

# Disease Modeling

## Contents

<b>1</b>	<b>Intro</b>	<b>2</b>
1.1	Comments . . . . .	2
<b>2</b>	<b>Intro to Two-Component Model</b>	<b>2</b>
2.1	Two-Component Model Notation . . . . .	2
2.2	Epidemic Component . . . . .	3
2.3	TK add references below . . . . .	3
2.4	Endemic Component . . . . .	3
2.5	Likelihood . . . . .	4
2.6	Priors . . . . .	5
<b>3</b>	<b>Graph Data Introduction</b>	<b>6</b>
3.1	Migration . . . . .	7
3.2	Graph Likelihood . . . . .	7
3.3	Graph Prior . . . . .	8
<b>4</b>	<b>Bayesian Inference</b>	<b>8</b>
4.1	Metropolis-Hastings Algorithm . . . . .	8
<b>5</b>	<b>MCMC/ Implementation</b>	<b>9</b>
5.1	Likelihood Computation . . . . .	10
5.2	change_gamma, change_lambda . . . . .	10
5.3	change_theta . . . . .	11
5.4	birth/death_theta reversible jump mcmc . . . . .	11
<b>6</b>	<b>Graph Proposals</b>	<b>13</b>
6.1	add and delete . . . . .	13
6.2	flip_edge . . . . .	14
6.3	Degree Preserving Swap . . . . .	14
6.4	Rewire . . . . .	14
<b>7</b>	<b>Results</b>	<b>14</b>
7.1	Two-Component Model on Simulated Data . . . . .	14
7.2	Graph Estimation on Fixed Two-Component Model . . . . .	14

## 1 Intro

Over May 2011, a strain *Escherichia coli* (*E. coli*) caused an outbreak of severe illness in Germany, with ~3,950 affected compared with ~200 cases a year typically seen. Of those infected 53 died.

In this - we extend a method presented by Held et al. (2006) for modeling parameters of infectious disease counts in a Bayesian framework. Their method models the count data as a branching Poisson process with a cyclical endemic parameter.

We then extend their model to incorporate a graph.

### 1.1 Comments

Expand and mention general ideas of epidemics spreading & how we model them (~1 page)

## 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  $\lambda$  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  $\lambda$  to change over time allowing for the disease to change infectivity over time.

### 2.1 Two-Component Model Notation

Let  $Z = (Z_0, Z_1, \dots, Z_n)$  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. The epidemic component is modeled as Poisson distribution as

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

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

Where  $\lambda_t$  is piecewise constant depending on an unknown number of changepoints  $K$  and unknown locations  $\theta_1 < \dots < \theta_K$ .

The endemic component is also Poisson distributed with

$$X_t \sim Pois(\nu_t)$$

Where  $\log \nu_t$  is modeled as a Fourier series:

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

## 2.2 Epidemic Component

The epidemic component is given by:

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

Where  $\lambda_t$  is the time varying infectivity parameter and  $Z_{t-1}$  is the the infected count in the previous time step.

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

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

## 2.3 TK add references below

For constant  $\lambda$ , this is known as a Poisson branching process. For such a process, when  $\lambda > 1$  the process explodes and the number of infected goes to infinity with probability 1. When  $\lambda < 1$  then the process “goes extinct” or reaches and remains at 0 with probability 1. 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  $\lambda_t$  to vary captures many scenarios, for example a particular infectious strain of the flu could cause  $\lambda_t$  to increase above 1 and cause an outbreak. Later, better or new vaccines and quarantine 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

$$X_t \sim Pois(\nu_t)$$

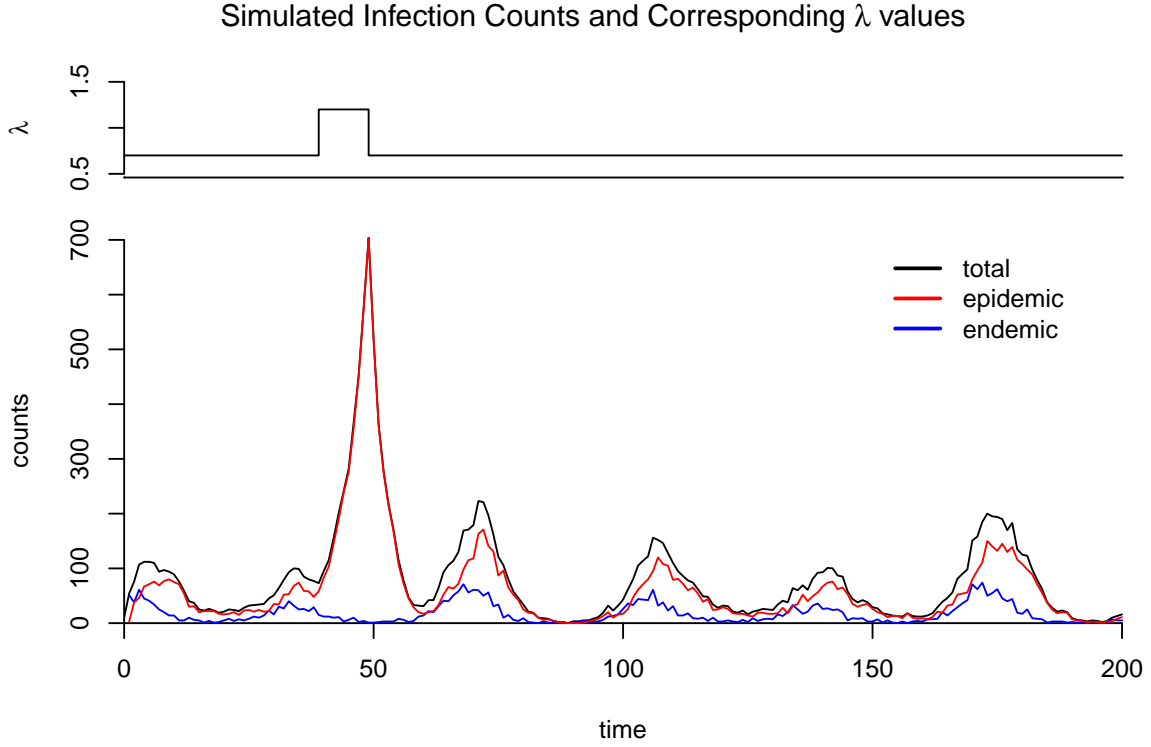


Figure 2.1: plotting example

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

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 cyclical 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  $Z_{t-1}$

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

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

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

where the endemic component  $\nu_t$  is described as

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

and  $\lambda_t$  piecewise function is described by

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

## 2.6 Priors

The prior values for the  $\gamma$  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  $\lambda$  values parameterize a Poisson distribution we set the prior to be  $\text{Gamma}(1, 1)$ , it's conjugate distribution. If Gamma is interpreted as the sum of exponentials, then the shape = 1, rate = 1 parameterization represents seeing a single occurrence in 1 unit of time and represents a vague prior.

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

The number of changepoints  $K$  takes values in  $\{1, \dots, N\}$  where  $N$  is the total number of time points of counts collected. In the (???) 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  $\lambda_t$  value.

The probability of a specific 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 unnormalized posterior is then the product of the likelihood and the priors

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

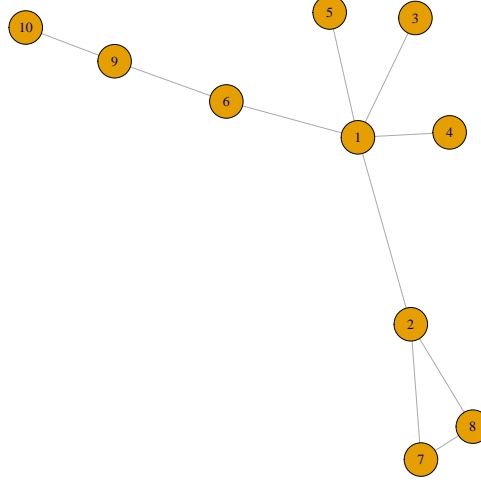


Figure 3.1: A graph configuration. The vertices  $i \in \{1, \dots, 10\}$  represent cities each with their own disease counts  $Z_{i,t}$ . The edges between the graph represent whether the counts between the cities can affect each other. In this example city 1's disease counts at time  $t$  are influenced 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.

### 3 Graph Data Introduction

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,  $i$  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  $G_{N_v, E}$  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  $V = \{1, \dots, N_v\}$  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. For clarity, the subscripts  $N_v, E$  will be dropped and it's assumed that the true graph has fixed number of vertices and fixed edges.

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,  $Z_{j,t-1}$ . 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

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

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

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

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

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

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

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

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  $m \sim \text{logit}(\beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \dots)$ . 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  $G_{N_v, E} \sim ER(p)$  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.

### 3.3 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

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

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  $P(G_{N_e}) = \binom{N_v}{N_e}$ . It may be desirable to place a *Beta* prior in  $p$  with higher mass on lower probabilities.

## 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,  $\theta$  given the data  $D$ .  $P(D|\theta)$  is the likelihood of the data  $D$  given the parameter value  $\theta$ .  $P(\theta)$  is the prior distribution of  $\theta$  and represents the belief that the parameters will take certain values. The prior distribution can be specified to be “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 is typically analytically intractable. To handle this Markov chain Monte-Carlo methods are used.

### 4.1 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,  $\theta$ . After initializing the Markov chain in some initial state  $\theta_0$ , 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  $\theta_0$
2. From the current state  $\theta$  at time  $n$  propose a new state  $j$  according to  $q$ . The probability of proposing a transition  $\theta \rightarrow \theta^*$  is  $q(\theta^*|\theta)$
3. Compute the acceptance probability

$$\alpha(\theta^*|\theta) = \min\left\{1, \frac{P(D|\theta^*)P(\theta^*)}{P(D|\theta)P(\theta)} \frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}\right\}$$

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  $\theta, \theta^*$  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  $\theta^*$  twice as frequently as  $\theta$ , then without the Hastings ratio there would be twice the frequency of  $\theta^*$ .

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.

## 5 MCMC/ 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 with some elements of the likelihood computation implemented in Rcpp. 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 pass the proposal to the `compute_log_like` 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 evaluation 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 (???) and (???) who perform separate likelihood computations for each three operators: birth (adding a changepoint), death (removing a changepoint) and changing a  $\lambda$  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  $\theta^*$  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 computation could feasibly be done following changes in the graph. Following a graph proposal any vertices whose component does not change, does not need its contribution to the likelihood recomputed. With sparse graphs this could save computational effort.

If we maintain a list of vertices (label them numerically) associated with the component they belong to (alphabetically/hex). We additionally maintain a list of components associated with a vector of vertices that comprise the component. Now assume we add an edge between two vertices. Then say a graph proposal is made to connect two vertices. We can look up the component that each of these vertices participate in (call them A and B), merge the components (A, B  $\rightarrow$  C). We then look up the vertices that participate in components formerly known as A, B and update them to the new component name (as well as update the component list) maintaining the data structure. Then we know which vertices to update.

Deleting an edge is problematic. The edge is within a component and that component needs to be updated. Updating the components becomes the issue. Either the component remains connected or it's split into two components. There's no way (?) to determine which is the case without () Wikipedia "There are also efficient algorithms to dynamically track the components of a graph as vertices and edges are added, as a straightforward application of disjoint-set data structures. These algorithms require amortized  $O(\alpha(n))$  time per operation, where adding vertices and edges and determining the component in which a vertex falls are both operations, and  $\alpha(n)$  is a very slow-growing inverse of the very quickly growing Ackermann function. A related problem is tracking components as all edges are deleted from a graph, one by one; an algorithm exists to solve this with constant time per query, and  $O(|V||E|)$  time to maintain the data structure; this is an amortized cost of  $O(|V|)$  per edge deletion. (Shiloach & Even 1981)."

## 5.2 change\_gamma, change\_lambda

The gamma and lambda parameters are updated in blocks via a standard Metropolis-Hastings step. The gamma and lambda proposals are each drawn from Multivariate Normal (MVN) distributions centered as the current parameter values. That is the proposed parameter vectors  $\gamma^*$  and  $\lambda^*$  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  $\sigma_\gamma$  and  $\sigma_\lambda$  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 ( $\sigma$ ) 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  $\theta_1, \dots, \theta_K$ . A change point  $\theta_k, k \in \{1, \dots, K\}$  is selected uniformly at random from all current change points. Then the proposed location  $\theta_k^*$  is selected from values between  $\{\theta_{k-1} + 1, \dots, \theta_{k+1} - 1\}$  uniformly at random. For  $\theta_1$  and  $\theta_K$  the proposed values are from  $\{0, \dots, \theta_1 - 1\}$  and  $\{\theta_K + 1, \dots, N\}$  respectively.

The  $\lambda$  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  $\theta_{k-1}$  and  $\theta_{k+1}$  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  $\lambda$  values are deterministically generated from current  $\lambda_k$  values and the  $\theta_k^*$  values. The Hastings ratio is then 1 and the acceptance rate is the ratio of the likelihoods.

### 5.4 birth/death\_theta reversible jump mcmc

The `birth_theta()` and `death_theta()` operators allow for increasing and decreasing the number of changepoints. Since the parameter space can jump between a collection of possible models  $\{M_K, K \in \{0, 1, \dots, n-1\}\}$  where  $K$  indexes the number of changepoints. However models from different  $M_K$ ’s have different length of  $\theta_k$  and the likelihoods are not directly comparable, they need to be defined on the same probability space. To handle this we use a Reversible Jump MCMC (RJMC MC) as proposed in (???).

#### 5.4.1 Birth

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

1. Draw  $u \sim Unif(0, 1)$

2. The following proposals for the new  $\lambda_1$  and  $\lambda_2$  values are from

$$\lambda_1 = \lambda_0 * \left(\frac{u}{1-u}\right)^{(\theta_1 - \theta_0)/(\theta_2 - \theta_0)}$$

$$\lambda_2 = \lambda_0 * \left(\frac{1-u}{u}\right)^{(\theta_2 - \theta_1)/(\theta_2 - \theta_0)}$$

That is the new  $\lambda$  values are a compromise between the original value  $\lambda_0$  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  $\lambda$  values are changed deterministically (as described later).

Then the new acceptance ratio is

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

Where

$$\begin{aligned} & \frac{P(death)}{P(birth)} \frac{q_{(K+1 \rightarrow K)}(\theta|\theta^*)}{q_{(K \rightarrow K+1)}(\theta^*|\theta) * P(u)} \\ &= 1 * \frac{\frac{1}{K+1}}{\frac{1}{N-K} * 1} = \frac{N-K}{K+1} \end{aligned}$$

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  $q_{(K+1 \rightarrow K)}(\theta|\theta^*)$  term is transition probability from a parameter state with  $K$   $\theta$ 's to  $K+1$ . The probability of being in the proposed state  $\theta^*$  and proposing jumping back to the current state  $\theta$  is the probability of selecting the newly added changepoint  $\theta_k^*$  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

For the death proposal we randomly select any of the current change points uniformly at random and remove it. Then the two  $\lambda$  values associated with the removed change point (call them  $\lambda_1$  and  $\lambda_2$  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  $\theta_m$  is the theta value that was removed. Then the acceptance rate is computed as

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

Where

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

The probability of proposing the death of change point  $\theta_m$  is  $1/K$  since it is chosen uniformly at random from  $\theta$  which has dimension  $K$ . The  $q_{(K-1 \rightarrow K)}(\theta|\theta^*)$  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  $\theta^*$  vector, the probability of birthing the  $\theta_m$  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 proposed graph is stored as an adjacency list. To create make proposals in the graph space the follow operators are used.

### 6.1 add and delete

The `add_edge_op()` functions proposes adding an edge to the graph by sampling uniformly at random from all edges in not currently in the graph. Let  $M = \binom{N_v}{2}$  be the maximum number of possible edges in the graph. Then  $q()$

The actual algorithm works as sfollows. The current number of edges  $N_e$  is tracked. Then we randomly sample from  $A = \{1, \dots, N_v(N_v - 1) - 2 * N_e\}$  then tracking the length of the adjacency lists, we decrement A by the  $N_v - length(adj[i])$  where  $length(adj[i])$  is the length of the adjacency list of vertex 1 and  $i \in \{1, \dots, N_v\}$ . If  $A >= 0$  then  $i = i + 1$  while while continuing to decrement A until  $A < 0$ . Then  $i$  is the index of the adj which has the edge that needs to be added. Then perform a linear search until the remaining i'th index is removed.

We need  $N\_v$

1.  $A = N_v(N_v - 1) - N_e$
2. for i in (1: $N\_v$ ):
3. if ( $A < length(adj[i])$ )
4. perform linear search to find the first  $A - length(adj[i])$  edge not in the list
5. else  $A = A - length(adj[i])$

Then the probability of selecting an edge is  $2/(N_v(N_v - 1) - 2 * N_e)$  by symmetry, each edge is represented twice.

$$\frac{\frac{1}{N_e+1}}{\frac{1}{M-E}} = \frac{M-E}{N_e+1}$$

and the Hastings ratio for removing an edge is given by

$$\frac{\frac{1}{N-(E-1)}}{\frac{1}{E}} = \frac{edges}{N_E - (E - 1)}$$

## Delete edge

The `delete_edge_op` samples uniformly at random an edge currently in the graph and proposes to remove it.

## 6.2 flip\_edge

Flip edge randomly samples an edge by randomly sampling two vertices at random without replacement. If the edge is currently in the adjacency list then it is removed, if it is not in the adjacency list then it is added. Since reversing a flip is equivalent to flipping that edge again, the Hastings ratio is 1.

This proposal does not appear to mix well and this maybe due to the fact that when the graph is sparse then the probability of adding an edge is higher than deleting an edge and vice versa.

## 6.3 Degree Preserving Swap

The degree preserving swap was implemented to allow changes to the graph structure while maintaining the degree of connectivity of each vertex. 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.

## 6.4 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

## 7.1 Two-Component Model on Simulated Data

## 7.2 Graph Estimation on Fixed Two-Component Model

# 8 Conclusion/Discussion

## Citations

Held, Leonhard, Mathias Hofmann, Michael Höhle, and Volker Schmid. 2006. “A Two-Component Model for Counts of Infectious Diseases.” *Biostatistics* 7 (3): 422–37. <https://doi.org/10.1093/biostatistics/kxj016>.