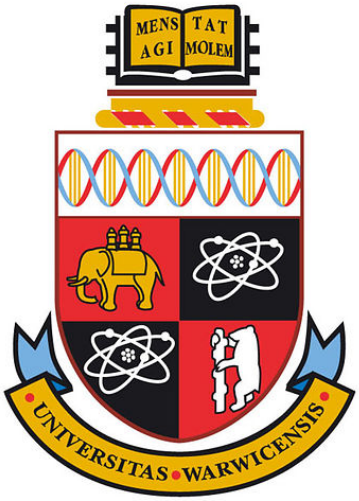
Spatial Outbreak Detection using Markov Chain Monte Carlo (MCMC) Simulations

by

Klein Wang

(ID:1823306)

Supervised by Dr Simon Spencer



Department of Statistics
University of Warwick
Coventry CV4 7AL
United Kingdom

Email: Klein.Wang@warwick.ac.uk

5 May 2022

Dissertation submitted in partial fulfillment of the requirements for the degree of Integrated Master of MORSE in the University of Warwick

Spatial Outbreak Detection

May 5, 2022

Abstract

The COVID-19 pandemic has facilitated an enhanced focus on infectious diseases and global epidemic control. Research has shown that spatial information plays an important role in the field of epidemiology, to predict potential outbreaks given historical infection data. This study aims to determine how to identify outbreaks of disease from a background of sporadic cases. We also discussed how public health centres could make use of those information to efficiently allocate medical resources to different regions. In particular, a spatial outbreak detection method is developed to use information from neighboring locations to improve the discrimination of an outbreak.

Based on historical research on outbreak prediction of different diseases, using Markov chain Monte Carlo simulations, two outbreak models were constructed in this study. After studying the detection power of both models given different choices of parameters, I implemented the spatial detection model on Maryland dataset (of COVID-19 cases) to evaluate the model predictability. It was found that when the estimation of isolated infectious effect is relatively small in log-scale ($\in (ln(0.5), ln(1.5))$), the spatial outbreak detection method has a better performance in predicting outbreaks. And as expected, there is an increasing detection accuracy when there is a larger difference of infectious cases between different locations within a time period.

Overview

This paper discusses the spatio-temporal effects on disease outbreak detection models, which is commonly used in the field of epidemiology. In particular, a comparison study between the spatially independent model 3.1 and spatially dependent model 3.2 was conducted to evaluate the spatial effect on predictability of an outbreak.

The MCMC methods and Bayesian statistics were introduced in Section 2 to study the statistical characteristics of the target outbreak distributions. By constructing the corresponding posterior distribution, I used different sampling methods (Gibbs Sampler 2.4.1, Metropolis-Hastings 2.4.2, Dirichlet random walk 2.4.3) to sample model parameters using MCMC. Then I used relevant plots and ROC curves 4 to evaluate the predictability of both outbreak models.

To measure the findings in a real-life scenario, I adopted the spatially dependent model on Maryland Covid data 5 (looking at positive cases only), collected from Mar 2020 to Mar 2022. In later section 6, the findings and interpretations that we draw from the MCMC simulated outputs are discussed in respect to the outbreak models. I also proposed several future study approaches to overcome the challenges and limitations in this project.

Acknowledgements

The relevant coding for data processing and MCMC methods can be found on **GitHub** repository: spatial outbreak detection.

Special thanks to my supervisor Dr Simon Spencer, who has offered me valuable advice and has shared his research experience with me during the project. Part of the methodology (Section 3) used in this paper is adopted from Simon's work in disease studies (leptospirosis [Ben21], campylobacteriosis [Spe11]). Also, I want to thank Professor Martyn Plummer, who has been offering me academic as well as mental guidance as my person tutor and also as the lead author of JAGS (Just Another Gibbs Sampler), a program for Bayesian modelling using Markov Chain Monte Carlo.

Contents

1 Introduction

1.1 What is an outbreak?

An outbreak is a cluster of disease cases that are close together in time and space. If outbreaks can be detected by the public health authorities then they can be investigated to determine if there is a common cause, such as a contaminated water supply or restaurant with poor food hygiene. Once a common cause has been established then action can be taken to prevent future cases.

1.2 Aims and Objectives

The aim of this paper is to identify outbreaks of disease from a background of sporadic cases. In particular, we aim to develop a spatial outbreak detection method that can use information from neighboring locations to improve the discrimination of an outbreak. In many applications there will be spatial and temporal variations in the background sporadic cases that also need to be captured by the model, for example due to seasonality. However, in this project we assume for simplicity that the distribution of sporadic cases is constant in both space and time.

In aspects of the overall purpose, we aim to explore:

- What is the grey area in a disease outbreak, that makes it difficult to be detection?
- By introducing spatial information to the outbreak model, what impact would it have to the probability of detecting an outbreak?
- How do we detect a spatial outbreak?

1.2.1 Project Structure

Timeline	Objectives
1st stage	Build simple independent outbreak model. Construct the MCMC algorithms. Estimate the level of increase in risk for an outbreak to occur.
2nd stage	Introduce adjacent matrix to reflect spatial information. Construct dynamic probability function for the spatially dependent model. Build spatial dependent outbreak model. Construct the modified MCMC algorithms. Estimate the level of risk and neighboring effects for an outbreak to occur.
3rd stage	Implement real-life disease data. Evaluate the outbreak pattern across regions. Propose future work.

Table 1: Project Structure

In the first stage of the project, I built a simple spatially independent outbreak model to estimate the probability for an outbreak to occur. From there, I started to observe the relationship between the estimated risk level and the grey area under simulation where it became difficult to detect whether an outbreak was occurred or not. Corresponding analysis was followed to visualise the simulation.

In the second stage, I constructed an adjacent matrix to introduce the localised information to the model, which would remove the spatial independence assumption among different regions. Using the updated spatially dependent outbreak model, I observed the dynamic outbreak pattern, and in particular, the probability distribution of the model.

In the final stage of the project, I further explored the open problems in the field of this study. To facilitate practical analysis, I applied the spatially dependent model to the coronavirus data from Maryland, US. In the end, a comparison study between our models and other existing methods is given, to propose potential research topics in the future work.

Motivation Given the context of the coronavirus, people may realise the significance and benefit of being able to detect potential outbreaks. Especially for areas that are lacking modern surveillance systems from public health agencies, researchers may find it more challenging to perform trend analysis of future infections, given limited individual health data. In such a case, by taking the spatial information into account, we may have better confidence in determining a potential outbreak event.

In this study, a parametric outbreak detection model (Section 3.1, 3.2) is constructed in Bayesian settings (discussion followed in Section 2.3) to represent real-life disease outbreaks within some locations.

1.3 Challenges

1.3.1 Measure of spatial information

The epidemiology can be complex with many possible transmission routes. Even when looking at a single disease such as campylobacteriosis, we would discover a large variation between years in the timing, duration and peak incidence of the seasonal pattern of the epidemic [Spe11].

We need to be cautious in constructing the adjacent metric among regions. As stated in historical research of spatial-temporal model, locations can be grouped using different selection approaches (via distance, water supply, census area [Spe11]). Any heterogeneity in selection would imply that some outbreaks would be affected less than others. This would make some detected outbreaks to be less representative, reducing the application of corresponding epidemiological inferences.

1.3.2 Computational expense of MCMC methods

In many past studies, researchers have been looking at the implications of the statistical large sample theory for computational complexity of Bayesian and quasi-Bayesian estimations [Bel12], which are derived by different random walk algorithms such as Metropolis-Hastings random walk. In particular, a polynomial complexity is established by exploiting the central limit theorem framework, giving a structural restriction on the methods (that the posterior density approaches to a normal density in large studied samples).

2 Topic Background

2.1 Outbreak Detection

In the history of applied statistics, there has been a surge in interest in early detection of infectious disease outbreaks [Unk11], using statistical algorithms, often based on Monte Carlo Simulation. It is essential for the National Public health [Buc08] to provide surveillance for accurate and timely outbreak detection, in order to achieve effective epidemic control.

Within the scope of epidemiology, the Poisson distribution has same practical applications in the study of disease incidence, such as analyzing counts of the number of disease outbreak in a given region or time period. In particular, the distribution would become dynamic when the regional spread is considered, which makes it challenging to find parameters of the model.

2.2 Parametric Model Approach

In the modern study of statistical outbreak models, researchers are trying to incorporate the location(spatial) information to potentially enable localized outbreaks of a disease to be detected, or variations in regional patterns to be identified. In past studies, several surveillance methods require only a cut-off value that categorizes pairs of observations as either being "close/neighboring" or 'not close' (e.g. Rogerson and Kulldorff(2001)). An appropriate adjacency metric or distance metric was then defined to represent the closeness of different geographic units.

In the area of epidemiology, there are many statistical methods being studied and implemented to provide spatial disease surveillance, including Regression, cumulative sum (CUSUM) charts, Space-time scan statistics. In this paper, we will focus on Spatio-temporal regression methods. In particular, for an area level model, Y_{ij} represents the number of cases at time period $i$in location j, where Y_{ij} follows a Poisson distribution. Interest would then be centred on estimating the Poisson mean (Lawson et al., 2003; Vidal Rodeiroand Lawson, 2006; Watkins et al., 2009; Zhou and Lawson, 2008), which varied with i and j.

2.2.1 Frequentist Approach

To estimate statistical information of our assumed outbreak distribution, many historical studies were established using Spatial Scan Statistics, in which the key idea is to search for areas of maximum disease activity by computing a score of a likelihood ratio of having an outbreak in each considered cluster versus no outbreak. Many modified methods were proposed from the idea of Spatial Scan statistics, such as FSS (flexible spatial scan statistics) and ULS (upper level set scan statistics).

However, in terms of real life disease data, we might find it difficult to construct full conditional distribution of some parametric models, from the sample observations. This leads to another research approach using the Bayesian setting.

2.2.2 Bayesian Approach

Bayesian-based spatial scan statistics was proposed to overcome the lack of information on distribution, using Bayes theorem. The Bayesian approach suggests the idea to identify a rectangular sub-region, which is composed of the cells within the highest posterior probability of having an outbreak. There are also other popular methods such as Bayesian-based multilevel spatial clustering and z-score based multilevel spatial clustering.

However, the use of real-life spatial information can be computationally demanding (as mentioned in Section 1.3.2), which results in being less practical for surveillance systems that monitor hundreds of disease organisms — computer limitations will restrict the complexity of calculations that can be performed for each disease. Hence in the main body of this project, the outbreak model is generalised under several assumptions of parameter distributions and spatial measurement.

2.3 Bayesian Statistics

In Bayesian Statistics, the prior distribution represents the uncertainty or background knowledge about a particular event or population prior to data sampling, and the posterior distribution represents updated conditional probability of a population attribute based on observed data as well as prior information [Bol07].

To compute and update conditional probabilities after sampling the data, we would construct a Markov chain Monte Carlo (MCMC) to simulate our outbreak data within a given time period and locations. this Bayesian statistical method is based on Bayes' theorem, which s first proposed by Thomas Bayes in a paper he published in 1763.

2.3.1 Bayes' theorem

The theorem is defined as below: Given two events A and B, the conditional probability of A given that B is true is expressed as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)},$$

where $P(B) \neq 0$.

In the implementation on simulating outbreak data, we have a subject belief about the distributions of the parameters (Θ), denoted as the prior density $P(\Theta)$, and a likelihood function of the data points D given the parameters, denoted as $L(D|\Theta)$.

It is difficult to compute the exact value of $P(D)$ (i.e. the full evidence from the data D). Therefore, the posterior density $P(\Theta|D)$ (i.e. the conditional probability of model parameters Θ given the evidence from the data) would be obtained without computing $P(D)$.

$$P(\Theta|D) \propto L(D|\Theta)P(\Theta)$$

2.3.2 Establishing posterior distribution

Under the assumption that the number of disease (infectious cases) Y_{ij}, given time i and location j, follows a Poisson distribution ($Y_{ij} \sim Poi(\lambda_{ij})$), we want to find the distribution parameter λ_{ij} for our model. By definition, posterior probability is the probability that an event will happen after all evidence or background information has been taken into account. In our case, the posterior distribution of λ_{ij} given Y_{ij}: $P(\lambda_{ij}|Y_{ij})$ can be derived from prior distribution $P(\lambda_{ij})$ and likelihood function $P(Y_{ij}|\lambda_{ij})$, using Bayes' rule.

Prior First, we need to define a prior distribution over the model parameters ϕ, which represents the whole parameter set. From the prior distribution, we are explicitly expressing our prior uncertainty about plausible values of λ_{ij}.

Given the definition of ϕ and a predictor variable $\mathbf{X}$ ($:= \{x_{ij}\}, \forall i, j$)), we can compute the prior density of λ_{ij} as below:

$$P(\lambda_{ij}) = P(\lambda_{ij}|\phi, x_{ij})$$

Likelihood Recall the observation Y_{ij} given parameter λ_{ij} follows a Poisson distribution ($Y_{ij} \sim Poi(\lambda_{ij})$). Then, we construct the likelihood function for the response variable $\mathbf{Y}$ and distribution parameter λ, where both $\mathbf{Y}$ and λ are in matrix form ($\mathbf{Y} := \{Y_{ij}\}, \lambda := \{\lambda_{ij}\}, \forall i, j$).

$$P(\mathbf{Y}|\lambda) = \prod_i \prod_j P(Y_{ij}|\lambda_{ij})$$

and,

$$\begin{aligned} P(Y_{ij}|\lambda_{ij}) &= f(Y_{ij}, \lambda_{ij}) \\ &= \frac{e^{-\lambda_{ij}} \lambda_{ij}^{Y_{ij}}}{Y_{ij}!} \end{aligned}$$

Posterior As stated in the Bayes' theorem: $P(A|B) \propto P(B|A)P(A)$, we can have approximated posterior as the following:

$$\begin{aligned} P(\lambda|\mathbf{Y}) &= \frac{P(\mathbf{Y}|\lambda)P(\lambda)}{P(\mathbf{Y})} \\ &\propto P(\mathbf{Y}|\lambda)P(\lambda) \\ &= \prod_i \prod_j P(Y_{ij}|\lambda_{ij})P(\lambda_{ij}|\phi, x_{ij}) \end{aligned}$$

2.3.3 Bayesian Hierarchical model

Bayesian hierarchical modelling is a statistical model established in multiple levels (hierarchical form) that estimates different parameters of the posterior distribution using the Bayesian statistics. In general, hierarchical Bayesian models can be represented as a directed acyclic graph (DAG) with successive conditional probabilities flowing from a prior assumptions on the base level of model parameters.

Unlike in the regular Bayesian model where we have that posterior density is proportional to the multiplication of the likelihood and prior density:

$$P(\Theta|D) \propto P(D|\Theta) \cdot P(\Theta),$$

in hierarchical Bayesian modeling, we have individual parameters and population parameters. For example, we can denote Θ_i as the individual parameter to observation D_i and α as the population parameter. Then we can compute the posterior density as following:

$$\begin{aligned} P(\alpha, \Theta|D) &\propto P(D|\Theta, \alpha) \cdot P(\Theta|\alpha) \cdot P(\alpha) \\ &= P(\alpha) \cdot \prod_i P(D_i|\Theta_i, \alpha) \cdot \prod_i P(\Theta_i|\alpha) \end{aligned}$$

2.4 Markov chain Monte Carlo (MCMC)

In Bayesian inference, the model parameters are regarded as random variables. As stated in Bayes' theorem, we can update the information about the parameters in terms of posterior density (defined as the normalised product of the prior density and the likelihood), from which one can obtain point and interval estimates (e.g. mode, mean, confidence interval).

As mentioned in the topic background, the difficulties arise when dealing with disease outbreak data. Due to the fact that many infection processes in real life are hard to observe, the use of data imputation method is naturally considered to overcome this problem. There are two widely used data imputation methods: EM algorithm and Markov chain Monte Carlo (MCMC) methods. A main drawback with EM algorithm for epidemic inference problems (see [Bec97]) is that the evaluation of the expectation step can be rather complicated.

In this study, we introduce Markov chain Monte Carlo (MCMC) methods to help estimate an unknown probability distribution for an outbreak process in a straightforward approach. Various simulations would be built to develop prospective surveillance on a timely basis across the locations [Ham13]. Mathematically speaking, the methods would allow us to generate various Markov chains, and each of whose stationary distributions is the distribution of our parameters of interest.

In combination with the Bayesian approach, MCMC enables analysis of the full model parameters. Posterior summaries such as means, medians, variances, intervals, etc. can all be easily obtained for individual parameters, or for joint distributions of parameters.

2.4.1 Gibbs sampling

Gibbs Sampling (also called *alternating conditional sampling*) was first proposed by Geman and Geman(1984) and further developed by Gelfand and Smith(1990). The target distribution is the invariant distribution of the Markov chain generated by the algorithm, to which it converges through iterations.

It is an efficient way of reducing a multi-dimensional problem to a lower-dimensional problem. The model parameter vector is subdivided into smaller subvectors (e.g. vectors with a single parameter). One iteration of the algorithm results in each subvector randomly sampled using the subvector's posterior density, conditional on the other subvector's current values (Duchateau & Janssen, [Duc11]). The conditional distributions used in the Gibbs sampler are often referred to as full conditionals: being conditional upon everything except the variable being sampled at each step.

Algorithm - systematic scan Given a n-dimensional variable $x : (x_1, x_2, ..., x_n)$, Gibbs sampler would start with $x^{(0)} := (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)})$ iterate for $t = 1, 2, ...$

1. Draw $x_1^{(t)} \sim f_{x_1|x_{-1}}(\cdot|x_2^{(t-1)}, ..., x_n^{(t-1)})$
   ...
2. Draw $x_i^{(t)} \sim f_{x_1|x_{-1}}(\cdot|x_1^{(t)}, ..., x_{i-1}^{(t)}, x_{i+1}^{(t-1)}, ..., x_n^{(t-1)})$
   ...
3. Draw $x_n^{(t)} \sim f_{x_n|x_{-n}}(\cdot|x_1^{(t)}, ..., x_{n-1}^{(t)})$

The method gives an accurate representation of joint posterior densities. In order to sample from all of the conditional distributions for all model parameters, the required conditional distributions will be sampled by **Metropolis Hastings** first.

2.4.2 Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm is a general Markov Chain Monte Carlo (MCMC) technique for sampling from a probability distribution that is difficult to sample from directly. The idea of the algorithm was developed from Metropolis et al. (1953) and Hastings (1970), which is based on proposing values sampled from an instrumental distribution, and the proposals are then accepted with a certain probability that reflects how likely it is that they are from the target distribution f.

Using this technique, it allows us to approximate the conditional distribution (e.g. to draw samples from the posterior) of the model. The algorithm allows for local updates, i.e. let the proposal value depend on the last accepted value. This makes it easier to come up with a suitable (conditional) proposal, at the price of yielding a Markov chain instead of a sequence of independent realisations. Since our models have multi-dimensional distributions, Metropolis-Hasting appears to be an efficient sampling technique, comparing with other sampling methods (e.g. rejection sampling, importance sampling).

Algorithm Given a n-dimensional variable $x : (x_1, x_2, ..., x_n)$, Metropolis-Hastings would start with $x^{(0)} := (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)})$ iterate for $t = 1, 2, ...$

1. Draw $x \sim q(\cdot|x^{(t-1)})$ (Here, $q(\cdot)$ is the proposal distribution).
2. Compute the acceptance probability/ratio

$$\alpha(x|x^{(t-1)}) = min\{1, \frac{f(x) \cdot q(x^{(t-1)}|x)}{f(x^{(t-1)}) \cdot q(x|x^{(t-1)})}\}$$

3. With probability $\alpha(x|x^{(t-1)})$, set $x^t = x$, and set $x^t = x^{(t-1)}$ otherwise.

2.4.3 Adaptive Dirichlet random walk

Gelman et al. (1997) propose to adjust the proposal distributions such as the overall acceptance rate is around 1/2 for one or two dimensional target distributions, and around 1/4 for proposals in higher dimensional spaces. In our case of study, since both spatially dependent and independent outbreak model are expected to have high-dimensional target distributions, we aim to propose MCMC methods to achieve an acceptance rate at around 25%.

In the field of epidemiology, an adaptive Dirichlet random walk scheme was first introduced in a recent study on leptospirosis notification data in New Zealand using a Bayesian model [Ben21]. To approach the target acceptance rate for the MCMC methods used in the paper, the authors applied this adaptive Dirichlet random walk scheme during the burn-in period of the MCMC.

In particular, the mixing parameter z is adjusted automatically under the following rules:

- Proposal $\theta \sim Dir(z \cdot \theta + z_\theta)$
- Accept the proposal with probability $min\{1, \frac{\pi(\theta|x) \cdot q(\theta^{(t-1)}|z \cdot \theta + z_\theta)}{\pi(\theta^{(t-1)}|x) \cdot q(\theta|z \cdot \theta^{(t-1)} + z_\theta)}\}$
- If a proposal is accepted, set $z^* = max\{0, z - 3\}$
- If a proposal is rejected, set $z^* = z + 1$

With this adjustment, the proposal would be concentrated around the current location if the acceptance rate is too low.

2.4.4 Diagnostic tools over the convergence

After discussing over various sampling-based algorithms implemented in MCMC methods, I would like to introduce some practical diagnostic tools of MCMC, that have been widely used to examine the convergence of the Markov chains.

In the recent studies of analyzing MCMC output ([Vat19]), there are two common questions to evaluate from the outputs:

1. Deciding when the Markov chain converges enough to our target distribution? (i.e. the stationary distribution of the chain has produced a representative sample from probability distribution of the model we proposed.)

2. When the iteration is large enough for us to estimate our parameter sets using the samples?

There are currently graphical approaches led by many practitioners to approach the questions based on trace plots of the simulation components, histograms representing the posterior distribution of the parameters and other ad-hoc convergence diagnostics (Cowles and Carlin, 1996).

In terms of deciding the proper number of iterations for MCMC, there's no theoretical approach to guide assessing the convergence of sums of the MCMC samples, but there are heuristic ones to suggest reasonable choice. One of these is Effective Sample Size (ESS) [Vat19]. The key idea behind ESS is to compute a "exchange rate" between dependent and independent samples, as to measure the number of effectively independent draws from the posterior distribution that the Markov chain is equivalent to.

The definition of ESS is as followings:

$$\text{ESS} = \frac{n}{1 + 2\sum_{k=1}^{\infty} \rho(k)},$$

where n is the number of samples and $\rho(k)$ is the correlation at lag k.

Another heuristic approach is to look at the Gelman-Rubin-Brooks statistic $\hat{R}$ [Kas98](Vats and Knudson, 2018), which is denoted as:

$$\hat{R} \approx \sqrt{1 + \frac{1}{\text{ESS}}}$$

Strictly speaking, complete convergence does not occur for any finite iterations n (Roy, 2019), but the choice of starting value (Rosenthal, 1995) would have a substantial impact on the rate of the convergence. In particular, when we start a MCMC simulation in an area of low probability of our target distribution, the convergence of the Markov chain may appear to be quite slow (Gilks et al., 1996). Therefore, choosing good starting values can save considerable time in terms of computation and increase confidence in the outputs.

2.4.5 Choosing starting values

In order to choose a proper starting value for our parameter set of interest, I decided to first use MCMC methods for a simple independent model (in Section 3.1).

The idea of starting with a simple Independent model is to aim for a low-dimensional target distribution, in which we can use an accept-reject sampler to produce one sample from the target, and pick reasonable values in a high probability region as the starting values. Here in the Bayesian inference setting, we also rely on the prior information of our parameters given their chosen prior distributions when we are choosing the reasonable starting values.

3 Models and Methodology

3.1 Spatially Independent Model

To monitor the spread of diseases, we want to measure the number of potential infectious cases within a given time. Recall in Section 2.3.2, we have defined that the number of infectious cases Y_{ij} follows a Poisson distribution: $Y_{ij} \sim Poi(\lambda_{ij})$, where λ_{ij} represents the frequency of having an infectious case of disease at time period i in location j.

To reflect the exponential growth of λ_{ij} based on x_{ij} (outbreak indicator), we start with a simple log regression model: $ln(\lambda_{ij}) = \alpha + \beta x_{ij}$.

For the parameters in our regression model, α represents the sporadic risk of having an infectious disease and β represents the additional risk cause by regional outbreaks, identified by a binary variable x_{ij} (in which $x_{ij} = 1$ leads to higher risk of having infectious diseases). Under the prior assumption, x_{ij} is following a Bernoulli distribution: $x_{ij} \sim Bernoulli(p)$ (also written as $x_{ij} \sim Binomial(1, p)$), in which an outbreak occurs when $x_{ij} = 1$ and no outbreak occurs otherwise($x_{ij} = 0$).

3.1.1 Assumptions

In this model, following assumptions are made:

- Each region has an independent outbreak distribution, i.e. spatial correlation are excluded in the model.
- Probability p of having an outbreak is constant across the time scope.

3.1.2 Parameters

Based on the spatial independence assumption, we have that x_{ij} is independent on all $x_{ik}|k \neq j$ (i.e. the outbreak predictability for each location doesn't depend on any other locations). Therefore, we can evaluate the outbreak predictability by only looking at one location instead, given the fact that theoretically it would yield the same result as we model with multiple locations.

Hence, we have the representation of the predictor variable $\mathbf{X}$ as $\{x_{ij}, \forall i \in (1, 2, \ldots, n), \forall j \in (1, \ldots, m)|m = 1\}$, which means that we are only focusing on one chosen location, indicated by location index 1. The predictor variable $\mathbf{X}$ can be simplified as: $\{x_{i1}, \forall i \in (1, 2, \ldots, n)\}$

Other model parameters can be simplified correspondingly as below:

- $Y_{i1} \sim Poisson(\lambda_{i1})$
- $ln(\lambda_{i1}) = \alpha + \beta x_{i1}$
- $x_{i1} \sim Bernoulli(p)$
- $p \sim Beta(a, b)$

Notice here probability p of having higher risk of an outbreak is assumed to be following a beta distribution $Beta(1, 12)$ with mean $\frac{a}{a+b} = \frac{4}{52}$, in which the beta distribution is known as the conjugate prior distribution for binomial(or Bernoulli) distribution. Here we assumed that the outbreak would occur around four times every year (52 weeks). The probability is independent among different locations and time(weeks) in the first model. That's why we refer this model as being **Spatially Independent**.

3.1.3 Generate data

Based on the assumptions that we have made for the Spatially Independent model, we can start to generate our sample data for predictor variable **X** and response variable **Y** based on different choices of the true values of model parameters (table 2). Discussion on the model performance (predictability on the outbreaks) is followed in section 4.1.2.

Trials	α	β	p
1	ln(1)	ln(2)	4/52
2	ln(1)	ln(2)	7/52
3	ln(1)	ln(3)	7/52
4	ln(1)	ln(4)	4/52
5	ln(1)	ln(4)	7/52
6	ln(2)	ln(2)	4/52
7	ln(2)	ln(2)	7/52
8	ln(2)	ln(3)	7/52
9	ln(2)	ln(4)	4/52
10	ln(2)	ln(4)	7/52
11	ln(4)	ln(2)	4/52
12	ln(4)	ln(2)	7/52
13	ln(4)	ln(4)	4/52
14	ln(4)	ln(4)	7/52
15	ln(4)	ln(10)	7/52
16	ln(4)	ln(10)	15/52

Table 2: Choice of true values in generated dataset

The true values under different choices are set to all model parameters $\phi = (\alpha, \beta, p)$. Here, we pick e^{α} to assume the fact that an infectious disease would normally occur e^{α} times within a given time period. And in the case of having a localised outbreak (i.e. $x_{i1} = 1$), we assume that the disease would occur e^{β} times more frequently every year, which gives:

$$\lambda_{i1} = \begin{cases} \mathrm{e}^{\alpha+\beta x_{i1}} = e^{\alpha} \cdot e^{\beta}, \text{ when } x_{i1} = 1 \\ \mathrm{e}^{\alpha+\beta x_{i1}} = e^{\alpha}, \text{ when } x_{i1} = 0 \end{cases}$$

Then given the total time length as n and outbreak probability as p, we use function $rbinom()$ in R to generate x_{i1} for each time i. In particular, we first generate x_{i1} for each time $i \in (1, 2, ..., n)$. After we successfully construct the sample data for binary variable **X**, we can computing the corresponding rate parameter λ_{i1} for the Poisson distribution that outcome variable Y_{i1} follows.

$$ln(\lambda_{i1}) = \alpha + \beta x_{i1}.$$

Using the information we get from λ_{i1}, we can further generate sample data for variable Y_{i1} with build-in function $rpois()$ in R.

3.2 Spatial Dependent Model

Based on the build-up of the initial Spatial Independent Model, we then begin to consider adding neighbouring effects of outbreak disease for different studies locations. In particular, we want to construct a

dynamic success probability p in terms of a function of both background isolated risk level and weighted spatio-temporal effects.

In the study of Spatial Scan Statistics, there are three key components being considered: the geometry of the area being scanned, the probability distribution generating events and the shapes, sizes of the scanning window. Here, we focus on constructing the dynamic probability function based on an adjacent distance metric.

3.2.1 Distance Metric

In real-life setting, designing a distance metric for the outbreak model can be very complicated (mentioned in Section 1.3.1). In particular, there are many different choices of measurement to reflect the spatial information among the regional infection.

One simple measurement on a two-dimension map would be using the Euclidean distance to compute the direct distance between the centroids of two locations. Based on the spreading pathways of different diseases, past researchers had also considered using a water supply map [Spe11] to group the nearby locations. Traffic routes among major airports, railway and bus stations could also help consider grouping two locations which are not geometrically nearby but are in fact closely connected via transportation.

In our case study, we are studying at the outbreak distribution among the inner London area, in which there are 14 boroughs involved. Inner London is smaller than Outer London both in terms of population and area, but the density of population is more than double that of Outer London. This would yield much higher risk for people living in inner London area to get exposed to various diseases, especially for those (e.g. coronavirus) can be easily spread through everyday human interactions. That's why we had our focus in this particular area.

List of boroughs In our generated data, the spatial information is derived based on the Euclidean distance among the 14 boroughs 3 in the inner London area.

location index	borough
1	Haringey
2	Camden Town
3	Islington
4	Hackney
5	Hammersmith and Fulham
6	Kensington and Chelsea
7	Westminster
8	City of London
9	Tower Hamlets
10	Newham
11	Wandsworth
12	Lambeth
13	Southwark
14	Lewisham

Table 3: Boroughs in the inner London area

The figure below 1 shows the geographic allocation of 14 boroughs within the inner London area.

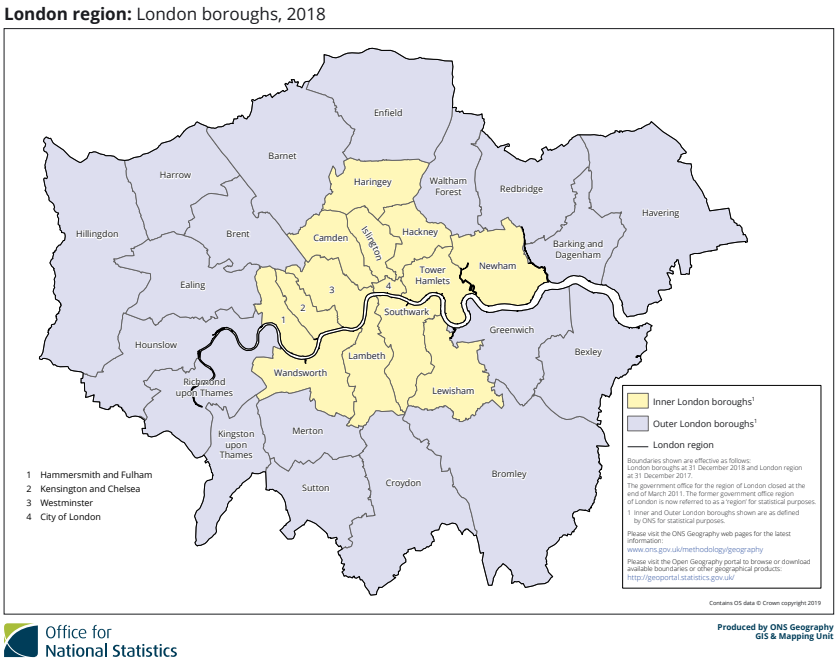


Figure 1: London region: London boroughs 2018

3.2.2 Assumptions

In Spatially Independent model, we have made two assumptions:

- Outbreak distribution is spatially independent in each location;
- Probability p of having higher risk of an outbreak is constant;

After considering neighbouring effects from nearby locations, we further relax those two assumptions:

- Outbreak distribution is spatially dependent in each location at a given time;
- Probability p of having higher risk of an outbreak is dependent on the number of outbreaks happening in the neighboring locations;

In additional to that, in terms of measuring the number of infectious cases Y_{ij} for different location j at a fixed time point i, we assume that:

- All locations are assumed to have the same population size,

which would make the outcome variable $\mathbf{Y}$ be completely dependent on the rate parameter λ, without considering any variation of population sizes in different locations.

3.2.3 Dynamic Outbreak Probability

Here in Spatial Dependent Model, we construct a dynamic probability function to reflect dynamic risk level. We denote p_{ij} as the probability of having an outbreak at time i in location j. Given time i, to consider all neighboring effects for location j, we derive our outbreak probability p_{ij} based on the spatial information from set $x_{i,-j} = (x_{i1}, \ldots, x_{i,j-1}, x_{i,j+1}, \ldots, x_{im})$, where m stands for the number of locations within the study region. i.e. the Inner London Area ($m = 14$). Hence, we have our modified probability distribution as $x_{ij} \sim Binomial(1, p_{ij}|x_{i,-j})$.

In our case, the dynamic outbreak probability will be consisted of two components: the background isolated risk term and weighted spatio-temporal risk term. When considering different neighboring locations, we assign the neighboring location with inverse distance weight (i.e. $d^{-1} := \{d_{kl}^{-1} | k \neq l\}, \forall k, l \in (1, 2, \ldots, m)$) to reflect the corresponding spatio-temporal effects.

Formulation Given the inverse distance metric, the dynamic probability for location j (by considering timely variable index i) can be formulated as the following:

$$p_{ij} = \gamma + \Delta \cdot \sum_{k=1, k\neq j}^{m} \left(\frac{d_{kj}^{-1}}{\sum_{l=1, l\neq j}^{m} d_{lj}^{-1}} \cdot x_{ik} \right),$$

where $x_{ik} \in \{0, 1\}$ is the binary variable that indicates an outbreak or not in location k given current time i. And d_{kj} is denoted as the Euclidean distance from location k to location j.

In many previous applications and studies of Spatial Scan Statistics (such as the circular spatial scan statistics proposed by Kulldorff, and the flexible spatial scan statistic proposed by Tango and Takahashi), the authors had considered detecting possible outbreak clusters using restricted likelihood ratio (frequentist approach) or posterior probability of having an outbreak (Bayesian approach). Therefore, in real application, we may adopt spatial scan statistics to identify sub-regions, and then assign different probabilities/risk levels for different spatial clusters.

For the purpose of reducing computational cost, we simplify the model here by assuming that γ is the background probability of having an isolated outbreak for all regions. We pick reasonable γ (i.e. background risk) from [0.1,0.2].

We have Δ as the spatio-temporal effect measured from the neighboring locations. In the worst case of scenario where all the neighboring locations are having an outbreak, the equation becomes the following:

$$\begin{aligned}
p_{ij} | (x_{i,-j} = 1) =& \gamma + \Delta \cdot \sum_{k\neq j}^{m} \left(\frac{d_{kj}^{-1}}{\sum_{l\neq j}^{m} d_{lj}^{-1}} \cdot 1 \right) \\
=& \gamma + \Delta \cdot \sum_{k\neq j}^{m} \frac{d_{kj}^{-1}}{\sum_{l\neq j} d_{lj}^{-1}} \\
=& \gamma + \Delta \cdot 1 \\
=& \gamma + \Delta
\end{aligned}$$

To simplify the representation, we denote $p_{ij} | (x_{i,-j} = 1)$ as $\widehat{p}$. We can compute the effect as $\Delta = \widehat{p} - \gamma$.

Dirichlet distribution In spatially independent model, outbreak probability p follows a beta distribution $Beta(a, b)$. However after introducing dynamic probability function, an additional hierarchical Bayesian level is added to our outbreak model, in which p is now depending on the choice of a vector of parameters:

$$\begin{pmatrix} \gamma \\ \Delta \end{pmatrix}$$

Note that p no longer follows a beta distribution.

Alternatively, we let the new parameter vector θ follows a multivariate beta distribution (also named as Dirichlet distribution). The Dirichlet distribution is a family of continuous multivariate probability distributions parameterized by a vector a of positive reals. We denote the distribution of θ as $Dir(a)$, where $\theta = (\gamma, \Delta, 1 - \gamma - \Delta)$, and a is a vector of the same length as θ.

In a Dirichlet distribution, the support over the probability simplex is defined as:

$$S_K = \{\theta : 0 \leq \theta_k \leq 1, \sum_{k=1}^{K} \theta_k = 1\}$$

The probability density function is defined as follows:

$$Dir(\theta|a) \triangleq \frac{1}{B(a)} \prod_{k=1}^{K} \theta_k^{a_k - 1} \mathbb{1}(\theta \in S_K),$$

where $B(a_1, ..., a_K)$ is the natural generalisation of the beta function to K variables:

$$B(a) = \frac{\prod_{k=1}^{K} \Gamma(a_k)}{\Gamma(a_0)},$$

where $a_0 = \sum_{k=1}^{K} a_k$.

We can also write as:

$$Dir(\theta|a) \triangleq \frac{\Gamma(\sum_{k=1}^{K} a_k)}{\prod_{k=1}^{K} \Gamma(a_k)} \prod_{k=1}^{K} \theta_k^{a_k - 1} \mathbb{1}(\theta \in S_K)$$

In our case, the parameter vector θ follows a degree 3 Dirichlet distribution ($K = 3$):

$$\theta \sim Dir_3(a),$$

where $a := \{a_1, a_2, a_3\}$ correspondingly with $\theta = (\gamma, \Delta, 1 - \gamma - \Delta)$

3.2.4 Parameters

Having constructed the dynamic probability function, we can re-establish several parameters within the dependent model.

- $Y_{ij} \sim Poisson(\lambda_{ij})$
- $ln(\lambda_{ij}) = \alpha + \beta x_{ij}$
- $x_{ij}|\theta \sim Bernoulli(p_{ij}|x_{i,-j})$
- $p_{ij} = \gamma + \Delta \cdot \sum_{k \neq j} \frac{d_{kj}^{-1}}{\sum_{l \neq j} d_{lj}^{-1}} \times x_{ik}$
- $\theta = (\gamma, \Delta, 1 - \gamma - \Delta) \sim Dir_3(a)$

3.2.5 Generate data

After building up the dynamic probability function, we can start to generate our sample data based on the assumptions that we have made.

As discussed previously in Section 3.1.3 on how to on generate data for spatially independent model, here I adopt the choices of those parameters α, β, p (listed in table 4), and use the corresponding findings to construct a comparison study between the two models. (Note that $\Delta = \widehat{p} - \gamma$, explained in 3.2.3.)

Given the conditional distribution that we have defined for each x_{ij} of location j at time i, each x_{ij} is dependent on $x_{i,-j}$. Due to the dependence assumption, it becomes impossible to write down the corresponding distribution of $\mathbf{X}$ and sample from it. Instead we use the following Gibbs sampler.

Trials	α	β	γ	Δ
1	ln(1)	ln(2)	7/52	0.665
2	ln(1)	ln(3)	7/52	0.665
3	ln(1)	ln(4)	7/52	0.665
4	ln(2)	ln(2)	7/52	0.665
5	ln(2)	ln(3)	7/52	0.665
6	ln(4)	ln(2)	7/52	0.665

Table 4: Choice of true values in generated dataset

For each choice of the model parameters, I start by picking two areas to have outbreaks, which are the City of London, and Southward because they are in the centre of the studies regions, and thus have a larger likelihood in passing the infections to their neighboring locations. Then, the localised outbreak probabilities are calculated for each locations whenever there is a location's outbreak status being updated. Using those locally updated probabilities, we use function $rbinom()$ in R to randomly generate new $\mathbf{X}$ in a new iteration. We keep iterating this computation for given times (e.g. 300 iterations) until the expected neighboring effect becomes stable in the region.

Next, given the total time length as n, we generate x_{ij} for each time i and location j. In particular, we first generate x_{ij} for $j \in (1, 2, ..., m)$ using the dynamic probability function (which depends on the inverse distance matrix $\mathbf{d}^{-1}$ and parameter set (γ, Δ)), and then iterate for n times for each time i ($i \in (1, 2, ..., n)$).

After we successfully construct the sample data for binary variable $\mathbf{X} := \{x_{ij}\}, \forall i, j$, we can computing the corresponding rate parameter $\lambda := \{\lambda_{ij}\}, \forall i, j$ for the Poisson distribution that outcome variable $\mathbf{Y} := \{Y_{ij}\}, \forall i, j$ follows.

$$\lambda_{ij} = e^{\alpha} \cdot e^{\beta x_{ij}},$$

where each $ln(\lambda_{ij}) = \alpha + \beta x_{ij}$.

Using the information we get from λ, we can further generate sample data for variable $\mathbf{Y}$ with build-in function $rpois()$ in R.

3.3 Constructing MCMC in independent model

As mentioned above, we have successfully establish a MCMC simulation for our spatial independent model. There are currently only three parameters included in our model, i.e. α, β, p. We have previously denoted the full parameter set as ϕ. We also have dependent variable $\mathbf{X}$ simplified as $\{x_{i1}\}, \forall i$, representing the existence of an outbreak disease at time i given the chosen location.

3.3.1 Posterior density

As discussed in Bayesian statistics, we can construct the posterior density (denoted as $\pi(\phi|\mathbf{Y})$), using the likelihood $L(\mathbf{Y}|\cdot)$ and prior density $p(\cdot)$. Hence, we have the following:

$$\begin{aligned}
\pi(\phi|\mathbf{Y}) &\propto L(\mathbf{Y}|\phi) \cdot p(\phi) \\
&= L(\mathbf{Y}|\alpha, \beta, \mathbf{X}) \cdot p(\alpha) \cdot p(\beta) \cdot p(\mathbf{X}|p) \cdot p(p) \\
&= L(\mathbf{Y}|\lambda) \cdot p(\alpha) \cdot p(\beta) \cdot p(\mathbf{X}|p) \cdot p(p) \text{ , where } ln(\lambda_{i1}) = \alpha + \beta x_{i1} \\
&= p(\alpha) \cdot p(\beta) \cdot p(p) \cdot p(\mathbf{X}|p) \cdot L(\mathbf{Y}|\lambda) \\
&= p(\alpha) \cdot p(\beta) \cdot p(p) \cdot \prod_{i=1}^{n} f_{Binomial}(\mathbf{X} = x_{i1}; 1, p) \cdot \prod_{i=1}^{n} f_{Poisson}(\mathbf{Y} = Y_{i1}; \lambda_{i1})
\end{aligned}$$

We have chosen the prior density of α, β to each follow an adjusted Gamma distribution:

$$\begin{aligned}
\alpha &\sim \alpha_{min} + \Gamma(1, 6) | \alpha_{min} = ln(0.5) \\
\beta &\sim \beta_{min} + \Gamma(1, 6) | \beta_{min} = ln(0.5)
\end{aligned}$$

We can also construct the prior densities of $x_{i1}|p$ and p based on their prior distributions:

$$\begin{aligned}
x_{i1}|p &\sim Binomial(1, p) \\
p &\sim Beta(a, b)
\end{aligned}$$

, where $\frac{a}{a+b}$ is the mean of outbreak within a given time period.

Hence, we can continue to compute our target posterior probability function using the prior information:

$$\begin{aligned}
\pi(\phi|\mathbf{Y}) \propto & L(\mathbf{Y}|\phi) \cdot p(\phi) \\
= & p(\alpha) \cdot p(\beta) \cdot p(p) \cdot \prod_{i=1}^{n} f_{Binomial}(\mathbf{X} = x_{i1}; 1, p) \cdot \prod_{i=1}^{n} f_{Poisson}(\mathbf{Y} = Y_{i1}; \lambda_{i1}) \\
= & f_{Gamma}(\alpha - \alpha_{min}; 1, 6) \cdot f_{Gamma}(\beta - \beta_{min}; 1, 6) \cdot f_{Beta}(p; a, b) \\
& \cdot \prod_{i=1}^{n} f_{Binomial}(x_{i1}; 1, p) \cdot \prod_{i=1}^{n} f_{Poisson}(\mathbf{Y} = Y_{i1}; \lambda_{i1})
\end{aligned}$$

The probability density function f for each statistical distributions discussed can be found in table 9 in Appendix.

3.3.2 Update α, β

After constructing the posterior distribution for the model, we begin our simulation by updating α,β with **Metropolis-Hasting** algorithm. We have set both proposal distributions of α, β to be Gaussian normal, i.e.

$$\begin{aligned}
q(\alpha_t) &\sim \mathcal{N}(\alpha_{t-1}, 0.2) \\
q(\beta_t) &\sim \mathcal{N}(\beta_{t-1}, 0.2)
\end{aligned}$$

Note that we have added extra conditions to reject β when $\beta < ln(2)$. Since we are using a log link function in our Poisson distribution, e^{β} represents the relative risk during an outbreak (Here we set a β minimum of $ln(2)$ to ensure that model has at least twice the number of cases during an outbreak).

Then, we can compute the acceptance ratio given the new proposed values of α, β.

$$\begin{aligned}
\text{Probability} &= \min\{1, \frac{\pi(\phi^*|\mathbf{Y}) \cdot q(\alpha^*, \beta^*|\alpha, \beta)}{\pi(\phi|\mathbf{Y}) \cdot q(\alpha, \beta|\alpha^*, \beta^*)}\} \\
&= \min\{1, \frac{L(\mathbf{Y}|\phi^*) \cdot p(\alpha^*) \cdot p(\beta^*) \cdot p(\mathbf{X}|p) \cdot p(p) \cdot q(\alpha^*, \beta^*|\alpha, \beta)}{L(\mathbf{Y}|\phi) \cdot p(\alpha) \cdot p(\beta) \cdot p(\mathbf{X}|p) \cdot p(p) \cdot q(\alpha, \beta|\alpha^*, \beta^*)}\} \\
&= \min\{1, \frac{L(\mathbf{Y}|\phi^*) \cdot p(\alpha^*) \cdot p(\beta^*) \cdot q(\alpha^*, \beta^*|\alpha, \beta)}{L(\mathbf{Y}|\phi) \cdot p(\alpha) \cdot p(\beta) \cdot q(\alpha, \beta|\alpha^*, \beta^*)}\}, \\
&\text{where } \phi^* = (\alpha^*, \beta^*, \mathbf{X}) \\
&= \min\{1, \frac{L(\mathbf{Y}|\alpha^*, \beta^*) \cdot p(\alpha^*) \cdot p(\beta^*)}{L(\mathbf{Y}|\alpha, \beta) \cdot p(\alpha) \cdot p(\beta)}\}, \\
&\text{where the proposal density } q(\alpha^*, \beta^*|\alpha, \beta) = q(\alpha, \beta|\alpha^*, \beta^*) \\
&\text{given the symmetric proposal density of } (\alpha, \beta) \\
&= \min\{1, \frac{\pi(\alpha^*, \beta^*|\mathbf{Y})}{\pi(\alpha, \beta|\mathbf{Y})}\}
\end{aligned}$$

After that, we generate a random number from a uniform distribution and compare it with our acceptance ratio. The idea is to accept our proposal values with probability (constructed above). Once we decide whether to accept or reject the proposal values, we move on to update p and indicator variable $\mathbf{X}$, which represents a set of variables $(x_{11}, x_{21}, ..., x_{n1})$. Variable n is representing the time length (in weeks) that we are observing using the model.

3.3.3 Update probability p

Since $\mathbf{X}$ has a conditional distribution on probability p, we first update the Beta distribution that p follows, from

$$p \sim Beta(a, b)$$

to

$$p|x \sim Beta(a + n_{x=1}, b + n_{x=0})$$

, We have $n_{\mathbf{X}=1}$ denoted as the number of x_{i1} that equals to 1 (i.e. $n_{\mathbf{X}=1} = \sum_i \mathbb{1}_{x_{i1}=1}$). Then the new mean of the Beta distribution is given by:

$$\frac{a + n_{\mathbf{X}=1}}{a + b + n}$$

The mean here represents the average probability that has a higher risk of outbreak in a given time period. Then we sample new p, which is proposed from the new Beta distribution.

3.3.4 Update variable X

After that, we can compute the marginal posterior density given $\mathbf{X}$, and use it to sample new $\mathbf{X}$ from Bernoulli distribution ($x \sim Bernoulli(p)$).

According to the Bayes' theorem, we can compute the posterior density of $x_{i1} = 1$ and $x_{i1} = 0$:

$$P(x_{i1} = 1|Y_{i1}, \phi) \propto \mathcal{L}(\mathbf{Y} = Y_{i1}|x_{i1} = 1, \alpha, \beta) \cdot p(x_{i1} = 1|p) \tag{1}$$

$$= f_{Poisson}(Y_{i1}; \lambda_{i1} = e^{\alpha+\beta}) \cdot p \tag{2}$$

$$P(x_{i1} = 0|Y_{i1}, \phi) \propto \mathcal{L}(\mathbf{Y} = Y_{i1}|x_{i1} = 0, \alpha, \beta) \cdot p(x_{i1} = 0|p) \tag{3}$$

$$= f_{Poisson}(Y_{i1}; \lambda_{i1} = e^{\alpha}) \cdot (1 - p) \tag{4}$$

Since we know that the sum of posterior density of $x_{i1} = 1$ and $x_{i1} = 0$ is 1, we can compute the probability easily, using the equations above:

$$P(x_{i1} = 1|Y_{i1}, \phi) = \frac{(1)}{(1) + (3)}$$

To find the marginal posterior density for $\mathbf{X}$, we can compute the normalising constant c first:

$$c = 1/(\mathcal{L}(\mathbf{Y} = Y_{i1}|x_{i1} = 1, \alpha, \beta) \cdot p(x_{i1} = 1|p) + \mathcal{L}(\mathbf{Y} = Y_{i1}|x_{i1} = 0, \alpha, \beta) \cdot p(x_{i1} = 0|p))$$

And then we can use the normalising constant c to further compute the posterior density of $x_{i1} = 1$.

$$\begin{aligned} P(x_{i1} = 1|Y_{i1}, \phi) &= \frac{(1)}{(1) + (3)} \\ &= c \cdot \mathcal{L}(\mathbf{Y} = Y_{i1}|x_{i1} = 1, \alpha, \beta) \cdot p(x_{i1} = 1|p) \end{aligned}$$

Using this marginal posterior density, we sample new x_{i1} from Bernoulli distribution ($x_{i1} \sim Bernoulli(p^*)$, where $p^* = P(x_{i1} = 1|Y_{i1}, \phi)$).

LogSumExp trick In terms of implementing the algorithm in programming language R, we need to cautious that posterior density of $\mathbf{X}$ would be very small, which makes it extremely difficult for the workspace in R to store those numeric values. Alternatively, a trick using the LogSumExp function (LSE) is adopted to overcome the numeric tolerance using the LogSumExp transformation.

In this case, we want to compute $P(x_{i1} = 1|Y_{i1}, \phi)$ using (1) and (2). For the sake of simplicity, here I denote (1), (2) as p_1 and p_2. Then, we have:

$$\begin{aligned} P(x_{i1} = 1|Y_{i1}, \phi) &= \frac{p_1}{p_1 + p_2} \\ &= \frac{e^{log(p_1)}}{e^{log(p_1)} + e^{log(p_2)}}, \end{aligned}$$

where we denote $M = \max(log(p_1), log(p_2))$

$$\begin{aligned} \frac{e^{log(p_1)}}{e^{log(p_1)} + e^{log(p_2)}} &= \frac{e^{log(p_1)+M-M}}{e^{log(p_1)+M-M} + e^{log(p_2)+M-M}} \\ &= \frac{e^M e^{log(p_1)-M}}{e^M(e^{log(p_1)-M} + e^{log(p_2)-M})} \\ &= \frac{e^{log(p_1)-M}}{e^{log(p_1)-M} + e^{log(p_2)-M}} \end{aligned}$$

without losing any generality, let $M = log(p_1)$. Then, we have

$$P(x_{i1} = 1|Y_{i1}, \phi) = \frac{1}{1 + e^{log(p_2)-M}}$$

Using this trick, R would be able to compute the posterior density with enough numeric tolerance.

3.4 Constructing MCMC in spatio-temporal dependent model

Comparing with the MCMC simulation we did for our spatial independent model above, we introduce two additional parameters γ, Δ in our dependent model to represent dynamic probability p. We also have $\mathbf{X} := \{x_{ij}\}$ as dependent variable, indicating the existence of an outbreak disease in each location j at time i.

3.4.1 Posterior density

As discussed in Bayesian statistics, we can construct the posterior density (denoted as $\pi(\phi|\mathbf{Y}), \phi = (\alpha, \beta, \gamma, \Delta)$), using the likelihood $L(\mathbf{Y}|\cdot)$ and prior density $p(\cdot)$. Hence, we have the following:

$$\begin{aligned}\pi(\phi|Y) &\propto L(Y|\phi) \cdot p(\phi) \\ &= L(Y|\alpha, \beta, \mathbf{X}, \theta) \cdot p(\alpha) \cdot p(\beta) \cdot p(\mathbf{X}|\theta) \cdot p(\theta)\end{aligned}$$

Likelihood Similarly to the likelihood we have for spatially independent model, we have the following:

$$L(Y|\alpha, \beta, x, \theta) = \prod_{i=1}^{n}\prod_{j=1}^{m} f_{Poisson}(\mathbf{Y} = Y_{ij}, \lambda_{ij} = \alpha + \beta x_{ij})$$

Note that here we treat $\mathbf{X}$ as the parameter of our Bayesian Hierarchical model instead of the sampled data.

Prior density We have chosen the prior density of α, β to each follow an adjusted Gamma distribution:

$$\begin{aligned}\alpha &\sim \alpha_{min} + \Gamma(1, 6)|\alpha_{min} = ln(0.5) \\ \beta &\sim \beta_{min} + \Gamma(1, 6)|\beta_{min} = ln(0.5)\end{aligned}$$

We can also construct the prior density of θ based on the prior distribution:

$$\theta \sim Dir_3(\theta, z_\theta)$$

However, since we have modified the outbreak probability $p|\theta$ to have a strong dependence with $\mathbf{X}$. In particular, we say $p_{ij}|\theta$ depends on $x_{i(-j)}|\theta$. Having such dependence would it much more difficult to compute the conditional probability $p(x_{ij}|\theta)$.

Instead we proposed a **pseudo likelihood** for $x_{ij}|\theta$ by assuming independence among different x_{ij}. (The concept is first proposed by Julian Besag in [Bes75] to measure the likelihood of spatial dependence data.) Therefore, we can write down the probability $p(x_{ij}|\theta)$ based on the prior distribution $x_{ij}|\theta \sim Binomial(1, p_{ij}|\theta)$, and $\theta = (\gamma, \Delta, 1 - \Delta)$.

$$p^*(\mathbf{X}|\theta) = \prod_{i=1}^{n}\prod_{j=1}^{m} f_{Binomial}(x_{ij}; 1, p_{ij}|\theta)$$

Computation Hence, we can continue to compute our target posterior probability function using the prior information:

$$\begin{aligned}\pi(\phi|Y) \propto& L(Y|\phi) \cdot p(\phi) \\ =& p(\alpha) \cdot p(\beta) \cdot p(\theta) \cdot p(\mathbf{X}|\theta) \cdot L(Y|\alpha, \beta, \mathbf{X}, \theta) \\ =& p(\alpha) \cdot p(\beta) \cdot p(\theta) \cdot \prod_{i=1}^{n}\prod_{j=1}^{m} f_{Binomial}(x_{ij}; 1, p_{ij}|\theta) \cdot \prod_{i=1}^{n}\prod_{j=1}^{m} f_{Poisson}(\mathbf{Y} = Y_{ij}; \lambda_{ij} = \alpha + \beta x_{ij}) \\ =& f_{Gamma}(\alpha - \alpha_{min}; 1, 6) \cdot f_{Gamma}(\beta - \beta_{min}; 1, 6) \cdot f_{Dir}(\theta; z_\theta) \\ & \cdot \prod_{i=1}^{n}\prod_{j=1}^{m} f_{Binomial}(x_{ij}; 1, p_{ij}|\theta) \cdot \prod_{i=1}^{n}\prod_{j=1}^{m} f_{Poisson}(Y_{ij}; \lambda_{ij})\end{aligned}$$

The probability density function f for each statistical distributions discussed can be found in table 9 in Appendix.

3.4.2 Update α, β

After constructing the posterior distribution for the model, we begin our simulation by updating α,β with **Metropolis-Hasting** algorithm. Similarly, we have added extra conditions to reject β when $\beta < ln(2)$.

Both proposal density of α, β are following a Gaussian normal distribution, i.e.

$$q(\alpha_t) \sim \mathcal{N}(\alpha_{t-1}, 0.2)$$
$$q(\beta_t) \sim \mathcal{N}(\beta_{t-1}, 0.2)$$

We can compute the acceptance ratio given the new proposed values of α, β.

$$\begin{aligned}
\text{Probability} &= min\{1, \frac{\pi(\phi^*|Y)}{\pi(\phi|Y)}\} \\
&= min\{1, \frac{L(Y|\phi^*) \cdot p(\alpha^*) \cdot p(\beta^*) \cdot p(x|\theta) \cdot p(\theta) \cdot q(\alpha^*, \beta^*|\alpha, \beta)}{L(Y|\phi) \cdot p(\alpha) \cdot p(\beta) \cdot p(x|\theta) \cdot p(\theta) \cdot q(\alpha, \beta|\alpha^*, \beta^*)}\} \\
&= min\{1, \frac{L(Y|\phi^*) \cdot p(\alpha^*) \cdot p(\beta^*) \cdot q(\alpha^*, \beta^*|\alpha, \beta)}{L(Y|\phi) \cdot p(\alpha) \cdot p(\beta) \cdot q(\alpha, \beta|\alpha^*, \beta^*)}\} \\
&\text{, where } \phi^* = (\alpha^*, \beta^*, \theta) \text{ and } \theta = (\gamma, \Delta, 1-\gamma-\Delta) \\
&= min\{1, \frac{L(Y|\alpha^*, \beta^*) \cdot p(\alpha^*) \cdot p(\beta^*)}{L(Y|\alpha, \beta) \cdot p(\alpha) \cdot p(\beta)}\} \\
&\text{, where the proposal density } q(\alpha^*, \beta^*|\alpha, \beta) = q(\alpha, \beta|\alpha^*, \beta^*) \\
&\text{given the symmetric proposal density of } (\alpha, \beta) \\
&= min\{1, \frac{\pi(\alpha^*, \beta^*|Y)}{\pi(\alpha, \beta|Y)}\}
\end{aligned}$$

Once we decide whether to accept or reject the proposal values, we move on to update γ, Δ and indicator variable $\mathbf{X} := \{x_{ij}, i \in (1, ..., n) \text{ and } j \in (1, ..., m)\}$. Variable n is the time length that we are observing using the model and variable m is the number of locations we are sampling from.

3.4.3 Update γ, Δ

The Dirichlet distribution is a conjugate prior for the multinomial distribution. The benefit of this is that (a) the posterior distribution is easy to compute and (b) it in some sense is possible to quantify how much our beliefs have changed after collecting the data.

With Dirichlet distribution, we can then represent the probabilities for the multinoomial probability vector $\theta = (\gamma, \Delta, 1-\gamma-\Delta)$. And we have the following properties:

- $\gamma + \Delta + (1-\gamma-\Delta) = 1$
- $\gamma, \Delta, 1-\gamma-\Delta \geq 0$

The density of the proposal θ^* can be written as $\theta^* \sim Dir(z \cdot \theta + z_\theta)$. To update θ (i.e. γ, Δ), we apply the adaptive dirichlet random walk algorithm (Section 2.4.3) to compute the acceptance probability:

$$\begin{aligned}
\text{Probability} &= \frac{\pi(\theta^*|\mathbf{X}) \cdot q(\theta|z \cdot \theta^* + z_\theta)}{\pi(\theta|\mathbf{X}) \cdot q(\theta^*|z \cdot \theta + z_\theta)} \\
&= \frac{\mathcal{L}(\mathbf{X}|\theta^*) \cdot p(\theta^*|z_\theta) \cdot q(\theta|z \cdot \theta^* + z_\theta)}{\mathcal{L}(\mathbf{X}|\theta) \cdot p(\theta|z_\theta) \cdot q(\theta^*|z \cdot \theta + z_\theta)} \\
&= \frac{\mathcal{L}(\mathbf{X}|\theta^*) \cdot Dir(\theta^*|z_\theta) \cdot Dir(\theta|z \cdot \theta^* + z_\theta)}{\mathcal{L}(\mathbf{X}|\theta) \cdot Dir(\theta|z_\theta) \cdot Dir(\theta^*|z \cdot \theta + z_\theta)},
\end{aligned}$$

where $\mathcal{L}(\cdot), p(\cdot), q(\cdot)$ represent the likelihood, prior density and proposal density respectively. Then, we can update the new proposal of θ with the acceptance probability.

3.4.4 Update variable X

After that, we can compute the marginal posterior density given **X**, and use it to sample new x_{ij} from Bernoulli distribution ($x_{ij} \sim Bernoulli(p_{ij}|\theta)$.

According to the Bayes' theorem, we can compute the posterior density of $x_{ij} = 1$ and $x_{ij} = 0$:

$$P(x_{ij} = 1|Y_{ij}, \phi) \propto \mathcal{L}(\mathbf{Y} = Y_{ij}|x_{ij} = 1, \alpha, \beta) \cdot p(x_{ij} = 1|p_{ij}) \tag{5}$$

$$= f_{Poisson}(Y_{ij}; \lambda_{ij} = e^{\alpha+\beta}) \cdot p_{ij} \tag{6}$$

$$P(x_{ij} = 0|Y_{ij}, \phi) \propto \mathcal{L}(\mathbf{Y} = Y_{ij}|x_{ij} = 0, \alpha, \beta) \cdot p(x_{ij} = 0|p_{ij}) \tag{7}$$

$$= f_{Poisson}(Y_{ij}; \lambda_{ij} = e^{\alpha}) \cdot (1 - p_{ij}) \tag{8}$$

Since we know that the sum of posterior density of $x_{ij} = 1$ and $x_{ij} = 0$ is 1, we can compute the probability easily, using the equations above:

$$P(x_{ij} = 1|Y_{ij}, \phi) = \frac{(6)}{(6) + (8)}$$

In terms of implementing the algorithm in programming language R, we can compute the normialising constant c first:

$$c = 1/(\mathcal{L}(\mathbf{Y} = Y_{ij}|x_{ij} = 1, \alpha, \beta) \cdot p(x_{ij} = 1|p_{ij}) + \mathcal{L}(\mathbf{Y} = Y_{ij}|x_{ij} = 0, \alpha, \beta) \cdot p(x_{ij} = 0|p_{ij}))$$

And then we can use the normalising constant c to further compute the posterior density of $x_{ij} = 1$.

$$P(x_{ij} = 1|Y_{ij}, \phi) = \frac{(6)}{(6) + (8)}$$

$$= c \cdot \mathcal{L}(\mathbf{Y} = Y_{ij}|x_{ij} = 1, \alpha, \beta) \cdot p(x_{ij} = 1|p_{ij})$$

Using this marginal posterior density, we sample new x_{ij} from Bernoulli distribution ($x_{ij} \sim Bernoulli(p^*)$, where $p^* = P(x_{ij} = 1|Y_{ij}, \phi)$). Here I adopt the same LogSumExp trick used in the simple model 3.3.4

4 Model Evaluation

4.1 Spatially Independent Model

Following different choices of the true values of the model parameters (listed in Section 3.1.3), data was generated for variables **X** and **Y** based on the distributions assumed for the spatially independent model. Simulations were then draw using the Markov Chain Monte Carlo methods, which simulated variables' values based on the posterior densities.

4.1.1 Results and convergence diagnosis

I investigated the convergence of the Markov chain using different number of iterations and burn-in period. Eventually, I have set the MCMC to have 30000 iterations with the first 5000 iterations as the burn-in period. The trace plots 2 (histograms and Markov chains) are plotted to visualise the convergence.

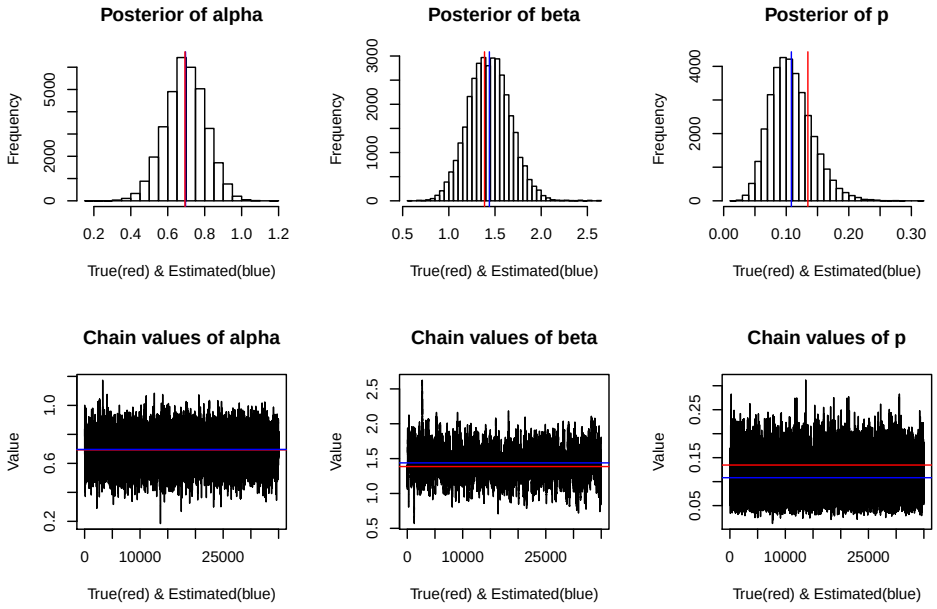


Figure 2: Trace Plot of Parameters for Spatial Independent Model. The red line represents the true value of the parameter, and the blue line represents the estimated value of the parameter.

4.1.2 ROC performance

By looking at the estimated outbreak probability plot for each x_{i1}, we can find that a higher value of true β would give a much better prediction on x_{i1}. Note in the plot 3, the red and blue lines represent for the (0.25%,99.75%) quantile value of outbreak probability, and the vertical green lines represent the week that is having an outbreak. For those weeks that have an outbreak (with true x_{i1} being 1), sampled x_{i1} would tend to have a very large posterior density indicating an outbreak. In the case of higher β, the ROC curve 4 that we plot would receive a higher AUC (area under curve) in between (0.98,1.00).

The summary table 5 shows a comparison of the AUC (area under ROC curve) for different generated data (in trials). Given the table, we can see that the model would have a better predictability on outbreaks with greater value of β in (ln(4),ln(10)) (i.e. greater discrimination between whether having an outbreak or not), resulting in a higher AUC.

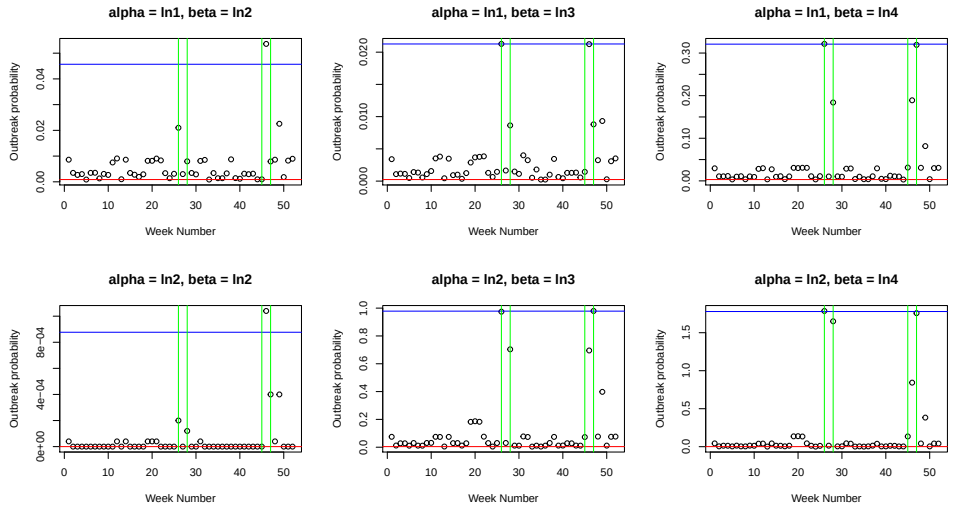


Figure 3: Estimated outbreak probability using Spatial Independent Model

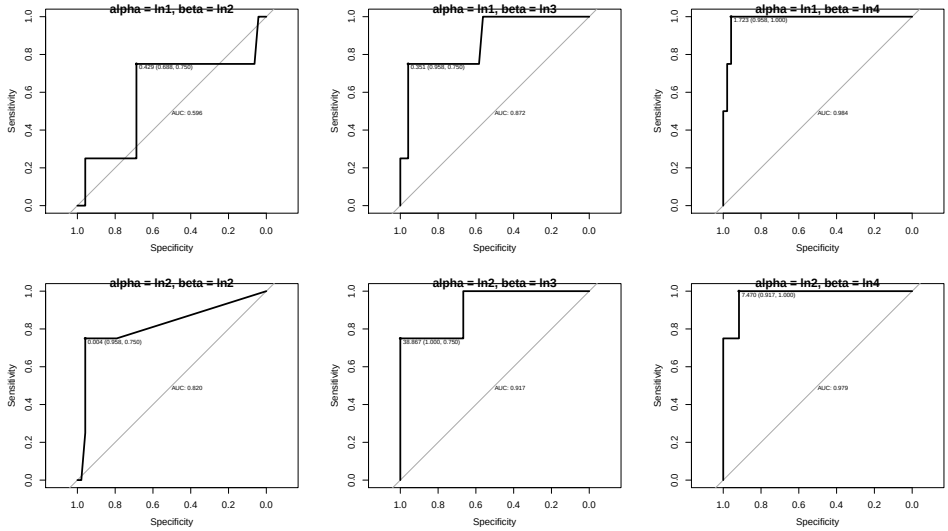


Figure 4: ROC curves for Spatial Independent Model

However, when I used a smaller value of true β in (ln(1),ln(3)), there were much more samples lying in the grey area, which is indicated by the quantile range that I set (0.25%,99.75%) based on posterior density of sampled x_{i1}. In such scenarios, it becomes much more difficult for the model to capture those outbreaks (in the grey area), resulting in a poorer outbreak predictability.

Also, we may find that the variation in the value of α doesn't appear to have a large effect to the overall outbreak predictability. Intuitionally, the reason is that e^{α} only represents as the number of infectious cases that would occur within a given time period (e.g. in this case, a week)

In order to study the additional effect that spatial information would have on predicting outbreaks, we want to focus our interest in those scenarios where it makes the model more difficult to detect an outbreak. In the next stage, by removing the independency among the regions, we want to enhance our detection using extra information from localised neighbors, where a region that is having an outbreak ($x_{i,j} = 1$) would increase the probability of its neighboring region to also have an outbreak ($x_{i,j+1}, x_{i,j-1} = 1$).

Trials	α	β	p	ROC performance
1	ln(1)	ln(2)	4/52	0.90
2	ln(1)	ln(2)	7/52	0.61
3	ln(1)	ln(3)	7/52	0.83
4	ln(1)	ln(4)	4/52	0.99
5	ln(1)	ln(4)	7/52	0.92
6	ln(2)	ln(2)	4/52	0.50
7	ln(2)	ln(2)	7/52	0.83
8	ln(2)	ln(3)	7/52	0.96
9	ln(2)	ln(4)	4/52	1.00
10	ln(2)	ln(4)	7/52	0.98
11	ln(4)	ln(2)	4/52	0.98
12	ln(4)	ln(2)	7/52	0.86
13	ln(4)	ln(4)	4/52	1.00
14	ln(4)	ln(4)	7/52	1.00
15	ln(4)	ln(10)	7/52	1.00
16	ln(4)	ln(10)	15/52	1.00

Table 5: ROC Performance for Spatially Independent Model

4.2 Spatially Dependent Model

Following different choices of the true values of the model parameters (listed in Section 3.2.5), data was generated for variables **X** and **Y** based on the distributions assumed for the spatially dependent model. Given the difficulties in computing likelihood for **X** (due to strong dependence among the locations $\{x_{ij}, \forall j \in (1, 2, \ldots, m)\}$), a pseudo-likelihood was introduced to simplify the computation on the posterior densities. Simulations were then draw using the Markov Chain Monte Carlo methods accordingly.

4.2.1 Results and convergence diagnosis

In the case of implementing MCMC methods on spatially dependent model, we have a much heavier burden in terms of the total computational cost (or time). After investigating the convergence of the Markov chain using different number of iterations and burn-in period, we have set our MCMC to have 3000 iterations and 500 burn-in period.

The trace plots 5 (Markov chains after the burn-in period) are plotted to visualise the convergence. It appears that all four parameters are mixing nicely after the burn-in period.

4.2.2 ROC performance

By looking at the ROC curves for Spatial Dependent models, we can compare the outbreak predictability under different trials (choice of parameters α, β) quantitatively.

From the table 6, similar findings as in spatial independent model (Section 4.1.2) were drawn, regarding the changes in the value of β.

Visually, all the corresponding ROC curves 6 are plotted for each trial, with computed AUC (area under curve) score. We could find that larger β would result in higher AUC score, meaning that the

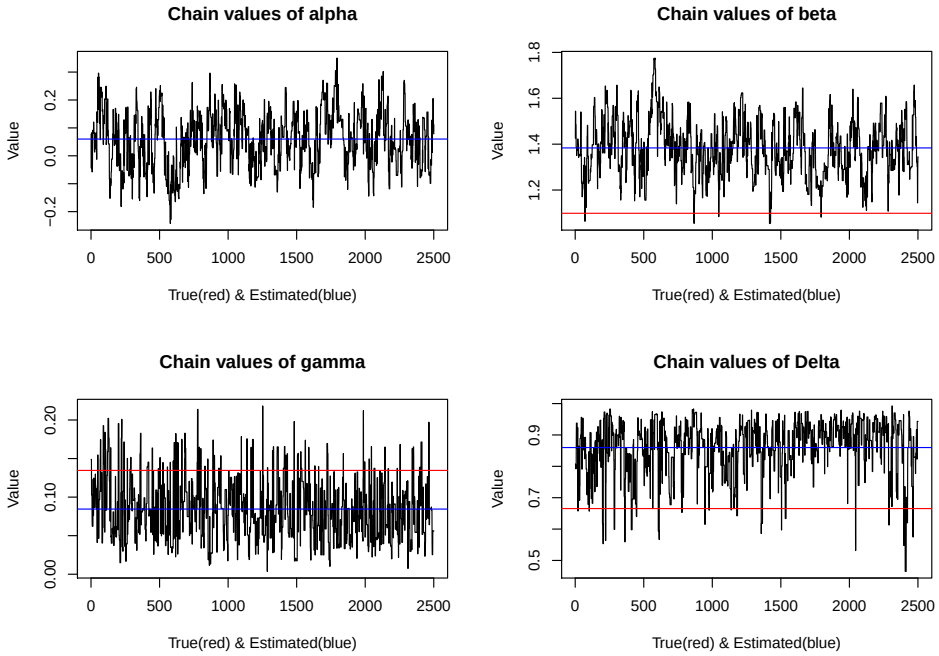


Figure 5: Trace plots of Parameters for Spatial Dependent Model. The red line represents the true value of the parameter, and the blue line represents the estimated value of the parameter.

Trials	α	β	γ	Δ	ROC (spatial)	ROC (simple)
1	ln(1)	ln(2)	7/52	0.665	0.701	0.668
2	ln(1)	ln(3)	7/52	0.665	0.822	0.809
3	ln(1)	ln(4)	7/52	0.665	0.923	0.922
4	ln(2)	ln(2)	7/52	0.665	0.797	0.797
5	ln(2)	ln(3)	7/52	0.665	0.910	0.913
6	ln(4)	ln(2)	7/52	0.665	0.864	0.861

Table 6: Choice of true values in generated dataset

model would be able to capture more spatial outbreaks.

Also when the same generated data is simulated in MCMC settings using spatial independent model, we may notice that the corresponding ROC(simple) performance appears to be slightly poorer than using the spatial dependent model. But as β tends to be larger (from ln(2) to ln(4)), the difference between the ROC performance is reduced. When $\alpha \in [ln(2), ln(4)]$ is chosen, the spatial independent model appears to perform evenly well as the spatial dependent model.

Recall an outbreak's significance is representing the difference in λ (number of infectious cases within a given time period) when having an outbreak and when there is no outbreak. Since we have:

$$\lambda = \begin{cases} e^{\alpha}, \text{ when there is no outbreak} \\ e^{\alpha}e^{\beta}, \text{ when there is an outbreak} \end{cases}$$

Hence the outbreak significance can be quantified as $e^{\alpha}(e^{\beta}-1)$. Intuitively, if α is small (e.g. $e^{\alpha} = 1$), then β would have to be much larger in order to represent an outbreak's significance.

In this case when large value of α is chosen, both models appear to have similar outbreak predictability through MCMC simulation. This finding is also shown in the cumulative sum plot for **X** 12 attached in

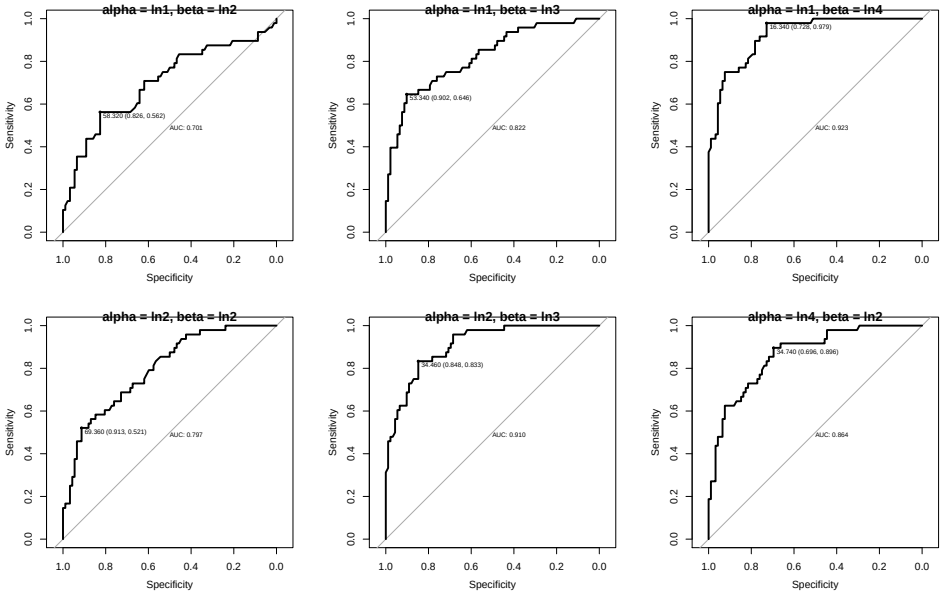


Figure 6: ROC curves for Spatial Dependent Model

appendix.

5 Case Study: Coronavirus in Maryland

5.1 Background

Given the context of the coronavirus, outbreak data in recent years has once again draw our attention to the infectious disease. In the public health sector, the state of Maryland, US has been monitoring a collection of positive COVID-19 test results that have been reported each day by the local health department via the ESSENCE system. The data is recorded for each individual counties in Maryland, from Mar 15th, 2020 til present.

5.1.1 Data

As mentioned above, the dataset was published and maintained by `opendata.maryland.gov`. It contains the cumulative number of positive COVID-19 cases among 24 Maryland counties within the Maryland jurisdiction (fig 7). New cases were updated in the system at 10:00am every day.

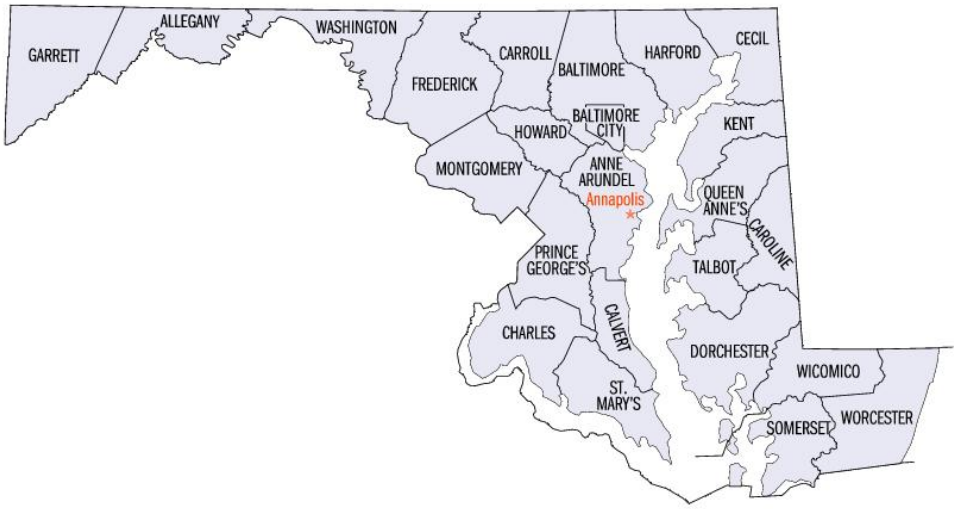


Figure 7: Map of Maryland Counties

There were some missing data (less than 1% of the whole dataset), in which I replaced them with 0. To study the distribution of the Covid disease in Maryland, I converted the cumulative amounts of cases into a weekly frequency format, in which I computed the number of new weekly cases for each county. By doing so, any extreme daily updates would not be too sensitive to the underlying model, and it also helps reduce the computational expense for MCMC methods.

5.1.2 Infection trend

When looking at the infection trends (fig 8,9) for different counties in Maryland (from year 2020 to 2022), a similar trend is found for all counties. In particular, there are two extreme peaks with large amount of positive cases being tested, which lies in around 2021 January to February and 2021 December to 2022 January. This phenomenon is also shared commonly with other western countries across the world. The red line in each plots represents the average number of cases for each county.

5.1.3 Time of interest

In order to conduct an explicit study on the spatial-temporal effect of the outbreak model, I chose a fixed time period (2021 November to 2022 March) to characterise the distribution. Given the possibility that

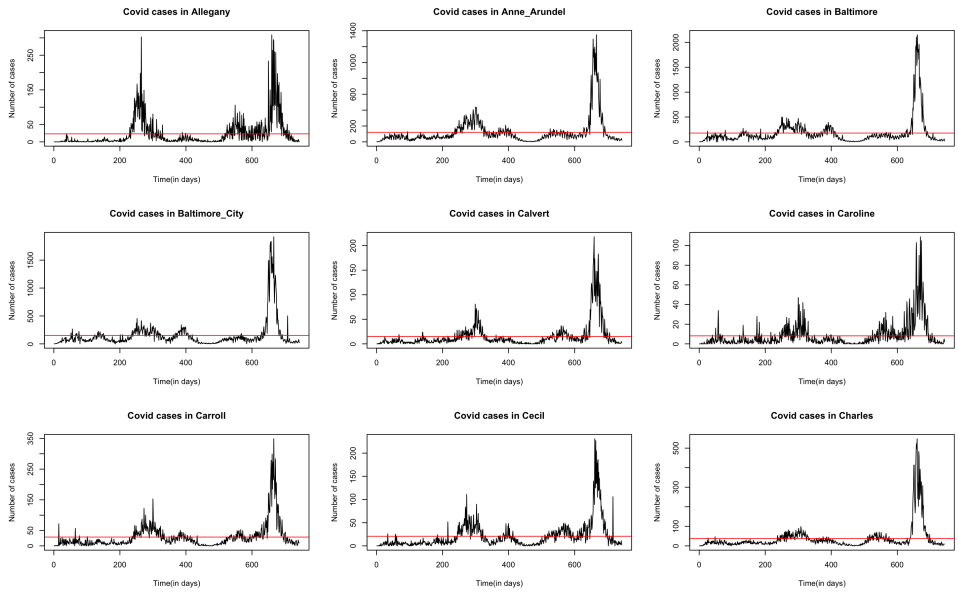


Figure 8: Infection trend for different counties (1)

the population size could get dynamic over the years (2020 to 2022), a fixed time period would restrict such variation. And MCMC methods are implemented over the chosen data.

5.2 Model formulation

5.2.1 Outbreak Indicator

Here, we are making an empirical assumption (for evaluation purposes only) that an outbreak is occurred only when it's larger than 86% quantile value of the number of positive cases for each county. Take Baltimore county for an example, if 86% quantile value is 200 cases per week, than any week that is having a larger number of cases would be marked as a temporal outbreak. The value of this threshold comes from the prior outbreak probability p in spatially independent model, or γ in spatially dependent model, which equals to $\frac{7}{52} = 0.134$. Therefore, I took a threshold at around $1 - \frac{7}{52} \simeq 86\%$.

5.2.2 Distance metric

Similar to the distance metric of the inner London area 3.2.1, I constructed a matrix to store the geographical distance between each two counties in Maryland. Given the fact that the most common way of transportation in Maryland, US is to drive in private vehicles, I am using the driving distance (in miles) from Google API database as the measurement.

5.2.3 Poisson rate parameter

In terms of the model parameters, one slight modification that I made is the initial rate parameter λ in the Poisson distribution which outcome variable $\mathbf{Y}$ follows. In the Maryland dataset, since the number of cases is highly correlated to the overall population size of each county, the rate parameter I use for the Poisson distribution is now represented as $\lambda_j n_j$, in which n_j denotes as the population size for county j and λ_j is representing the rate parameter of having positive cases per each person.

Note here that each county j is assumed to have the same λ_j for each individual. In other words, the population size n_j is treated as an offset in the rate parameter, and the underlying Poisson distribution

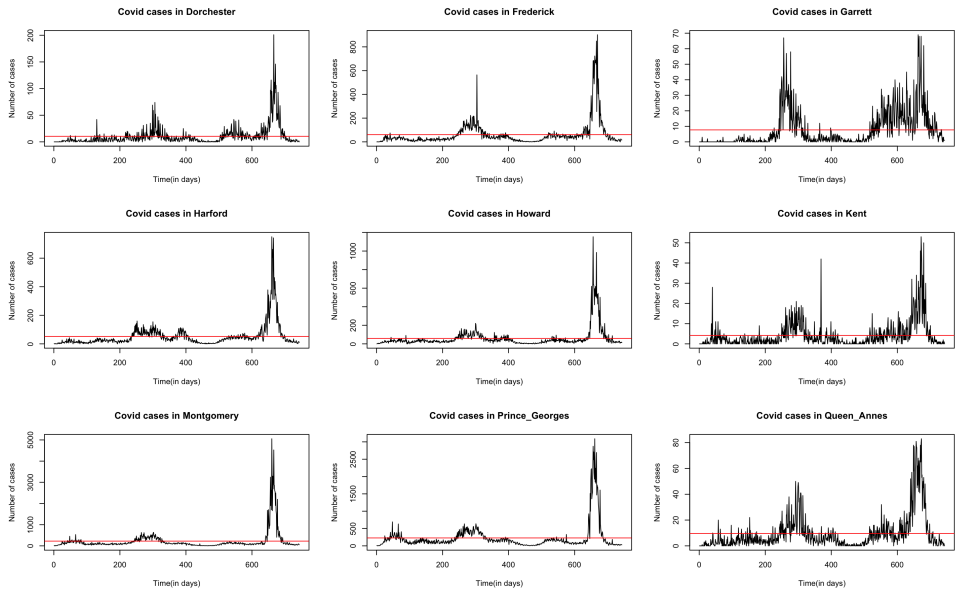


Figure 9: Infection trend for different counties (2)

varies with the population size in the county. Hence, by considering the difference in the population size, the threshold of having an outbreak would be different for each county.

5.3 Choosing startvalues

Given the high computational expense of implementing the MCMC methods in Bayesian hierarchical models, here I aim to find estimations of the model parameters based on the principle of Maximum Likelihood estimation, and use those estimates as the starting values in the MCMC simulation. The starting values chosen for the model parameters are as follows:

$$\begin{cases} \alpha = ln(0.52) \\ \beta = ln(4.36) \\ \gamma = 0.22 \\ \Delta = 0.58 \end{cases}$$

5.3.1 Choosing α, β

Based on the setting of our outbreak models, the value of λ given $\mathbf{X}$ is assumed to be constant, which means in the case of Maryland data, both $\lambda_{per,x=0}$ (when there is no outbreak) and $\lambda_{per,x=1}$ (when there is an outbreak) are fixed. And we know that:

$$\begin{aligned} \lambda_{per,x=0} &= e^{\alpha} \\ \lambda_{per,x=1} &= e^{\alpha} \cdot e^{\beta} \end{aligned}$$

Note that here λ_{per} refers to the number of cases that each person would have within a week. This gives:

$$\begin{aligned} \alpha &= \ln(\lambda_{per,x=0}) \\ \beta &= \ln(\frac{\lambda_{per,x=1}}{\lambda_{per,x=0}}) \\ &= \ln(\lambda_{per,x=1}) - \ln(\lambda_{per,x=0}) \end{aligned}$$

Using the principle of Maximum Likelihood estimation, we can estimate α, β as the following:

$$\begin{aligned}\alpha &= \ln(\text{mean}(\lambda_{per,x=0})) \\ &= \ln(0.52) \\ \beta &= \ln(\text{mean}(\lambda_{per,x=1})) - \alpha \\ &= \ln(4.36)\end{aligned}$$

Notice here our estimation on α is approximately negative, which means that the additional lower bound α_{min} that we have set for spatial dependent model is necessary to restrict the proposal of α, so that the prior distribution ($\alpha \sim \alpha_{min} + \Gamma(1, 6)$) still holds.

5.3.2 Choosing γ, Δ

Since our outbreak probability is consist of γ, Δ, where γ is representing the independent/background probability that an isolated outbreak occurs in one of the counties. Therefore, I took the average of the outbreak frequency in each county (,i.e. the isolated outbreak frequency) as the starting value for γ:

$$\gamma = \frac{1}{m}\sum_{j=1}^{m}\frac{\sum_{i=1}^{n} x_{ij}}{n},$$

where m is the number of counties, and n is the length of weeks. The numerator $\sum_{i=1}^{n} x_{ij}$ represents the number of outbreaks in county j within the given time period.

Rearrange the equation above, we would have:

$$\begin{aligned}\gamma &= \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}}{n \cdot m} \\ &= 0.22,\end{aligned}$$

which is the total outbreak frequency given the entire dataset.

After the estimation on γ, we can further derive Δ based on the relationship (mentioned in Section 3.2.3: formulation of dynamic outbreak probability) that:

$$\widehat{p} = \gamma + \Delta,$$

where $\widehat{p}$ stands for the maximum outbreak probability of location j given all the other neighboring locations are having outbreaks.

Here, we set this maximum probability $\widehat{p}$ to be 0.8, which gives us $\Delta = \widehat{p} - \gamma = 0.58$.

5.4 Bivariate normal proposal distribution

In both spatially independent model and spatially dependent model, α, β are proposed individually, each following a Normal distribution. When the same proposal distributions are applied to propose α, β based on coronavirus data from Maryland, US, the acceptance rate for α, β becomes very low [0.5%,1.5%], indicating poor performance of the mixing of the Markov chain.

Therefore, a Bivariate normal distribution is introduced to propose α, β simultaneously based on the 2×2 covariance matrix of α, β:

$$\Sigma = \begin{bmatrix} \text{var}(\alpha) & \text{cov}(\alpha, \beta) \\ \text{cov}(\beta, \alpha) & \text{var}(\beta) \end{bmatrix}.$$

The covariance matrix Σ is first computed from one trial of MCMC simulation where the acceptance rate for α, β is low, using $cov()$ function in R. Given the practical concern of very high computational expense, we want to aim for an optimal acceptance rate for α, β around $[10\%, 30\%]$ by tuning the covariance matrix in the proposal distribution.

According to the theorem of *Diffusion Limits for Metropolis* ([Rob01]), for a Metropolis algorithm in d dimensions, the optimal Gaussian proposal distribution $Q(x, \cdot)$ as $d \to \infty$ is:

$$Q(x, \cdot) = N(x, \frac{2.38^2}{d}\Sigma_t),$$

where Σ_t is the target covariance. Then we update Σ as the following:

$$\Sigma^* = \Sigma \cdot \frac{2.38^2}{2},$$

In this case, since there are two variables α, β, we have $d = 2$.

This method of proposal multiple variables from one joint distribution is commonly referred as the 'block sampling ([Tur17])' technique used in MCMC simulation. Using this approach, the proposal distribution now takes additional correlation information between α and β. In the practical comparison with initial individual distributions, bivariate normal distribution largely improves the efficiency of the Metropolis-Hastings algorithm, in terms of the mixing (time used for convergence).

5.5 Results and convergence diagnosis

In the case of implementing MCMC methods (using spatially dependent model) on coronavirus data from Maryland, US, we have a very similar performance from the simulation, comparing with the results from self-generated data (Section 4.2). After investigating the convergence of the Markov chain using different number of iterations and burn-in period, we have set our MCMC to have 3000 iterations and 500 burn-in period. The mixing of the model parameters $(\alpha, \beta, \gamma, \Delta)$ can be seem from the trace plots 10.

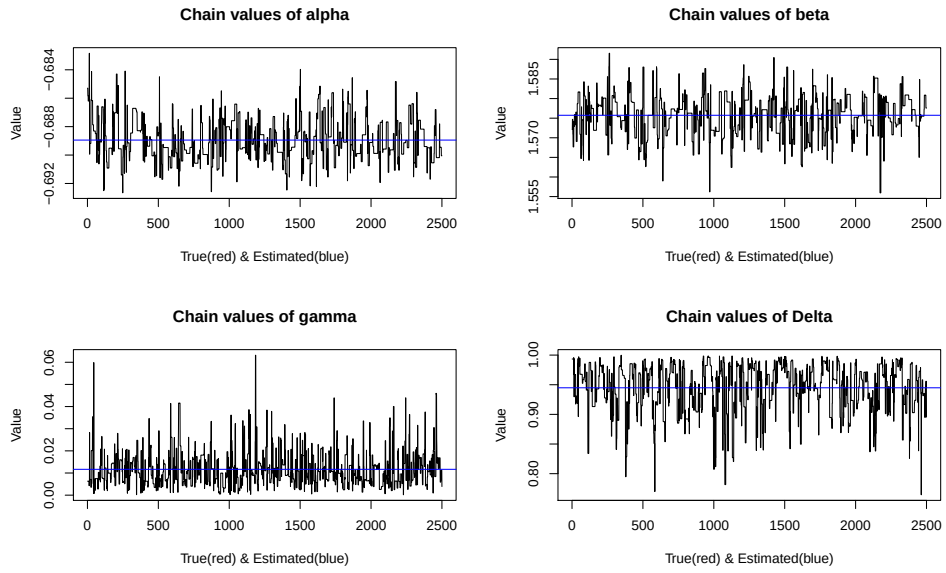


Figure 10: Trace Plot of Parameters for Spatial Dependent Model on Coronavirus data from Maryland. The blue line represents the estimated value of the parameter.

The table 8 contains the estimated values for each model parameter, computed from the MCMC outputs on Maryland dataset. The interpretation of a comparison between the starting values and MCMC estimation is followed in Section 6.2.

Parameter	Starting value	MCMC estimation
α	ln(0.52)	ln(0.50)
β	ln(4.36)	ln(4.84)
γ	0.22	0.0113
Δ	0.58	0.937

Table 7: MCMC estimation on the model parameters

Using the construction of the spatially dependent model, the overall Covid-19 predictability (for positive cases) is pretty strong at around 0.960 (indicated by ROC curve 11), The estimated outbreak probabilities for each week i and county j have also been computed and visualised in the cumulative plot (on the right).

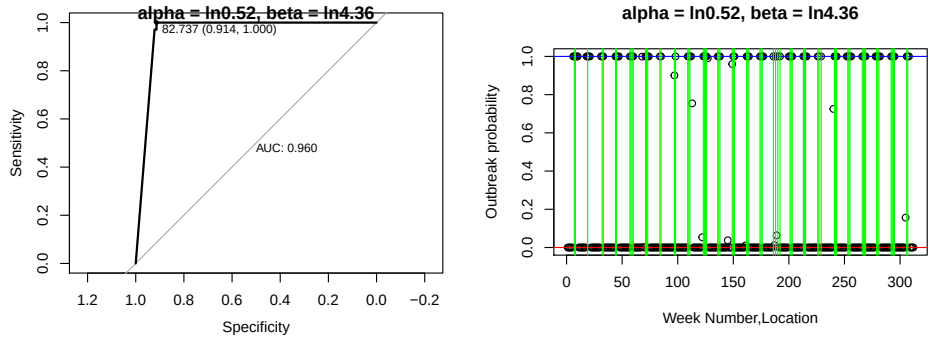


Figure 11: ROC curve (left) and estimated outbreak probability (right) of Spatial Dependent Model on Coronavirus data from Maryland

6 Conclusion

In this section, we would discuss the findings 6.1, interpretations 6.2 based on the MCMC outputs that we evaluated in Section 4 and 5. Then, we would also mention some of the limitations 6.3 that are present in this project, which provides further study approaches 6.4 of this topic in the future.

6.1 Findings

6.1.1 Spatially independent model

In the setting of a spatially independent model, we have found that the overall outbreak predictability largely depends on the choice of β, as the additional infectious effect during an outbreak, measured in log-scale. As the value of β tends to ln(4) or even larger, it would become much easier for model to capture outbreaks within a given time period. Conversely, given a smaller value of β, there would some real outbreak being left as undetected, which lay in the grey area that we defined using posterior density of having an outbreak.

6.1.2 Spatially dependent model

After removing the assumption of independence among the neighboring locations, the outbreak model is extended into a spatial-temporal setting. In the spatial dependent model, where the outbreak probability for each location at a given time is also taking neighboring effect into consideration, we have found that the model results in having a slightly better predictability on outbreaks (given the same choice of parameters and generated data), particularly when $\alpha \in [ln(0.5), ln(1.5)]$ is chosen. When a larger α is chosen, the difference between the two models is reduced in terms of their ROC performance.

6.2 Interpretations

In the case of real Covid-19 infectious data in Maryland, US, we estimated (discussed in Section 5.3) the starting value of the parameters used in the setting of our outbreak models. Given the findings that we have for the two models, and the fact that the estimated parameter α is relatively small as ln(0.52), the spatial dependent model is preferred in this scenario.

Recall that the maximum likelihood estimation of the model parameters $(\alpha, \beta, \gamma, \Delta)$ has been chosen as the starting values of the MCMC simulation. Based on the MCMC outputs (estimation on the parameters in table 8), I have found that the estimated values of α, β are very close to the maximum likelihood estimation. Surprisingly, there is a larger difference between the starting values and MCMC estimation for γ, Δ.

Parameter	Starting value	MCMC estimation
α	ln(0.52)	ln(0.50)
β	ln(4.36)	ln(4.84)
γ	0.22	0.0113
Δ	0.58	0.937

Table 8: MCMC estimation on the model parameters

In particular, γ (the background probability of having an isolated outbreak for all regions) has a much smaller estimated value (0.0113) than the starting value. And conversely, Δ (the spatio-temporal effect

measured from the neighboring locations) has a much larger estimated value (close to 1) than the starting value (0.58). This estimation suggests that a location would almost surely have an outbreak when all the neighboring locations are having outbreaks. We may draw a further interpretation that the model is heavily relying on the spatial information among the neighboring locations, to predict regional outbreaks.

6.3 Limitations

6.3.1 Self-generated data

In terms of implementing MCMC methods for both spatially independent (Section 3.1) and dependent model (Section 3.2), I am using self-generated (simulated) data based on the assumed parameters' true values and distributions. A comparison study on the spatio-temporal effects is established explicitly by excluding any randomised noise in real-life disease data.

The main drawback of using self-generated data is that the corresponding models would be less robust when implementing in real-life datasets.

6.3.2 Dependence among the parameters

The performance of Gibbs sampler is negatively effected by the fact that $\mathbf{X}$ is highly dependent in the posterior. When constructing the likelihood of x_j, it is difficult to compute the full conditional probability of it given other $x_{k|k\neq j}$ (the neighboring information).

To cope with this issue, a pseudo likelihood computation is adopted by assuming independence among $\mathbf{X}$. This eases the overall computation of MCMC algorithms, at the costs of possible slow convergence of the Markov chains to our target distributions.

6.3.3 Measurement

Dynamic probability function In our spatial function $p_i(\gamma, \Delta, x_i)$, the probability of having an outbreak for each location is depending on all the other locations, in which the effect is weighted based on the Euclidean distance.

However in real life scenarios, infection is most likely spread in regions that have highly dense population. In such cases, using distance between locations to measure the neighboring effects might no longer be reliable in terms of model prediction and interpretation.

Dynamic population For large modern metropolis, the population could get quite mobile from time to time, which makes the population estimation for each county (or location) less reliable.

By considering the epidemic nature of most infectious disease like coronavirus, it could be reasonable to use population size of people not staying at home, which in the case of Maryland, relevant public data can be found at the U.S. Bureau of Transportation Statistics. The data source also records the dynamic population size for each month.

6.3.4 MCMC diagnosis

There are so far many heuristic approaches (discussed in Section 2.4.4) to diagnose and monitor the performance of a MCMC simulation, which makes it quite model-dependent without any theoretical criterion.

6.4 Future work

6.4.1 Model extension

In our current models (both spatially independent and dependent), the construction of our rate parameter λ only relies on a two-dimensional parameter set:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix},$$

where α is the intercept coefficient which stands for the averaged sporadic frequency of an infectious disease, and β is the additional infectious effect cause by an external risk factor $\mathcal{X}$ (which leads to higher risk of having an outbreak).

$$\begin{bmatrix} \alpha \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}$$

In the following future studies, more advanced models can be constructed by considering a higher dimensional parameter set. Using those models, we can aim to increase the overall model flexibility as well as interpretability of observing disease outbreaks. Possible parameter extensions are listed below:

Spatio-temporal term The distribution variable λ in the studied models, or the individual λ_{per} for people living in Maryland, U.S. (Section 5) is assumed to be the same for all locations or counties. In a more complex setting, we can adopt the methodology used in the study of *campylobacteriosis notification data* [Spe11], by introducing the spatio-temporal term u_j , which follows a normal distribution $u_j \sim \mathcal{N}(0, \sigma_u^2)$. This would give us:

$$ln(\lambda_j) = \alpha + \beta x + u_j,$$

where j is the location index.

Timely-temporal term In our study, we have only considered the spatial effects of disease outbreak. In spatial dependent model where we have constructed a dynamic probability function $p_i(\gamma, \Delta, x_i)$, the probability p_i is only considering the spatial effect at time i, indicated by $x_i := \{x_{ij}, j \text{ is the location index}\}$.

In spatially and temporally dependent model, we can the expand the spatial effect to a timely dimension, where a location that is having an outbreak would last for a given time period. This means other neighboring locations would likely be effected by this outbreak location over a time period, instead just at time i.

6.4.2 Hamiltonian Monte Carlo

Recently, convenient implementations of a powerful MCMC technique called Hamiltonian Monte Carlo (HMC: also called hybrid MC) have become available [LiM18]. HMC uses the concept of Hamiltonian dynamics to create a proposal distribution for the Metropolis-Hastings algorithm, together with the leap-frog algorithm and the No U-Turn sampler. HMC requires more computational effort per sample step compared to other MCMC techniques, but because subsequent steps are less strongly correlated it also produces more effective samples per sample step.

To cope with additional computational requirement, we can consider the software platforms that implement the automatic construction of MCMC samplers for user-defined models. Some of the widely used platforms [LiM18] include JAGS(Just Another Gibbs Sampler), NIMBLE (Numerical Inference for Statistical Models for Bayesian and Likelihood Estimation), Stan (,which provides full Bayesian inference for continuous-variable models based on the No-U-Turn sampler, an adaptive form of HMC).

7 Summary

In this dissertation I have introduced a new model for outbreak detection that uses information from nearby locations to improve the identification of outbreaks. On simulated data I showed that there was greater power to detect outbreaks (higher predictability), resulting in an improved ROC score.

I then applied my novel approach to coronavirus data from Maryland, US. The estimation on the model parameters (γ, Δ) suggests a crucial evidence that the spatial information can be relied heavily to improve the discrimination of an outbreak.

References

[Bec97] Britton.T Becker.N.G. "Uses of the EM algorithm in the analysis of data on HIV/AIDS and other infectious diseases". In: *Stat Methods Med Res* 6.1 (Mar. 1997), pp. 24–37. DOI: 10.1177/096228029700600103. URL: https://pubmed.ncbi.nlm.nih.gov/9185288/.

[Bel12] Chernozhukov.V Belloni.A. "On the Computational Complexity of MCMC-based Estimators in Large Samples". In: *arXiv preprint* 37.4 (Jan. 2012). DOI: 10.1214/08-AOS634. URL: https://arxiv.org/abs/0704.2167v3.

[Ben21] Spencer.SEF Benschop.J Nisa.S. "Still 'dairy farm fever'? A Bayesian model for leptospirosis notification data in New Zealand". In: *J. R. Soc. Interface* 18 (Jan. 2021). DOI: 10.1098/rsif.2020.0964. URL: https://doi.org/10.1098/rsif.2020.0964.

[Bes75] Besag.J. "Statistical Analysis of Non-Lattice Data". In: *Royal Statistical Society* 24 (Sept. 1975). URL: https://www.jstor.org/stable/2987782?seq=1.

[Bol07] W. Bolstad. *The Frailty Model*. 2007.

[Buc08] Buckeridge.DL. "Predicting Outbreak Detection in Public Health Surveillance: Quantitative Analysis to Enable Evidence-Based Method Selection". In: (2008). URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2656053/.

[Duc11] P. Duchateau L. Janssen. "Understanding Computational Bayesian Statistics". In: (2011).

[Ham13] Richardson D. Hamra G MacLehose R. "Markov chain Monte Carlo: an introduction for epidemiologists". In: (2013). URL: https://academic.oup.com/ije/article/42/2/627/738896.

[Kas98] Gelman.A Kass.E.Robert Carlin.P.B. "Markov Chain Monte Carlo in Practice: A Roundtable Discussion". In: *The American Statistician* 52.2 (May 1998), pp. 93–100. DOI: 10.2307/2685466. URL: https://www.jstor.org/stable/2685466.

[LiM18] Bolker.M.B Li.M Dushoff.J. "Fitting mechanistic epidemic models to data: A comparison of simple Markov chain Monte Carlo approaches". In: *Stat Methods Med Res* 27.7 (July 2018). DOI: 10.1177/0962280217747054. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6027774/.

[Rob01] Rosenthal.J Roberts.G. “Optimal Scaling for Various Metropolis–Hastings Algorithms”. In: *Statistical Science* 16 (Nov. 2001). URL: http://www.jstor.org/stable/3182776.

[Spe11] Spencer.SEF. “The detection of spatially localised outbreaks in campylobacteriosis notification data”. In: (Sept. 2011). URL: https://pubmed.ncbi.nlm.nih.gov/22748176/.

[Tur17] Paciorek.C Turek.D Valpine.P. “Automated Parameter Blocking for Efficient Markov Chain Monte Carlo Sampling”. In: *Bayesian Analysis* 12 (June 2017). URL: https://projecteuclid.org/journals/bayesian-analysis/volume-12/issue-2/Automated-Parameter-Blocking-for-Efficient-Markov-Chain-Monte-Carlo-Sampling/10.1214/16-BA1008.full.

[Unk11] Steffen Unkel. “Statistical methods for the prospective detection of infectious disease outbreaks: a review”. In: (July 2011). URL: https://rss.onlinelibrary.wiley.com/doi/10.1111/j.1467-985X.2011.00714..

[Vat19] Flegal.M.J Vats.D Robertson.N. “Analyzing MCMC output”. In: *arXiv preprint* 2 (Dec. 2019). URL: https://arxiv.org/pdf/1907.11680.pdf.

8 Appendix

8.1 Plots

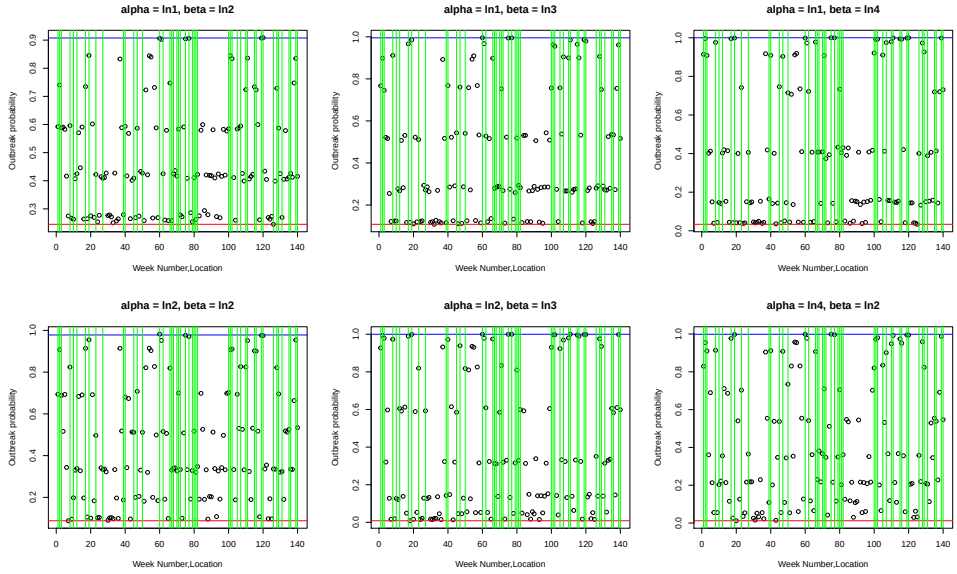


Figure 12: Estimated outbreak probability using Spatial Dependent Model

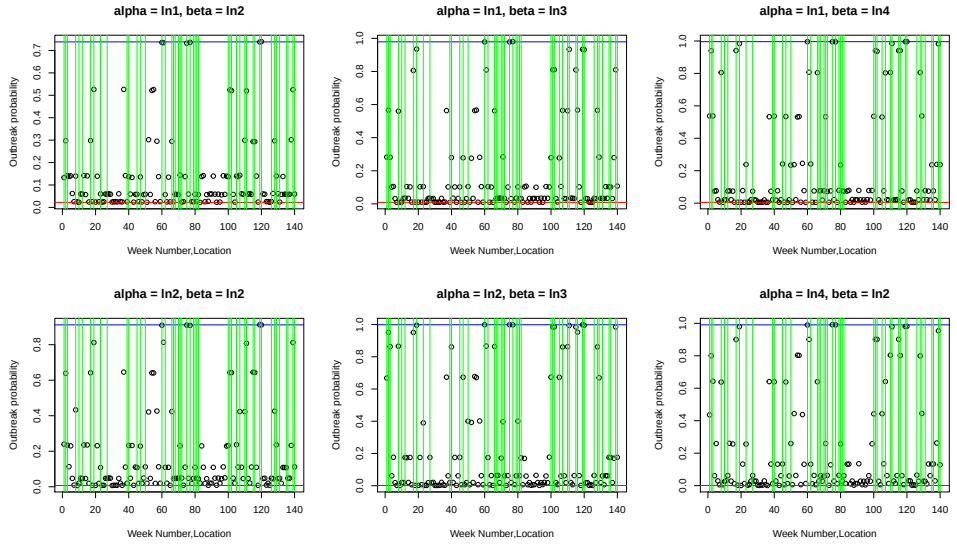


Figure 13: Estimated outbreak probability using Spatial Independent Model using spatial data

8.2 Distribution table

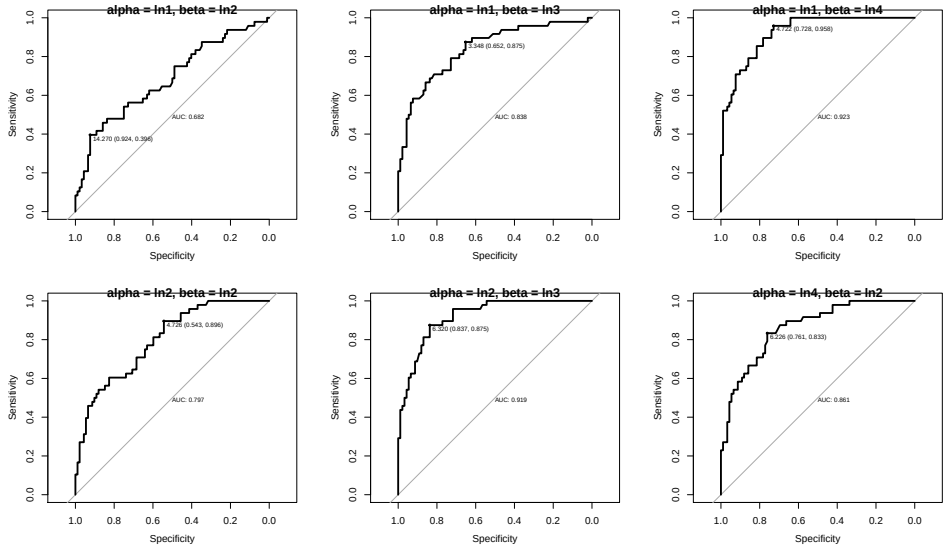


Figure 14: ROC curves for Spatial Independent Model using spatial data

Distribution	parameters	support	PDF
Poisson	$\lambda \in (0, \infty)$	$k \in \mathbb{N}_0$	$f(k) = \frac{\lambda^k e^{-\lambda}}{k!}$
Binomial	n, p	$k \in 0, 1, \ldots, n$	$f(k) = \binom{n}{k} p^k (1-p)^{n-k}$
Gamma	$\alpha > 0, \beta > 0$	x$\in (0, \infty)$	$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$
Beta	$\alpha > 0, \beta > 0$	x$\in [0, 1]$	$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$ where $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$
Dirichlet	$K \geq 2$	$x_i, \ldots, x_K \in (0, 1)$	$f(x) = \frac{1}{B(\alpha)} \prod_{i=1}^K x_i^{\alpha_i - 1}$ where $B(\alpha) = \frac{\prod_{i=1}^K \Gamma\alpha_i}{\Gamma(\sum_{i=1}^K \alpha_i)}$
	$\alpha_1, \ldots, \alpha_K > 0$	$\sum_{i=1}^K x_i = 1$	

Table 9: Probability distribution table for different distributions

8.3 R Code

Construct distance metric and dynamic function of p_{ij} given θ and $x_{i,-j}$:

```
# load distance information
library("readxl")
d <- read_excel('london/inner_london.xlsx', sheet=2) # distance (in miles)
d <- d[-1] #drop the 1st index column
d_inv <- 1/d # inverse distance metric
names(d_inv) <-
    c('Her','Cam','Isl','Hac','Ham','Ken','Wes','Cit','Tow','New','Wan','Lam','Sou','Lew')
m <- dim(d)[1] # number of boroughs


# define probability function at each time t
local_prob <- function(x_t,d_inv,theta){
  # x_t: a vector of Xs indicating an outbreak or not for each location at time t
  # d_inv: an inverse distance metrix among all locations within the region
  # j: the index of the location
  # theta: a vector of gamma, Delta and 1-gamma-Delta
```

```
  gamma = theta[1]
  Delta = theta[2]
  m <- dim(d_inv)[1] # number of locations
  prob_t <- rep(0,m) # probability for each location at time t

  if (length(x_t) != 1) {
    for (j in 1:m){
      x_t = x_t[-j] # removing x_j
      d_j = d_inv[j,-j] # the jth row excluding the jth location, giving a vector
          where j!=k
      weighted_d_j = d_j/sum(d_j) # compute the weight to each entry

      prob_t[j] <- gamma+Delta*sum(weighted_d_j*x_t)
    }
  }
  else {prob_t = gamma+Delta*x_t} # if x_t is single value
  return(prob_t)
}
```

Generate data using Gibbs Sampler for the spatially dependent model:

```
##################### Generate data for Spatially dependent model
    ######################
# Gibbs sampling
set.seed(2022)
x_0 <- rep(0,14) # looking at 14 boroughs in inner London area at time 0
x_0[c(8,13)] = c(1,1) # City of London-8, Southwark-13
x_i <- x_0
p_i <- local_prob(x_i,d_inv,a_theta)

# running time: 2mins
for (k in 1:300){ # let p converge to have neighboring effect after 300 updates
  for (j in 1:m){
    x_i[j] <- rbinom(1,size = 1,prob = p_i[j]) #p in location j at time i
    p_i <- local_prob(x_i,d_inv,a_theta) # update p_i
  }
  # print(sum(x_i))
}

### Generate data
x <- x_i
p <- p_i

for (i in 1:n){
  for (j in 1:m){
    x_i[j] <- rbinom(1,size = 1,prob = p_i[j]) #p in location j at time i
    p_i <- local_prob(x_i,d_inv,a_theta) # update p_i
  }
  x <- rbind(x,x_i)
  # update p_i using x_i
  p_i <- local_prob(x_i,d_inv,a_theta)
  p <- rbind(p,p_i)
```

```
}

x <- x[-1,] # a matrix of x_it with time and location, remove the 1st empty column
p <- p[-1,] # a matrix of p_it with time and location, remove the 1st empty column
# rbind(apply(x,1,sum),apply(p,1,mean))


# set parameters
lambda = exp(alpha + beta*x) # excluding the noise, returns a matrix
y <- matrix(data = NA, nrow = 1, ncol = m)
for (i in 1:n){
  y_i <- rpois(1,lambda = lambda[i,1]) # sampled observations for the 1st location
      at time i
  for (j in 2:m){
    y_i[j] = rpois(1,lambda = lambda[i,j]) # sampled observations for one location
  }
  y <- rbind(y,y_i)
}

y <- y[-1,] # a matrix of y_it with time and location, remove the 1st empty row
```

Construct Posterior density for the spatially dependent model:

```
################ Construct Posterior Distribution #################
# log density function of the Dirichlet distribution
lddirichlet <- function(x,a) {
  # x: current value of theta, a: a_theta
  wh<-which(a!=0) # allow a for having 0s for some a_i
  return(lgamma(sum(a[wh]))+sum((a[wh]-1)*log(x[wh])-lgamma(a[wh])))
}

# Log Prior density
prior <- function(param,x,sum=TRUE){
  # x: a matrix of x_it, outbreak indicator in location i at time t
  alpha = param[1]
  beta = param[2]
  gamma = param[3]
  Delta = param[4]
  theta = c(gamma,Delta,1-gamma-Delta)

  alpha_prior = dgamma(alpha-alpha.min, shape = 6, log = T) # shape size
      determined the support range of alpha
  beta_prior = dgamma(beta-beta.min, shape=6, log = T) # shape size determined the
      support range of beta

  x <- as.matrix(x)
  x_prior <- matrix(data = NA, nrow = 1, ncol = dim(x)[2])
  for (t in 1:dim(x)[1]){
    p_t = local_prob(x[t,],d_inv,theta) # compute localized probability at time t
    x_t_prior <- dbinom(x[t,],1,p_t,log = T) # a vector of density for each x_t
    x_prior <- rbind(x_prior,x_t_prior)
  }
```

```
  x_prior <- x_prior[-1,] # remove the 1st empty row

  gamma_Delta_prior = lddirichlet(theta,a_theta)

  if (sum==T){res = alpha_prior+beta_prior+sum(x_prior)+gamma_Delta_prior}
  else {res = alpha_prior+beta_prior+x_prior+gamma_Delta_prior}
  return(res)
}


# Log Likelihood
# likelihood of y given x
likelihood <- function(param,x){
  alpha = param[1]
  beta = param[2]
  pred = exp(alpha + beta*x)

  singlelikelihoods = dpois(y, lambda = pred, log = TRUE) # predicted lambda, y
      comes from above
  return(singlelikelihoods) # a matrix of likelihoods
}

# likelihood of x given p(theta)
likelihood_x <- function(x,theta){
  likelihood_x <- matrix(data = NA, nrow = 1, ncol = dim(x)[2])
  for (i in 1:dim(x)[1]){
    res = dbinom(x[i,], 1, local_prob(x[i,],d_inv,theta), log = T)
    likelihood_x <- rbind(likelihood_x,res)
  }
  likelihood_x <- likelihood_x[-1,] # remove the 1st empty row
  return(sum(likelihood_x)) # likelihood for x given theta
}


# Posterior
posterior <- function(param,x,sum=TRUE){
  if (sum==T){res = sum(likelihood(param,x)) + prior(param,x,T)}
  else {res = likelihood(param,x) + prior(param,x,F)}
  return (res) # matrix + matrix
}
```

Parameter Estimation using MCMC methods on Maryland Covid Data:

```
### Metropolis Hasting algorithm ###

load("trials_mcmc/maryland/maryland_ln(0.52)_ln(4.36).RData")
sigma <- cov(chain[-1,c(1,2)])
Sigma <- (2.38^2/dim(sigma)[1])*sigma
proposalfunction <- function(param){
  # param: alpha,beta,gamma,Delta
  tmp = mvrnorm(n=1,mu=param[c(1,2)],Sigma=Sigma) # proposal distribution of
      alpha,beta
```

```
  param[1] = tmp[1]
  param[2] = tmp[2]
  # print(paste(param[c(1,2)]))
  return(param)
}



run_metropolis_MCMC <- function(startvalue,x,iterations,burnIn){
  # create chain to store alpha,beta,gamma,Delta
  chain = matrix(NA,iterations+1,length(startvalue)) # each row:
      c(alpha,beta,gamma,Delta)
  chain[1,] = startvalue # 1st row

  # create chain_x to store x
  t <- dim(x)[1] # total time
  m <- dim(x)[2] # total locations
  chain_x <- array(NA, c(iterations+1,t,m)) # create a 3 dimensional array
  chain_x[1,,] <- x # startvalue for x at time 0
  a <- 1 # scale parameter in Dirichlet distribution Dir(theta+a*theta_)
  prob_alpha_beta = 0
  prob_theta = 0

  # iterations
  for (i in 1:iterations){
    if(i %% 100==0){print(paste('iteration',i))}

    # MH for alpha, beta
    proposal = proposalfunction(chain[i,]) # propose new alpha, beta
    if (proposal[2]>beta.min & proposal[1]>alpha.min){ # reject if it's smaller
        than min
      probab = exp(posterior(proposal,x,T) - posterior(chain[i,],x,T)) # ratio
      prob_alpha_beta[i] = probab # record probability ratio
      if (is.na(probab) || runif(1) >= probab){
        chain[i+1,] = chain[i,] # reject
      } else {
        chain[i+1,] = proposal # accept alpha,beta
      }
    }
    else{
      prob_alpha_beta[i] = 0
      chain[i+1,] = chain[i,] # reject
    }

    # Dirichlet random walk
    theta <- c(chain[i,3],chain[i,4],1-chain[i,3]-chain[i,4]) # current theta
    theta_ = rdirichlet(1,a*theta+a_theta) # proposal for new theta
    prob1 <- likelihood_x(x,theta_) - likelihood_x(x, theta) # ratio of likelihoods
    prob2 <- lddirichlet(theta_, a_theta)-lddirichlet(theta, a_theta) # ratio of
        proposal density
    prob3 <- lddirichlet(theta,a_theta+a*theta_) -
```

```
      lddirichlet(theta_,a_theta+a*theta) # ratio of proposal density
    full_prob <- exp(prob1+prob2+prob3)
    prob_theta[i] = full_prob # record probability ratio
    if (is.na(full_prob) || runif(1) >= full_prob){
      a = a+1
    } else {
      chain[i+1,3] = theta_[1] # accept gamma
      chain[i+1,4] = theta_[2] # accept Delta
      a = max(0,a-3) # adaptive approach on choosing scaling parameter a, to get
          close to 25% ratio
    }

    # update x, Gibbs sampler
    param <- chain[i+1,]
    # normalized_c <- 1/(exp(posterior(param,1,F))+exp(posterior(param,0,F)))
    log_p1 <- posterior(param,1,F)
    log_p0 <- posterior(param,0,F) # R = exp(log_p1)/(exp(log_p1)+exp(log_p0)),
        i.e. p1/(p1+p0)
    M <- pmax(log_p0,log_p1)
    R <- exp(log_p1-M)/(exp(log_p1-M)+exp(log_p0-M)) # re-write R, using
        log-sum-exp trick

    for (j in 1:m){
      chain_x[i+1,,j] = rbinom(t,1,R[,j]) # vector of proposed x for one location
    }
  }

  # return acceptances
  acceptance_alpha_beta = 1-mean(duplicated(chain[-(1:burnIn),c(1:2)]))
  acceptance_theta = 1-mean(duplicated(chain[-(1:burnIn),c(3:4)]))

  # return(chain)
  setClass(Class="res",
           representation(
             chain="matrix",
             chain_x="array",
             prob_alpha_beta="numeric",
             prob_theta="numeric",
             acceptance_alpha_beta="numeric",
             acceptance_theta="numeric"
           )
  )
  return(new('res',
             chain = chain,
             chain_x = chain_x,
             prob_alpha_beta = prob_alpha_beta,
             prob_theta = prob_theta,
             acceptance_alpha_beta = acceptance_alpha_beta,
             acceptance_theta = acceptance_theta))
}
```

R functions to produce trace plot and histograms from MCMC for model parameters:

```
# visualization for model parameters

# plot the histogram of the parameter distributions
plot_hist <- function(burnIn,param_index,param_name,param_true){
  hist(chain[-(1:burnIn),param_index],nclass=30, main=paste("Posterior
      of",param_name[param_index]), xlab="True value = red line" )
  abline(v = mean(chain[-(1:burnIn),param_index]),col='blue')
  abline(v = param_true[param_index], col="red" )
}
# plot the MCMC chains for the parameter
plot_chain <- function(burnIn,param_index,param_name,param_true){
  plot(chain[-(1:burnIn),param_index], type = "l", xlab="True value = red line" ,
      main = paste("Chain values of",param_name[param_index]))
  abline(h = param_true[param_index], col="red" )
  abline(h = mean(chain[-(1:burnIn),param_index]),col='blue')
}
```

Visualising X with cumulative probability plot and ROC curve:

```
# visualizing x
plot_x_sum <- function(chain_x,model="simple"){
  # chain_x: simulated x from MCMC, 1st row being the true x in dataset

  if (model=="simple" | model=="simple_with_spatial"){
    x = chain_x[1,]
    chain_x = chain_x[-1,]
    s = apply(chain_x[-(1:burnIn),],2,mean) # accumulating column sum for each
        week of x
  } else if (model=="spatial"){
    x = chain_x[1,,]
    chain_x = chain_x[-1,,]
    s = apply(chain_x[-(1:burnIn),,],2:3,mean) # average x over iterations
        (row:time;col:location)
    s = as.vector(s)
  }

  quantile.no = 400
  segments <- quantile(s,probs = seq(0,1,1/quantile.no))
  point <- unname(segments[c(1,quantile.no-1)])
  plot(s, xlab="Week Number,Location" , ylab="Outbreak probability")
  abline(h = point[1], col="red")
  abline(h = point[2], col='blue')
  # compare with real x
  wh <- which(x==1)
  for (k in wh){ abline(v = k, col='green') }

  #plot(0:1,0:1,t="n") # make the plot transparent using axes
  #legend("topright",c('0.25% quantile','99.75% quantile','real outbreak
      week'),col=c('red','blue','green'),lty=1)
}
```

```
plot_roc <- function(chain_x,model="simple"){
  # ROC curve
  if (model=="simple" | model=="simple_with_spatial"){
    x = chain_x[1,]
    chain_x = chain_x[-1,]
    s = apply(chain_x[-(1:burnIn),],2,mean) # accumulating column sum for each
        week of x
  } else if (model=="spatial"){
    x <- as.vector(chain_x[1,,])
    chain_x = chain_x[-1,,]
    s = apply(chain_x[-(1:burnIn),,],2:3,mean) # average x over iterations
        (row:time;col:location)
    s = as.vector(s)
  }

  df <- data.frame(s*100,x)
  names(df) <- c('chain_x','x')
  library('pROC')
  roc(df$x,df$chain_x,plot=TRUE, print.thres=TRUE, print.auc=TRUE)
}
```