Bayesian Hierarchical Models for Spatial Count Data with Application to Fire Frequency in British Columbia

by

Hong Li
B.Sc. University of Victoria 2006

A Thesis Submitted in Partial Fullfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Mathematics and Statistics

Bayesian Hierarchical Models for Spatial Count Data with Application to Fire Frequency in British Columbia

by

Hong Li
B.Sc. University of Victoria 2006

Supervisory Committee

Dr. Farouk Nathoo (Department of Mathematics and Statistics)
Supervisor

Dr. Julie Zhou (Department of Mathematics and Statistics)
Departmental Member

Dr. Min Tsao (Department of Mathematics and Statistics)
Departmental Member

Supervisory Committee

Dr. Farouk Nathoo (Department of Mathematics and Statistics)
Supervisor

Dr. Julie Zhou (Department of Mathematics and Statistics)
Departmental Member

Dr. Min Tsao (Department of Mathematics and Statistics)
Departmental Member

Abstract

This thesis develops hierarchical spatial models for the analysis of correlated and overdispersed count data based on the negative binomial distribution. Model development is motivated by a large scale study of fire frequency in British Columbia, conducted by the Pacific Forestry Service. Specific to our analysis, the main focus lies in examining the interaction between wildfire and forest insect outbreaks. In particular, we wish to relate the frequency of wildfire to the severity of mountain pine beetle (MPB) outbreaks in the province. There is a widespread belief that forest insect outbreaks lead to an increased frequency of wildfires; however, empirical evidence to date has been limited and thus a greater understanding of the association is required. This is critically important as British Columbia is currently experiencing a historically unprecedented MPB outbreak. We specify regression models for fire frequency incorporating random effects in a generalized linear mixed modeling framework. Within such a framework, both spatial correlation and extra-Poisson variation can be accommodated through random effects that are incorporated into the linear predictor of a generalized linear model. We consider a range of models, and conduct model selection and inference within the Bayesian framework with implementation based on Markov Chain Monte Carlo.

Table of Contents

List of Figures

Acknowledgments

I would first like to acknowledge my gratitude to my supervisor Dr. Farouk Nathoo who has guided me into the field of Bayesian Statistics and spatial analysis. During my study at the University of Victoria, he has provided infinite help and patience. I would also like to express my thanks to my committee members, Dr. Julie Zhou, Dr. Min Tsao and Dr. Jason L. Loeppky, who have been very helpful in assisting in the completion of the thesis. Thanks also goes to the Pacific Forest Service for providing the data set. In the end, I would like to thank my family members for their support.

Chapter 1

Introduction

Fire plays an important role in forested ecosystems. It can have a significant impact on both forest health and diversity, which in turn, plays a role in timber supply and habitat availability for many plant and animal species (Taylor *et al.*, 2005). Forest fire helps to maintain forest health and diversity by reducing the build-up of dead leaves on the forest floor and eliminating the overhead forest canopy to increase the sunlight that stimulates new growth from seeds and roots. On the other hand, forest fire can have undesirable negative effects on public safety, health and property. As a result, it is important to study the distribution of fire in order to develop effective management strategies that balance the positive ecological aspects of fire with the negative social and economic impacts. Forest fire can have numerous causes such as dry weather, careless human behavior and lightning among many others. Moreover, it is thought that large areas of dead pine trees due to mountain pine beetle (MPB) outbreak *may* lead to an increased frequency of wildfires and we investigate this issue through our modeling and analysis.

The mountain pine beetle is a small, dark-colored, cylindrical beetle that will generally complete its life cycle in one year. Figure 1.1 shows the life cycle of a mountain pine beetle. During the mid-summer and early fall, female beetles mate with males and lay tiny, pearl white eggs under the bark. Eggs hatch into larvae in 10 to 14 days and stay in the larval stage underneath the bark during winter. They resume feeding in spring and grow up to 7 mm in length. By late June to early July, the larvae finish pupating and become adult beetles. After making an exit hole, adult beetles (Figure 1.2) fly to attack new trees in mid-July to mid-August. Mountain

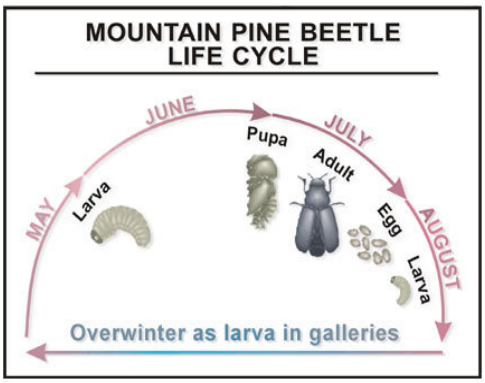

Figure 1.1: Life Cycle of a MPB

Figure 1.2: Adult MPB

pine beetle typically attacks mature or weakened lodgepole pine trees and kills them within a few weeks of successful attack. Due to the unusual hot, dry summers and mild winters in central British Columbia during the recent few years, the current outbreak of mountain pine beetle in the west-central interior of British Columbia is the largest that Canada has ever recorded. As recently reported by the Canadian Forest Service, at the current rate of spread, 50% of the mature pine trees will be dead by 2008 and 80% by 2013. With this increase in infestation, a greater need to understand and quantify the association between infestation and fire frequency has developed. The aim of this thesis is to relate the frequency of wildfires across British Columbia over a 44 year study period to the severity of mountain pine beetle outbreaks through regression modeling. Challenges arise for statistical analysis in this setting as count data on fire frequency exhibit spatial correlation and valid inference on regression coefficients must accommodate this correlation structure.

Our data arise from an observational study monitoring wildfire occurrence across the province of British Columbia. For monitoring fire occurrence, the province is divided into I=1712 homogeneous subregions, with each subregion having the same

total area. Here, each subregion constitutes a grid cell having area $(25\text{km})^2$ and the total number of fires occurring in each cell, during the period 1963 to 2006 has been recorded. Thus, for the i^{th} region, our data consists of a fire count N_i, $i = 1, ..., 1712$ along with region specific covariates $\boldsymbol{x_i}$ that contain auxiliary information that we wish to relate to the mean of fire count. Specifically, for each region we have obtained information on (1) the total area affected by mountain pine beetle, (2) the total area of forest covering, (3) the total area of pine leading stands, (4) the total number of roadways and (5) the drought code, a measure of the moisture content in the floor of the forest, averaged over the 44 year study period, in the region. Our primary focus lies with examining association between fire frequency and mountain pine beetle infestation; however, covariate information on (2) through (5) are included in our analysis as these are known factors related to fire frequency.

The spatial distribution of the counts N_i, $i = 1, .., 1712$ is illustrated in Figure 1.3. There are more fires in the south-east part of the province with relatively fewer towards the north. Figure 1.4 shows that mountain pine beetle outbreaks are clustered in the center of the province. Figure 1.5 presents the spatial distributions of the remaining covariates.

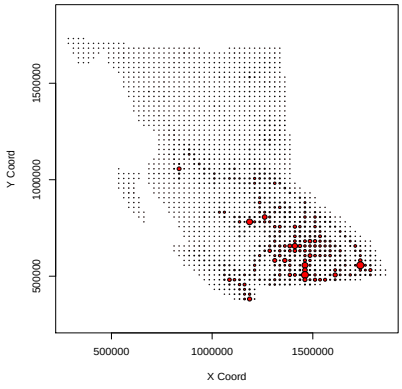

Figure 1.3: Map of Total Fire Counts

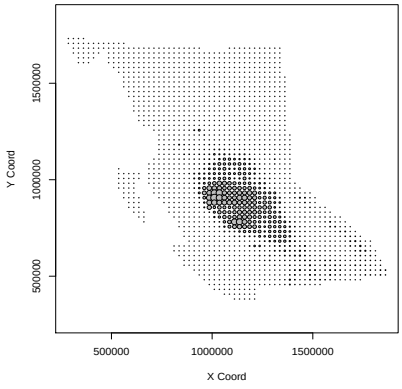

Figure 1.4: Map of MPB Affection

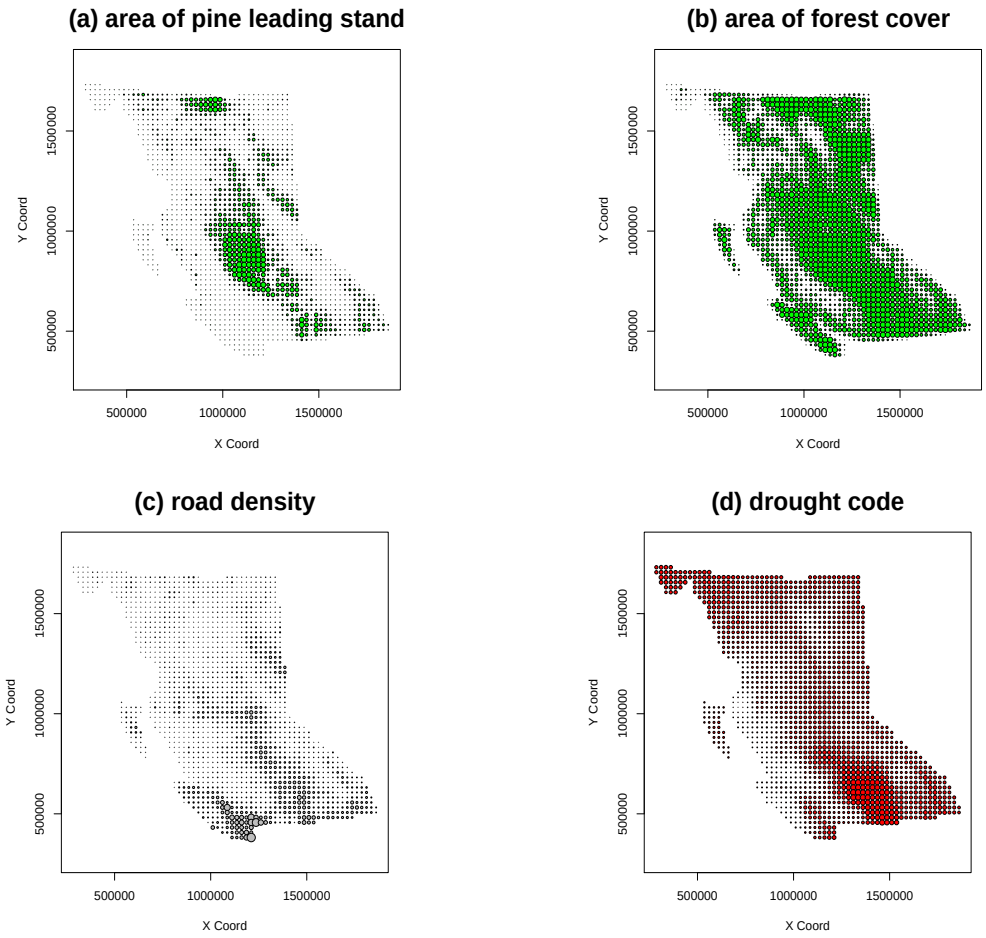

Figure 1.5: Map of Other Covariates

Figure 1.3 indicates a clear spatial structure in the fire frequencies. This spatial structure can likely be explained, in part, through regression modeling incorporating the covariates which, as indicated in Figure 1.4 and 1.5, are themselves spatially varying. A standard regression framework based on generalized linear models (GLM) could thus be considered. Unfortunately, this framework can not accommodate residual spatial variability. That is, spatial structure that can not be explained by the available covariates. Ignoring this residual spatial structure, if it exists, can have drastic consequences on model inference and in particular inference with respect to

parameters in regression models. In particular, using the standard GLM structure can lead to spurious associations resulting from under-estimating variability. In this thesis we will therefore adopt the generalized linear mixed modeling (GLMM) framework, where residual spatial structure is accommodated through random effects that are assigned spatially correlated priors in a hierarchical modeling framework. In particular, we employ Markov random field priors and examine their suitability through simulation.

The remainder of this thesis is structured as follows: In Chapter 2, regression modeling, spatial data and models for spatial data are reviewed. In addition, Bayesian inference and computational methods for implementation are reviewed, discussed extensively and investigated through examples. An exploratory analysis of our data based on simple methods is carried out in Chapter 3, and this analysis motivates the development of our hierarchical spatial model that accommodates special features exhibited by the data. A Negative Binomial regression model with spatial random effects is defined, and detailed parameter estimation methods are proposed in Chapter 4 where we also test our Markov Chain Monte Carlo estimation algorithm using synthetic data sets. Chapter 5 discusses the results of fitting our overdispersed spatial count model to the fire frequency data and Chapter 6 lists directions for future work.

Chapter 2

Literature Review

2.1 Regression Models for Count Data

Regression models are commonly used to assess the relationship between response variables and explanatory variables. They can be used for prediction, inference, and hypothesis testing. A standard regression specification for continuous data is the linear model (Dobson, 1990)

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \text{MVN}(0, \sigma^2 I), \tag{2.1}$$

where $\boldsymbol{y} = [y_1, ..., y_n]^T$ is a vector of response variables, X is a $n \times p$ matrix of covariates and $\boldsymbol{\beta}$ is a p-vector of regression coefficients. The vector $\boldsymbol{\varepsilon}$ is an error term of length n and the elements of $\boldsymbol{\varepsilon}$ are assumed to be independent and identically distributed based on a Gaussian specification $\varepsilon \overset{i.i.d}{\sim} \text{N}(0, \sigma^2)$. This model is appropriate when response variables arise from a normal distribution; however, the normality assumption is not strictly appropriate for discrete data on counts, such as the fire frequencies we consider in our application. For example, if $y_i \overset{ind}{\sim} \text{Poisson}(\lambda_i)$, $i = 1, ..., n$, then it is well known that

$$\text{E}(y_i) = \text{Var}(y_i) = \lambda_i.$$

This is easily seen upon considering the probability mass function

$$f(y_i; \lambda_i) = \frac{\text{e}^{-\lambda_i} \lambda^{y_i}}{y_i!} \qquad \text{for } i = 1, ..., n.$$

from which we can show that

$$\begin{aligned}
\mathrm{E}(y_i) &= \textstyle\sum_{y_i=0}^{\infty} y_i \frac{\exp(-\lambda_i)\lambda_i^{y_i}}{y_i!} \;=\; \sum_{y_i=1}^{\infty} y_i \frac{\exp(-\lambda_i)\lambda_i^{y_i}}{y_i!} \\
&= \textstyle\sum_{y_i=1}^{\infty} \frac{\exp(-\lambda_i)\lambda_i^{(y_i-1)}\lambda_i}{(y_i-1)!} \;=\; \sum_{y_i=0}^{\infty} \frac{\exp(-\lambda_i)\lambda_i^{y_i}\lambda_i}{y_i!} \\
&= \lambda_i \textstyle\sum_{y_i=0}^{\infty} \frac{\exp(-\lambda_i)\lambda_i^{y_i}}{y_i!} \;=\; \lambda_i \\
\\
\mathrm{E}(y_i^2) &= \textstyle\sum_{y_i=0}^{\infty} y_i^2 \frac{\exp(-\lambda_i)\lambda_i^{y_i}}{y_i!} \;=\; \sum_{y_i=1}^{\infty} y_i \frac{\exp(-\lambda_i)\lambda_i^{(y_i-1)}\lambda_i}{(y_i-1)!} \\
&= \lambda_i \textstyle\sum_{y_i=0}^{\infty} (y_i+1) \frac{\exp(-\lambda_i)\lambda_i^{y_i}}{y_i!} \\
&= \lambda_i \underbrace{\sum_{y_i=0}^{\infty} y_i \frac{\exp(-\lambda_i)\lambda_i^{y_i}\lambda_i}{y_i!}}_{=\lambda_i} + \lambda_i \underbrace{\sum_{y_i=0}^{\infty} \frac{\exp(-\lambda_i)\lambda_i^{y_i}}{y_i!}}_{=1} = \; \lambda_i^2 + \lambda_i \\
\mathrm{Var}(y_i) &= \mathrm{E}(y_i^2) - \{\mathrm{E}(y_i)\}^2 \;=\; (\lambda_i^2 + \lambda_i) - \lambda_i^2 \;=\; \lambda_i
\end{aligned}$$

Thus, an appropriate regression model for Poisson distributed data should accommodate the mean/variance relationship $\mathrm{E}(y_i) = \mathrm{Var}(y_i)$, $i = 1, ..., n$; whereas, the linear model (2.1) does not. The variance-stabilizing transformation (Dobson, 1990) is sometimes used to make the linear model more applicable for count data. Suppose $Y \sim \text{Poisson}(\lambda)$ and consider the transformation

$$u = f(y) = \sqrt{y}.$$

The first and second derivative of $f(y)$ are

$$f'(y) = \frac{1}{2\sqrt{y}} \qquad \text{and} \qquad f''(y) = -\frac{1}{4} y^{-\frac{3}{2}}.$$

Then by the delta method (Oehlert, 1992), we can find that

$$\begin{aligned}
\mathrm{E}(u) = \mathrm{E}(f(y)) &\approx f(E[Y]) + \tfrac{1}{2} Var[Y] f''(E[Y]) \\
&= \sqrt{\lambda} + \tfrac{1}{2}\lambda(-\tfrac{1}{4}\lambda^{-\frac{3}{2}}) \\
&= \sqrt{\lambda} - \tfrac{1}{8\sqrt{\lambda}}
\end{aligned}$$

and

$$\begin{aligned}\mathrm{Var}(u) = \mathrm{Var}(f(y)) &\approx [f''(\mu)]^2\sigma^2 \\ &= [\tfrac{1}{2\sqrt{\lambda}}]^2\lambda \\ &= \tfrac{1}{4}\end{aligned}$$

so that the variance is constant after transformation and the mean-variance relationship implied by the linear model is appropriate for the transformed response. Unfortunately, inference is then conducted on a transformed scale; moreover, a one-to-one transformation of a discrete random variable is nothing more than another discrete random variable. Alternatively, a generalized linear model proposed by Nelder and Wedderburn (1972) is more generally applicable for response variables arising from the exponential family of distributions.

2.1.1 Generalized Linear Models

Let us consider a random variable Y that has p.d.f. (p.m.f.) $f(y;\theta)$ which depends on parameter θ. If $f(y;\theta)$ can be written in the form

$$f(y;\theta) = s(y)t(\theta)\exp(a(y)b(\theta)) \tag{2.2}$$

or equivalently,

$$f(y;\theta) = \exp(a(y)b(\theta) + c(\theta) + d(y)), \tag{2.3}$$

where $c(\theta) = \ln(t(\theta))$ and $d(y) = \ln(s(y))$, then $f(y;\theta)$ belongs to the exponential family of distributions (Barndorff-Nielsen, 1978) provided the support of y does not depend on θ. The expected value and variance of $a(Y)$ can be expressed as

$$\mathrm{E}[a(Y)] = -\frac{c'(\theta)}{b'(\theta)} \tag{2.4}$$

and

$$\mathrm{Var}[a(Y)] = \frac{b''(\theta)c'(\theta) - c''(\theta)b'(\theta)}{[b'(\theta)]^2} \tag{2.5}$$

respectively (Dobson, 1990). The distribution in (2.3) is said to be in the canonical form if $a(y) = y$. In this case, (2.4) and (2.5) are the mean and variance for Y.

Given a set of independent random variables $Y_1, ..., Y_n$ from the exponential family of distributions (2.3), we model the expected value, μ_i, of y_i as a linear function of covariates $\boldsymbol{x_i}$ after transformation by

$$\eta_i = g(\mu_i) = \boldsymbol{x_i^T \beta}, \tag{2.6}$$

where $g(\cdot)$ is known as the link function which is usually a nonlinear, monotonicly increasing function transforming μ_i to the real line. The link function provides the relationship between the linear predictor and the mean. When $b(\theta)$ in (2.3) is equal to the linear predictor η_i, the link function $g(\cdot)$ in (2.6) is referred to as the canonical link. The normal regression model is a special case of the generalized linear model where an identity link

$$g(\mu_i) = \mu_i = \boldsymbol{x_i^T \beta}$$

is used. When $Y_i \sim \text{Bin}(n_i, \mu_i)$, the binomial logistic regression is obtained through the logit link

$$g(\mu_i) = \log \frac{\mu_i}{1 - \mu_i} = \boldsymbol{x_i^T \beta}$$

that constrains μ_i between 0 and 1. Often, count data N_i are assumed to follow a Poisson distribution with mean λ_i. The Poisson distribution belongs to the exponential family of distributions in the canonical form since the p.m.f. can be written as

$$\begin{aligned} f(N_i; \lambda_i) &= \frac{\exp^{-\lambda_i} \lambda_i^{N_i}}{N_i!} \\ &= \exp(-\lambda_i + N_i \log \lambda_i - \log N_i!), \end{aligned}$$

where $a(N_i) = N_i$, $b(\lambda_i) = \log \lambda_i$, $c(\lambda_i) = -\lambda_i$ and $d(N_i) = -\log N_i!$. By (2.4) the

expected value of N_i is

$$\begin{aligned}\mu_i = \mathrm{E}[N_i] &= -\tfrac{c'(\lambda_i)}{b'(\lambda_i)} \\ &= -\tfrac{-1}{1/\lambda_i} \\ &= \lambda_i.\end{aligned}$$

The canonical link is $g(\cdot) = \log(\cdot)$ leading to the well known log-linear model:

$$g(\lambda_i) = \log\lambda_i = \boldsymbol{x_i^T \beta}$$

that ensures $\lambda_i > 0$. Other specifications of link functions are possible and inference on $\boldsymbol{\beta}$ typically proceeds through maximum likelihood (Dobson, 1990) as the resulting estimators will have good large sample properties assuming that the model has been correctly specified.

The likelihood function for independent responses $Y_1, ..., Y_n$ in the canonical form of exponential family of distributions can be written as

$$L(\boldsymbol{\theta}; \boldsymbol{y}) = \prod \exp(y_i b(\theta_i) + c(\theta_i) + d(y_i))$$

and the log-likelihood function is

$$l(\boldsymbol{\theta}; \boldsymbol{y}) = \sum y_i b(\theta_i) + \sum c(\theta_i) + \sum d(y_i).$$

In this case,

$$\mathrm{E}(Y_i) = \mu_i = -\frac{c'(\theta_i)}{b'(\theta_i)}$$

and the link function is identified in form (2.6). The global maximum of the log-likelihood function $l(\boldsymbol{\theta}; \boldsymbol{y})$ is given uniquely by solving $\partial l/\partial \boldsymbol{\theta} = 0$ or $\partial l/\partial \boldsymbol{\beta} = 0$ under certain regularity conditions (Cox and Hinkley, 1974). Dobson (1990) showed that

$$\frac{\partial l}{\partial \beta_j} = U_j = \sum_{i=1}^{n} \frac{(y_i - \mu_i)x_{ij}}{\mathrm{Var}(Y_i)} \Big(\frac{\partial \mu_i}{\partial \eta_i}\Big) \tag{2.7}$$

where x_{ij} is the j^{th} element of $\boldsymbol{x}_i^T$. Iterative numerical methods are used to solve the non-linear estimating equations $U_j = 0$ for $j = 1, ..., p$. Beginning with starting

values $\boldsymbol{\beta} = \boldsymbol{b}^{(0)}$, the m^{th} approximation is given by the Newton-Raphson update

$$\boldsymbol{b}^{(m)} = \boldsymbol{b}^{(m-1)} - [A^{(m-1)}]^{-1}\boldsymbol{U}^{(m-1)} \tag{2.8}$$

where

$$A^{(m-1)} = \Big[\frac{\partial^2 l}{\partial\beta_j \partial\beta_k}\Big]_{\boldsymbol{\beta}=\boldsymbol{b}^{(m-1)}}$$

is the matrix of second order derivatives of log-likelihood function evaluated at $\boldsymbol{\beta} = \boldsymbol{b}^{(m-1)}$ and $\boldsymbol{U}^{(m-1)}$ is the vector of U_j in (2.7) evaluated at $\boldsymbol{\beta} = \boldsymbol{b}^{(m-1)}$.

The method of scoring is an alternative procedure for obtaining estimates, where the matrix of second order derivatives A in (2.8) is replaced by the matrix of expected values

$$\mathrm{E}\Big[\frac{\partial^2 l}{\partial\beta_j \partial\beta_k}\Big]$$

which is equal to the negative of the information matrix $I = \mathrm{E}[\boldsymbol{U}\boldsymbol{U}^T]$ that has elements

$$I_{jk} = \mathrm{E}[U_j U_k] = \mathrm{E}\Big[\frac{\partial l}{\partial\beta_j}\frac{\partial l}{\partial\beta_k}\Big] = -\mathrm{E}\Big[\frac{\partial^2 l}{\partial\beta_j \partial\beta_k}\Big].$$

Therefore, the scoring method replaces (2.8) with

$$\boldsymbol{b}^{(m)} = \boldsymbol{b}^{(m-1)} + [I^{(m-1)}]^{-1}\boldsymbol{U}^{(m-1)} \tag{2.9}$$

where $I^{(m-1)}$ denotes the information matrix evaluated at $\boldsymbol{b}^{(m-1)}$. With some algebra, the iterative equation for the scoring method (2.9) can be rewritten as

$$X^T P X \boldsymbol{b}^{(m)} = X^T P \boldsymbol{z} \tag{2.10}$$

where P is the $n \times n$ diagonal matrix with elements

$$P_{ii} = \frac{1}{\mathrm{Var}(Y_i)}\Big(\frac{\partial\mu_i}{\partial\eta_i}\Big)^2$$

and $\boldsymbol{z}$ has elements

$$z_i = \sum_k x_{ik} b_k^{(m-1)} + (y_i - \mu_i)\Big(\frac{\partial\eta_i}{\partial\mu_i}\Big)$$

with μ_i and $\partial\eta_i/\partial\mu_i$ evaluated at $\boldsymbol{b}^{(m-1)}$. The maximum likelihood estimators are obtained by first using some initial values $\boldsymbol{b}^{(0)}$ to evaluate P and $\boldsymbol{z}$. Solving (2.10) gives $\boldsymbol{b}^{(1)}$ and provides a better approximation for P and $\boldsymbol{z}$. Then $\boldsymbol{b}^{(2)}$, $\boldsymbol{b}^{(3)}$ and so on can be found following the same process. Finally $\boldsymbol{b}^{(m)}$ is taken as the maximum likelihood estimate when the difference between $\boldsymbol{b}^{(m)}$ and $\boldsymbol{b}^{(m-1)}$ is sufficiently small so that a relative convergence criteria is satisfied.

2.2 Overdispersion and the Negative Binomial Model

The Poisson distribution is a fundamental distribution used for the analysis of count data. As mentioned earlier, an important assumption for applying the Poisson model to count data is the mean-variance relationship, $\mathrm{E}(y_i) = \mathrm{Var}(y_i)$ whenever y_i has Poisson distribution. This relationship can fail to hold in practice leading to overdispersion $\mathrm{Var}(y_i) > \mathrm{E}(y_i)$ or underdispersion $\mathrm{Var}(y_i) < \mathrm{E}(y_i)$. Using the Poisson model for regression in these situations will result in biased estimates of variability and, in particular, standard errors are underestimated when overdispersion, also known as extra-Poisson variability, exists in the data (Dobson 1990). A natural generalization of the Poisson distribution is the Negative Binomial distribution with two parameters $\lambda_i \geq 0$ and $a \geq 0$, where the dispersion parameter a quantifies extra-Poisson variation (Lawless, 1987). The p.m.f. can be written as

$$f(N_i; \lambda_i, a) = \frac{\Gamma(N_i + \frac{1}{a})}{N_i!\Gamma(1/a)}(\frac{\lambda_i a}{\lambda_i a + 1})^{N_i}(\frac{1}{\lambda_i a + 1})^{1/a}, \qquad N_i = 0, 1, 2, ... \tag{2.11}$$

and a log-linear regression specification is based on

$$\log(\lambda_i) = \boldsymbol{x_i^T}\boldsymbol{\beta}, \qquad i = 1, ..., n$$

where $\boldsymbol{x_i}$ is a $p \times 1$ vector of explanatory variables associated with N_i. Under this parameterization, $\mathrm{E}(N_i) = \lambda_i$ and $\mathrm{Var}(N_i) = \lambda_i + a\lambda_i^2$. The model implies a quadratic mean variance relationship. It is clear that the Negative Binomial model is more

flexible than the Poisson model since the second parameter a allows us to adjust the variance without adjusting the mean. A smaller value of a corresponds to less extra-Poisson variability, with the limiting case $a = 0$ corresponding to the Poisson distribution. The likelihood function for a random sample $N_i \overset{ind}{\sim} \text{NegBin}(\lambda_i, a)$, $i = 1, ..., n$ is proportional to

$$\text{L}(\boldsymbol{\beta}, a) = \prod_{i=1}^{n} \frac{\Gamma(N_i + \frac{1}{a})}{\Gamma(1/a)} (\frac{\lambda_i a}{\lambda_i a + 1})^{N_i} (\frac{1}{\lambda_i a + 1})^{1/a}. \tag{2.12}$$

The maximum likelihood estimates, $\hat{\boldsymbol{\beta}}$ and $\hat{a}$, can be obtained by first maximizing $l(\boldsymbol{\beta}, a) = \log(\text{L}(\boldsymbol{\beta}, a))$ with respect to $\boldsymbol{\beta}$ for a fixed a. This gives the estimates $\hat{\boldsymbol{\beta}}(a)$ that can be found by solving the equation $\partial l(\boldsymbol{\beta}, a)/\partial \boldsymbol{\beta} = 0$ and $\hat{a}$ can be determined through the equation $\partial l(\hat{\boldsymbol{\beta}}(a), a)/\partial a = 0$ where $l(\hat{\boldsymbol{\beta}}(a), a)$ is called the profile likelihood leading to an iterative procedure. Alternatively, given estimates $\tilde{\lambda_i}$, moment estimation for a is obtained by solving the equation

$$\sum_{i=1}^{n} \frac{(N_i - \tilde{\lambda_i})^2}{\tilde{\lambda_i}(1 + a\tilde{\lambda_i})} = n - p, \tag{2.13}$$

and the corresponding estimator $\tilde{\boldsymbol{\beta}}$ of the regression coefficients can be obtained from $\partial l(\boldsymbol{\beta}, \tilde{a})/\partial \boldsymbol{\beta} = 0$. Breslow (1984) suggested the initial value of $\tilde{\lambda_i}$ can be obtained by fitting the Poisson model ($a = 0$). Then the process can be iterated until convergence is reached. The efficiency and robustness properties of inference procedures based on these two estimators have been examined by Lawless (1987). Lawless (1987) showed that when the Negative Binomial model is correct, the estimator $\tilde{\boldsymbol{\beta}}$ is asymptotically equivalent to the maximum likelihood estimator $\hat{\boldsymbol{\beta}}$ and the moment estimation for a is likely to be more robust but less efficient than the maximum likelihood estimation for a. When the Negative Binomial model is wrong, that is, the regression specification is correct but the distribution of N_i given $\boldsymbol{x_i}$ is not Negative Binomial, in general, we get consistent estimation for $\boldsymbol{\beta}$ but the maximum likelihood estimation has the covariance

matrix wrong. Suppose $\text{Var}(N_i \mid \boldsymbol{x_i}) = \sigma_i^2$, the correct asymptotic covariance matrix for $\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})$ is $I_1^{-1} B_1 I_1^{-1}$; however, the asymptotic covariance matrix for $\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})$ in this case converges in probability to I_1^{-1}, where

$$(I_1)_{r,s} = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \frac{\lambda_i x_{ir} x_{is}}{1 + a\lambda_i} \qquad \text{and} \qquad (B_1)_{r,s} = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \frac{\sigma_i^2 x_{ir} x_{is}}{(1 + a\lambda_i)^2}.$$

Under the same model specification, the moment estimation gives consistent estimation of a. It also yields an asymptotically correct covariance matrix for $\tilde{\boldsymbol{\beta}}$ which appears to be an advantage of the moment estimator over the maximum likelihood estimator under model misspecification. Lawless (1987) also mentioned that for obtaining test and confidence intervals about $\boldsymbol{\beta}$ and a, the likelihood-ratio statistics with their distributions approximated by χ^2 distributions are satisfactory even for small samples.

2.3 Spatial Modeling

Generalized linear models defined in Section 2.1.1 focus on analyzing data under the independence assumption. Often, data from diverse areas such as climatology, ecology, environmental monitoring, and health science are spatially correlated. This implies that the dependence structure underlying the data is some function of location information. Typically, observations located closer together will be more similar than those farther apart and thus the data exhibit spatial variability. It is well established that ignoring spatial dependence in the data when working with regression models will result in biased estimates of variation and inefficient statistical inference (Cressie, 1993). Therefore, in order to accurately assess the association between response and covariates, it is important to allow for spatial dependence when developing regression models for data that may be spatially correlated. This is crucial in our application where we want to assess the association between mountain pine beetle infection and fire.

Spatial data are typically classified into one of three types: point-referenced data, point pattern data or areal data (Banerjee *et al.*, 2004). The first case, point-referenced data is often referred to as geocoded or geostatistical data. The response $Y(s)$ is a random variable observed at location $s \in R^r$ where s varies continuously over some fixed region $D \subseteq R^r$ and the index set D contains an r-dimensional rectangle of positive volume. Taking the air pollution monitoring as an example, data are collected at some fixed stations. Interest lies in predicting the response $Y(s^*)$ at some unobserved location s^* based on observed value at a fixed set of locations $Y(s_1)$, $Y(s_2)$, ..., $Y(s_n)$. We model the dependence between $Y(s_i)$ and $Y(s_{i'})$ through a covariance function

$$\text{Cov}[Y(s_i), Y(s_{i'})] = c(s_i, s_{i'}) = c(d_{ii'})$$

where $d_{ii'}$ is the distance between s_i and $s_{i'}$. If the covariance function only depends on $d_{ii'}$, it is called isotropic covariance function. One frequently used specification of $c(d_{ii'})$ is the exponential model

$$c(d_{ii'}) = \begin{cases} \sigma^2 \exp(-\phi d_{ii'}) & \text{if } d_{ii'} > 0 \\ \tau^2 + \sigma^2 & \text{otherwise} \end{cases} \tag{2.14}$$

where τ^2, σ^2 and ϕ are positive parameters. When $d_{ii'} = 0$ that is s_i and $s_{i'}$ denote the same location,

$$c(0) = \text{Var}[Y(s_i)] = \tau^2 + \sigma^2$$

where τ^2 and σ^2 are non-spatial (nugget effect) and spatial variance component respectively. Many other choices are possible for $c(d_{ii'})$ such as the spherical, the Gaussian and the Matérn (Banerjee *et al.*, 2004). To estimate parameters in $c(d_{ii'})$ based on data $\boldsymbol{Y} = [Y(s_1), ..., Y(s_n)]^T$, we typically make a stationary Gaussian process assumption for $Y(s)$, $s \in D$ in which case

$$\boldsymbol{Y} \mid \mu, \boldsymbol{\theta} \sim \text{MVN}(\mu \mathbf{1}, \Sigma(\boldsymbol{\theta})),$$

where

$$(\Sigma(\boldsymbol{\theta}))_{ii'} = c(d_{ii'}),$$

and in the case of exponential covariance function (2.14), $\boldsymbol{\theta} = (\sigma^2, \tau^2, \phi)^T$. Having estimated $\boldsymbol{\theta}$ and μ, classical spatial prediction of $Y(s^*)$ proceeds through the best linear unbiased predictor (BLUB) also known as kriging (Cressie, 1993).

In the second type of spatial data, point pattern data, it is the index set D itself that is random and gives the locations of random events that are the point pattern. In the simplest setting, $Y(s) = 1$ for all $s \in D$ that indicates the occurrence of the event. This process marks the location of random events, for examples: location of lightening strikes, location of disease outbreaks and location of earthquake epicenters. In this setting we are typically interested in determining the spatial pattern of the process. When an event is equally likely to occur at any point in the observation area regardless of the locations of other events, we term it as complete spatial randomness. The stochastic process which characterizes complete spatial randomness is the homogeneous Poisson process. There are two alternatives to complete spatial randomness. Spatial clustering implies that event points tend to be spatially close to other points, and spatial regularity implies that event points space themselves out as much as possible. For analysis, plots of the data are typically a good place to start and the Ripley's K function (Banerjee *et al.*, 2004) is commonly used for measuring clustering and is given by

$$K(r) = \frac{1}{\lambda}\text{E}[\text{number of points within radius } r \text{ of an arbitrary point}], \qquad (2.15)$$

where λ is the intensity parameter that represents the mean number of points per unit area. This quantity (2.15) can be estimated by

$$\hat{K}(r) = n^{-2}|A| \sum \sum_{i \neq j} p_{ij}^{-1} I_r(d_{ij}),$$

where A is the observation area, n is the number of points observed in A and d_{ij} is the distance between point i and point j. The function $I_r(d_{ij}) = 1$ if $d_{ij} < r$ and equals to 0 otherwise. The value of p_{ij} denotes the proportion of the circle centered at i and passing through j that lies within A. We compare $\hat{K}(r)$ to the theoretical value $K(r) = \pi r^2$ since E[events within radius r of an arbitrary event] $= \lambda \pi r^2$ for a homogeneous Poisson process. If the data are clustered we expect $\hat{K}(r) > \pi r^2$ while $\hat{K}(r) < \pi r^2$ suggests that data points follow some regularly space pattern.

The third and final type of spatial data is termed areal data, where responses are observed on a regular or irregular lattice typically consisting of a set of geographical regions with well-defined boundaries. The fire count data examined in this thesis falls into this class, with counts observed over $I = 1712$ subregions dividing the province. Here, we observe Y_1, Y_2, ..., Y_n associated with geographical regions (also called areal units) 1, 2, .., n. The spatial structure underlying the observations is often summarized through an adjacency matrix W, whose entries code, in some sense, the connectivity (also called the neighbourhood structure) of the underlying map. This adjacency matrix is a primary component to spatial models for areal data. A typical definition for the adjacency matrix, spatially connecting units i and j is

$$W_{ij} = \begin{cases} 0 & \text{if } i = j \\ 0 & \text{if } i \text{ and } j \text{ are not neighbours} \\ c_{ij} & \text{if } i \text{ and } j \text{ are neighbours.} \end{cases}$$

Here $c_{ij} > 0$ quantifies the strength of the neighbour relationship between areal units i and j. Many forms for the connectivity weights are possible, for example

$$c_{ij} = \exp\{-d_{ij}\},$$

where d_{ij} is the intercentroidal distance between region i and region j. The most commonly employed connectivity weights, and the form that we adopt in this thesis

is simply based on adjacency, where we set

$$c_{ij} = \begin{cases} 1 & \text{if region } i \text{ and region } j \text{ share some common boundary} \\ 0 & \text{otherwise.} \end{cases}$$

For example, if we divide a region into 3×3 grid cells as in Figure 2.1, then the

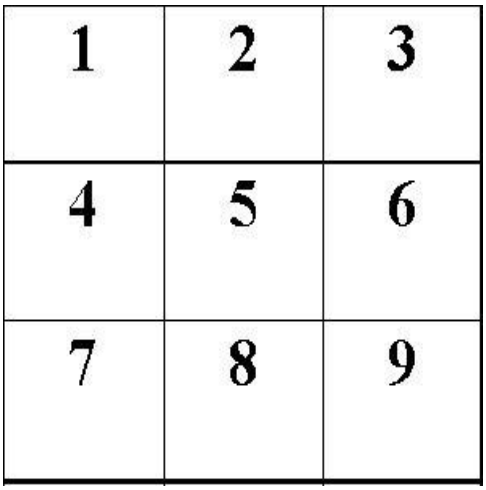

Figure 2.1: 3×3 grid cells

corresponding 9×9 adjacency matrix takes the form

$$W = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{2.16}$$

In order to model spatial dependence in the data Y_1, Y_2, ..., Y_n we shall construct the joint distribution $f(y_1, y_2, ..., y_n)$ through the specification of a set of simple full conditional distributions $f(y_i \mid y_j, j \neq i), i = 1, ..., n$. This idea of constructing a complicated joint distribution for spatial correlated random variables through a set of simple local specifications was pioneered by Besag (1974), and has been applied extensively in image analysis and disease mapping. In this context, it is clear that given a joint distribution, the full conditional distributions are always uniquely determined; however the converse is not always true. In general, we say that the set of full conditional distributions is compatible if they determine a unique and valid joint distribution. If we have a set of compatible full conditional distributions $f(y_i \mid y_j, j \neq i), i = 1, ..., n$, the form of resulting joint distribution can be constructed using Brook's Lemma (Brook, 1964). Brook's Lemma notes that

$$\begin{aligned} f(y_1, y_2, ..., y_n) = \; & \frac{f(y_1|y_2,...,y_n)}{f(y_{10}|y_2,...,y_n)} \cdot \frac{f(y_2|y_{10},y_3,...,y_n)}{f(y_{20}|y_{10},y_3,...,y_n)} \\ & \cdots \frac{f(y_n|y_{10},...,y_{n-1,0})}{f(y_{n0}|y_{10},...,y_{n-1,0})} \cdot f(y_{10}, ..., y_{n0}), \end{aligned} \tag{2.17}$$

where $\boldsymbol{y_0} = (y_{10}, ..., y_{n0})^T$ is any fixed point in the support of $f(y_1, ..., y_n)$. This gives us a joint distribution up to a normalizing constant say

$$f(y_1, ..., y_n) \propto p(y_1, ..., y_n)$$

since $f(y_{10}, ..., y_{n0})$ is an unknown constant. If $p(y_1, ..., y_n)$ is improper that is

$$\int_S p(y_1, ..., y_n) d\boldsymbol{y} = \infty,$$

where S is the sample space of $\boldsymbol{Y}$, then this is the best we can do. If $p(y_1, ..., y_n)$ is proper that is

$$\int_S p(y_1, ..., y_n) d\boldsymbol{y} < \infty,$$

then we can find some $A < \infty$ so that

$$\frac{1}{A}\int_S p(y_1, ..., y_n)d\boldsymbol{y} = 1$$

and $1/A$ is the desired normalizing constant for the joint distribution. A simple bivariate example will illustrate the idea of finding the joint distribution through compatible conditional distributions using Brook's Lemma.

Example 1 (Banerjee *et al.*, 2004): For two binary variables Y_1 and Y_2, suppose $Y_1 \mid Y_2 \sim \text{Bin}(1, \Pi_1)$ and $Y_2 \mid Y_1 \sim \text{Bin}(1, \Pi_2)$ where $\Pi_1 = \Pr(Y_1 = 1 \mid Y_2)$ and $\Pi_2 = \Pr(Y_2 = 1 \mid Y_1)$ and suppose that Π_1 and Π_2 are defined through the conditional logit models:

$$\log\frac{\Pi_1}{1-\Pi_1} = \alpha_0 + \alpha_1 Y_2 \tag{2.18}$$

and

$$\log\frac{\Pi_2}{1-\Pi_2} = \beta_0 + \beta_1 Y_1 \tag{2.19}$$

Solving (2.18), we can find that

$$\Pi_1 = \frac{\exp(\alpha_0 + \alpha_1 Y_2)}{1 + \exp(\alpha_0 + \alpha_1 Y_2)}. \tag{2.20}$$

Similarly,

$$\Pi_2 = \frac{\exp(\beta_0 + \beta_1 Y_1)}{1 + \exp(\beta_0 + \beta_1 Y_1)} \tag{2.21}$$

can be found by solving (2.19). By Brook's Lemma (2.17), the joint distribution of Y_1 and Y_2 can be written as

$$f(y_1, y_2) = \frac{f(y_1 \mid y_2)}{f(y_1 = 0 \mid y_2)}\frac{f(y_2 \mid y_1 = 0)}{f(y_2 = 0 \mid y_1 = 0)}f(0, 0).$$

Since $Y_1 \mid Y_2 \sim \text{Bin}(1, \Pi_1)$ and $Y_2 \mid Y_1 \sim \text{Bin}(1, \Pi_2)$,

$$f(y_1, y_2) = \frac{\Pi_1^{y_1}(1-\Pi_1)^{1-y_1}}{\Pi_1^0(1-\Pi_1)^{1-0}}\frac{(\Pi_2')^{y_2}(1-\Pi_2')^{1-y_2}}{(\Pi_2')^0(1-\Pi_2')^{1-0}}f(0, 0) \tag{2.22}$$

where

$$\Pi_2' = \Pr(Y_2 = 1 \mid Y_1 = 0) = \frac{\exp(\beta_0)}{1 + \exp(\beta_0)}. \tag{2.23}$$

After substituting (2.20) and (2.23) into (2.22) and simplifying, we have

$$f(y_1, y_2) = [\exp(\alpha_0 + \alpha_1 y_2)]^{y_1} [\exp(\beta_0)]^{y_2} f(0, 0). \tag{2.24}$$

By using the fact that $\sum_{(y_1,y_2)} f(y_1, y_2) = 1$, we can find the constant $f(0, 0)$ in (2.24) is

$$f(0, 0) = [1 + \exp(\beta_0) + \exp(\alpha_0) + \exp(\alpha_0 + \alpha_1)\exp(\beta_0)]^{-1}.$$

Therefore,

$$f(y_1, y_2) = \frac{[\exp(\alpha_0 + \alpha_1 y_2)]^{y_1} [\exp(\beta_0)]^{y_2}}{1 + \exp(\beta_0) + \exp(\alpha_0) + \exp(\alpha_0 + \alpha_1)\exp(\beta_0)}$$

is the joint distribution of binary Y_1 and Y_2 defined through (2.18) and (2.19).

In this way we shall specify a joint distribution through a set of compatible full conditional distributions, and use Brook's lemma to derive the resulting form of the joint distribution. The conditional distributions will depend on the neighbourhood structure underlying the map $\mathbf{W}$, and spatial dependence is thus built into the joint model specification. Having defined a neighbourhood structure, the full conditional distribution of Y_i can be written as

$$f(y_i \mid y_j, j \neq i) = f(y_i \mid y_j, j \in \partial_i), \qquad i = 1, ..., n,$$

where $\partial_i = \{j \mid W_{ij} \neq 0\}$ denotes the neighbours for region i. We use a set of simple local specifications that depend only on lattice adjacencies to develop a spatial dependence structure. This sort of specification is referred to as a Markov random field (MRF) (Besag, 1974). In this thesis we will focus on a specific type of Markov random field model known as the Gaussian conditionally autoregressive model.

A conditionally autoregressive (CAR) model (Besag, 1974) for modeling areal data $\boldsymbol{y} = [y_1, ..., y_n]^T$ is, in a sense, the simplest non-trivial special case of MRF that can be used for modeling spatial data. This model is specified through the set of full conditional distributions

$$[y_i \mid y_j, i \neq j] \sim \mathrm{N}\Big(\sum_j \frac{W_{ij}}{W_{i+}} y_j, \frac{\sigma^2}{W_{i+}}\Big), \qquad i = 1, .., n,$$

where $W_{i+} = \sum_{j=1}^{n} W_{ij}$ denotes the sum of i^{th} row in W and σ^2 is the variance component. The mean of $[y_i \mid y_j, i \neq j]$ is nothing more than a weighted average, obtained from the y_j's corresponding to the neighbouring regions of region i. Note also that the variance is inversely proportional to the number of neighbours that region i has, which seems intuitive. By Brook's Lemma (2.17), the joint distribution can be written as

$$P(y_1, ..., y_n) \propto \exp\Big\{ -\frac{1}{2\sigma^2} \mathbf{y}^T (D_W - W)\mathbf{y} \Big\}, \tag{2.25}$$

or with a little algebra (2.25) can be rewritten as

$$P(y_1, ..., y_n) \propto \exp\Big\{ -\frac{1}{2\sigma^2} \sum_{i \neq j} W_{ij}(y_i - y_j)^2 \Big\},$$

where $D_W = \mathrm{diag}\{W_{1+}, W_{2+}, ..., W_{n+}\}$ is a diagonal matrix. Note that since $(D_W - W)\mathbf{1} = \mathbf{0}$, (2.25) is a singular multivariate normal distribution. There does not exist a valid normalizing constant, so it is an improper distribution and is often referred to as an intrinsically autoregressive model (IAR). Because data could not arise under an improper stochastic mechanism, we can not use (2.25) as a model for data, but it can be used as a prior model for spatial random effects (Banerjee *et al.*, 2004). For example, the Poisson mixed model is commonly used in spatial epidemiology and can be simply written as:

$$\begin{aligned} Y_i \mid \mu_i &\overset{ind}{\sim} \mathrm{Poisson}(\mu_i), \qquad i = 1, ..., n, \\ \log(\mu_i) &= \boldsymbol{x_i^T \beta} + b_i, \\ \boldsymbol{b} &= [b_1, ..., b_n]^T \sim \mathrm{CAR}(\sigma^2). \end{aligned} \tag{2.26}$$

This kind of model is called a generalized linear mixed spatial model. Fitting the model (2.26) can be difficult using standard techniques based on maximum likelihood estimation as the likelihood function will involve an integral of dimension n. A Bayesian approach is often adopted for inference in such hierarchical spatial models. Here, inference is based on computing a posterior distribution of unknowns, given the data, and exact inference is obtained, without reliance on asymptotics, which is a nice feature of the Bayesian approach. In the next section we review some basic ideas related to Bayesian inference and computation.

2.4 Bayesian Inference and Computation

Statistical inference concerns the learning of some unknown aspect of the population from which the data were drawn. Bayesian inference fits a probability model to observed data and summarizes the result through a probability distribution on the unknown parameters $\boldsymbol{\theta}$ or unobserved data $\tilde{y}$ we are interested in. In other words, Bayesian inferences are made in terms of probability statements conditional on the observed data $\boldsymbol{y}$. The Bayesian method offers potentially attractive advantages over the frequentist statistical approach for modeling spatial data (Banerjee *et al.*, 2004). First, the Bayesian approach allows us to induce specific spatial correlation among random effects through prior distributions. Second, the marginal likelihood function can be complex in multidimensional and constrained settings even though some numerical procedures such as the EM algorithm have been introduced to handle this. Under the Bayesian setting, computational challenges associated with computing posterior distributions can be overcome by applying Markov Chain Monte Carlo methods which will be introduced in section 2.4.2. Third, hierarchical Bayesian models provides a mechanism to specify a complicated model for non-Gaussian, dependent data through several layers, each of which can be easily understood and computed. Fi-

nally, the Bayesian method explicitly acknowledges the uncertainty of the model and parameters.

2.4.1 Framework for Bayesian Inference

A Bayesian statistical model is made of a sampling distribution (likelihood function), $P(\boldsymbol{y} \mid \boldsymbol{\theta})$, for observed data conditional on unknowns $\boldsymbol{\theta}$, and a prior distribution, $p(\boldsymbol{\theta})$, that reflects various degrees of belief on the likely values of unknowns (Robert, 2001). Given these two distributions, the joint distribution or full probability model can be written as

$$p(\boldsymbol{\theta}, \boldsymbol{y}) = P(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$$

and the posterior distribution is obtained via Bayes' rule (Gelman *et al.*, 2004)

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{y})}{p(\boldsymbol{y})} = \frac{P(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{y})}, \tag{2.27}$$

where $p(\boldsymbol{y}) = \sum_{\boldsymbol{\theta}} P(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$ in the case of discrete $\boldsymbol{\theta}$ and $p(\boldsymbol{y}) = \int L(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$ in the case of continuous $\boldsymbol{\theta}$. Since $p(\boldsymbol{y})$ does not depend on $\boldsymbol{\theta}$, it can be considered as a constant with fixed $\boldsymbol{y}$. Therefore, (2.27) can be obtained up to normalizing constant as,

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) \propto P(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}), \tag{2.28}$$

that is proportional to the likelihood function times the prior. Using numerical methods described in the next section, we can work with (2.28) for model estimation and avoid computing the normalizing constant which is not easily obtained.

Complex models can be built through the specification of several simple stages under a Bayesian hierarchical framework. A hierarchical Bayes model (Robert, 2001) is a Bayesian statistical model where the prior distribution $p(\boldsymbol{\theta})$ is decomposed into several conditional levels of distributions

$$p_1(\boldsymbol{\theta} \mid \boldsymbol{\theta_1}),\ p_2(\boldsymbol{\theta_1} \mid \boldsymbol{\theta_2}),\ ...,\ p_n(\boldsymbol{\theta_{n-1}} \mid \boldsymbol{\theta_n})$$

and a marginal distribution

$$p_{n+1}(\boldsymbol{\theta_n})$$

such that

$$p(\boldsymbol{\theta}) = \int_{\boldsymbol{\Theta_1}\times\ldots\times\boldsymbol{\Theta_n}} p_1(\boldsymbol{\theta} \mid \boldsymbol{\theta_1}) \cdots p_n(\boldsymbol{\theta_{n-1}} \mid \boldsymbol{\theta_n}) p_{n+1}(\boldsymbol{\theta_n}) d\boldsymbol{\theta_1} \cdots \boldsymbol{\theta_n}.$$

The parameters $\boldsymbol{\theta_i}$ are called hyper-parameters of level i, for $1 \leq i \leq n$. The most common hierarchical Bayesian model is the case when $n = 2$. At the first stage, a distribution for the data given parameters is specified. At the second stage, prior distributions for parameters given hyper-parameters are specified and distributions for hyper-parameters are specified at the third stage. The prior distributions at the first level may correspond to the structural information about the model such as uncertain linear restrictions on the parameters of a regression model, whereas the prior distributions at the second level correspond to the more subjective information that accounts for the imprecision of these restrictions. The hierarchical modeling improves the robustness of the resulting Bayes estimators, since uncertainty regarding the model structure can be incorporated into additional prior distributions. In addition, the decomposition of a prior distribution into its components in the hierarchical Bayes approach simplifies Bayesian calculations and allows for an easier approximation of some posterior quantities by simulation.

The choice of prior distribution is critical for Bayesian inference (Gelman *et al.*, 2004). If prior information is available from external knowledge, this information can be used to construct a prior distribution for unknowns. The mechanism for converting prior information to prior probability distributions is often unclear; moreover, prior information will typically not induce a unique prior distribution. Often, there is little prior information regarding model unknowns, in which case a noninformative or vague prior distribution can be employed. Such priors typically arise in the form of a parametric distribution with large or infinite variance. For large data sets, this

approach is reasonable as the likelihood will dominate the prior, and inference will be primarily data-driven. For small data sets, this approach is not reasonable and inference will be sensitive to prior choice.

When the posterior distribution follows the same parametric form as the prior distribution, the prior is called a conjugate prior. Probability distributions that belong to the exponential family of distributions always have conjugate prior distributions (Robert, 2001). Suppose $f(y_i \mid \theta)$'s are from the exponential family of distributions and have form (2.2) for $i = 1, ..., n$. The likelihood function for a random sample is then

$$L(\boldsymbol{y}; \theta) = \phi(\boldsymbol{y}) t(\theta)^n \exp(w(\boldsymbol{y}) b(\theta)), \tag{2.29}$$

where

$$\phi(\boldsymbol{y}) = \prod_{i=1}^{n} s(y_i) \qquad \text{and} \qquad w(\boldsymbol{y}) = \sum_{i=1}^{n} a(y_i).$$

If the prior distribution of θ is specified as

$$p(\theta) \propto t(\theta)^{\eta} \exp(b(\theta) v),$$

then the posterior distribution is

$$p(\theta \mid \boldsymbol{y}) \propto t(\theta)^{n+\eta} \exp(b(\theta)(w(\boldsymbol{y}) + v))$$

which has the same density form as the prior distribution. This choice of prior density is conjugate and is often called the natural conjugate prior. For example, the Gamma(α, β) distribution with p.d.f.

$$\begin{aligned} p(\theta) &= \tfrac{\beta^{\alpha}}{\Gamma(\alpha)} \theta^{\alpha-1} \exp(-\beta\theta) \\ &\propto \theta^{\alpha-1} \exp(-\beta\theta) \end{aligned}$$

is a natural conjugate prior for a Poisson distribution in the form

$$f(x \mid \theta) = \frac{\theta^x \exp(-\theta)}{x!}$$

In this case, the posterior distribution can be written as

$$\begin{aligned} p(\theta \mid x) &\propto f(x \mid \theta) \times p(\theta) \\ &\propto \theta^x \exp(-\theta)\theta^{\alpha-1}\exp(-\beta\theta) \\ &\propto \theta^{x+\alpha-1}\exp(-\theta(1+\beta)), \end{aligned}$$

which is the kernel of Gamma$(x+\alpha, 1+\beta)$ distribution. Table 2.1 lists natural conjugate priors for some common exponential families. Conjugate priors can be convenient to work with and can simplify computation as we shall see in the next section which discusses Bayesian computation through the Metropolis-Hastings algorithm.

$f(x \mid \theta)$	$p(\theta)$	$p(\theta \mid x)$
Normal(θ, σ^2)	Normal(μ, τ^2)	Normal$(\frac{\sigma^2\mu+\tau^2 x}{\sigma^2+\tau^2}, \frac{\sigma^2\tau^2}{\sigma^2+\tau^2})$
Poisson(θ)	Gamma(α, β)	Gamma$(\alpha+x, \beta+1)$
Gamma(ν, θ)	Gamma(α, β)	Gamma$(\alpha+\nu, \beta+x)$
Binomial(n, θ)	Beta(α, β)	Beta$(\alpha+x, \beta+n-x)$
NegBin(m, θ)	Beta(α, β)	Beta$(\alpha+m, \beta+x)$
Multinomial$(\theta_1, ..., \theta_k)$	Dirichlet$(\alpha_1, ..., \alpha_k)$	Dirichlet$(\alpha_1+x_1, ..., \alpha_k+x_k)$
Normal$(\mu, 1/\theta)$	Gamma(α, β)	Gamma$(\alpha+0.5, \beta+(\mu-x)^2/2)$

Table 2.1: Natural conjugate priors for some common exponential families

2.4.2 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a general numerical framework for generating dependent realizations from possibly high dimensional probability distributions. This framework is used for Bayesian inference where it is of interest to summarize the posterior distribution (2.27) and the corresponding normalizing constant can not be obtained analytically. In what follows we review some basic theory related to Markov

chains, in particular the limit theorems which justify the use of MCMC in practice. We will then discuss the Gibbs sampler and Metropolis-Hastings algorithms, illustrating these algorithms with some simple examples. Finally, practical implementation issues are discussed.

In general, a stochastic process (Gamerman and Lopes, 2006) is defined as a collection of random quantities denoted $\theta^{(t)} \in S$ where $t \in T$. The index set T takes nonnegative integers and S is known as the state space. For simplicity of presentation, we will start by assuming S is discrete. We will then briefly discuss results for general state spaces. A Markov chain is a special type of stochastic process where the past and future states are conditionally independent given the current state. This property can be stated as

$$Pr(\theta^{(n+1)} \in A \mid \theta^{(n)} = x, \theta^{(n-1)} \in A_{n-1}, ..., \theta^{(0)} \in A_0) = Pr(\theta^{(n+1)} \in A \mid \theta^{(n)} = x) \tag{2.30}$$

for sets $A, A_{n-1}, ..., A_0 \subset S$ and $x \in S$. In the case of homogeneous Markov chain where (2.30) does not depend on n, a transition kernel $P(x, A)$ can be defined as:

1. $P(x, .)$ is a probability distribution over S for all $x \in S$;

2. the function $x \mapsto P(x, A)$ can be evaluated for all $A \subset S$.

When dealing with discrete state space, we often work with the set A of the form $A = \{y\}$ and the transition probability $P(x, \{y\}) = P(x, y)$ is defined as:

1. $P(x, y) \geq 0$ for $\forall x, y \in S$;

2. $\sum_{y \in S} P(x, y) = 1$ for $\forall x \in S$.

For a discrete state space S with r elements, a $r \times r$ transition matrix P can be

established as

$$P = \begin{bmatrix} P(x_1, x_1) & \cdots & P(x_1, x_r) \\ \vdots & & \vdots \\ P(x_r, x_1) & \cdots & P(x_r, x_r) \end{bmatrix}.$$

The transition probability from state x to state y over m steps can be obtained as the matrix product of P m-times denoted as P^m (Gamerman and Lopes, 2006). We let $\pi^{(n)}$ with components $\pi_n(x_i) = Pr(\theta^{(n)} = x_i)$ denote a row vector containing marginal probabilities associated with $\theta^{(n)}$. The recursive relationship between successive marginal distributions of the chain can be written as $\pi^{(n)} = \pi^{(0)}P^{n-1}P = \pi^{(n-1)}P$. We let ρ_{xy} denote the probability of the chain, starting from state x, eventually reaching state y. A state $y \in S$ is said to be recurrent if $\rho_{yy} = 1$ and is said to be transient if $\rho_{yy} < 1$. For a recurrent state y, if $\mathrm{E}[T_y \mid \theta^{(0)}] < \infty$ where $T_y = \min\{n \geq 1 : \theta^{(n)} = y\}$ denotes the hitting time of y, the state y is said to be positive recurrent which is an important property for establishing limiting results. Within the context of iterative simulation algorithms, asymptotic behavior of the chain as the number of iterations $n \to \infty$ is the most important area of the Markov chain theory. A distribution π is said to be a stationary distribution of a chain with transition probabilities $P(x, y)$ if

$$\sum_{x \in S} \pi(x)P(x, y) = \pi(y) \qquad \text{for} \qquad \forall y \in S \tag{2.31}$$

which in matrix form is $\pi = \pi P$. If the stationary distribution π exists and

$$\lim_{n\to\infty} P^{(n)}(x, y) = \pi(y),$$

then the sequence of marginal distributions $\pi^{(n)}$ will approach π as $n \to \infty$, independently of the initial distribution of the chain. In this sense, π is also referred to as the limiting distribution. There are situations where stationary distributions exist but limiting distributions do not (Gamerman and Lopes, 2006). In order to establish limiting results, we will introduce the notion of periodicity. The period of a state x

is the largest common divisor of the set $\{n \geq 1 : P^{(n)}(x,x) > 0\}$ denoted by d_x. A state is said to be ergodic if the state is positive recurrent and aperiodic ($d_x = 1$). In addition, a chain is ergodic if all its states are ergodic. Given this, an important limiting theorem, the ergodic theorem, can be stated based on the ergodicity of the chain. Suppose $\theta^{(n)}$ is ergodic with stationary distribution π and $t(\theta)$ is a real valued function with $\mathrm{E}_\pi[t(\theta)] < \infty$. Then the ergodic average

$$\bar{t}_n = (1/n)\sum_{i=1}^{n} t(\theta^{(i)}) \overset{a.s}{\to} \mathrm{E}_\pi[t(\theta)] \qquad \text{as} \qquad n \to \infty.$$

That is, the Markov chain is equivalent to the strong law of large numbers and it is this theorem that justifies the use of MCMC for estimating expectations taken with respect to the posterior distribution for Bayesian inference.

In practice, when using Markov Chain simulation to fit statistical models in a Bayesian framework, the state space S corresponds to a parameter space that in general will not be a discrete set. Nevertheless, the ergodic theorem described above can be extended and applied to more general state spaces. Assuming S is a continuous state space, the transition kernel is defined through a conditional probability density function

$$p(x,y) = \frac{\partial P(x,y)}{\partial y}$$

where

$$P(x,y) = Pr(\theta^{(n+1)} \leq y \mid \theta^{(n)} = x) = Pr(\theta^{(1)} \leq y \mid \theta^{(0)} = x), \qquad \text{for } x, y \in S.$$

Then the continuous version of (2.31) can be written as

$$\pi(y) = \int_{-\infty}^{\infty} \pi(x)p(x,y)dx \tag{2.32}$$

where π is the stationary distribution of the chain. With these definitions, the limiting results considered in the discrete case will carry over to the continuous case, though,

a thorough technical presentation of these general results is beyond the scope of this thesis.

The key to MCMC simulation for Bayesian inference is to simulate realizations $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\theta}^{(1)}$, $\boldsymbol{\theta}^{(2)}$, ... from an ergodic Markov chain whose stationary distribution is the posterior distribution of interest. Starting from an initial state $\boldsymbol{\theta}^{(0)}$, realizations of the chain are produced successively until the chain 'forgets' this initial state and begins to exhibit its steady state behavior. If the chain reaches approximate stationarity at iteration T, the set of sampled values, $\boldsymbol{\theta}^{(0)}$, ..., $\boldsymbol{\theta}^{(T)}$, is discarded as a 'burn-in' period and successive realizations $\boldsymbol{\theta}^{(T+1)}$, $\boldsymbol{\theta}^{(T+2)}$, $\boldsymbol{\theta}^{(T+3)}$, ... are approximate draws from the posterior distribution (which is the stationary distribution of the Markov chain). Bayes inference can be based on summarizing the posterior distribution using a Monte Carlo sample of size J draws after the burn-in period. For a given Bayesian inference problem, there are many ways to construct the required Markov chain. We will introduce the two most widely used MCMC algorithms, the Gibbs sampler and the Metropolis-Hastings algorithm, in Sections 2.4.2.1 and 2.4.2.2. We will then, in Section 2.4.3, discuss convergence diagnostics for Markov chain simulation.

2.4.2.1 Gibbs Sampling

Gibbs sampling is a useful MCMC simulation scheme that samples iteratively from the full conditional distribution of each parameter, given all the other parameters and the data. The original work is presented by Geman and Geman (1984) within the context of image analysis but has since been applied in far more general contexts. Gelfand and Smith (1990), in a landmark paper, present many ways of applying the Gibbs sampler to a wide variety of Bayesian inference problems. Suppose our posterior distribution $[\boldsymbol{\theta}|\boldsymbol{y}]$ is k-dimensional, where $\boldsymbol{y}$ denotes the observed data. For any component θ_i of $\boldsymbol{\theta}$, the full conditional distribution is defined as $[\theta_i|\theta_1, ..., \theta_{i-1}, \theta_{i+1}, ..., \theta_k, \boldsymbol{y}] = [\theta_i|\boldsymbol{\theta}_{-i}, \boldsymbol{y}]$. Given this, Gibbs sampling is preformed

according to the following iterative scheme:

1. Set iteration counter t to 1 and choose a starting point for $\boldsymbol{\theta}$ in the parameter space; that is, set
$$\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, ..., \theta_k^{(0)})^T;$$

2. At iteration t, draw a new value $\boldsymbol{\theta}^{(t)}$ by successively generating each component of $\boldsymbol{\theta}$ from its corresponding full conditional distribution:
$$[\theta_i^{(t)} | \theta_1^{(t)}, ..., \theta_{i-1}^{(t)}, \theta_{i+1}^{(t-1)}, ..., \theta_k^{(t-1)}, \boldsymbol{y}];$$
A single 'sweep' of the Gibbs sampler consists of a cycle through all k components of $\boldsymbol{\theta}$.

3. Change iteration counter from t to $t+1$ and repeat steps 2 and 3, to produce successive values of the Markov chain.

Note that Gibbs sampling reduces the problem of simulating from a high dimensional distribution to the problem of simulating from a sequence of lower dimensional (usually scalar) distributions.

To illustrate the workings of the Gibbs sampler, we demonstrate using a simple case of two Bernoulli random variables X and Y (Casella and George, 1992). Suppose the joint probability function is

$$\begin{bmatrix} f_{x,y}(0,0) & f_{x,y}(1,0) \\ f_{x,y}(0,1) & f_{x,y}(1,1) \end{bmatrix} = \begin{bmatrix} p_1 & p_2 \\ p_3 & p_4 \end{bmatrix} \quad \text{with } p_1 + p_2 + p_3 + p_4 = 1.$$

Then the conditional distributions of $f(X \mid Y = y)$ and $f(Y \mid X = x)$ can be easily calculated and are summarized with two matrices

$$A_{y|x} = \begin{bmatrix} \frac{p_1}{p_1+p_3} & \frac{p_3}{p_1+p_3} \\ \frac{p_2}{p_2+p_4} & \frac{p_4}{p_2+p_4} \end{bmatrix}$$

and

$$A_{x|y} = \begin{bmatrix} \frac{p_1}{p_1+p_2} & \frac{p_2}{p_1+p_2} \\ \frac{p_3}{p_3+p_4} & \frac{p_4}{p_3+p_4} \end{bmatrix},$$

which, in the context of the Gibbs sampler, can be viewed as the transition matrices that give the probabilities of getting to y state from x and vice versa. Suppose we are interested in the marginal distribution of X which is given by

$$f_x = [f_x(0)\ f_x(1)] = [p_1 + p_3,\ p_2 + p_4]. \tag{2.33}$$

Instead of generating samples from f_x, the Gibbs sampler generates a sequence of random variables

$$Y_0',\ X_0',\ Y_1',\ X_1',\ ...,\ Y_k',\ X_k' \tag{2.34}$$

where $X_j' \sim f(x \mid Y_j' = y_j')$ and $Y_{j+1}' \sim f(y \mid X_j' = x_j')$. The transition matrix for the X' sequence in (2.34) is $A_{x|x} = A_{y|x}A_{x|y}$. Furthermore, if we denote the marginal distribution of X_k' as

$$f_k = [f_k(0),\ f_k(1)],$$

then for any k,

$$f_k = f_{k-1}A_{x|x} = f_0 A_{x|x}^k$$

where f_0 is the initial probability. Hoel *et al.* (1972) showed that as long as all the entries of $A_{x|x}$ are positive, f_k will converge to the unique stationary distribution f that satisfies

$$fA_{x|x} = f \tag{2.35}$$

regardless of f_0. The marginal distribution f_x defined as (2.33) satisfies (2.35) that is

$$f_x A_{x|x} = f_x A_{y|x} A_{x|y} = f_x.$$

Therefore, for large k, the distribution f_k, is approximately f_x, and thus we see, through this simple example, how the Gibbs sampler generates approximate, dependent realizations from a pre-specified target distribution f_x. A general proof that

the Gibbs sampler produces the required ergodic Markov chain under very general conditions can be found in Robert and Casella (2004).

Once a Gibbs sampler has been implemented, there are several ways to form a sample size M from the desired distribution (posterior distribution). The first approach is the independent sampling procedure (Gelfand and Smith, 1990) that generates M independent Gibbs sequences until convergence, say after k iterations, and uses the final value X'_k in each of the sequences to form the sample. This approach requires Mk generations, but provides a sample with independent values, provided the M chains are initialized independently. The second approach advocated by Geyer (1992) considers a long single chain. A sample of size M can be formed by M successive values from the chain after reaching convergence at iteration k. This generation method requires $k + M$ iterations; however, if the chain's autocorrelation is too high, it may take too long for a single chain to adequately cover the entire parameter space appropriately. That is, the effective sample size may be far less than M due to the high autocorrelation. A third approach takes every l^{th} iteration after the burn-in period (Raftery and Lewis, 1992), thereby thinning the chain and reducing autocorrelation between those values of the chain that are recorded. This approach requires $k + lM$ iterations. It reduces the autocorrelation between sampled values and is advantageous if computer storage of values is limited. Combinations of these approaches are also adapted and methods of assessing convergence will be discussed in detail in section 2.4.3. Examples 2 and 3 serve to further illustrate how the Gibbs sampler works, again, considering only simple settings for now.

Example 2 (Casella and George, 1992): Suppose the joint distribution of X and Y is given by

$$f(x, y) \propto \binom{n}{x} y^{x+\alpha-1}(1-y)^{n-x+\beta-1}, \ x = 0, 1, ..., n \text{ and } 0 \leq y \leq 1.$$

This is a non-standard bivariate distribution; however, the full conditional distributions are easily recovered as

$$f(x \mid y) \sim \text{Binomial}(n, y) \tag{2.36}$$

$$f(y \mid x) \sim \text{Beta}(x + \alpha, n - x + \beta). \tag{2.37}$$

If we are interested in the marginal distribution $f(x)$ of X, the Gibbs sequence (2.34) can be iteratively generated by simulations from (2.36) and (2.37). In this example, $M = 500$ parallel chains are produced and the 10^{th} value from each chain is used to form the sample (see Appendix A for the R code for this example). Figure 2.2 displays the histogram obtained from the Gibbs sampling output with $n = 16$, $\alpha = 2$ and $\beta = 4$. The solid line represents the density of the true marginal distribution of $f(x) = \int f(x, y)dy$, which can be shown to have a Beta-Binomial form

$$f(x) = \left(\begin{array}{c} n \\ x \end{array} \right) \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(x + \alpha)\Gamma(n - x + \beta)}{\Gamma(\alpha + \beta + n)} \qquad x = 0, 1, ..., n.$$

It is apparent that samples generated using the Gibbs sampler recover the properties of the true marginal distribution very well in this example.

Example 3 (Casella and George, 1992): Suppose the conditional distributions $X \mid Y$ and $Y \mid X$ are exponential distributions restricted to the interval $(0, B)$, that is

$$f_{X|Y}(x \mid y) \propto y\mathrm{e}^{-yx},\ 0 < x < B < \infty \tag{2.38}$$

and

$$f_{Y|X}(y \mid x) \propto x\mathrm{e}^{-xy},\ 0 < y < B < \infty. \tag{2.39}$$

Similar to Example 2, Gibbs sampling is applied based on (2.38) and (2.39) and $M = 500$ parallel chains are produced, and the 15^{th} value from each chain is used to form the sample (see Appendix B for the R code for this example). Figure 2.3 displays

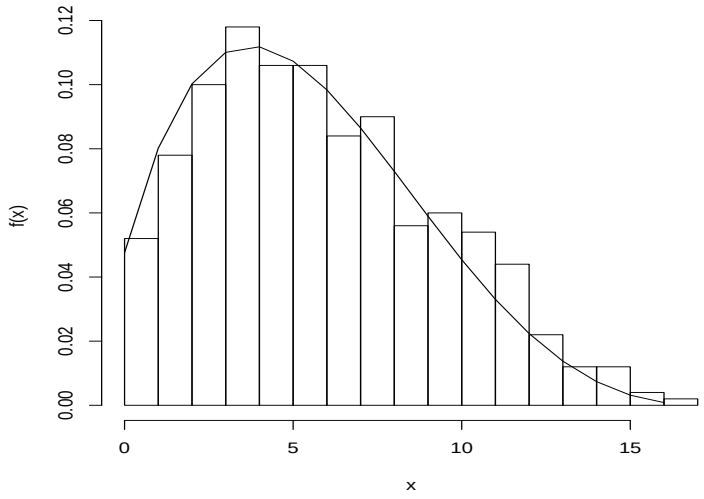

Figure 2.2: Histogram of Gibbs sampling for $f(x)$ in Example 2

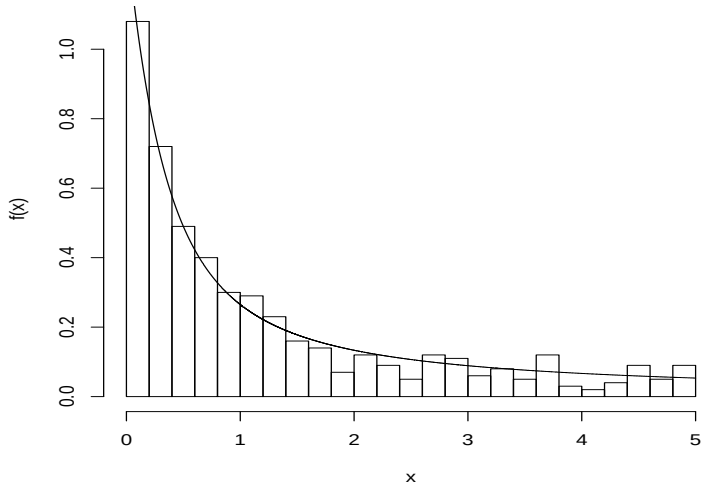

Figure 2.3: Histogram of Gibbs sampling for $f_X(x)$ in Example 3

histogram of the simulated X values with $B = 5$. The restriction that $B < \infty$ ensures that the marginal distribution $f(x)$ exists. When we employ the restriction $B < \infty$,

$$f_{X|Y}(x \mid y) = \frac{y\mathrm{e}^{-yx}}{1 - y\mathrm{e}^{-yB}} \qquad \text{and} \qquad f_{Y|X}(y \mid x) = \frac{x\mathrm{e}^{-xy}}{1 - x\mathrm{e}^{-xB}}.$$

Moreover,

$$f_X(x) \propto \frac{1 - x\mathrm{e}^{-xB}}{x} \tag{2.40}$$

is the true marginal distribution for X in this example. For the purpose of comparison, the density (2.40) after proper normalization is the solid line in Figure 2.3. Once again, samples generated using the Gibbs sampling recover the true density fairly well.

To employ Gibbs sampling for a given problem, it is clear that we must be able to simulate from the required low-dimensional full conditional distributions. If we recognize the full conditional density as some standard distribution, such as normal or gamma, we can simulate from it directly. Otherwise, we will need to employ techniques that enable sampling from arbitrary one (or low) dimensional distributions. There are many ways to complete this task, for example, through rejection sampling, inverse-probability sampling (based on the probability integral transform), the ratio-of-uniforms method just to name a few. In this thesis we will employ the Metropolis-Hastings algorithm for this task.

2.4.2.2 The Metropolis-Hastings Algorithm

The Metropolis-Hastings Algorithm is another MCMC algorithm that can be used to generate simulations from an arbitrary distribution. The objective here is to generate samples from a target distribution $\pi(x) = f(x)/K$ where K is the (possibly unknown) normalizing constant. Suppose $h(x)$ is a known density and for a known constant $c = \sup_x \frac{f(x)}{h(x)}$, $f(x) \leq ch(x)$ for all x. To obtain a random value from $\pi(.)$, a

classical simulation method named acceptance-rejection sampling can be performed as follows:

1. generate a candidate value z from $h(.)$;
2. calculate the ratio $r = f(z)/[ch(z)]$;
3. generate a value $u \sim \mathrm{U}(0, 1)$;
4. if $u \leq r$, return z; otherwise, goto step 1.

It is easily shown that the accepted value z comes from $\pi(.)$; however, finding a constant c that does the trick may be difficult in many applications and this method may result in an undesirably large number of rejections (Chib and Greenberg, 1995).

As in acceptance-rejection sampling, the Metropolis-Hastings is a rejection algorithm that allows us to generate samples from a non-standard distribution, $\pi(x)$, and requires only knowledge of the corresponding density up to a normalizing constant. The original idea is presented in papers written by Metropolis *et al.* (1953) and Hastings (1970). This algorithm generates a proposed value x^* from some candidate distribution, $q(x^*|x^{(t-1)})$, conditional on the previous value $x^{(t-1)}$, and either accepts or rejects this proposed value with a certain probability. More precisely, given a candidate distribution, the Metropolis-Hastings algorithm proceeds as follows:

1. Set iteration counter t to 1 and choose a initial value $x^{(0)}$ in the parameter space;
2. At iteration t, generate x^* from the candidate distribution $q(x^*|x^{(t-1)})$;
3. Compute the acceptance ratio r as:

$$r = \frac{\pi(x^*)q(x^{(t-1)}|x^*)}{\pi(x^{(t-1)})q(x^*|x^{(t-1)})};$$

4. Set
$$x^{(t)} = \begin{cases} x^* & \text{with probability } \min(1, r) \\ x^{(t-1)} & \text{with probability } 1 - \min(1, r) \end{cases}$$

5. Change iteration counter from t to $t + 1$;

6. Repeat steps 2-5 until convergence is reached.

Notice that we don't need to know the normalizing constant associated with the distribution function of $\pi(x)$, since the normalizing constant is canceled in the calculation of the acceptance ratio.

To implement the Metropolis-Hastings algorithm, it is necessary to specify a suitable candidate distribution (Chib and Greenberg, 1995). Typically, candidate distributions are selected from a family of distributions that require the specification of tuning parameters such as the location and scale. The first method given by Metropolis *et al.* (1953) produces a random walk chain. The candidate value $\boldsymbol{x^*}$ is drawn according to the process

$$\boldsymbol{x^*} = \boldsymbol{x} + \boldsymbol{z}$$

where $\boldsymbol{x}$ is the current value of the chain and $\boldsymbol{z}$ is called the increment random variable that is generated from a multivariate density $q_1(.)$. In the single variable case, the most commonly used candidate distribution is a normal distribution (Gamerman and Lopes, 2006) centered at the current value, $x^* \sim \mathrm{N}(x, \sigma^2)$. Here, σ^2 is a prechosen constant, commonly referred to as a tuning parameter, that is chosen so that the algorithm performs adequately. Note that when the candidate density $q_1(.)$ is symmetric, that is $q_1(\boldsymbol{z}) = q_1(-\boldsymbol{z})$, the acceptance ratio can be reduced to

$$\boldsymbol{r} = \min\Big\{\frac{\pi(\boldsymbol{x^*})}{\pi(\boldsymbol{x})}, 1\Big\}.$$

The second method suggested by Tierney (1994) is represented by a vector autoregressive process of order 1. In this case,

$$\boldsymbol{x}^* = \boldsymbol{a} + B(\boldsymbol{x} - \boldsymbol{a}) + \boldsymbol{z}$$

where $\boldsymbol{z}$ is generated from some distribution $q_2(.)$. The vector $\boldsymbol{a}$ and matrix B are both conformable with $\boldsymbol{x}$. Setting $B = -I$ where I is the identity matrix produces an autoregressive chain that has values reflected about the point $\boldsymbol{a}$ and induces negative correlation between successive elements of the chain. There are also other possible methods to choose candidate distributions such as using independent candidate functions (Hastings, 1970), exploiting the form of $\pi(.)$ (Chib and Greenberg, 1994) and using the acceptance-rejection sampling method with a pseudodominating density (Tierney, 1994). To illustrate the Metropolis-Hastings algorithm, we present an example based on simulating the bivariate normal distribution and consider three different candidate distributions.

Example 4 (Chib and Greenberg, 1995): Consider the bivariate normal distribution $\mathrm{N}_2(\boldsymbol{\mu}, \Sigma)$ where $\boldsymbol{\mu} = (1, 2)^T$ is the mean vector and

$$\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$$

is the 2×2 covariance matrix. The density function can be written as

$$\pi(\boldsymbol{x}) = \frac{1}{2\pi|\Sigma|^{-1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\}.$$

Therefore, the acceptance ratio for a symmetric candidate distribution is

$$r = \min\Big\{ \frac{\exp\{-\frac{1}{2}(\boldsymbol{x}^* - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}^* - \boldsymbol{\mu})\}}{\exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\}},\ 1 \Big\}.$$

Three candidate distributions are proposed. The first one is a random walk generating density

$$\boldsymbol{x}^* = \boldsymbol{x} + \boldsymbol{z}, \tag{2.41}$$

where the i^{th} component of $\boldsymbol{z}$ is uniformly distributed on the interval $(-\delta_i, \delta_i)$ for $i = 1,\ 2$. We set $\delta_1 = 0.75$ and $\delta_2 = 1$. The second candidate distribution is also a random walk generating density (2.41) but $\boldsymbol{z}$ is distributed as $\mathrm{N}_2(0, D)$ where

$$D = \begin{pmatrix} 0.6 & 0 \\ 0 & 0.4 \end{pmatrix}.$$

Therefore, each component of $\boldsymbol{z}$ can be easily generated independently from a normal distribution. The third candidate distribution is an autoregressive generating density

$$\boldsymbol{x}^* = \boldsymbol{\mu} - (\boldsymbol{x} - \boldsymbol{\mu}) + \boldsymbol{z},$$

where components of $\boldsymbol{z}$ are distributed independently as Unif$(-1, 1)$. To illustrate the characteristics of the output, Figure 2.4 (panels (b), (c) and (d)) are the scatter plots of 6000 simulated values using the three candidate densities (see Appendix C for R code). The top left panel of Figure 2.4 (panel (a)) is the scatter plot of 4000 values simulated directly from the desired bivariate normal distribution. It is clear that the Metropolis-Hastings algorithm based on either of the three candidate distributions reproduces the shape of target bivariate distribution very well in this simple example.

Choosing the tuning parameters which represent the spread and scale of the candidate distribution is an important matter. Choice of tuning parameter will effect the acceptance rate of the chain. Consider the simple situation where the candidate distribution is a Normal distribution centered at the current value that has variance σ^2 which is the tuning parameter. A large value for the variance will allow the candidate to propose moves that are distant from the current value, but it is at the likely cost of having a very small acceptance rate. On the other hand, a small value for the variance allows only a close move around the current value, which may give a high acceptance rate but at the expense of higher chain autocorrelation. In this case, the

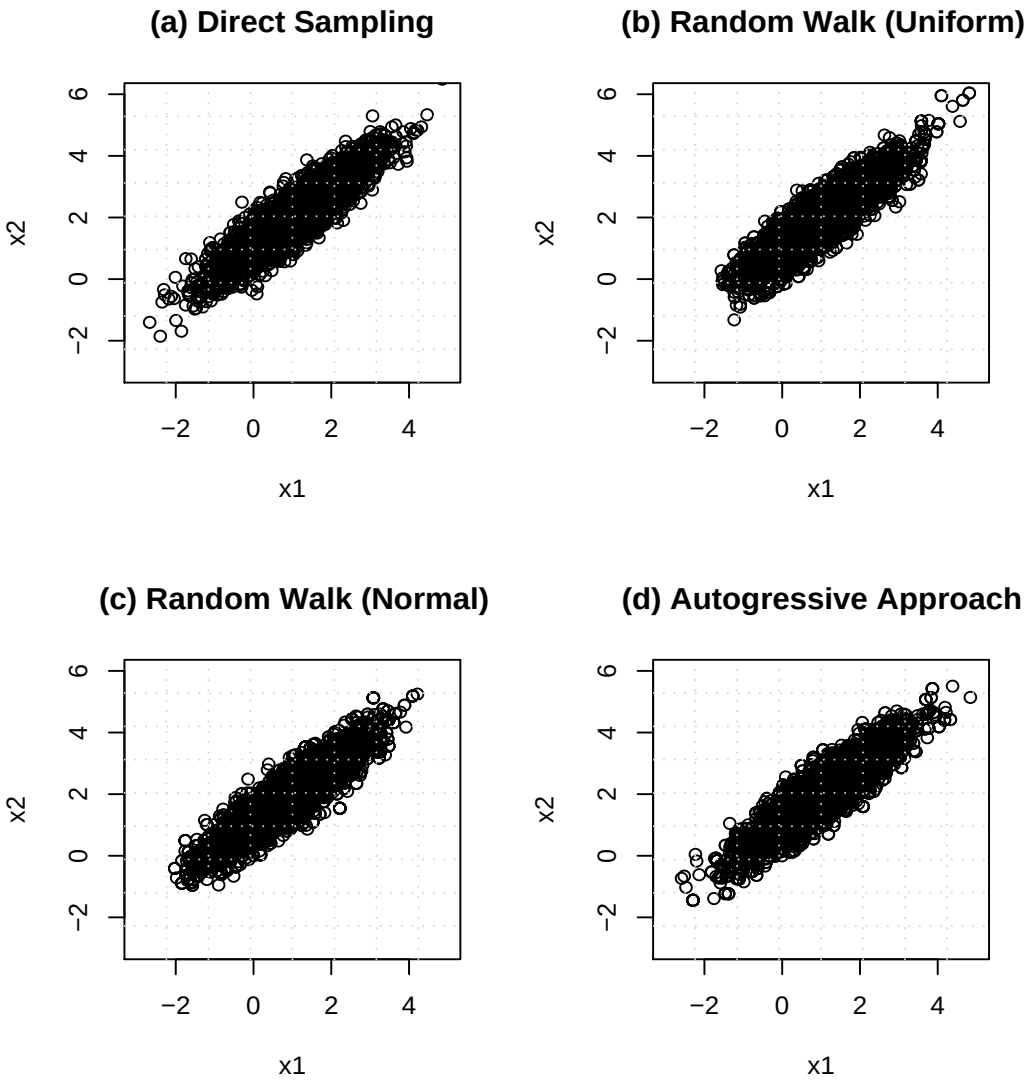

Figure 2.4: Scatter Plots of Simulated Draws for Example 4

chain will take longer to traverse the support of the density. Autocorrelations across sample values are likely to be high in both situations. Müller (1993) recommended that the acceptance rate for a random walk chain should be around 0.5. The work by Roberts, Gelman and Gilks (1994) suggested that if the candidate distributions are normal, the ideal acceptance rate is approximately 0.45 in one-dimensional case and approximately 0.23 as the number of dimensions approaches infinity. In general, the ideal candidate distribution will yield an acceptance rate between 0.2 to 0.5 (Besag *et al.*, 1995). In practice, we can run the algorithm for several iterations in a preliminary tuning phase in order to achieve a reasonable acceptance rate. If the acceptance rate is too low, we decrease the value for the variance of the candidate distribution. If the acceptance rate is too high, we increase the value for the variance of the candidate distribution. Once a reasonable acceptance rate is achieved, the tuning parameter and hence the candidate distribution is held fixed to produce successive realizations from the Markov chain.

Our overall strategy for Markov chain simulation will be based on a combination of the Gibbs sampler and the Metropolis-Hastings algorithm. Working within a Gibbs sampling framework, we will successively draw realizations from full conditional distributions. If, at a given step in the algorithm, the corresponding full conditional distribution is of standard form, this will be a simple task, which we shall refer to as a 'Gibbs step'. If, on the other hand, the full conditional distribution is not of standard form we could apply the the Metropolis-Hastings algorithm to obtain the required draw. One obvious disadvantage of this approach is that it requires simulating a Markov chain *within* another Markov chain; moreover, this would be required at every sweep of the Gibbs sampler which seems computationally too intensive for practical application. Fortunately, this is not required. Upon encountering a non-standard full conditional distribution within a Gibbs sampler, applying only a single iteration of the Metropolis-Hastings algorithm (termed a 'Metropolis-Hastings step'),

and subsequently moving to the next component of the Gibbs sampler is sufficient for producing an ergodic Markov chain with the required stationary distribution (see for example, Robert and Casella (2004) for detailed discussions and technical results).

2.4.3 Diagnosing Convergence

MCMC algorithms provide a way to generate realizations from a distribution without knowing the normalizing constant of that distribution. The next question that arises is how one can assess convergence of the MCMC algorithm. That is, we need to decipher at which iteration the MCMC sequence has reached approximate stationarity. Further to this we need to consider the number of subsequent iterations required so that the chain will exhibit all the features of the target distribution. The theoretical foundations of MCMC algorithms show that the chains constructed by MCMC algorithms are ergodic under fairly general conditions. Therefore, the ergodic theorem guarantees the estimation consistency of various aspects of the target distribution. For practical implementation, we must of course settle for a finite Monte Carlo sample size. Thus we must decide a suitable number of post burn-in iterations at which point it safe to stop these algorithms. Monte Carlo sample size calculations based on a prespecified error bounds are very difficult or impossible to perform in most practical situations. Thus, empirical methods based on MCMC output $\boldsymbol{\theta}^1, ..., \boldsymbol{\theta}^t, ...$ are typically used for convergence assessment, to determine when the chain has reached approximate stationarity and has exhibited all the features of the posterior distribution.

For typical applications, two different types of convergence need to be assessed. The first type is convergence to the stationary distribution. This can be done through a trace plot (time series plot) of the sampled values, which can indicate when the Markov chain has forgotten its initial state and has begun to exhibit its steady state behavior. For most MCMC algorithms, convergence to the stationary distribution

is not the major issue. Instead, the speed of exploration of the target distribution and the degree of correlation between sampled values within the chain are most important. Therefore, it is crucial to examine the autocorrelation plot of the chain. A useful technique is to monitor convergence of ergodic averages, $\frac{1}{T}\sum_{t=1}^{T} h(\boldsymbol{\theta}^t)$, to their asymptotic values for a function $h(.)$ such that $\mathrm{E}_\pi[h(\boldsymbol{\theta})] < \infty$. A plot of ergodic averages after each iteration can be used to check convergence. When $\boldsymbol{\theta}$ is high dimensional, it will not be possible to examine trace plots, ergodic average plots and autocorrelation plots for each component of $\boldsymbol{\theta}$. In this case, a representative subset of model parameters, in addition to certain functions of model parameters is monitored. One useful summary of all model parameters is the logarithm of the posterior distribution at each state of the chain (up to an additive constant not depending on $\boldsymbol{\theta}$) which can always be examined as an overall summary of the chain.

Instead of diagnosing convergence based on a single chain, multiple chains initialized from different starting values are often used. By simulating several independent parallel chains, the variability and dependence on initial values are reduced and convergence is easier to assess by plotting multiple chains onto the same axis; however, there are dangers in a naive implementation of the multiple chains principle. The slowest chain will govern convergence and it is extremely important that the different chains are initialized at points that are well dispersed over the parameter space.

Aside from the examination of trace plots, the Gelman-Rubin diagnostic statistic (Gelman and Rubin, 1992) is another useful tool for deciding how long the Markov chain should run. In this context, we begin by running $2N$ iterations of M parallel chains each initialized at dispersed points in the target distribution. After discarding the first N iterations, we compute the between and within sequence variances, denoted

as B and W respectively. For the parameter of interest ϕ,

$$B = \frac{N}{M-1}\sum_{j=1}^{M}(\bar{\phi}_{.j} - \bar{\phi}_{..})^2$$

and

$$W = \frac{1}{M}\sum_{j=1}^{M} S_j^2 \text{ where } S_j^2 = \frac{1}{N-1}\sum_{i=1}^{N}(\phi_{ij} - \bar{\phi}_{.j})^2.$$

In these formulas, ϕ_{ij} denotes the i^{th} value from j^{th} chain, $\bar{\phi}_{.j} = \frac{1}{N}\sum_{i=1}^{N}\phi_{ij}$ and $\bar{\phi}_{..} = \frac{1}{M}\sum_{j=1}^{M}\bar{\phi}_{.j}$. The marginal posterior variance $Var[\phi \mid y]$, where y is the data, can be estimated using a weighted average of B and W:

$$\widehat{Var}^{+}(\phi \mid y) = \frac{N-1}{N}W + \frac{1}{N}B.$$

This estimator is biased; however, it is asymptotically unbiased under stationarity. For finite N, $\hat{Var}^{+}(\phi \mid y)$ overestimates $Var[\phi \mid y]$ while W underestimating $Var[\phi \mid y]$ since the individual sequences have not had time to explore the target. Therefore, the Gelman and Rubin diagnostic statistic

$$\hat{R} = \left[\frac{\hat{Var}^{+}(\phi \mid y)}{W}\right]^{1/2}$$

is always greater than 1 and declines to 1 as $N \to \infty$. Further simulations are required when $\hat{R}$ is high. It is typically recommended that simulations continue until $\hat{R} < 1.1$. Note that this diagnostic is univariate, and hence must be applied to each component of the parameter vector separately and monitored for all parameters of interest.

It is generally safe to conclude convergence if the autocorrelation function drops quickly, the trace plot and ergodic plot indicate stability of Monte Carlo estimates and the Gelman and Rubin diagnostic statistics close to 1 for all parameters of interest. If this is not the case, running a longer chain may solve the problem but techniques based on alternations to the model or algorithm can be more efficient.

Reparameterizations such as centering covariates and hierarchical centering can improve the algorithm (Gamerman and Lopes, 2006). When high correlations exist between components, blocking techniques that update groups of parameters (based on their joint full conditional distribution) can be very beneficial in improving computational performance (in the sense of lowering chain autocorrelations). In this case, slow componentwise moves are replaced by moves dictated by the joint full conditional distribution for the block of parameters considered (Liu *et al.*, 1994). Carefully considering the identifiability of parameters in the model, and seeking a better proposal distributions can also be helpful. Having discussed methods for posterior computation, we complete the literature review by discussing Bayesian methods for model selection and goodness-of-fit. We focus on methods that are applicable to hierarchical models and that are easily implemented via MCMC.

2.4.4 Bayesian Model Selection using the Deviance Information Criterion

For a given problem in data analysis, there will usually be several models under consideration. In general, a larger model has more flexibility and therefore has the advantage of fitting the data better; however, these large models become more difficult to compute and interpret. Choosing between competing, perhaps non-nested models is thus an important issue in any data analysis.

The standard Bayesian approach to model selection arises through the Bayes factor (BF) which, given a prior over a set of competing models, is obtained as the ratio of the posterior and prior model odds. The BF can also be written as a ratio of the marginal likelihoods of observed data $\boldsymbol{y}$ under each model defined as

$$BF = \frac{p_1(\boldsymbol{y})}{p_2(\boldsymbol{y})} = \frac{\int L_1(\boldsymbol{y} \mid \boldsymbol{\theta_1})\pi_1(\boldsymbol{\theta_1})d\boldsymbol{\theta_1}}{\int L_2(\boldsymbol{y} \mid \boldsymbol{\theta_2})\pi_2(\boldsymbol{\theta_2})d\boldsymbol{\theta_2}},$$

where $L_i(\boldsymbol{y} \mid \boldsymbol{\theta_i})$ is the likelihood function under model i and $\pi_i(\boldsymbol{\theta_i})$ is the prior specification assigned to model i, $i = 1, 2$. the Bayes factor has a nice interpretation

in terms of posterior model probabilities; however, the calculation of the marginal likelihoods is difficult for complex models. The Akaike Information Criterion (AIC) is another technique for model comparison taking the form

$$AIC = -2l_{M_i}(\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}) + 2p$$

and the Bayesian Information Criterion (BIC) is yet another criteria having the form

$$BIC = -2l_{M_i}(\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}) + \log(n)p.$$

Both are easily computable alternatives to the BF. Here, $l_{M_i}(\hat{\boldsymbol{\theta}}_{\boldsymbol{i}})$ is the log-likelihood for model M_i, $\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}$ is the MLE of $\boldsymbol{\theta}$ under model M_i, n is the number of observations and p is the number of parameters. Models with lower AIC or BIC values are preferred. The BIC is particularly relevant for Bayesian model selection as it can be asymptotically related to posterior model probabilities derived under a uniform prior over the model space. Both the AIC and BIC impose penalties for model complexity based on the number of model parameters p. Unfortunately, these methods are not appropriate for hierarchical spatial model where parameters include correlated random effects. In this context, correlation between parameters becomes an issue and hence the *effective* number of parameters is not entirely clear. As a result, penalizing model complexity becomes a more subtle issue.

To address this concern, the deviance information criterion (DIC) proposed by Spiegelhalter et al. (2002) is an extension of the AIC approach that may be applied for choosing between hierarchical spatial models. The criteria is based on the deviance statistic which is defined as

$$D(\boldsymbol{\theta}) = -2\log L(\boldsymbol{y} \mid \boldsymbol{\theta}),$$

where $L(\boldsymbol{y} \mid \boldsymbol{\theta})$ is the likelihood function of the data given parameters under the model. Then the DIC is defined as

$$DIC = \overline{D(\boldsymbol{\theta})} + p_D,$$

where $\overline{D(\boldsymbol{\theta})} = \mathrm{E}[D(\boldsymbol{\theta}) \mid \boldsymbol{y}]$ is the posterior mean of the deviance, a measure of fit with lower values indicating superior fit to the data. The quantity p_D is a penalty term that measures the complexity of the model and is defined by

$$p_D = \overline{D(\boldsymbol{\theta})} - D(\bar{\boldsymbol{\theta}}),$$

where $D(\bar{\boldsymbol{\theta}})$ is the deviance evaluated at the posterior mean of $\boldsymbol{\theta}$. Rather than simply penalizing the model depending on the total 'raw' number of parameters appearing in the model, the p_D penalty accounts for spatial correlation or shrinkage among correlated parameters and can be interpreted as an estimate of the effective number of model parameters. Spiegelhalter et al. (2002) present a detailed justification for the definition of p_D and illustrate its use with several examples. There, it is also shown that in the special case of non-hierarchical generalized linear models (models without random effects), p_D is approximately equal to the raw parameter count p and that this approximation is exact in the case of the normal linear model. With models incorporating correlated parameters, p_D will be smaller than the raw number of parameters. Overall, just as with the AIC and BIC, the model with lower DIC score is preferred since it reaches the best combination of fit and parsimony.

2.4.5 Goodness-of-fit for Bayesian Models using Posterior Predictive Model Checking

A fundamental aspect of any model based data analysis involves checking the fit of the proposed model to the data. Conclusions drawn from any analysis are of course conditional on such checks of model adequacy. As with model fitting, the Bayesian approach to goodness-of-fit relies heavily on simulation based methods. We focus here on methods based on the posterior predictive distribution. The posterior predictive checking technique (Gelman et al., 2004) is based on an examination of the fit of a model to the observed data by drawing replicated data values from the posterior

predictive distribution defined as

$$p(\boldsymbol{y}^{rep} \mid \boldsymbol{y}) = \int p(\boldsymbol{y}^{rep} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \boldsymbol{y}) d\boldsymbol{\theta},$$

where $\boldsymbol{y}$ is a vector of the observed data, $\boldsymbol{\theta}$ is a vector of parameters and $\boldsymbol{y}^{rep}$ represents a hypothetical replicate data set, which is assumed to be drawn under the same conditions as the observed data, and also assumed to be conditionally independent of $\boldsymbol{y}$ given $\boldsymbol{\theta}$.

In practice, the posterior predictive distribution is computed using simulation. Having obtained samples from the posterior distribution for a given model, simulation from the posterior predictive distribution is straightforward using one-for-one composition sampling. The replicated data, drawn from the predictive distribution induced by the model, should look similar to the observed data when the model fits. Any obvious departures from the observed data indicate potential failings of the model (in a predictive sense). To quantify the lack of fit, the posterior predictive p-value measures the probability that the replicated data could be more extreme than the observed data (where the probability is computed W.R.T the posterior predictive distribution). To illustrate further, let $T(.)$ be a checking function (a statistic) used to summarize some aspect of the data. This is usually chosen to be a specific feature of the data, that is of interest. The posterior predictive p-value is calculated as:

$$p_B = \Pr(T(y^{rep}) \geq T(\boldsymbol{y}) \mid \boldsymbol{y}).$$

Using carefully constructed checking functions, one can search for specific discrepancies between observed and simulated data.

Chapter 3

Exploratory Data Analysis

This chapter examines the British Columbia fire frequency data through an exploratory analysis. Mountain pine beetle is an insect that attacks and kills mature or weakened pine trees. It is thought that infection *may* increase the number of forest fires. Moreover, a large number of mountain pine beetle outbreaks have been observed recently in the west-central interior of British Columbia and a huge proportion of pine trees will be killed by mountain pine beetle in next 5 years at the current rate of spread. As a result, understanding the association between fire frequency and the severity of mountain pine beetle outbreaks has important implications for fire management strategies.

As mentioned in the introduction, our data is collected over the province which has been subdivided into $I = 1712$ homogeneous subregions of equal area, where each subregion has an area of $(25\text{km})^2$. We are thus dealing with areal spatial data. Our data set contains aggregated fire counts from a 44 year study, and we let N_i denote the total number of fires occurring in region i, during the entire study period for $i = 1, ..., I$. For simplicity, the data are time-aggregated so that we focus on a purely spatial, rather than a spatiotemporal analysis. Note, with this time aggregation, we must take great care in not over-interpreting the results of our analysis. Certainly, causation can not be inferred from this analysis of an observational study; nevertheless, examination of associations is an important first step in understanding the complex ecological processes governing the spatial distribution of wildfire.

In addition to the area affected by mountain pine beetle outbreaks, several other regional specific covariates, including area of forest cover, area of pine leading stands,

the number of roadways, and a drought climate code are also available. Figure 1.3, 1.4 and 1.5 provide an overview of the spatial distribution underlying the data set. It is clear from these figures that the data exhibit spatial variability. For example, high values of fire counts are clustered in the southern part of British Columbia, while high values of area affected by MPB are clustered in the central part of British Columbia. A primary question arising in an exploratory analysis of this data relates to the existence of residual spatial variability. That is, do the available covariates, which are themselves spatially-varying, explain all of the spatial variability in the response? If this is the case, then a standard generalized linear model may be used for regression analysis. If this is not the case, then we must accommodate the residual spatial variability in order to make valid inference on regression coefficients.

Note that spatial information for this data set is contained in an adjacency matrix W, which indicates the neighbourhood structure of each cell in the following way:

$$W_{ij} = \begin{cases} 0 & \text{if } i = j \\ 0 & \text{if } i \text{ and } j \text{ are not neighbours} \\ 1 & \text{if } i \text{ and } j \text{ are neighbours,} \end{cases}$$

where neighbours are defined as two regions whose centroids are separated by no more than 25km. This distance for defining neighbours was based on consultation with researchers at the Pacific Forestry Service, who felt that this was an appropriate spatial scale for analysis. Our primary goal is to develop a statistical model to assess the association between covariates and the mean fire frequency, while simultaneously accommodating spatial dependence. It is also of interest to examine and characterize any residual spatial structure, as this residual structure may give clues on covariates that are missing from the model.

As an initial step, a standard log-linear Poisson regression model is fit to the total

fire counts based on the five covariates mentioned above (using the *glm* function in **R**). The model can be specified as:

$$\begin{aligned} N_i &\overset{ind}{\sim} \text{Poisson}(\lambda_i) \qquad i = 1, ..., I, \\ \log(\lambda_i) &= \boldsymbol{\beta}^T \boldsymbol{x_i}, \end{aligned} \tag{3.1}$$

where N_i is the total number of fires occurring in region i over the study period, $\boldsymbol{\beta}$ is the vector of regression coefficients and $\boldsymbol{x_i}$ is a vector of covariates for region i. Table 3.1 gives the maximum likelihood estimators of the regression coefficients, $\hat{\boldsymbol{\beta}}$,which may provide a general idea of the relationship between these covariates and the mean fire count. In particular, the model fit and corresponding confidence interval indicate a positive association between mean fire frequency and area infected by mountain pine beetle. Note, however, we must be careful in interpreting this result as the model assumptions have not been verified.

Estimation of Parameters in Standard Log-Linear Poisson Model

	Estimate	Std. Error	95% CI
β_0 (Intercept)	3.6163	0.0045	(3.6075, 3.6250)
β_1 (MPB)	0.2490	0.0037	(0.2417, 0.2562)
β_2 (Drought Code)	0.4439	0.0028	(0.4384, 0.4495)
β_3 (forest)	0.7062	0.0045	(0.6974, 0.7150)
β_4 (pine)	-0.4925	0.0045	$(-0.5014, -0.4836)$
β_5 (road)	0.4016	0.0021	(0.3975, 0.4057)

Table 3.1: MLE of $\boldsymbol{\beta}$ from *glm* Function in *R*

The deviance residual (Dobson, 1990) measures the individual observation's contribution to the lack of fit of the model. Under the model defined in (3.1), the deviance

residual takes the following form:

$$r_i^D = \text{sign}(N_i - \hat{\lambda}_i)\sqrt{d_i^2},$$

where $d_i = 2\Big(\big(N_i\log(N_i) - N_i\big) - \big(N_i\log(\hat{\lambda}_i) - \hat{\lambda}_i\big)\Big)$, $\hat{\lambda}_i = \exp(\hat{\boldsymbol{\beta}}^T \boldsymbol{x_i})$ and the sign() function is defined as

$$\text{sign}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0. \end{cases}$$

If the proposed model is correct, the deviance residuals should be independently distributed and asymptotically normal. Figure 3.1 presents a map of the deviance residuals associated with each fire count. From the map, it is clear that the deviance residuals are not randomly distributed as they should be under a standard Poisson log-linear regression model. High values are clustered in southeastern part of British Columbia, which gives evidence that the data exhibit residual spatial variation and the independence assumption underlying the standard Poisson regression model has not been satisfied.

To further examine the spatial dependence among residuals, we consider the Moran's I statistic (Banerjee *et al.*, 2004) which can be used for exploratory analysis in measuring strength of spatial correlation among data observed over areal units. Moran's I takes the form

$$I = \frac{n \sum_i \sum_j W_{ij}(Y_i - \bar{\boldsymbol{Y}})(Y_j - \bar{\boldsymbol{Y}})}{(\sum_{i \neq j} W_{ij}) \sum_i (Y_i - \bar{\boldsymbol{Y}})^2}$$

where $\bar{\boldsymbol{Y}}$ is the mean of the vector of observations $\boldsymbol{Y}$, n is number of observations and W_{ij} is the element from the adjacency matrix. Moran's I is asymptotically normally distributed with mean $I_0 = -\frac{1}{n-1}$ under the null model where the Y_i's are not spatially

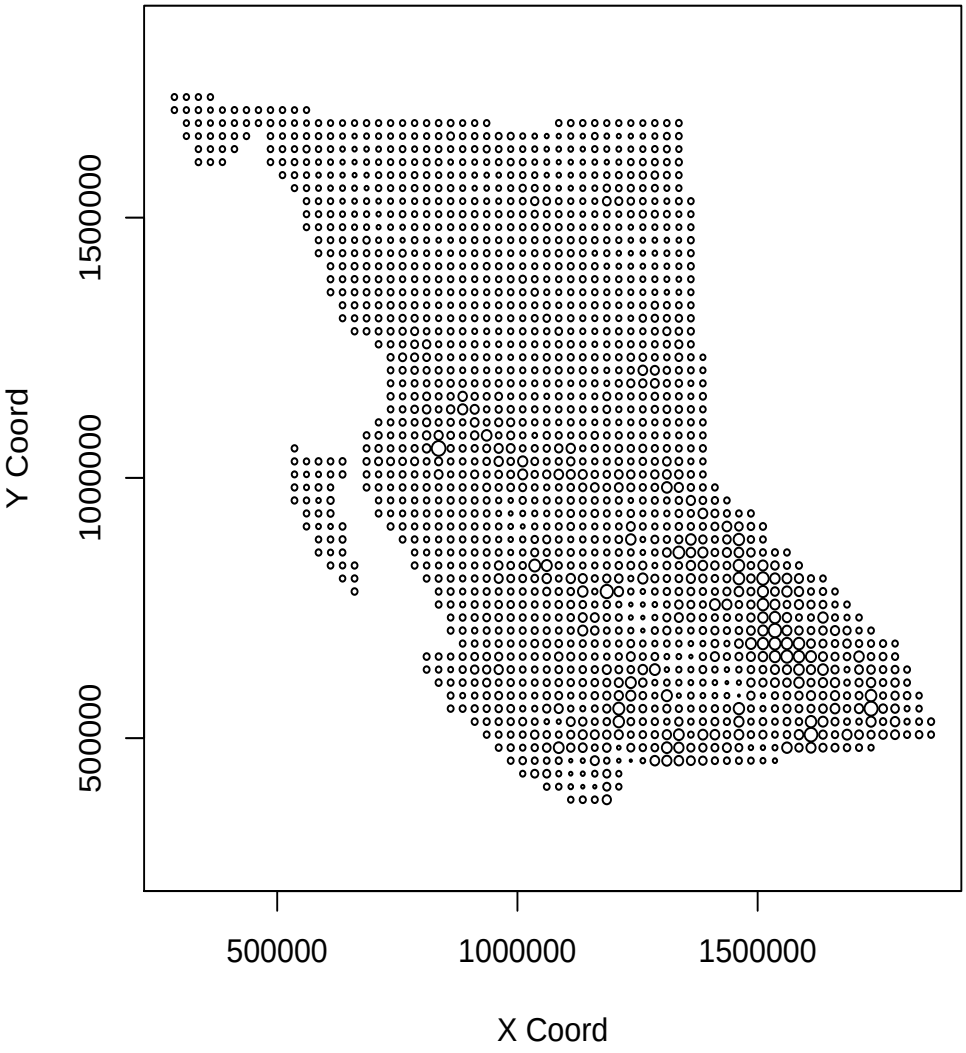

Figure 3.1: Deviance Residuals under the standard Poisson log-linear regression model (larger circle indicates higher value)

correlated. The variance can be estimated by

$$\mathrm{Var}(I) = \frac{n^2(n-1)S_1 - n(n-1)S_2 - 2S_0^2}{(n+1)(n-1)^2 S_0^2}$$

where $S_0 = \sum_{i \neq j} W_{ij}$, $S_1 = \frac{1}{2}\sum_{i \neq j}(W_{ij} + W_{ji})^2$ and $S_2 = \sum_k (\sum_j W_{kj} + \sum_i W_{ik})^2$. The test statistic based on I is then

$$Z = \frac{I - I_0}{\sqrt{\mathrm{Var}(I)}}$$

which is asymptotically $\mathrm{N}(0,1)$ when the null hypothesis of independence is true and a P-value can be obtained. For the deviance residuals as mapped in Figure 3.1, Moran's I is 0.5392 with estimated standard error 0.0124. The corresponding test statistic is 43.6428 with P-value closes to 0 which suggests very strong evidence against the null hypothesis of no spatial correlation in the data. Of course, with a large data set such as this, we must be careful in interpreting p-values; however, this in conjunction with Figure 3.1 seems to indicate significant residual spatial structure in the data. Thus the presence of spatial correlation in our data is not ignorable, even after taking covariates into consideration.

In order to check for extra-Poisson variability, we first compute the ratio of the mean and variance of our total fire counts. This yields a value of 0.0047, certainly not close to 1 which it should be in the absence of over-dispersion. To examine this further, we fit the fire count data using the *glm* function in **R** under a Negative Binomial specification. The glm software in **R** uses a slightly different parameterization than that presented in Section 2.2. In particular, the dispersion parameter is defined as k which is equal to $\frac{1}{a}$ in the earlier description of the negative binomial model. In this case, $\mathrm{E}(N_i) = \lambda_i$ and $\mathrm{Var}(N_i) = \lambda_i + \lambda_i^2/k$, so that a larger value of k corresponds to less extra-Poisson variability and $k \to \infty$ corresponds to the Poisson model. The estimated dispersion parameter for our fire frequency data, $\hat{k}$, is 0.8045 (s.e. $= 0.0279$)

which seems to indicate extra-Poisson variability. This extra-Poisson variation may arise from two separate sources. First, spatially structured variation as indicated in Figure 3.1 and second, unstructured variation representing deviation from the Poisson model (3.1), but not related to the spatial structure of the data. In the next chapter, we develop a hierarchical model specification that can accommodate and quantify both sources of residual variation.

Chapter 4

Overdispersed Spatial Count Model

4.1 Model Specification

In order to address the over-dispersion and residual spatial variation apparent in the exploratory analysis, we propose a mixed negative binomial regression model. At the first level, our Bayesian hierarchical model is specified as

$$N_i \mid \lambda_i, a \overset{ind}{\sim} \text{Negbin}(\lambda_i, a),\ i = 1, ..., I, \tag{4.1}$$

where N_i is the total fire count in region i, λ_i describes the mean for region i, and $a \geq 0$ is the dispersion parameter. The likelihood function based on this first level model specification can be written as

$$L_{NB}(\boldsymbol{N} \mid \boldsymbol{\lambda}, a) = \prod_{i=1}^{I} \frac{\Gamma(N_i + \frac{1}{a})}{N_i! \Gamma(1/a)} (\frac{\lambda_i a}{\lambda_i a + 1})^{N_i} (\frac{1}{\lambda_i a + 1})^{1/a}. \tag{4.2}$$

We adopt a log-linear mixed regression specification for the mean

$$\log(\lambda_i) = \boldsymbol{\beta}^T \boldsymbol{x_i} + b_i \tag{4.3}$$

that links the mean λ_i to regional covariates $\boldsymbol{x_i}$ and in addition incorporates a spatial random effect b_i to model the residual spatial variability indicated in Figure 3.1. At the second level of the model, we assign a spatially correlated prior distribution to the random effects. In particular, we adopt the conditional autoregressive model

$$\mathbf{b} \mid \sigma^2_{\mathbf{b}} \sim CAR(\sigma^2_{\mathbf{b}})$$

with the variance component $\sigma^2_{\mathbf{b}}$ ($\tau = \frac{1}{\sigma^2_{\mathbf{b}}}$ is the precision parameter) quantifying residual spatial variability. Note that this is in contrast to the dispersion parameter

a which quantifies the degree of spatially unstructured residual variability. In both cases the notion of residual variation can be linked to covariates that are missing from the model. Having both parameters in the model allows us to tease out the various forms of residual variability, as arising from spatial as well as inherently non-spatial factors. The model specification is completed by assigning prior distributions to a, $\sigma^2_{\mathbf{b}}$ and $\boldsymbol{\beta}$. We will adopt, vague, weakly-informative priors for these parameters and the specific prior forms are discussed below.

For the purpose of implementation, working with the negative binomial likelihood (4.2) can be computationally inconvenient. As a result, we shall make use of the well known representation of the negative binomial distribution as a Poisson-Gamma mixture

$$N_i \mid \nu_i, \lambda_i \overset{ind}{\sim} \text{Poisson}(\nu_i \lambda_i) \tag{4.4}$$

with

$$\nu_i \mid a \overset{i.i.d}{\sim} \text{Gamma}(\frac{1}{a}, a). \tag{4.5}$$

This representation will allow us to work with the much simpler Poisson likelihood, in conjunction with a latent variable framework implemented through MCMC.

Theorem: Let $N_i \mid \lambda_i, a \overset{ind}{\sim} \text{Negbin}(\lambda_i, a)$ with p.m.f

$$f(N_i \mid \lambda_i, a) = \frac{\Gamma(N_i + \frac{1}{a})}{N_i! \Gamma(1/a)} (\frac{\lambda_i a}{\lambda_i a + 1})^{N_i} (\frac{1}{\lambda_i a + 1})^{1/a}, \tag{4.6}$$

then it can be derived as a mixture of $N_i \mid \nu_i, \lambda_i \overset{ind}{\sim} \text{Poisson}(\nu_i \lambda_i)$ and $\nu_i \mid a \overset{i.i.d}{\sim} \text{Gamma}(\frac{1}{a}, a)$.

Proof: Let $f(N_i \mid \nu_i, \lambda_i)$ and $f(\nu_i \mid a)$ denote the p.m.f. of (4.4) and p.d.f. of (4.5) respectively.

Marginalizing $f(N_i, \nu_i \mid \lambda_i, a)$ over ν_i yields

$$\begin{aligned}
f(N_i \mid \lambda_i, a) &= \int_0^\infty f(N_i \mid \nu_i, \lambda_i) \times f(\nu_i \mid a) d\nu_i \\
&= \int_0^\infty \tfrac{(\nu_i\lambda_i)^{N_i}\exp(-\nu_i\lambda_i)}{N_i!} \times \tfrac{\nu_i^{\frac{1}{a}-1}\exp(-\nu_i/a)}{\Gamma(1/a)a^{1/a}} d\nu_i \\
&= \tfrac{\lambda_i^{N_i}}{N_i!\Gamma(1/a)a^{1/a}} \int_0^\infty \nu_i^{N_i}\exp(-\nu_i\lambda_i)\nu_i^{\frac{1}{a}-1}\exp(-\nu_i/a) d\nu_i \\
&= \tfrac{\lambda_i^{N_i}}{N_i!\Gamma(1/a)a^{1/a}} \int_0^\infty \nu_i^{N_i+\frac{1}{a}-1}\exp(-\nu_i(\lambda_i + \tfrac{1}{a})) d\nu_i \\
&= \tfrac{\lambda_i^{N_i}\Gamma(N_i+\frac{1}{a})}{N_i!\Gamma(1/a)a^{1/a}} (\tfrac{a}{\lambda_i a+1})^{N_i+\frac{1}{a}} \\
&= \tfrac{\Gamma(N_i+\frac{1}{a})}{N_i!\Gamma(1/a)} (\lambda_i a)^{N_i} (\tfrac{1}{\lambda_i a+1})^{N_i+\frac{1}{a}} \\
&= \tfrac{\Gamma(N_i+\frac{1}{a})}{N_i!\Gamma(1/a)} (\tfrac{\lambda_i a}{\lambda_i a+1})^{N_i} (\tfrac{1}{\lambda_i a+1})^{1/a}
\end{aligned}$$

which is the p.m.f. of a Negative Binomial variable as in (4.6). □

Using the Poisson-Gamma mixture representation, the likelihood function, conditional on Gamma distributed latent variables can be written as

$$L(\mathbf{N} \mid \boldsymbol{\beta}, \boldsymbol{\nu}, \mathbf{b}) = \prod_{i=1}^{I} \frac{(\nu_i \exp(\boldsymbol{\beta}^T \boldsymbol{x}_i + b_i))^{N_i} \exp(-\nu_i \exp(\boldsymbol{\beta}^T \boldsymbol{x}_i + b_i))}{N_i!}. \tag{4.7}$$

Under the Bayesian hierarchical framework, vague prior distributions are adopted for the remaining parameters. The prior distribution of $\nu_i \mid a$ is i.i.d Gamma$(\frac{1}{a}, a)$ with p.d.f

$$P(\nu_i \mid a) = \frac{\nu_i^{\frac{1}{a}-1}\exp(-\nu_i/a)}{\Gamma(1/a)a^{1/a}},$$

where $\mathrm{E}(\nu_i) = 1$ and $\mathrm{Var}(\nu_i) = a$. The prior distribution for regression coefficients β_k is i.i.d $\mathrm{N}(\mu_{\beta_k}, \sigma^2_{\boldsymbol{\beta}})$ with p.d.f

$$P(\beta_k) = \frac{1}{\sqrt{2\pi}\sigma_{\boldsymbol{\beta}}} \exp(-\frac{(\beta_k - \mu_{\beta_k})^2}{2\sigma^2_{\boldsymbol{\beta}}})$$

where taking a large value for $\sigma^2_{\boldsymbol{\beta}}$ yields a vague prior. The prior distribution of $\boldsymbol{b}$ follows the CAR spatial model which has the form

$$P(\boldsymbol{b} \mid \tau) \propto \exp\Big\{ -\frac{\tau}{2} \boldsymbol{b}^T (D_W - W) \boldsymbol{b} \Big\},$$

or

$$P(\boldsymbol{b} \mid \tau) \propto \exp\Big\{ -\frac{\tau}{2} \sum_{i \neq j} W_{ij}(b_i - b_j)^2 \Big\}$$

where $D_W = \text{diag}\{W_{1+}, W_{2+}, ..., W_{n+}\}$ is a diagonal matrix with $W_{i+} = \sum_{j=1}^{I} W_{ij}$ representing the total number of neighbours of region i. Moreover, in order to obtain fully Bayesian inference, hyper-priors for a and τ are also specified. The Gamma$(\frac{1}{\varepsilon}, \varepsilon)$ distribution is taken to be the hyper-prior distribution for a which has the form

$$P(a) = \frac{a^{\frac{1}{\varepsilon}-1} exp(-a/\varepsilon)}{\Gamma(1/\varepsilon)\varepsilon^{1/\varepsilon}},$$

where $\text{E}(a) = 1$ and $\text{Var}(a) = \varepsilon$ and taking a large value for ε yields a vague prior. The hyper-prior for the precision parameter τ is chosen as Gamma$(2, 1)$ with the p.d.f

$$P(\tau) = \tau exp(-\tau).$$

This is equivalent to assigning an Inverse-Gamma$(2, 1)$ distribution as the prior for the variance component $\sigma^2_{\mathbf{b}}$ and gives $\text{E}(\sigma^2_{\mathbf{b}}) = 1$ and $\text{Var}(\sigma^2_{\mathbf{b}}) = \infty$.

The posterior distribution of $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{\nu}, a, \sigma^2_{\mathbf{b}})$ given the data under our hierarchical model specification takes the form

$$\begin{aligned} f(\boldsymbol{\theta} \mid \boldsymbol{N}) &= L(\mathbf{N} \mid \boldsymbol{\beta}, \boldsymbol{\nu}, \mathbf{b}) \times \textstyle\prod_{k=0}^{5} P(\beta_k) \times \prod_{i=1}^{I} P(\nu_i \mid a) \times P(a) \times P(\boldsymbol{b} \mid \tau) \times P(\tau) \\ &/ \int_{\boldsymbol{\theta}} L(\mathbf{N} \mid \boldsymbol{\beta}, \boldsymbol{\nu}, \mathbf{b}) \times \textstyle\prod_{k=0}^{5} P(\beta_k) \times \prod_{i=1}^{I} P(\nu_i \mid a) \times P(a) \times P(\boldsymbol{b} \mid \tau) \times P(\tau) d\boldsymbol{\theta} \end{aligned}$$

Computing the posterior distribution analytically is not tractable since the calculation of the corresponding normalizing constant involves high dimensional integration (the fire count data has $\dim(\boldsymbol{\theta}) = 3432$ which is the total number of parameters in the model). Also, we are interested in the marginal distribution of each component of $\boldsymbol{\theta}$ where integrations are essential. Therefore, a Markov Chain Monte Carlo algorithm is designed to draw samples from the posterior and summarize aspects of the posterior distribution.

4.2 Computational Implementation

As discussed in Section 2.4.2, MCMC algorithms allow us to generate realizations from a distribution without knowing the normalizing constant and avoid the difficulty in calculation of high dimensional integration. Gibbs sampling in conjunction with Metropolis-Hastings steps are used for generating realizations from the posterior distribution. For Metropolis-Hastings updates, we will use Gaussian random walk candidate distributions; which may be applied after a suitable transformation to the real line for parameters taking values on proper subsets of the real line. In such cases, the Metropolis-Hastings acceptance ratio must be modified to accommodate the Jacobian of this transformation. For the overdispersed spatial count model specified in Section 4.1 based on the Poisson-Gamma mixture representation, model parameters are updated in the following way:

1. We update each scalar element β_k of $\boldsymbol{\beta}$ separately. The full conditional distribution for β_k based on the prior $\beta_k \overset{i.i.d}{\sim} \mathrm{N}(0, \sigma_\beta^2)$ has p.d.f. proportional to

$$L(\mathbf{N} \mid \boldsymbol{\beta}, \boldsymbol{\nu}, \mathbf{b}) \times \exp(-\frac{\beta_k^2}{2\sigma_\beta^2}),$$

   where $L(\mathbf{N} \mid \boldsymbol{\beta}, \boldsymbol{\nu}, \mathbf{b})$ is the likelihood function in the form (4.7). A random walk Metropolis step with candidate generated from a normal distribution is used.

2. The full conditional distribution for the overdispersion parameter a based on the prior $a \sim \mathrm{Gamma}(\frac{1}{\varepsilon}, \varepsilon)$ has p.d.f. proportional to

$$g(a) = \prod_{i=1}^{n} \frac{\nu_i^{\frac{1}{a}-1} \exp(-\nu_i/a)}{\Gamma(1/a) a^{1/a}} \times a^{\frac{1}{\varepsilon}-1} \exp(-a/\varepsilon).$$

   A candidate value a^* is obtained by transforming a to the real line and applying a random walk Metropolis step as follows:

- Obtain a draw, x, from a $\mathrm{N}(\log(a), c)$ distribution, where $c > 0$ is a tuning parameter.
- Let $a^* = \exp(x)$.

The Metropolis-Hastings acceptance probability for this move is $p = \mathrm{Min}(1, A)$ where

$$A = \frac{a^*}{a} \times \frac{g(a^*)}{g(a)}$$

and the term a^*/a comes from the Jacobian due to the change of variables.

3. The full conditional distribution for each latent variable ν_i based on the gamma mixing distribution $\nu_i \mid a \overset{i.i.d}{\sim} \mathrm{Gamma}(\frac{1}{a}, a)$ is

$$\mathrm{Gamma}\Big(N_i + \frac{1}{a}, (\exp(\boldsymbol{\beta}^T \boldsymbol{x}_i + b_i) + \frac{1}{a})^{-1}\Big)$$

from which we can draw directly in a Gibbs step.

4. With the prior distribution $\tau \sim \mathrm{Gamma}(2, 1)$, the resulting full conditional distribution for the precision parameter τ (inverse of the spatial variance component) is

$$\mathrm{Gamma}\Big(\frac{n-1}{2} + 2, (\frac{1}{2}\mathbf{b}^T(D_W - W)\mathbf{b} + 1)^{-1}\Big)$$

from which we can also draw directly in a Gibbs step.

5. We update each element b_i of the spatial random effects $\boldsymbol{b}$ separately. The full conditional distribution for b_i based on the prior $\mathbf{b} \mid \tau \sim \mathrm{CAR}(\tau)$ has p.d.f. proportional to

$$\exp(N_i b_i)\exp(-\nu_i \exp(\boldsymbol{\beta}^T \boldsymbol{x}_i + b_i)) \times \mathrm{N}\Big(\frac{\sum(W_{ij} b_i)}{W_{i+}}, \frac{1}{\tau W_{i+}}\Big).$$

We use a random walk Metropolis step with candidate generated from a normal distribution and re-center $\boldsymbol{b}$ around its own mean after each iteration in order

to identify an overall intercept in $\boldsymbol{\beta}$ (since the CAR prior we use arises from a singular normal distribution that is invariant to shifts in the mean).

In addition, the random effects b_i and ν_i defined in section 4.1 are not uniquely identified; however, the sum $\alpha_i = b_i + \log(\nu_i)$ is indeed identifiable (Eberly and Carlin, 2003). The issue of running an MCMC sampler over non-identified parameter spaces is a subtle issue, but one that has been discussed extensively, beginning with Besag and Green (1993) and more recently by Gustafson (2005). We avoid a technical discussion here and simply note that, in general, posterior inference obtained from samplers running over non-identified spaces is valid, provided that posterior summaries and inference are made with respect to the identifiable part of the model only. Thus we do not monitor or attempt to make inference on ν_i and b_i separately, but only on α_i, which is well identified as the random intercept in a generalized linear mixed model (Breslow and Clayton, 1993).

4.3 Analysis of Synthetic Data

Before applying our MCMC algorithm to fitting the B.C. fire frequency data, we test our implementation using synthetic data. We also use this simulation exercise to examine the adequacy of the spatial CAR model in detecting an underlying spatial signal arising through residual spatial variation.

The synthetic data are generated based on a region divided into 25×25 equally spaced grid cells and the neighbourhood structure is based on an adjacency matrix W where $W_{ij} = 1$ if region i and j share a common boundary for $i \neq j$ and $W_{ij} = 0$ otherwise (similar to Figure 2.1 and adjacency matrix 2.16). Suppose three covariates $\boldsymbol{x_1}$, $\boldsymbol{x_2}$ and $\boldsymbol{x_3}$ are generated independently from a normal distribution with mean 0 and variance 1. We then simulate the synthetic data in the following way: first, a vector of spatial random effects $\boldsymbol{b}$ is generated from a CAR model with $\tau = 3$,

and a vector $\boldsymbol{\nu}$ of latent variables representing extra-Poisson variability is generated from Gamma$(\frac{1}{a}, a)$ distribution where we set the dispersion parameter $a = 2$. Second, we set the intercept $\beta_0 = 1$ and set the regression coefficients corresponding to each covariate to be -1.5, 2, -1 respectively. The response Y_i of cell i is then generated from a Poisson distribution with mean $\nu_i \exp(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + b_i)$. We fit the overdispersed spatial count model as described in section 4.1 to the simulated data. If the MCMC algorithm mentioned in section 4.2 performs as it should, estimated model parameters $\hat{\boldsymbol{\beta}}$, $\hat{a}$, $\hat{\tau}$ and $\hat{\boldsymbol{\alpha}}$ (where $\alpha_i = b_i + \log(\nu_i)$) should be similar to the true values of these parameters as used for generating the synthetic data. We program our MCMC sampler in Matlab. In order to reduce the autocorrelation of the chain and to save computer memory space, each chain is thinned by 100. The precision parameter τ is the hyper-parameter of the model which has the slowest rate of convergence. We will only display plots which are related to τ for convergence assessment. Figure 4.1 shows that the autocorrelation of the chain drops quickly after thin. The algorithm has reached stationarity after 10000 iterations based on the ergodic average plot of τ (Figure 4.2). Therefore, we use a Monte Carlo sample of 10000 after the burn-in to produce posterior summaries.

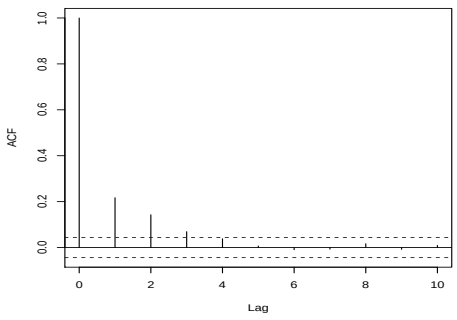

Figure 4.1: Autocorrelation Plot of τ

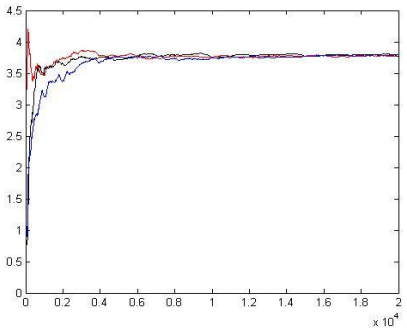

Figure 4.2: Ergodic Average Plot of τ

Table 4.1: Parameter Summaries of Synthetic Data

	True Value	Matlab Estimation	WinBugs Estimation
β_0	1	1.10 (0.10)	1.10 (0.08)
		(0.91, 1.29)	(0.94, 1.27)
β_1	-1.5	-1.36 (0.10)	-1.37 (0.08)
		$(-1.55, -1.19)$	$(-1.53, -1.21)$
β_2	2	1.84 (0.11)	1.84 (0.10)
		(1.61, 2.05)	(1.65, 2.05)
β_3	-1	-0.99 (0.09)	-0.99 (0.08)
		$(-1.17, -0.81)$	$(-1.15, -0.83)$
τ	3	3.74 (1.49)	3.75 (1.48)
		(1.29, 6.83)	(1.32, 6.82)
a	2	2.13 (0.18)	2.13 (0.18)
		(1.79, 2.49)	(1.79, 2.49)

Table 4.1 is a summary of model parameters including the posterior mean, standard deviation and 95% credible interval. Three sets of values are listed in the table: the true values, estimated values using our algorithm coded in Matlab, and estimated values obtained from WinBugs (an off-the-shelf software package for fitting Bayesian hierarchical models). Note that while WinBugs uses MCMC to sample from the posterior, it does not use our algorithm; rather, a Gibbs sampler based on adaptive rejection sampling (Gilks and Wild, 1992) is employed. This allows us to further check our algorithm and implementation as posterior summaries obtained from different sampling schemes should still produce similar results (up to Monte Carlo error) if both sampling schemes are valid, and have been correctly implemented. Examining the results, it is clear that estimates from the two implementations are extremely

similar and the 95% credible interval contains the true value of each parameter. We feel, based on this exercise that it is safe to conclude that our computational implementation is without major error.

Next, we examine the ability of the spatial CAR model to represent residual spatial structure under various settings. We work on the same 25×25 grid and use the same covariates as above; however, different underlying spatial structures are generated. In the first case, $\boldsymbol{b}$ is simulated from a CAR model with $\tau = 3$. In the second case, a simple north/south divide is used where $\boldsymbol{b} = -1$ for grid cells in the northern half of the grid and $\boldsymbol{b} = 1$ for grid cells in the south. In the third case, a linear function $f(x_i, y_i) = 0.1 * x_i + 0.1 * y_i$ is used to represent the residual spatial variability where x_i and y_i are the coordinates of the centroid of cell i. Specifically, we set $b_i = f(x_i, y_i) - \bar{\boldsymbol{c}}$ where $\bar{\boldsymbol{c}} = \frac{1}{625}\sum_{i=1}^{625} c_i$. For simplicity, we set $\boldsymbol{\nu} = \mathbf{1}$ for this simulation exercise, as we are primarily concerned with examining the CAR model and the spatial structure. We thus fit the data with a Poisson model with CAR random effects which, of course, is a special case of Negative Binomial spatial model. We fit our model to all three simulated data sets and results displayed correspond to a total sample size of 15000 after burn-in of 5000 iterations using three chains initialized at different points in the parameter space. Figure 4.3, 4.4 and 4.5 present maps of the true random effects in comparison with the estimated values (posterior mean) of the random effects for each of the three different cases described above. In case one, we get a a good re-construction of the original spatial surface. This is not entirely unexpected since the simulated data come from the CAR model (thus we are fitting a correctly specified model). The posterior mean of the CAR precision τ is 3.384 ($std = 0.3757$) and 95% CI is $(2.707, 4.163)$ which contains the true value $\tau = 3$. Similarly, in case three, where the linear function is used for generating the spatial surface, the CAR model successfully captures the smooth change over space. Finally in the second case, that of the North-South boundary, again the CAR model

performs adequately. Some over-smoothing of the boundary is evident; however, this is not particularly worrisome as exact boundary detection is not a goal in our forest fire analysis. Table 4.2 lists posterior mean, standard deviation and 95% credible intervals for regression coefficients, for the three simulated data sets. In all cases, the posterior distribution is heavily concentrated about the true values. Note that since the estimated values are extremely close to the values chosen for generating the data, it seems as though the inference is strongly driven by the observed data and that the vague prior distributions do not play a strong role in the estimation.

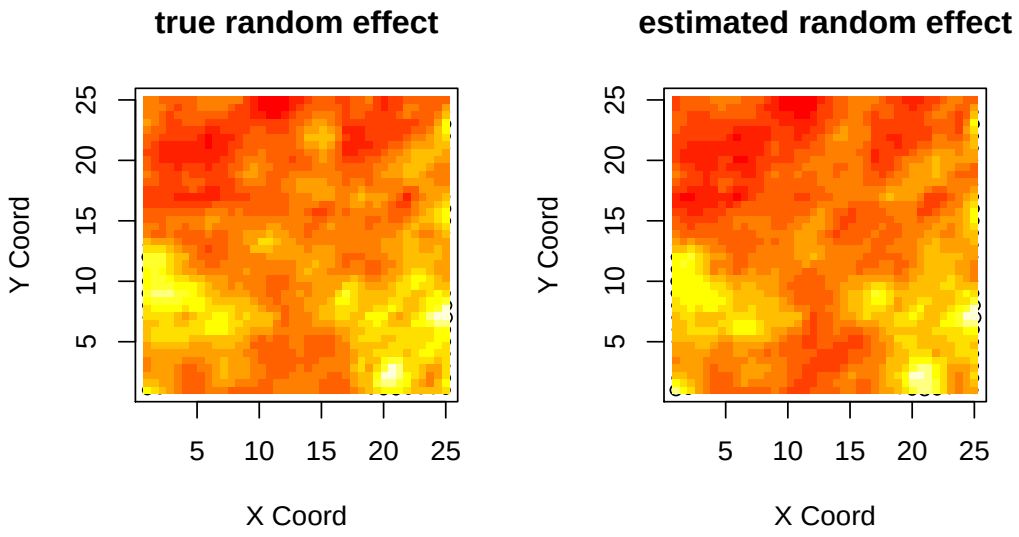

Figure 4.3: Map of Case 1 (CAR Random Effects)

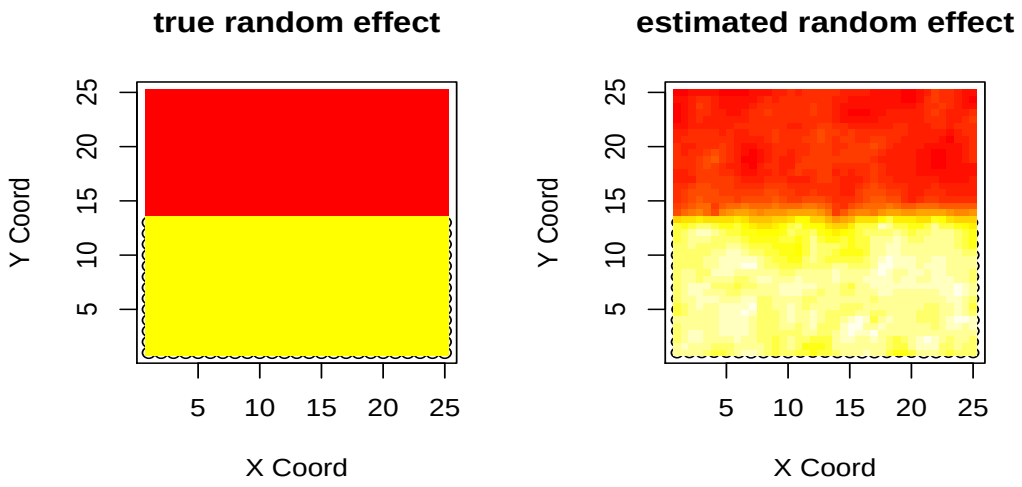

Figure 4.4: Map of Case 2 (North-South divide Random Effects)

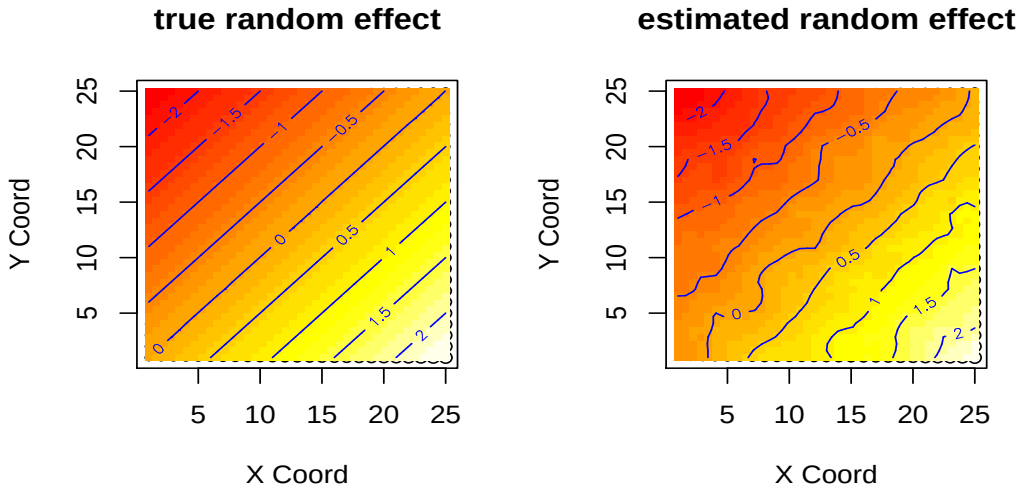

Figure 4.5: Map of Case 3 (Linear Random Effects)

Table 4.2: Regression Coefficients of Three Random Effects Cases

	True Value	case 1: CAR	case 2: Boundary	case 3: Linear
β_0	1	1.03 (0.06)	1.07 (0.05)	1.01 (0.01)
		(0.93, 1.14)	(0.98, 1.17)	(0.92, 1.08)
β_1	-1.5	-1.50 (0.04)	-1.50 (0.03)	-1.51 (0.03)
		$(-1.57, -1.42)$	$(-1.57, -1.44)$	$(-1.56, -1.46)$
β_2	2	2.00 (0.06)	1.97 (0.05)	2.01 (0.03)
		(1.90, 2.09)	(1.86, 2.06)	(1.96, 2.08)
β_3	-1	-0.99 (0.03)	-0.96 (0.03)	-0.99 (0.03)
		$(-1.05, -0.92)$	$(-1.01, -0.90)$	$(-1.05, -0.95)$

Chapter 5

Study of Fire Frequency

5.1 Model Estimation

In this section we apply the overdispersed spatial count model to the fire frequency data. In applying our computational algorithm, we center the covariate values about their means, which also improves MCMC convergence, and in addition, we perform all computations involving the likelihood on the log-scale, to avoid problems with numerical underflow. We set the prior variance for regression coefficients $\sigma_\beta^2 = 1000$, to get a normal prior whose variance is large for each component of $\boldsymbol{\beta}$, and set $\varepsilon = 1000$ to get a Gamma prior with mean 1 and variance 1000 for the dispersion parameter a. Seven different initial values spread over the parameter space for each parameter are used for convergence checking. Since sampled values within each chain are highly correlated, it takes a long time for the algorithm to converge. In this case each chain is thinned by 200 to reduce the autocorrelation and to save computer memory space. As shown in Figure 5.1 and 5.2, the autocorrelation function for hyper-parameters τ and a drop quickly after thinning. The trace plots (Figure 5.3, 5.5(a) and 5.5(c)) and the ergodic plots (Figure 5.4, 5.5(b) and 5.5(d)) indicate that the MCMC algorithm has reached stationarity and posterior mean estimates are very stable well before 2000 ($\times$200) iterations. In addition, the Gelman and Rubin diagnostic statistic for all parameters are close to 1 after 2000 ($\times$200) iterations. We remove the first 2000 ($\times$200) iterations as the burn-in period and posterior inference is based on the remaining 2000 thinned samples.

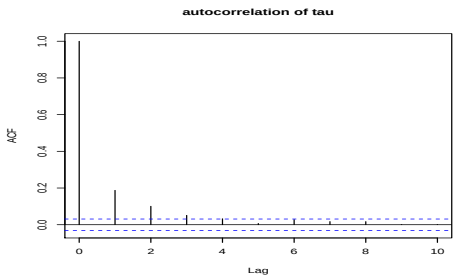

Figure 5.1: Autocorrelation Plot of τ

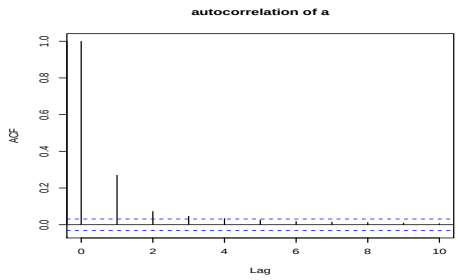

Figure 5.2: Autocorrelation Plot of a

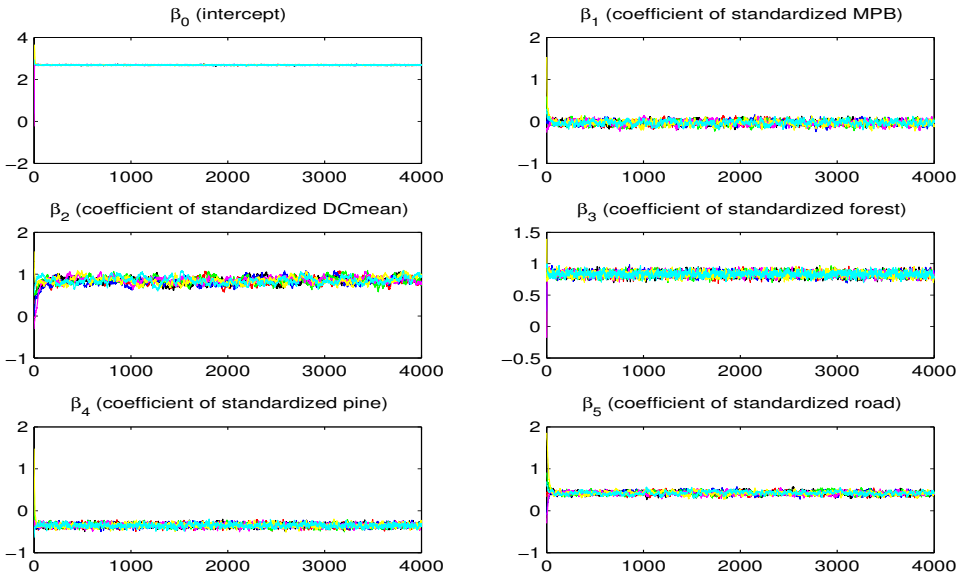

Figure 5.3: Trace plots of each component of $\boldsymbol{\beta}$ by plotting seven chains onto the same axis for convergence checking

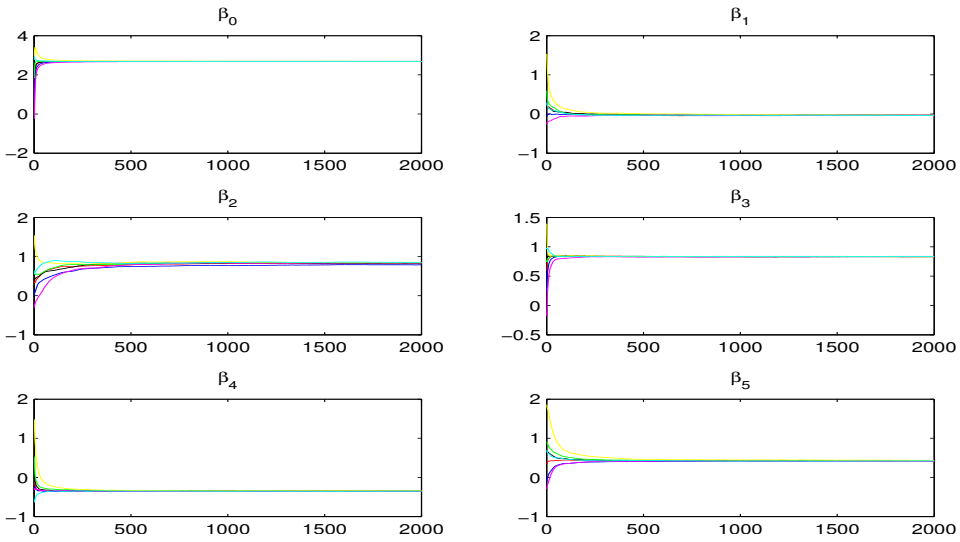

Figure 5.4: Ergodic average plots of each component of $\boldsymbol{\beta}$ by plotting seven chains onto the same axis for convergence checking

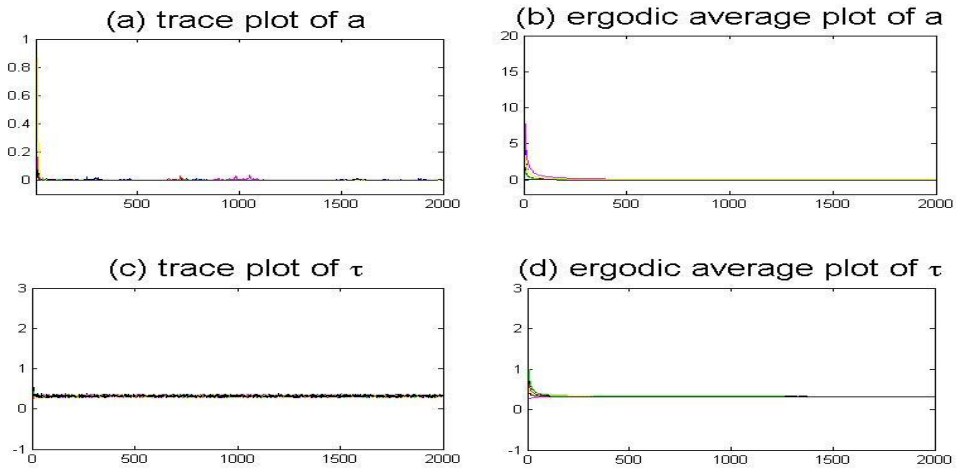

Figure 5.5: Trace plots and ergodic average plots of a and τ by plotting seven chains onto the same axis for convergence checking

Posterior summaries are presented in Table 5.1. We assess the impact of each covariate through examination of the 95% credible interval, and in particular a significant association is evident when the value $\boldsymbol{\beta} = 0$ is not contained within this interval. The covariates representing drought code, forest covering, number of roadways and number of pine leading stands are all significantly associated with fire frequency. This is not surprising as association between these factors and fire frequency is well understood. These covariates are included in the model to avoid confounding. Interestingly, under the fit of our spatial model, the area affected by mountain pine beetle is not significantly associated with fire frequency. This is in stark contrast to the results obtained from the standard log-linear Poisson regression model (3.1) used in exploratory analysis (see Table 5.2) which indicated a positive association between mountain pine beetle outbreak and fire frequency. The fundamental difference between the two sets of results can be seen clearly by comparing the estimates of variability appearing in Tables (5.1) and (5.2). The estimated standard errors for the regression coefficients in the non-spatial model are approximately one order of magnitude smaller than the corresponding estimates obtained from the hierarchical spatial model. Thus we see very clearly in our analysis how ignoring spatial correlation and over-dispersion can lead to spurious associations. A more complicated regression structure incorporating interaction between covariates was not considered and we leave investigation of interactions to future work. In addition, a more careful examination of the relationship between mountain pine beetle infection and fire frequency will necessarily involve a spatiotemporal analysis. Again, we leave this as an avenue for future work; however, extensions of the current model formulations to suite such an analysis are discussed in Chapter 6. We also note that the estimated over-dispersion parameter $\hat{a}$ is close to 0 which suggests that most of the residual variability can be attributed to spatially structured factors. This may also because the prior distribution of a is sharply peaked near zero that may distort posterior inferences (Gelman, 2006). Figure 5.6 and Fig-

ure 5.7 are maps of the posterior means and standard deviations of $\alpha_i = b_i + \log(\nu_i)$ summarized based on our fitted model. The map presented in Figure 5.6 is essentially estimating the residual variability in the observations. Such maps prove useful as exploratory tools in that any apparent 'hot-spots' can be used to suggest covariates that are missing from the model. Examination of Figure 5.7 also proves interesting. In particular, we can see that high variability of estimation exists across the boundary of the province. This pattern essentially illustrates an 'edge-effect' arising from lack of information for boundary cells which have fewer neighbours than interior cells. Posterior standard deviations are also largest towards the north of the province, where data on fire frequency is most sparse. Moreover, our model only takes the first-order neighbours into consideration; however, the western side of British Columbia is the Pacific ocean and the other sides of the province are connected with other provinces (e.g. Alberta). We may able to deal with the boundary effects in our model in the future by taking the second-order neighbours into consideration.

Table 5.1: Posterior Summaries of Negative Binomial Model with Spatial Frailty

	Mean	Std. Deviation	95% Credible Interval
β_0 (Intercept)	2.6879	0.0145	(2.6587, 2.7157)
β_1 (MPB)	−0.0335	0.0553	(−0.1446, 0.0746)
β_2 (DCmean)	0.8457	0.0811	(0.6941, 1.0119)
β_3 (forest)	0.8316	0.0377	(0.7577, 0.9058)
β_4 (pine)	−0.3491	0.0418	(−0.4313, −0.2655)
β_5 (road)	0.4220	0.0425	(0.3376, 0.5058)
$\tau = 1/\sigma^2_{\mathbf{b}}$ (CAR precision)	0.3221	0.0143	(0.2949, 0.3508)
a (overdispersion)	0.0002	0.0010	(0.0000, 0.0020)

Table 5.2: Parameters Summaries of Standard Log-Linear Poisson Model

	Estimate	Std. Error	95% Confidence Interval
β_0 (Intercept)	3.6163	0.0045	(3.6075, 3.6250)
β_1 (MPB)	0.2490	0.0037	(0.2417, 0.2562)
β_2 (DCmean)	0.4439	0.0028	(0.4384, 0.4495)
β_3 (forest)	0.7062	0.0045	(0.6974, 0.7150)
β_4 (pine)	−0.4925	0.0045	(−0.5014, −0.4836)
β_5 (road)	0.4016	0.0021	(0.3975, 0.4057)

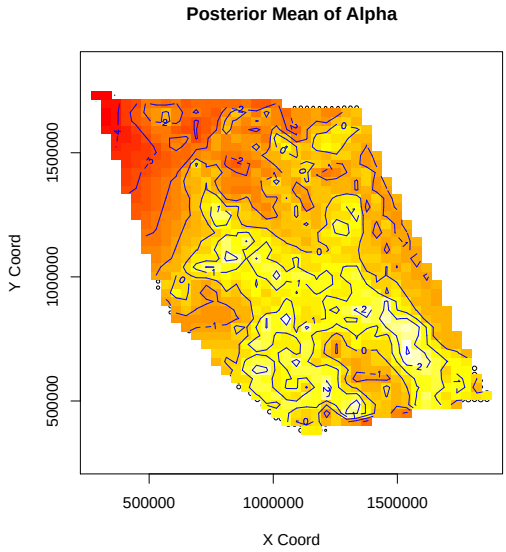

Figure 5.6: Map of posterior mean of $\boldsymbol{\alpha}$ based on Negative Binomial CAR model

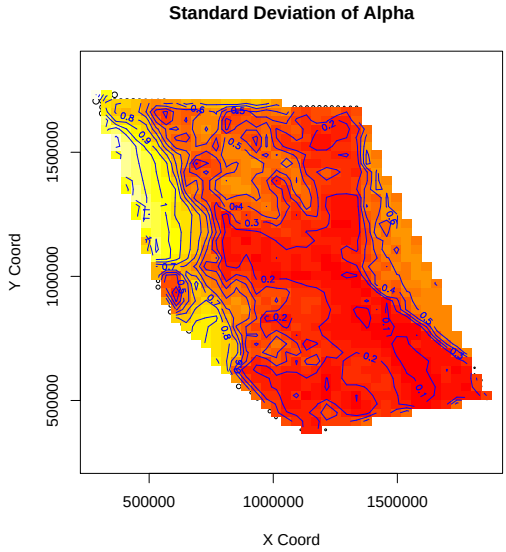

Figure 5.7: Map of posterior standard deviation of $\boldsymbol{\alpha}$ based on Negative Binomial CAR model

5.2 Model Selection

Besides fitting our Negative Binomial spatial model, simpler models arising as special cases are also fit to the data. In total we fit five models to the data, and we make comparisons using the DIC. The first special case is the simple Poisson log-linear regression model with no random effects, which was used in the exploratory analysis. The second model extends the first by incorporating region specific random effects that are assumed independent (as opposed to spatially correlated). A third model includes spatial random effects, but uses a Poisson, rather than a negative binomial error structure. Finally the fourth submodel uses a negative binomial model, but without spatial random effects.

Table 5.3 lists the effective number of parameters p_D and the DIC scores for all the models considered. As expected, the Poisson model with no random effects has the smallest p_D value which, again as expected, is equal to the actual number of parameters (number of regression coefficients). We note that this model has by far the highest DIC score indicating that the simple parsimonious model structure does not out-weigh the very poor fit of this simple model. By adding random effects onto the simple Poisson model, the effective number of parameters p_D increases greatly but the DIC scores substantially decrease illustrating improvements in model fit. Note also, the p_D values for the models with spatial random effects are smaller than the p_D values for the independent random effects model, since random effects are correlated in CAR models; reducing their effective count. Overall, the Negative Binomial model with spatial random effects has the smallest DIC value, followed closely by the spatial Poisson model. This is not entirely surprising as it seems most of the residual variation is spatially structured, and thus a large difference between the spatial models based on the negative binomial and Poisson error structures is not expected. We can explore the difference between these two models in more detail in the future by simulating

realizations from Negative Binomial model with CAR random effects and fitting the synthetic data using Poisson model with CAR random effects. Nevertheless, based on the suggestion from Spiegelhalter et al. (2002), differences in the DIC score of greater than 5 indicate a meaningful difference between models and thus it seems that our negative binomial spatial model is the preferred model according to the DIC criteria.

Table 5.3: p_D and DIC for the 5 Models considered for Fire Frequency Data

model	p_D	DIC
Poisson(λ_i), $\log(\lambda_i) = \boldsymbol{\beta}^T \boldsymbol{x}_i$	6	95292
Poisson(λ_i), $\log(\lambda_i) = \boldsymbol{\beta}^T \boldsymbol{x}_i + b_i$, $b_i \overset{i.i.d}{\sim} \mathrm{N}(0, 1/\tau)$	1476	10849
Poisson(λ_i), $\log(\lambda_i) = \boldsymbol{\beta}^T \boldsymbol{x}_i + b_i$, $\mathbf{b} \sim \mathrm{CAR}(1/\tau)$	1315	10697
NegBin(λ_i, a), $\log(\lambda_i) = \boldsymbol{\beta}^T \boldsymbol{x}_i$	1403	10710
NegBin(λ_i, a), $\log(\lambda_i) = \boldsymbol{\beta}^T \boldsymbol{x}_i + b_i$, $\mathbf{b} \sim \mathrm{CAR}(1/\tau)$	1295	10678

5.3 Model Checking

We check the fit of our model to various important aspects of the data using posterior predictive checking based on $L = 4000$ replicated data sets generated from posterior predictive distribution. We denote N_{ij}^{rep} as the i^{th} element in the j^{th} replicated data set. Figure 5.8 displays a histogram of the observed data along with histograms for 9 replicated data sets. The overall distributions of the observed data and the data predicted from the model seem similar. The scatter plots of 9 replicated fire count data sets shown in Figure 5.9 also show similarity to the observed data. In addition, Figure 5.10 directly displays contour plots of the observed data and 3 replicated data sets. No systematic differences can be found compared to the observed data, and the model seems to be describing the spatial distribution fairly well. Next we take a close look into three aspects of the data and quantify the lack-of-fit using

posterior predictive p-values. The first aspect examines the ability of our model to capture extreme values. To this end we base our test statistic on the largest order statistic

$$T_1 = N_{(I)} = \text{the largest value in the data set.}$$

Figure 5.11 displays the posterior predictive distribution of this statistic under our model along with the observed value. There is not an indication of a lack-of-fit with the corresponding $p - value = 0.575$. The second aspect of the data we examine is the total number of regions with zero fire counts, based on the test quantity

$$T_2 = \sum_{i=1}^{I} \mathrm{I}(N_{ij}^{rep} = 0),$$

where $I(\cdot)$ is the indicator function taking either 0 when the statement is false or 1 when the statement is true. Figure 5.12 displays the posterior predictive distribution of this statistic under our model along with the observed value. Here we see a clear indication of a lack-of-fit, with the actual number of zeros falling far in the right tail of the predictive distribution (p-value = 0.003). Our model is unable to describe adequately this feature of the data. A possible remedy involves an extension of our model to incorporate a point mass on zero values, and this is discussed in Chapter 6. Finally, we check the model's ability to describe the ratio of mean to variance with

$$T_3 = \frac{\text{mean}(\mathbf{N}_j^{rep})}{\text{variance}(\mathbf{N}_j^{rep})},$$

where $\mathbf{N}_j^{rep}$ is the j^{th} replicated data set. Figure 5.13 displays the posterior predictive distribution of this statistic under our model along with the observed value. There is no indication of a lack-of-fit ($p - value = 0.3875$) to this aspect of the data.

Overall, our model is able to describe several key features of this complicated data set; however, it is clear that the 'zero-heavy' nature of the data needs further investigation. The conclusions of our analysis are admittedly subject to this flaw

in our model. Nevertheless, model development, particularly in complicated studies of environmental monitoring, is necessarily an iterative process, and we intend to address the model adequacy issues raised in this section as future work.

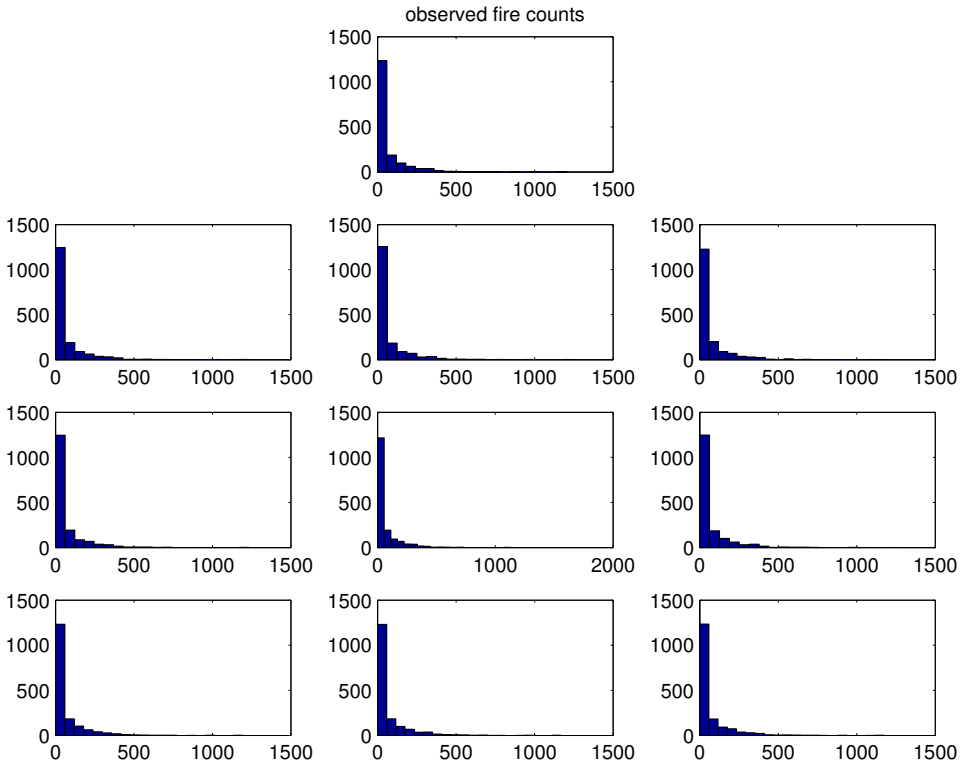

Figure 5.8: Histograms of Observed Data and Replicated Data Sets
(x-axis presents the index of observations, y-axis presents the data value)

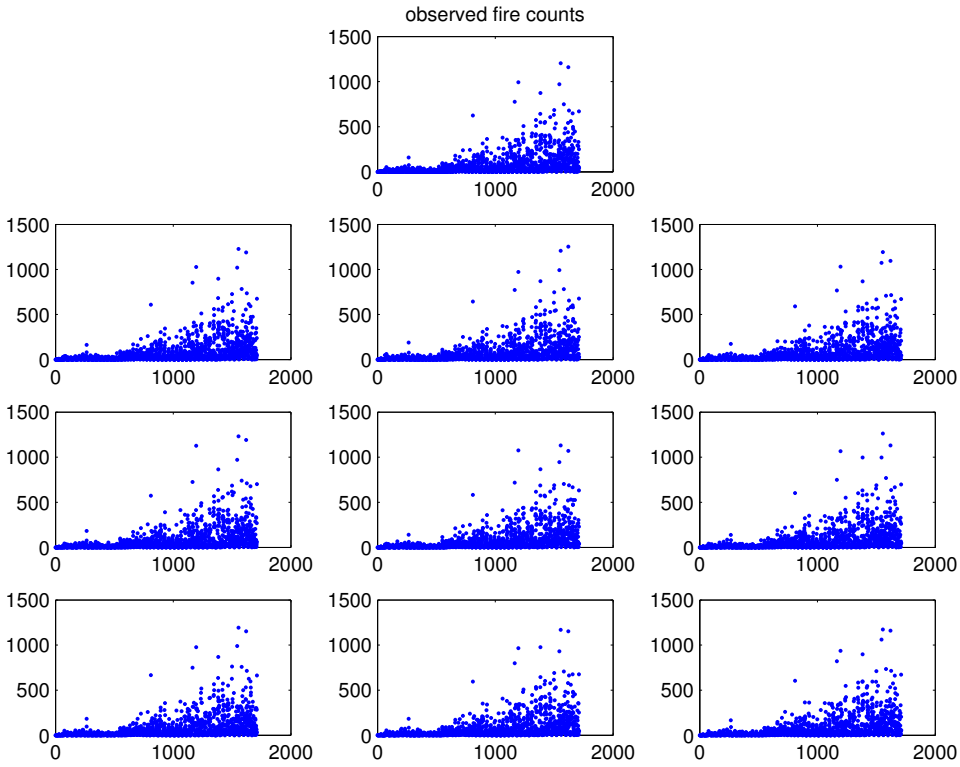

Figure 5.9: Scatter Plots of Observed Data and Replicated Data Sets
(x-axis presents the index of observations, y-axis presents the data value)

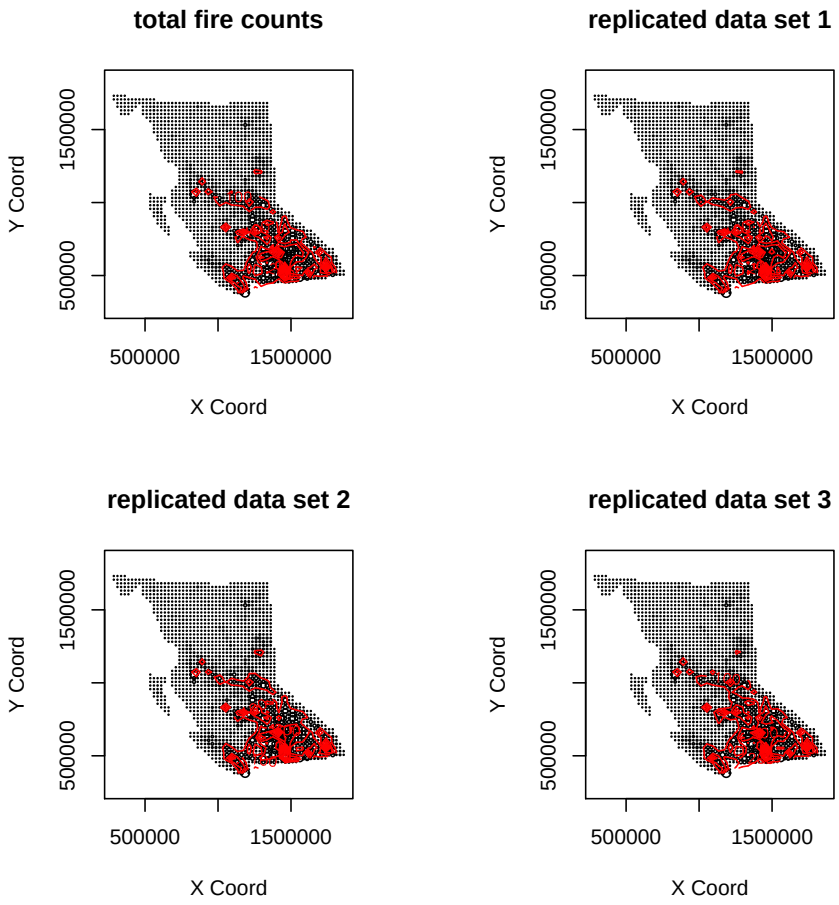

Figure 5.10: Contour Plots of Observed Data and Replicated Data

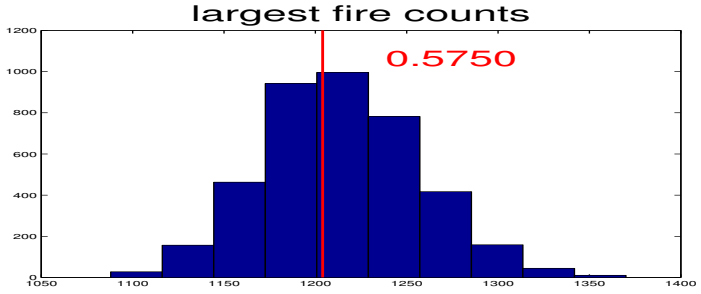

Figure 5.11: The Largest Fire Counts

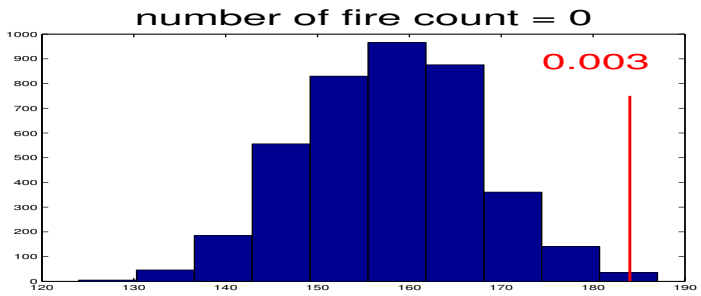

Figure 5.12: Number of Zeros

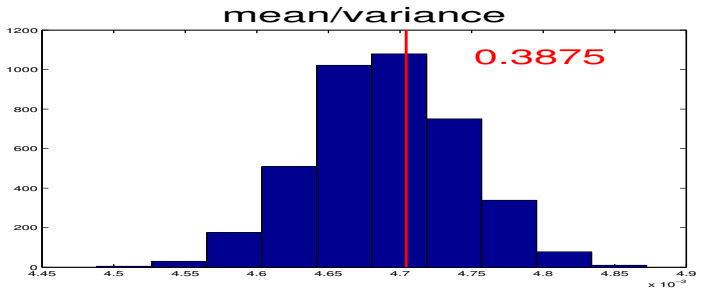

Figure 5.13: Overdispersion with Represented Using Mean and Variance Ratio

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we have developed a Negative Binomial regression model with spatial random effects under the Bayesian hierarchical framework. This model takes care of overdispersion apparent in the data while incorporating spatial information through random effects. We implemented the model using Markov Chain Monte Carlo algorithms. The methods were validated through the analysis of synthetic data. Also, from analyzing three different sets of random effects, we can conclude that the CAR model works well in capturing smooth change over space. On the other hand, the CAR model is not good at exact boundary detecting and finding extrema since the CAR model heavily relies on the neighbourhood information and is a Gaussian model. Therefore, we will see some over-smoothing in these two cases.

The application of the Negative Binomial model with CAR random effects is not limited to fire count data, it also can be used for other count data in different fields, such as epidemiology and ecospecies. For example, if we have the death count of lung cancer as the response variable along with several regional covariates and neighbourhood structure of these regions, then we can perform analysis to the data using our model in the same way as what we did to the fire count data. Moreover, since the model is implemented based on the Poisson-Gamma mixture representation, we can easily reduce the Negative Binomial model to the Poisson model by setting the latent variable $\boldsymbol{\nu} = \mathbf{1}$. In the case where dispersion is not apparent in the data, this is very handy. We can still apply the model with slight modification of the algorithm.

6.2 Future Work

As discovered in the previous chapter, the Negative Binomial regression model with spatial random effects does not capture the large number of zeros presented in the data. In order to deal with this heavy tail property of the data, a zero heavy Poisson mixture (Hougaard et al., 1997) could be considered. Instead of simply having $N_i \mid \nu_i, \lambda_i \sim \text{Poisson}(\nu_i \lambda_i)$ in the model, we define the p.m.f. of N_i, at the first level of the model, to take a mixture form

$$f^*(N_i \mid \nu_i, \lambda_i, \rho) = \rho f(N_i \mid \nu_i, \lambda_i) + (1-\rho)\text{I}\{N_i = 0\},$$

where $\rho \in [0, 1]$, is a mixing probability, $f(N_i \mid \nu_i, \lambda_i)$ is the Poisson p.m.f. and $I(\cdot)$ is the indicator function. Since our main focus is looking at the relationship between MPB outbreaks and fire frequency, a simple alternative method to the zero heavy Poisson mixture model is that we can simply ignore the northern half area of the province. Mountain pine beetle outbreaks are clustered in the center interior of the province and the most subregions in the northern part of the province have zero fire counts. Therefore, ignoring these area may solve the problem of large number of zeros.

Figure 6.1 and 6.2 display the temporal distribution of the mountain pine beetle infection data. Our analysis has ignored this temporal variation and we recognize this as a drawback in this, our initial investigation of the data. A very simple extension of our model, denoting N_{it} as the number of fires in region i at time t, could be based on

$$N_{it} \mid \lambda_{it}, a \sim \text{Negbin}(\lambda_{it}, a), \tag{6.1}$$

where λ_{it} describes the mean and $a \geq 0$ is the dispersion parameter. A log-link function

$$\log(\lambda_{it}) = \boldsymbol{\beta}^T \boldsymbol{x_{it}} + b_i \tag{6.2}$$

is used, where $\boldsymbol{x_{it}}$ is a vector of time-dependent covariates, which could represent the past history of mountain pine beetle severity, and b_i is the spatial effect as before. This model would allow us to examine 'time-lagged' associations, between the severity of mountain pine beetle and mean fire frequency over time. This would be far more informative than the time-aggregated analysis performed so far. Indeed, the aggregation over time may well have masked associations that could only be uncovered in a space-time examination of this data. Note also that the association between pine beetle infection and fire frequency may itself exhibit a changing pattern over time, so that dynamic models (Pole et al., 1999), allowing the regression coefficients $\boldsymbol{\beta}$ to evolve over time may prove a useful avenue for future work.

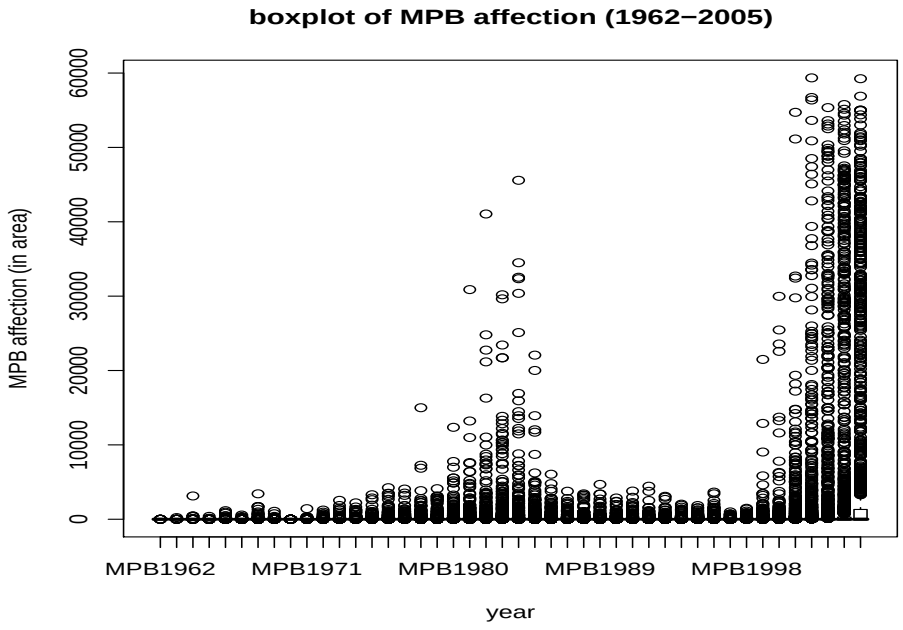

Figure 6.1: Boxplot of Yearly Area affected by MPB

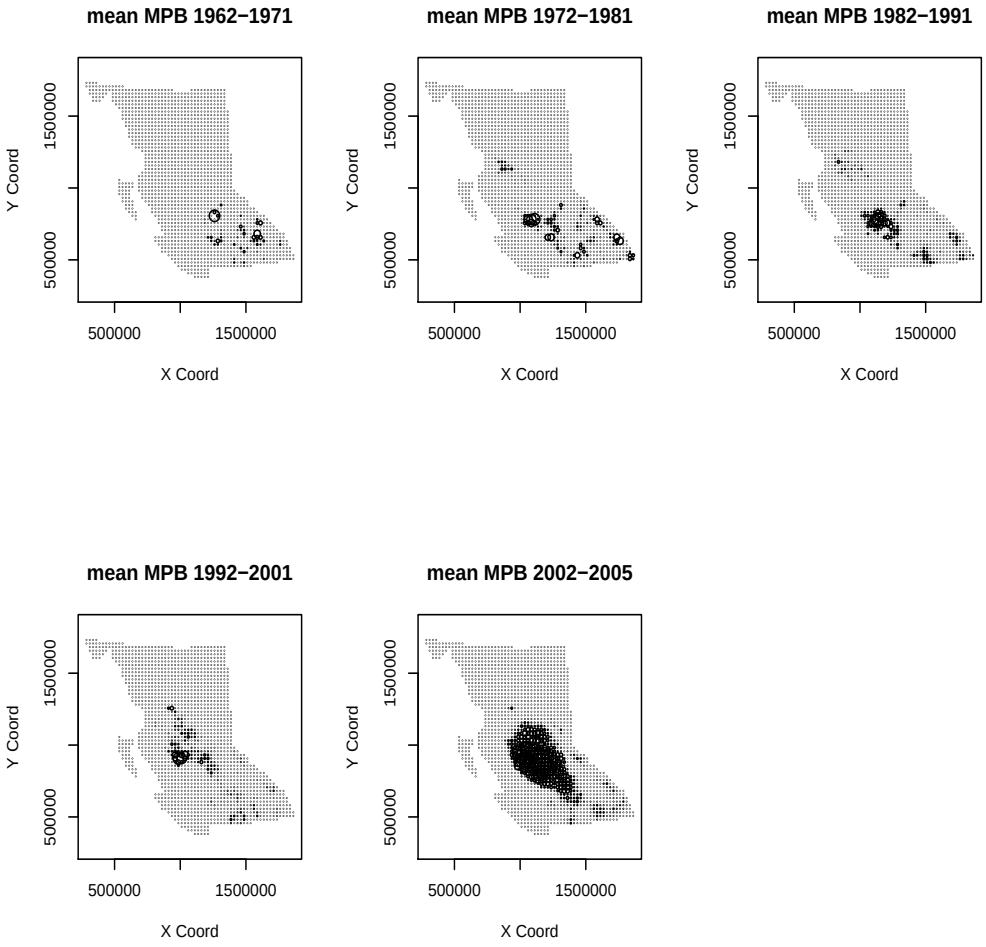

Figure 6.2: Mean Area affected by MPB Outbreaks for Each Ten Year Period

References

Banerjee, S., Carlin, P. B. and Gelfand, E. A. (2004). *Hierarchical Modeling and Analysis for Spatial Data*. New York: CRC.

Barndorff-Nielsen, O. (1978). *Information and Exponential Families in Statistical Theory*. New York: Wiley.

Bastos, S. L. and Gamerman, D. (2006). Dynamic survival models with spatial frailty. *Lifetime Data Analysis* **12**, 441-460.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society. B* **36**, 192-236.

Breslow, N. (1984). Extra-Poisson variation in log-linear models. *Applied Statistics* **33**, 38-44.

Brook, D. (1964). On the distinction between the conditional probability and the joint probability approaches in the specification of nearest-neighbour systems. *Biometrika* **51**, 481-483.

Casella, G. and George, E. I. (1992). Explaining the gibbs sampler. *The American Statistician* **46**, 167-174.

Chib, S. and Greenberg, E. (1994). Bayes inference for regression models with ARMA(p, q) errors. *Journal of Econometrics* **64**, 183-206.

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. *The American Statistician* **49**, 327-335.

Cox, D. R. and Hinkley, D. V. (1974). *Theoretical Statistics*. London: Chapman & Hall.

Cressie, N. A. C. (1993). *Statistics for Spatial Data*, 2nd edn. New York: Wiley.

Dobson, A. J. (1990). *An Introduction to Generalized Linear Models*. London: Chapman & Hall.

Gamerman, D. and Lopes, H. F. (2006). *Markov Chain Monte Carlo*, 2nd edn. New York: CRC.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* **85**, 398-409.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis* **1**, 515-533.

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (2004). *Bayesian Data Analysis*, 2nd edn. New York: CRC.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science* **7**, 457-72.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 721-724.

Geyer, C. J. (1992). Practical Markov Chain Monte Carlo (with discussion). *Statistical Science* **7**, 473-511.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov Chains and their applications. *Biometrika* **57**, 97-109.

Hougaard, P., Lee, T. M. and Whitmore, A. G. (1997). Analysis of overdispersed count data by mixtures of poisson variables and poisson processes. *Biometrics* **53**, 1225-1238.

Lawless, F. J. (1987). Negative binomial and mixed poisson regression. *The Canadian Journal of Statistics* **15**, 209-225.

Liu, J. S., Wong, W. H. and Kong, A. (1994). Correlation structure and convergence rate of the Gibbs sampler: applications to the comparison of estimators and augmentation schemes. *Biometrika* **81**, 27-40.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika* **57**, 97-109.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953). Equation of state calculations by fast computing machine. *Journal of Chemical Physics* **21**, 1087-1091.

Müller, P. (1993). A generic approach to posterior integration and gibbs sampling, manuscript.

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society. A* **135**, 370-384.

Oehlert, W. G. (1992). A Note on the Delta Method. *The American Statistician* **46**, 27-29.

Pole, A., West, M. and Harrison, J. (1999). *Applied Bayesian Forecasting and Time Series Analysis*. New York: CRC.

Raftery, A. E. and Lewis, S. (1992). How many iterations in the Gibbs sampler?. In *Bayesian Statistics 4* (eds J. M. Bernardo *et al.*). Oxford: Oxford University Press, 763-773.

Robert, C. P. (2001). *The Bayesian Choice*, 2nd edn. New York: Springer-Verlag.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1994). Weak convergence and optimal scaling of random walk metropolis algorithms, Technical report, University of Cambridge.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and van der Linde, A. (2002). Bayesian

measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B* **64**, 583-639.

Taylor, S. W., Parminter, J. and Thandi, G. (2005). Logistic regression models of wildfire probability in British Columbia.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Annuals of Statistics* **22**, 1701-1762.

Appendix

A. R Code for Example 2

```
M <- 500        # number of chains
K <- 10         # number of iterations per chain
X <- matrix(0, M, K)        # simulated value, one chain per row
Y <- matrix(0, M, K)

# parameter values
n = 16
alpha = 2
beta = 4

# initial values
x0 <- rep(0:n, ceiling(M/n))
X[,1] <- x0[1:M]
y0 <- runif(M)
Y[,1] <- y0

Gibbs1 <- function(M, K){
    for(m in 1:M){
        for(k in 2:K){
            Y[m,k] <- rbeta(1, X[m, k-1]+alpha, n-X[m, k-1]+beta)
            X[m,k] <- rbinom(1, n, Y[m,k])
        }
    }
```

```
    return(X)
}

X <- Gibbs1(M, K)
hist(X[,K], freq=FALSE, right=FALSE, xlab = "x", ylab = "f(x)", main = "")

# true marginal density
fx <- rep(0, n+1)
for(x in 1:(n+1)){
    c1 <- factorial(alpha+beta-1) / (factorial(alpha-1) * factorial(beta-1))
    c2 <- factorial((x-1)+alpha-1)* factorial(n-(x-1)+beta-1)
    c3 <- factorial(alpha + beta + n-1)
    fx[x] <- choose(n, x-1) * c1 * c2 / c3
}
lines(0:n, fx, lty=1)
```

B. R Code for Example 3

```
M <- 500        # number of chains
K <- 15         # number of iterations per chain
X <- matrix(0, M, K)        # simulated value, one chain per row
Y <- matrix(0, M, K)
B = 5        # parameter value

# initial values
x0 <- runif(M, 0, B)
X[,1] <- x0
```

```
y0 < − runif(M, 0, B)
Y[,1] < − y0

for(m in 1:M){
    for(k in 2:K){
        u < − runif(1)
        Y[m,k] < − log(1-u*(1-exp(-X[m,k-1]*B)))/(-X[m,k-1])
        u < − runif(1)
        X[m,k] < − log(1-u*(1-exp(-Y[m,k]*B)))/(-Y[m,k])
    }
}
hist(X[,K], 30, freq=FALSE, xlab = "x", ylab = "f(x)", main = "")

# true marginal density
w < − 0.0001
x1 < − seq(0.01, B, w)
fx < − rep(0, length(x1))
for(t in 1:length(x1)){
    fx[t] < − (1-exp(-B*x1[t]))/x1[t]
}
A < − sum(fx)*w       # normalizing constant
lines(x1, fx/A)
```

C. R Code for Example 4

```
# parameter values mu < − matrix(c(1,2), 2, 1)
Sigma < − matrix(c(1,0.9,0.9,1), 2, 2)
```

```
par(mfrow=c(2,2))

# the Choleski approach (Direct Sampling)
nsim <- 4000
x <- matrix(NA, 2, nsim)
P <- chol(Sigma)        # Choleski factorization: PP^T = Σ
u <- matrix(NA, 2, 1)
for(t in 1:nsim){
    u <- rnorm(2, 0, 1)
    x[,t] <- mu + t(P) %*% u
}
plot(x[1,], x[2,], xlim=c(-3,5), ylim=c(-3,6), xlab="x1", ylab="x2", main="(a) Direct Sampling")
grid(8,9)

nsim <- 6000

# Candidate generating density 1 (Random Walk (Uniform))
x1 <- matrix(NA, 2, nsim)
x1[,1] <- matrix(c(0,0), 2, 1)
z <- matrix(c(0,0), 2, 1)
accept <- 0
for(t in 2:nsim){
    z[1] <- runif(1, -0.75, 0.75)
    z[2] <- runif(1, -1, 1)
    xstar <- x1[,(t-1)] + z
```

```
    ratio <- exp(-0.5*t(xstar-mu)%*%solve(Sigma)%*%(xstar - mu))
            / exp(-0.5*t(x1[,(t-1)]-mu)%*%solve(Sigma)%*%(x1[,(t-1)] - mu))
    u <- runif(1)
    if(u <= ratio){
        x1[,t] <- xstar
        accept <- accept + 1
    }
    else {
        x1[,t] <- x1[,(t-1)]
    }
}
acceptratio1 <- accept / nsim
plot(x1[1,], x1[2,], xlim=c(-3,5), ylim=c(-3,6),xlab="x1", ylab="x2", main="(b) Random Walk (Uniform)")
grid(8,9)

# Candidate generating density 2 (Random Walk (Normal))
x2 <- matrix(NA, 2, nsim)
x2[,1] <- matrix(c(0,0), 2, 1)
accept <- 0
for(t in 2:nsim){
    z[1] <- rnorm(1, 0, 0.6^0.5)
    z[2] <- rnorm(1, 0, 0.4^0.5)
    xstar <- x2[,(t-1)] + z
    ratio <- exp(-0.5*t(xstar-mu)%*%solve(Sigma)%*%(xstar - mu))
            / exp(-0.5*t(x2[,(t-1)]-mu)%*%solve(Sigma)%*%(x2[,(t-1)] - mu))
    u <- runif(1)
```

```
    if(u <= ratio){
        x2[,t] <- xstar
        accept <- accept + 1
    }
    else {
        x2[,t] <- x2[,(t-1)]
    }
}
acceptratio2 <- accept / nsim
plot(x2[1,], x2[2,], xlim=c(-3,5), ylim=c(-3,6), xlab="x1", ylab="x2", main="(c)
Random Walk (Normal)")
grid(8,9)

# Candidate generating density 4 (Autoregressive Approach)
x4 <- matrix(NA, 2, nsim)
x4[,1] <- matrix(c(0,0), 2, 1)
accept <- 0
for(t in 2:nsim){
    z[1] <- runif(1, -1, 1)
    z[2] <- runif(1, -1, 1)
    xstar <- mu-(x4[,(t-1)]-mu) + z
    ratio <- exp(-0.5*t(xstar-mu)%*%solve(Sigma)%*%(xstar - mu))
            / exp(-0.5*t(x4[,(t-1)]-mu)%*%solve(Sigma)%*%(x4[,(t-1)] - mu))
    u <- runif(1)
    if(u <= ratio){
        x4[,t] <- xstar
        accept <- accept + 1
```

```
    }
    else {
        x4[,t] <- x4[,(t-1)]
    }
}
acceptratio4 <- accept / nsim
plot(x4[1,], x4[2,], xlim=c(-3,5), ylim=c(-3,6), xlab="x1", ylab="x2", main="(d)
Autoregressive Approach")
grid(8,9)
```

D. MATLAB Code for Fitting Negative Binomial Spatial Model

a. Main Function

```
% load data
load totalcounts.txt;       % fire counts in each grid cell
N = totalcounts;
load covariates.txt;        % matrix of covariates
X = covariates;
load W.txt;        % adjacency matrix
load Dw.txt;       % diagonal matrix of number of neighbours

% initial values (vary for different chains)
beta(1:6,1) = 0;        % regression coeeficients
nu(1:length(N),1) = 0.5;        % extra-poisson variations
a = 0.5;        % overdispersion parameters
b(1:length(N),1) = 0;        % spatial random effects
tau = 0.5;        % precision parameter of CAR model
```

% initial values for parameters of prior distribution

sigmabeta = 1000; % $\beta_p \overset{i.i.d}{\sim} \mathrm{N}(0, \sigma^2_{\boldsymbol{\beta}})$

eps = 1000; % $a \sim \mathrm{Gamma}(1/\varepsilon, \varepsilon)$

theta = 2; % $\tau = 1/\sigma^2_{\boldsymbol{b}} \sim \mathrm{Gamma}(theta, k)$

k = 1;

% metroplis algorithm jump step (adjust based on acceptance ratio)

sigbeta = [0.01;0.01;0.01;0.01;0.01;0.01];

siga = 0.05;

sigb = 0.1;

nsim = 800000; % number of interations

thin = 200;

% MCMC Algorithms Start Here

[resultbeta, resultnu, resulta, resultb, resulttau, accept] = NegBinIterationThin(N, X, W, Dw, beta, nu, a, b, tau, sigmabeta, eps, theta, k, sigbeta, siga, sigb, nsim, thin);

b. NegBinIterationThin Function

function [beta, nu, a, b, tau, accept] = NegBinIterationThin(N, X, W, Dw, beta, nu, a, b, tau, sigmabeta, eps, theta, k, sigbeta, siga, sigb, nsim, thin)

rbeta = length(beta(:,1));

lbeta = length(beta(1,:));

rb = length(b(:,1));

```
accept(1:(rbeta+1+rb),1) = 0;       % beta: 1-6; a: 7; b: remaining
count = 1;       % storage counter
tempbeta = beta(:, lbeta);
tempa = a(:, lbeta);
tempnu = nu(:, lbeta);
temptau = tau(:, lbeta);
tempb = b(:, lbeta);

for t=(lbeta+1):nsim
    % updating regression coefficients beta
    for i=1:rbeta
        newbeta = NegBinMetrop1(N, X, tempnu, tempb, tempbeta, i, sigmabeta,
sigbeta);
        if newbeta(i) ~= tempbeta(i)
            accept(i) = accept(i)+1;
        end
        tempbeta = newbeta;
    end
    % updating overdispersion paramter a
    newa = NegBinMetrop2(tempa, tempnu, eps, siga);
    if newa ~= tempa
        accept(7) = accept(7)+1;
    end
    tempa = newa;
    % updating extra-poisson variation nu
    intercept(1:length(N),1) = 1;
    X0 = cat(2, intercept, X);
```

```
    lambda = exp(X0*tempbeta+tempb);
    shapenu = N+1/tempa;
    scalenu = (lambda + 1/tempa).^(−1);
    tempnu = gamrnd(shapenu, scalenu);
    % updating precision parameter, tau, of CAR model
    shapetau = (length(N)-1)/2+theta;
    scaletau = 1/(0.5*tempb'*(Dw-W)*tempb+1/k);
    temptau = gamrnd(shapetau, scaletau);
    % updating CAR random effects b
    for i=1:rb
        newb = NegBinMetrop3(N(i), X(i,:), W, Dw, tempbeta, tempnu(i), tempb, i,
temptau, sigb);
        if newb(i) ∼= tempb(i)
            accept(rbeta+1+i) = accept(rbeta+1+i)+1;
        end
        tempb = newb;
    end
    tempb = tempb - mean(tempb);        % recentralize
    % store values
    if t == thin*count
        beta(:,count) = tempbeta;
        a(:,count) = tempa;
        nu(:,count) = tempnu;
        tau(:,count) = temptau;
        b(:,count) = tempb;
        count = count+1;
    end
```

```
end
accept = accept / nsim;        % calculate acceptance ratio
```

c. NegBinMetrop1 Function

```
% Function for updating regression coefficient beta
function [coef] = NegBinMetrop1(N, X, nu, b, coef, i, sigmabeta, sigbeta)
% star[i] ~ N(coef[i],sigbeta[i])
star = coef(i) + sigbeta(i)*randn(1);
coefstar = coef;
coefstar(i) = star;
% calculate the acceptance ratio
priorstar = normpdf(coefstar(i),0,sigmabeta);
if priorstar == 0
    priorstar = 0.00000000000000000000001;
end
prior = normpdf(coef(i),0,sigmabeta);
if prior == 0
    prior = 0.00000000000000000000001;
end
ratio = exp(NegBinLoglikelihood(coefstar,N,X,nu,b)-NegBinLoglikelihood(coef,N,X,nu,b))*
priorstar/prior;
% generate a number from uniform(0,1)
u = rand(1);
% accept xstar if the random number not larger than acceptance ratio
if u <= ratio
    coef(i) = star;
```

```
end
```

d. NegBinMetrop2 Function

```
% Function for updating dispersion parameter a
function [olda] = NegBinMetrop2(olda, v, eps, siga)
% log(star) ~ N(log(olda),siga)
logstar = log(olda) + siga*randn(1);
star = exp(logstar);
if star == 0
    star = 0.00000000000000000000000000001;
end
if star == Inf
    star = 10^300;
end
% calculate the acceptance ratio
top = log(star) + sum(log(gampdf(v,1/star,star))) + log(gampdf(star,1/eps,eps));
bottom = log(olda) + sum(log(gampdf(v,1/olda,olda))) + log(gampdf(olda,1/eps,eps));
if top == -Inf
    top = -10^300;
end
if top == Inf
    top = 10^300;
end
if bottom == -Inf
    bottom = -10^300;
end
```

```
if bottom == Inf
    bottom = 10^300;
end
ratio = exp(top-bottom);
u = rand(1);
if u <= ratio
    olda = star;
end
```

e. NegBinMetrop3 Function

```
% Function for updating spatial random effects b
function [b] = NegBinMetrop3(N, X, W, Dw, beta, v, b, i, tau, sigb)
% star ~ N(b[i],sigb)
star = b(i) + sigb*randn(1);
bstar = b;
bstar(i) = star;
% calculate the acceptance ratio
par1 = W(i,:)*bstar/Dw(i,i);        % mean
par2 = 1/(tau*Dw(i,i))^0.5;         % standard deviation
priorstar = normpdf(bstar(i),par1,par2);
if priorstar == 0
    priorstar = 0.00000000000000000000001;
end
par3 = W(i,:)*b/Dw(i,i);
prior = normpdf(b(i),par3,par2);
if prior == 0
```

```
    prior = 0.00000000000000000000001;
end
ratio = exp(NegBinLoglikelihood(beta, N, X, v, bstar(i))-NegBinLoglikelihood(beta,
N, X, v, b(i)))* priorstar/prior;
u = rand(1);
if u <= ratio
    b(i) = star;
end
```

f. NegBinLoglikelihood Function

```
% Function for calculating the log-likelihood of the model
function [logL] = NegBinLoglikelihood(beta, N, X, v, b)
intercept(1:length(N),1) = 1;
X = cat(2, intercept, X);
lambda = exp(X*beta+b);
zeroTFlambda =(lambda==0);
lambda(zeroTFlambda==1) = 0.000000000001;
InfTFlambda = (lambda==Inf);
lambda(InfTFlambda==1) = 10^300;
zeroTFv =(v==0);
v(zeroTFv==1) = 0.000000000001;
InfTFv = (v==Inf);
v(InfTFv==1) = 10^300;
logL = N'*log(v) + N'*log(lambda) - v'*lambda;
```