta_1, \ldots, \theta_K$. A change point$\theta_k, k \in \{1, \ldots, K\}$ is selected uniformly at random from all current change points. Then the proposed location θ_k^* is selected from values between $\{\theta_{k-1}+1, \ldots, \theta_{k+1}-1\}$ uniformly at random. For θ_1 and θ_k the proposed values are from $\{0, \ldots, \theta_1 - 1\}$ and $\{\theta_K + 1, \ldots, N\}$ respectively.

The λ are then updated to match the newly proposed location as:

$$\lambda_t = \begin{cases} \lambda^{(1)} & t < \theta_0 \\ \lambda^{(k^*)}, & \theta_{k^*} \leq t < \theta_{(k^*-1)} \\ \lambda^{(K+1)}, & t \geq \theta_K \end{cases}$$

Since the proposal is generated uniformly at random between θ_{k-1} and θ_{k+1} the proposal probability is $q(\theta_{k^*}|\theta_k) = 1/(\theta_{k+1} - \theta_{k-1} - 2) = q(\theta_k|\theta_k^*)$. Furthermore the the λ values are deterministically generated from current λ_k values and the θ_k^* values. The Hastings ratio is then 1 and the acceptance rate is the ratio of the likelihoods.

5.4 `birth/death_theta` and Reversible Jump MCMC

The `birth_theta()` and `death_theta()` operators allow for an increase and decrease in the number of changepoints. Then the parameter space can jump between a collection of possible models $\{M_K, K \in \{0, 1, \cdots, n-1\}\}$ where K indexes the number of changepoints. However models from different M_K's have different dimensions of $\boldsymbol{\theta_k}$ and the likelihoods

are not directly comparable (since they are not defined on the same probability space). To handle this we use a Reversible Jump MCMC (RJMCMC) as proposed in Green (1995).

5.4.1 `birth_theta()`

For a birth step a new changepoint is chosen uniformly at random from all possible time steps $\{1, \ldots, N\}$ that aren't currently change points $\{\theta_1, \ldots, \theta_K\}$ and then added to the current set of change points

1. Draw $u \sim Unif(0, 1)$
2. The following proposals for the new λ_1 and λ_2 values are from
$$\lambda_1 = \lambda_0 * (\frac{u}{1-u})^{(\theta_1-\theta_0)/(\theta_2-\theta_0)}$$
$$\lambda_2 = \lambda_0 * (\frac{1-u}{u})^{(\theta_2-\theta_1)/(\theta_2-\theta_0)}$$
That is the new λ values are a compromise between the original value λ_0 of the interval that was split. This compromise is a function of the location of the split of the interval.
3. In order to determine the acceptance probability for the proposal, the corresponding death move must also be determined. In death move a current changepoint is selected uniformly at random and then removed. Then the λ values are changed deterministically (as described later).

Then the new acceptance ratio is

$$\alpha_{birth}(\theta^*|\theta) = \frac{P(death)}{P(birth)} \frac{q_{(K+1\to K)}(\theta|\theta^*)}{q_{(K\to K+1)}(\theta^*|\theta) * P(u)} |J_{birth}|$$

Where

$$\frac{P(death)}{P(birth)} \frac{q_{(K+1\to K)}(\theta|\theta^*)}{q_{(K\to K+1)}(\theta^*|\theta) * P(u)}$$
$$= 1 * \frac{\frac{1}{K+1}}{\frac{1}{N-K} * 1} = \frac{N-K}{K+1}$$

The $P(birth)$ and $P(death)$ are the probabilities of proposing a birth and death step respectively and are selected such that $P(birth) = P(death)$. The $q_{(K+1\to K)}(\theta|\theta^*)$ term is transition probability from a parameter state with K θ's to $K+1$. The probability of being in the proposed state θ^* and proposing jumping back to the current state θ is the probability of selecting the newly added changepoint θ_{k^*} and deleting it. This occurs with probability $1/K+1$ since the changepoints are selected uniformly at random and there are $K+1$ changepoints in the proposed state. Similarly the probability of the current proposal state is $1/N-K$ since there are $N-K$ possible timesteps to add. Since $u \sim U(0,1)$ then $P(u) = 1$ and the reverse move is deterministic so it's probability is also 1 (and omitted from the equation). Finally the Jacobian is given by

$$|J_{birth}| = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_0}$$

.

5.4.2 `death_theta()`

For the death proposal we randomly select any of the current change points uniformly at random and remove it. Then the two λ values associated with the removed change point (call them λ_1 and λ_2 to match the above notation) are recombined deterministically as

$$\lambda_0 = \lambda_1^{\frac{\theta_m - \theta_{m-1}}{\theta_{m+1} - \theta_{m-1}}} * \lambda_2^{\frac{\theta_{m+1} - \theta_m}{\theta_{m+1} - \theta_{m-1}}}$$

where θ_m is the theta value that was removed and λ_0 is the new λ value for the merged interval. Then the acceptance rate is computed as

$$\alpha_{death}(\theta^*|\theta) = \frac{P(birth)}{P(death)} \frac{q_{(K-1 \to K)}(\theta|\theta^*) * P(u)}{q_{(K \to K-1)}(\theta^*|\theta)} \frac{1}{|J_{death}|}$$

Where

$$\frac{P(birth)}{P(death)} \frac{q_{(K-1 \to K)}(\theta|\theta^*) * P(u)}{q_{(K \to K-1)}(\theta^*|\theta)} = 1 * \frac{\frac{1}{N-K+1}}{\frac{1}{K}} = \frac{K}{N-K+1}$$

The probability of proposing the death of change point θ_m is $1/K$ since it is chosen uniformly at random from $\boldsymbol{\theta}$ which has dimension K. The $q_{(K-1 \to K)}(\theta|\theta^*)$ term is the probability of adding back the change point. Since the change points are birthed uniformly at random from all change point not currently in the $\boldsymbol{\theta^*}$ vector, the probability of birthing the θ_m that was deleted which is $1/(N-(K-1)) = 1/(N-K+1)$ And the Jacobian is

$$|J_{death}| = 1/|J_{birth}| = \lambda_0/(\lambda_1 + \lambda_2)^2$$

since the function is invertible.

6 Graph Proposals

The data structure for the graph is an adjacency list and is implemented using a list of numeric vector in R. While not a true hashmap, numerically indexing the list allows for near $O(1)$ look-ups (Horner 2015). To make proposals in the graph space the following operators are used.

6.1 `add_edge_op()` and `delete_edge_op()`

The `add_edge_op()` functions proposes adding an edge to the graph by sampling uniformly at random from all edges not currently in the graph. Let N_v be the number of vertices (cities) in the graph and N_e the current number of edges. Then the algorithm proceeds as follows

1. $A_0 = N_v(N_v - 1) - 2N_e$
2. $A = A_0$
3. for $i \in (1, \ldots, N_v)$:
4. if $(A <= N_v - \text{length}(\text{adj}[i]))$
    - Find the first $A - \text{length}(\text{adj}[i])$ edge not in the list

- add edge to the proposed adjacency list
- break

5. else $A = A - \text{length(adj}[i])$

   - $i = i + 1$
   - continue

Then the probability of selecting an edge to add is $2/(N_v(N_v-1)-2N_e)$ by symmetry. $N_v(N_v-1)$ is the total possible size of the adjacency list and N_e is the number of unique edges currently in the list. Then $N_v(N_v-1)-2N_e$ is the remaining "slots" in the adjacency list. We draw uniformly at random from $A \in \{1,\ldots,N_v(N_v-1)-2N_e\}$ which represents an index of the edges not present in the graph. Now we interate along the adjacency list `adj` whose length is the number of vertices N_v. Starting from $i = 1$ if $A <= N_v - length(\text{adj}[i])$ then we know that the edge to be added is in adj[i] and we can search for the correct edge in $O(|V|)$. If $A > N_v - length(\text{adj}[i])$ then we recompute $A = A - N_v + length(\text{adj}[i])$ increment $i = i + 1$ and repeat. Then $q(G^*_{N_e+1}|G_{N_e})$ the probability of proposing adding that particular edge occurs with probability $2/(N_v(N_v-1)-2N_e)$, since A_0 is chosen uniformly at random from $\{1,\ldots,N_v(N_v-1)-2N_e\}$ and each edge e_{ij} is represented twice (once in adj[i]: $\{\ldots$,j,$\ldots\}$) and again in adj[j] : $\{\ldots,i,\ldots\}$.

The probability of $q(G_{N_e}|G^*_{N_e+1})$ of deleting the edge (given the graph where the proposed edge was added) occurs with probability $2/2(N_e+1) = 1/(N_e+1)$

$$\frac{q(G_{N_e}|G^*_{N_e+1})}{q(G^*_{N_e+1}|G_{N_e})} = \frac{\frac{1}{N_e+1}}{\frac{2}{N_v(N_v-1)-2N_e}} = \frac{N_v(N_v-1)-2N_e}{2(N_e+1)}$$

and the Hastings ratio for removing an edge is given by

$$\begin{aligned}\frac{q(G_{N_e}|G^*_{N_e-1})}{q(G^*_{N_e-1}|G_{N_e})} &= \frac{\frac{2}{N_v(N_v-1)-2(N_e-1)}}{\frac{1}{N_e}} \\ &= \frac{2N_e}{N_v(N_v-1)-2(N_e-1)}\end{aligned}$$

6.2 `degree_preserving()`

The degree preserving swap was implemented to allow changes to the graph structure while maintaining the degree of connectivity of each vertex. This is because with large migration rates, the degree of the cities strongly influences the likelihood. Let's assume that the migration rate $m = 1$, then in Figure 6.1, city 5 and 2 have double the counts of cities 1, 3, 4, and 6. Let's also assume that the configuration on the right is the true graph. If the MCMC algorithm proposes the graph on the left first, to get to the graph on the right would require say, disconnecting $1-5$ as seen in the middle. With such a high migration rate, the cities 5 and 2 would never be swapped (it would require transitioning to a graph where city 5 would have degree 2 and city 1 degree 0). The degree preserving swap handles directly swapping between the left and right graphs without passing through the middle.

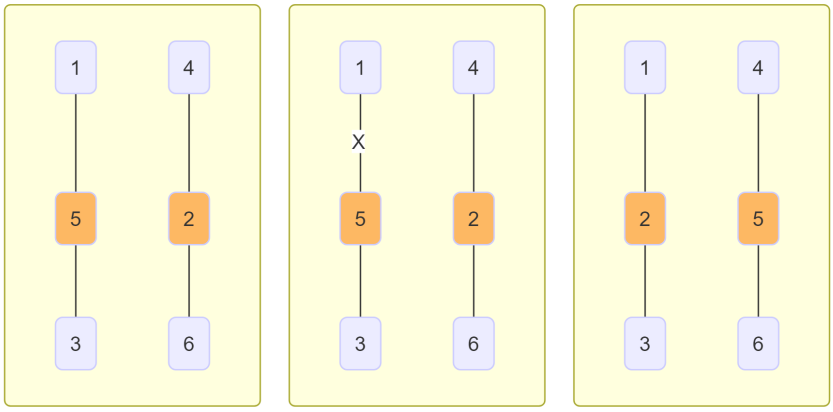


Figure 6.1: An example graph configuration that might necessitate a degree preserving swap. **Left** and **right**: two configurations of graphs whose vertices have the same degree, extbfmiddle: a potential intermediary proposal. With one-step edge adding and deleting the likelihood of switching between these two graph is low if the migration rate is high enough. The degree preserving swap operator would attempt to switch between these two graphs in 1 step. Deleting the edge between city 5 and 1 (middle figure) would occur with low probability. With a migration rate of 1, this would lower city 5's count by approximately 1/3 from the true value and by half for city 1.

This was implemented by selecting two edges (v11, v12) and (v21, v22) and attempting to form the edges (v11, v22) and (v12, v21). If both edges are not already in the graph then the swap is made. If either or both edges exists, then the proposed swap is rejected. Since the proposed swap is reversed by randomly selecting (v11, v22) and (v12, v21), the Hastings ratio is 1 and the acceptance ratio is the ratio of the likelihoods.

6.3 `rewire()`

Rewire randomly selects an edge (v1, v2) and deletes it from the adjacency list. It then randomly samples a vertex v3 and forms the edge (v2, v3). Since the reverse move is selecting the edge (v2, v3) and then rewiring (v1, v2), the Hastings ratio is 1.

7 Results/Discussion

7.1 Two-Component Model

parameters	simulation values
K: number of changespoints	2
$\boldsymbol{\lambda}$: epidemic parameters	0.7, 1.2, 0.7
$\boldsymbol{\gamma}$: endemic parameters	log(10), 0.5, 1.5
$\boldsymbol{\theta}$: changepoints	39, 49

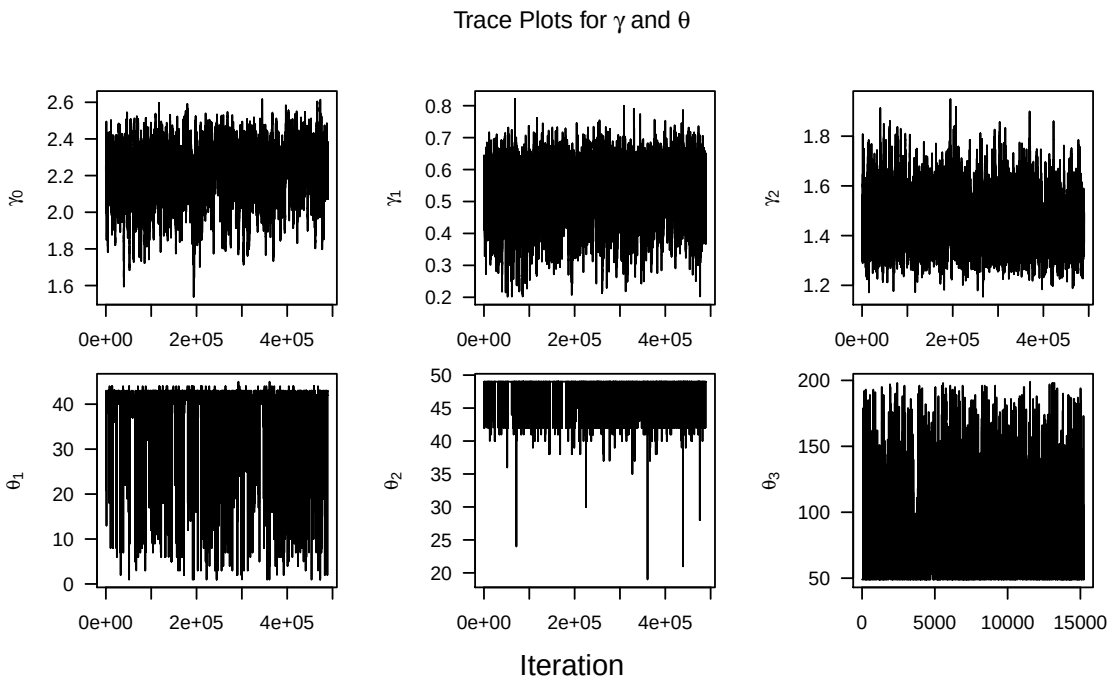


Figure 7.1: Trace plots for $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$. The trace plots seem relatively well mixed. The autocorrelation is likely a function of the correlation between the each set of $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$ respectively. Thinning could potentially reduce the autocorrelation between sample draws.

In this paper, I have implemented an MCMC algorithm that estimates the Two-Component Model from simulated data. The simulation parameters are taken from Held et al. (2006)

Figure 7.1 are trace plots, which are time series of the parameter draws. Typically a "good" trace plot shows no obvious autocorrelation (which can indicate a proposal step that is too small) or flat portions (proposal steps that are too large). Trace plots with these patterns indicate that the sampler has not converged to the posterior distribution of the parameter. "Tuning" or adjusting the proposal jump size can help fix these patterns, with the end goal of being able to efficiently explore the posterior distribution of the parameter (Gelman et al. 2013,p. 296). The trace plots for the γ parameters seem to rather well mixed though there is a slight pattern to the γ_0 trace plot.

This is likely due to the correlation (see Figure 7.2) between γ_0, γ_1 and γ_2 due to their nature of parameterizing a cyclical function. The same is true for the θ parameters; they are constrained such that $\theta_1 < \theta_2 < \theta_3$. The dependence can cause inefficient sampling by proposing moves that are outside these narrow regions/combinations of parameter values (Turner et al. 2013). This autocorrelation seen in the trace plots could possibly be reduced via thinning i.e. taking every k samples instead of every sample. Gelman et al. (2013) mentions that they have found it useful to store no more than a total of 1000 iterations.

The θ_3 trace plot is of note because it is only valid for when $K = 3$; the θ_3 does not exist otherwise.\

The $\boldsymbol{\lambda}$ trace plots have interesting patterning, specifically the λ_2 and λ_3 traces appear to fix tightly at 1.2 and 0.7 respectively while also jumping to 0.7 and 1.2 respectively. This is an artifact of the dimension change in the model. When $K = 2$, λ_2 is tightly fixed

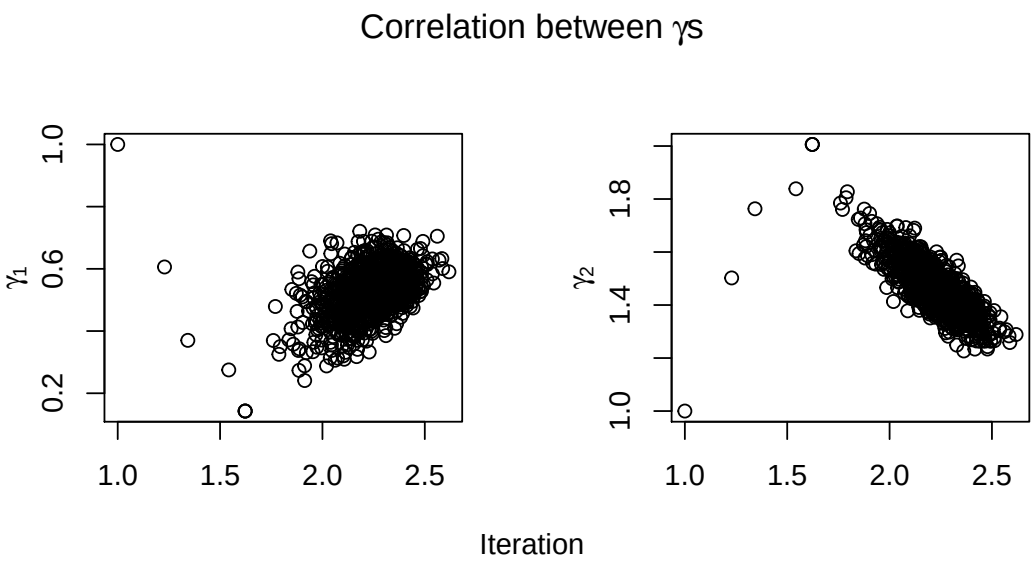


Figure 7.2: Plots of γ_0 vs γ_1 and γ_0 vs γ_2. Each point represents the corresponding parameter value at the same sample draw, i.e., the values of each parameter when the sample was accepted. The strong correlation between the parameters can lead to poor mixing. Note the tails are due to the initial values of the MCMC chain.

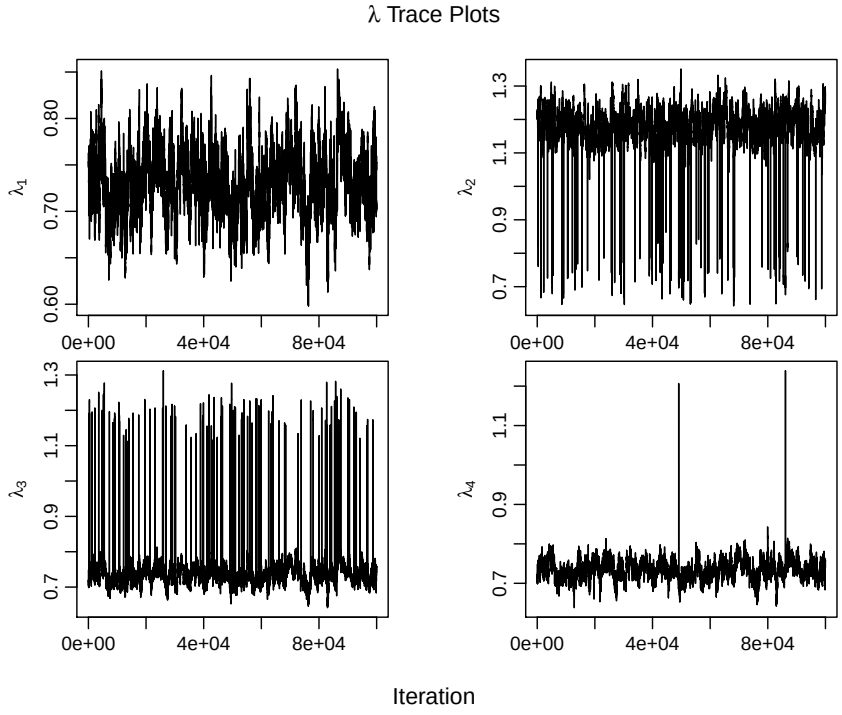


Figure 7.3: Trace plots for all λ values. Overall the trace plots show that the samplers are mixing well with some minor autocorrelation that could potentially be resolved with thinning or more samples. The raw trace plots include the λ values when there are 3 and 4 separate λ's and is the cause of the random jumps.

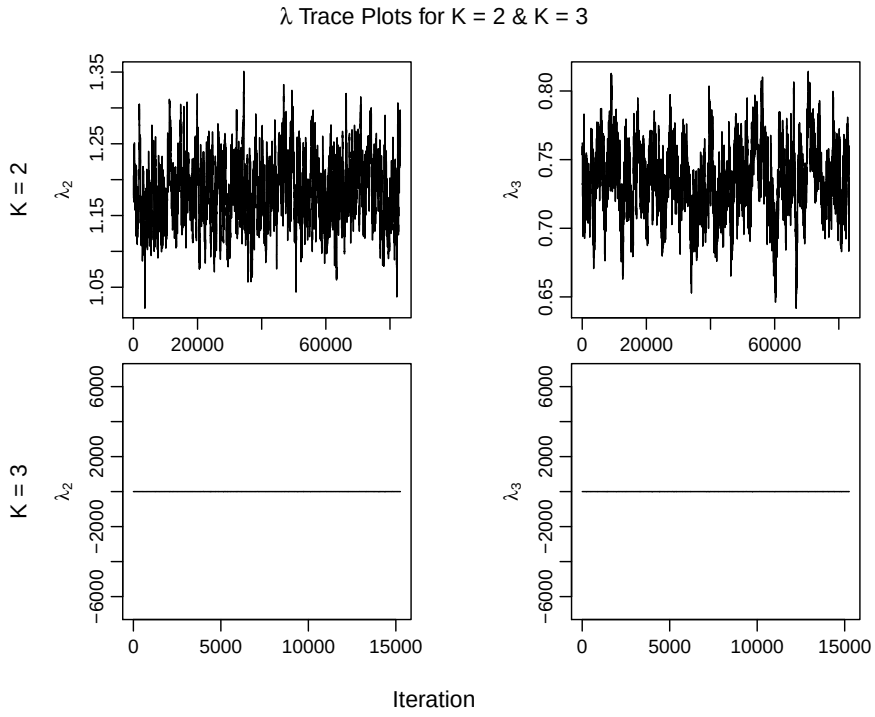


Figure 7.4: Trace plots filtered for K = 2 and K = 3 showing how the different number of changepoints affects the traces of the individual λ values.

around 1.2 and λ_3 at 0.7 but when $K = 2$, λ_3 can take the role of fitting the outbreak while λ_4 models to the return 0.7, the baseline.

In assessing the inference on the data, I examine the density plots. Here we see that the density plots cover the true values.

7.1.1 Prior Checks

I also performed prior checks where the likelihood ratio is set to 1 so that the posterior distributions should match the prior.

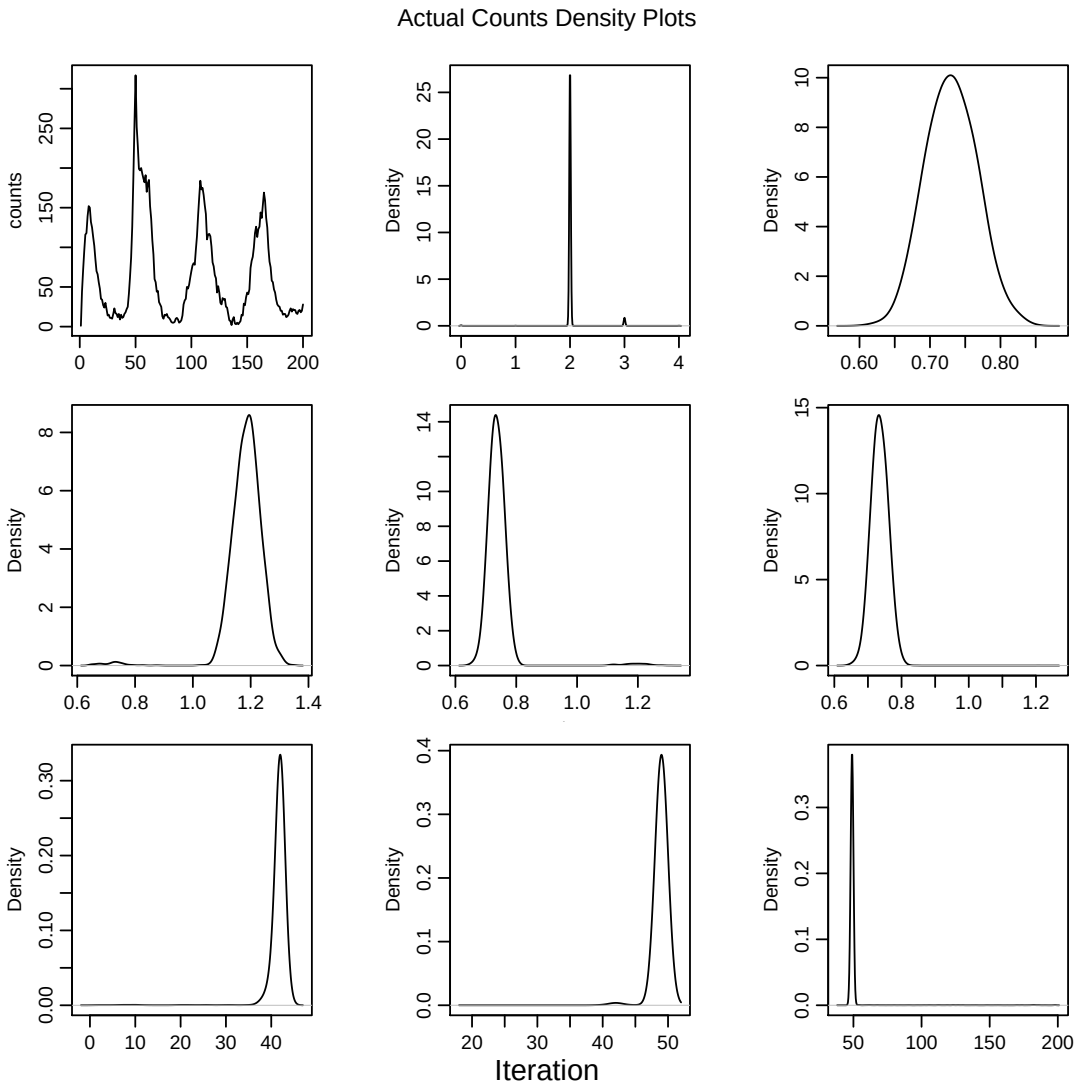
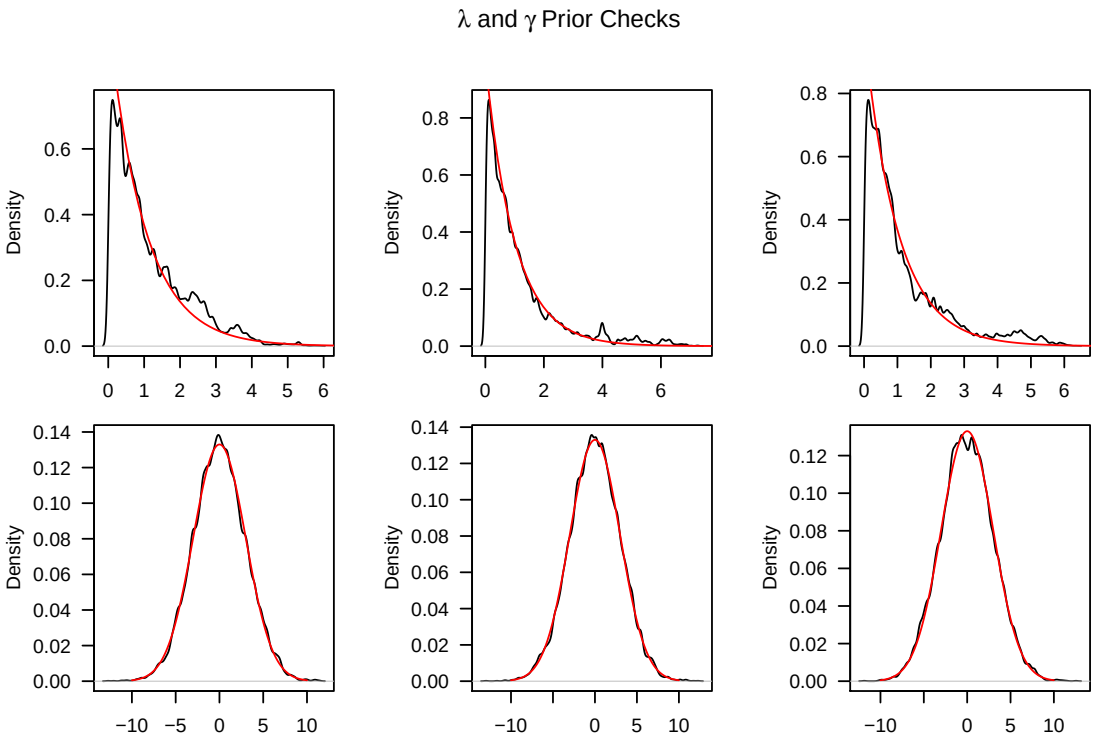


Figure 7.5: Plots for the count data and density plots showing K



The following data was collected by running 100,000 iterations (thinned to 10,000 samples) of proposing to update either the λ vector, γ vector, adding a change point ($K \to K+1$) or removing a change point ($K \to K-1$). The likelihood ratio for set to 1 so the only factor in accepting or rejecting a state was the priors. As such we would expect to see the parameter values posterior distribution to be their prior distributions. The priors are

$$\vec{\lambda} \sim \text{Gamma}(1, 1)$$

$$\vec{\gamma} \sim \text{Norm}(0, 3)$$

$$K \sim \text{Unif}(0, N)$$

In Figure **??** we have plots of the samples of the λ values (first 3) and the γ values. Each is overlayed with the density of their respective priors in red. Overall there appears to be a good match between the sampled values and their prior distributions.

In figure 7.6 we have a frequency histogram of the sampled values of K (the number of change points) overlayed with a red circle. The red circle represents the appropriate density value from the prior distribution of K. Overall we see a good qualitative fit between the values sampled from the posterior and the prior distribution. The following table gives the sample frequency and the actual probability (from the prior distibution). If these probabilities did not match, then there is likely a bug in the code or some unknown prior assumption on the model. Note that the sample frequency was computed from a thinned sample of 10,000 hence why 11, 25, and 51 have sample probabilities of 0.0001, there was 1 occurence of each.

K	sample	actual
0	0.136	0.135

K	sample	actual
1	0.270	0.270
2	0.2681	0.271
3	0.185	0.180
4	0.0899	0.902
5	0.0365	0.361
6	0.0113	0.012
7	0.0029	0.0034
8	0.0007	0.0008
9	0.0001	0.00019
11	0.0001	6.94e-6
25	0.0001	2.97e-19
51	0.0001	1.96e-52

7.2 Graph Estimation on Fixed Two-Component Model

Progress on the graph. The graph proposals are a challenge to the computational speed and mixing becomes a problem. For example the original base graph operator was `flip_edge()` which had much poorer mixing than the `add/del_edge_op()` combination. To test the correctness of the operators, we also set the likelihood to always equal to 1. With the correct derivation about the priors, any significant deviation of the posterior distributions from the prior distributions is likely to be due to implementation errors in the code.

7.3 Graph Priors

7.3.1 Posterior distribution of number of edges matches prior distribution

Here we set the prior ratio and the likelihood ratio to always equal to 1. This places a uniform prior distribution on p and a uniform prior on $G|p$. Then the probability of each graph is equally likely and the posterior distribution of the number of edges should be Binom$(105, 0.5)$. Figure 7.7 shows a thinned posterior sample of the number of edges overlayed with the appropriate Binom$(105, 0.5)$ in red.

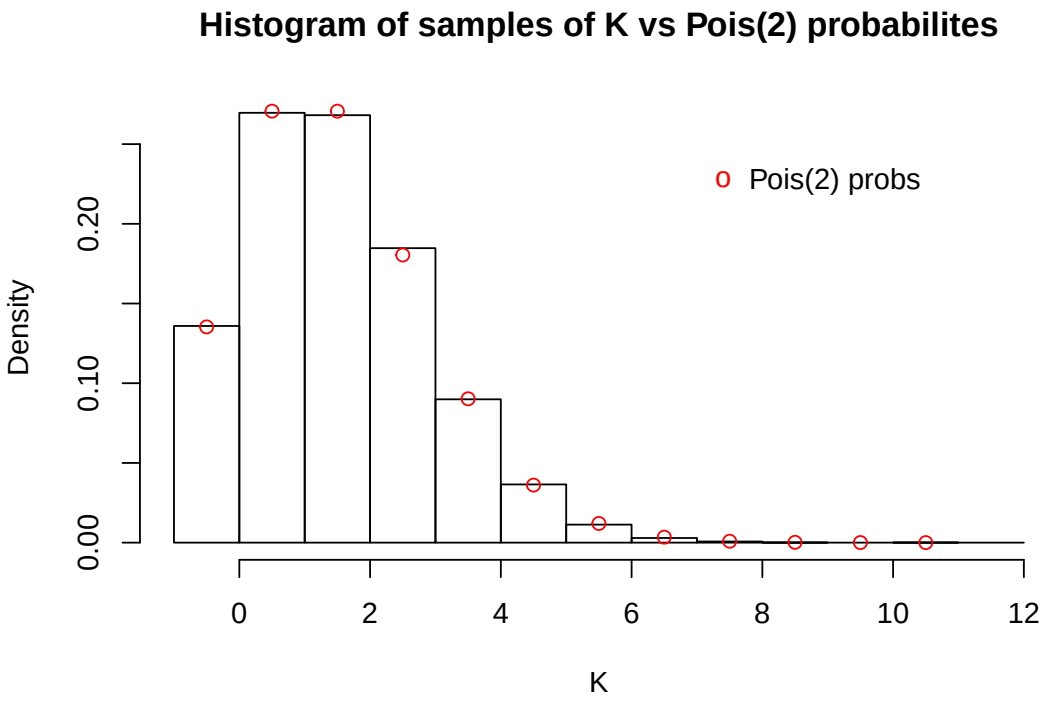


Figure 7.6: Sample probabilities of K overlayed with true probabilities. Overall this shows a good agreement between the sample probabilities and the true probabilties.

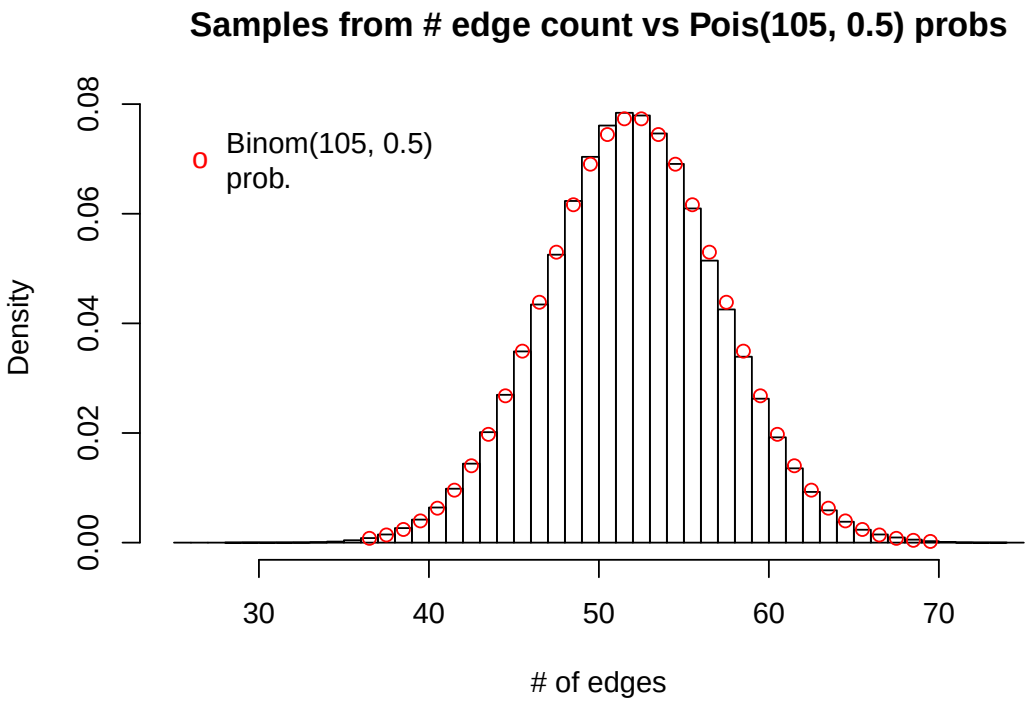


Figure 7.7: Posterior probabilities of the number of edge counts, overlayed with true probabilities. Overall this shows a good agreement between the posterior probabilities and the true probabilties.

8 Conclusion

Conclusion here

Citations

Carpenter, B. (2017), "Bayesian Posteriors are Calibrated by Definition," *Statistical Modeling, Causal Inference, and Social Science*.

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, 675–692.

Diggle, P., Knorr-Held, L., Rowlingson, B., Su, T., Hawtin, P., and Bryant, T. (2003), "On-line Monitoring of Public Health Surveillance Data," pp. 233–266. https://doi.org/10.1093/acprof:oso/9780195146493.003.0009.

Eddelbuettel, D., and François, R. (2011), "Rcpp: Seamless R and C++ Integration," *Journal of Statistical Software*, 40, 1–18. https://doi.org/10.18637/jss.v040.i08.

EHEC O104:H4 Outbreak Germany 2011 (2011), report, Robert Koch-Institut; Robert Koch-Institut, Infektionsepidemiologie. https://doi.org/http://dx.doi.org/10.25646/88.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013), *Bayesian Data Analysis, Third Edition*, CRC Press.

Green, P. J. (1995), "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination." https://doi.org/10.1093/biomet/82.4.711.

Grimmett, G., and Stirzaker, D. (2004), *Probability and random processes*, Oxford Oxford University Press.

Groendyke, C., Welch, D., and Hunter, D. R. (2011), "Bayesian Inference for Contact Networks Given Epidemic Data," *Scandinavian Journal of Statistics*, 38, 600–616. https://doi.org/10.1111/j.1467-9469.2010.00721.x.

Held, L., Hofmann, M., Höhle, M., and Schmid, V. (2006), "A two-component model for counts of infectious diseases," *Biostatistics*, 7, 422–437. https://doi.org/10.1093/biostatistics/kxj016.

Horner, J. (2015), "Hash Table Performance in R: Part I," *R-bloggers*.

Keeling, M. J., and Rohani, P. (2008), *Modeling Infectious Diseases in Humans and Animals*, Princeton University Press.

Kruschke, J. (2014), *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan*, Academic Press.

Pawitan, Y. (2001), *In All Likelihood: Statistical Modelling and Inference Using Likelihood*, OUP Oxford.

R Core Team (2019), *R: A Language and Environment for Statistical Computing*, Vienna, Austria: R Foundation for Statistical Computing.

Turner, B. M., Sederberg, P. B., Brown, S. D., and Steyvers, M. (2013), "A Method for Efficiently Sampling From Distributions With Correlated Dimensions," *Psychological methods*, 18, 368–384. https://doi.org/10.1037/a0032222.